ISBN: 978-602-72004-3-2

PROSIDING

5th Indonesian Symposium on Robotic Systems and Control 2017 (ISRSC 2017)

Isola Resort, 6 Juli 2017







Publisher:
Fakultas Pendidikan Teknologi dan Kejuruan
Universitas Pendidikan Indonesia

SUSUNAN PANITIA

GENERAL CHAIRMAN
Dr. Ir. Endra Pitowarno, M.Eng. (Politeknik Elektronika Negeri Surabaya)

STEERING COMMITTEE
Dr. Kusprasapta Mutijarsa (Institut Teknologi Bandung)
Prof. Dr. Budi Mulyanti (Universitas Pendidikan Indonesia)
Dr. Ade Gafar Abdullah (Universitas Pendidikan Indonesia)
Dr. Ana (Universitas Pendidikan Indonesia)

ORGANIZING COMMITTE
Didin Wahyudin, Ph. D. (Universitas Pendidikan Indonesia) – Chair
Iwan Kustiawan, Ph. D. (Universitas Pendidikan Indonesia)
Dr. Dadang Lukman Hakim, M.T. (Universitas Pendidikan Indonesia)
Dr. Aip Saripudin, M.T. (Universitas Pendidikan Indonesia)
Dr. Jaja Kustija, M.Sc. (Universitas Pendidikan Indonesia)
Dr. Hasbullah, M.T. (Universitas Pendidikan Indonesia)

REVIEWER
Ir. Heru Santoso BR, M.Eng., Ph.D. (Universitas Gadjah Mada) – Chair
Dr. Ir. Wahidin Wahab, M.Sc. (Universitas Indonesia)
Ir. Gigih Prabowo, M.T. (Politeknik Elektronika Negeri Surabaya)
Dr. Eril Mozef, M.S., DEA. (Politeknik Negeri Bandung)
Prof. Dr. Mauridhi Hery Purnomo (Institut Teknologi Sepuluh Nopember)
Prof. Dr. Benyamin Kusumoputro (Universitas Indonesia)
Dr. Ir. Indrawanto (Institut Teknologi Bandung)
Dr. Ir. Djoko Purwanto, M.Eng. (Institut Teknologi Sepuluh Nopember)
Dr. Ir. Dian Retno Sawitri, M.T. (Universitas Dian Nuswantoro)

ISBN: 978-602-72004-3-2

Editor:
Didin Wahyudin
Kusprasapta Mutijarsa
Iwan Kustiawan

Desain sampul dan tata letak
Renaldi dan M. Husni

Cetakan pertama, Mei 2018

Penerbit:
Fakultas Pendidikan Teknologi dan Kejuruan UPI
Jl. Dr. Setiabudhi No. 207 Bandung 40154, Tel.: +62 2013163, Fax.: +62 2011576,
Email: fptk@upi.edu

KATA PENGANTAR

The Indonesian Symposium on Robotic Systems and Control (ISRSC) merupakan acara ilmiah tahunan bidang robotika yang diselenggarakan sebagai bagian dari rangkaian kegiatan Kontes Robot Indonesia Tingkat Nasional. Kegiatan ini merupakan kesempatan bagi komunitas akademik, peneliti, dan industri di bidang robotika untuk bertemu serta mendiskusikan berbagai solusi, hasil penelitian, dan metoda-metoda dalam menyelesaikan berbagai masalah serta tantangan pada bidang robotika.

5th Indonesian Symposium on Robotic Systems and Control (ISRSC 2017) dilaksanakan di ISOLA Resort, Universitas Pendidikan Indonesia pada tanggal 6 Juli 2017. Adapun topiknya mencakup pengembangan ilmu dan teknologi robotika, aplikasi dan rekayasa robotika, kompetisi robotika, serta aspek-aspek lainnya yang terkait dengan bidang robotika.

Prosiding ISRSC 2017 ini berisi makalah hasil *refereeing process*. Terdapat 53 makalah yang dipresentasikan peserta dari berbagai perguruan tinggi di Indonesia. ISRSC 2017 menghadirkan narasumber utama yaitu Prof. Dr. Ir. Bambang Riyanto Trilaksono, M.Eng. dari Institut Teknologi Bandung.

Kami mengucapkan terima kasih kepada semua fihak yang telah membantu terlaksananya kegiatan ini sehingga berjalan dengan baik. Terima kasih kepada Kementerian Ristek Dikti, para reviewer, dan seluruh civitas akademika Universitas Pendidikan Indonesia yang telah memberikan banyak bantuan kepada kami. Tak lupa kami ucapkan terima kasih kepada para peserta yang telah mengirimkan makalah dan berpartisipasi dalam kegiatan ini.

Akhir kata, mohon maaf atas segala kekurangan. Kritik dan saran konstruktif sangat kami nantikan. Semoga bermanfaat.

Editor

Didin Wahyudin
Kusprasapta Mutijarsa
Iwan Kustiawan

SAMBUTAN GENERAL CHAIRMAN

Assalaamu'alaikum Warohmatullaahi Wabarakaatuh.

Sukses penyelenggaraan dan meningkatnya minat pegiat di ISRSC sejak tahun 2013 hingga 2016 makin menegaskan arah peta yang benar bahwa kontes robot di Indonesia ini bukanlah ajang untuk sekedar berhura-hura di bidang teknologi, namun dapat menjadi pelaku utama dalam dunia riset teknologi masa depan. Untuk itu pada penyelenggaraan tahun 2017 ini kepanjangan dari nama ISRSC yang semula adalah Indonesian Symposium on Robot Soccer Competition dikembangkan dan disempurnakan menjadi 5th Indonesian Symposium on Robotic Systems and Control 2017 (ISRSC 2017). Dengan tanpa mengubah nama singkatan yang sudah populer ini diharapkan dalam waktu yang singkat ajang ISRSC dapat mendunia dan menjadi salah satu ajang konferensi terakreditasi yang menjadi rujukan para peneliti.

Dengan menggunakan kata kunci yang merupakan jantung dari para perekayasa teknologi dunia modern: robotic systems & control, maka hampir semua disiplin ilmu rekayasa masa depan masuk dalam kategori pengguna robotic systems & control. Kata "robot" – yang dalam konotasi lain adalah identik dengan intelligent/kecerdasan – mengisyaratkan bahwa dimensi aplikasi teknologi dalam kehidupan kita sehari-hari telah makin menyatu tanpa batas. Mari, kita tuntaskan pekerjaan besar kita: memakmurkan negeri dengan mencerdaskan dan meningkatkan kehidupan berteknologi.

Selamat bersimposium di 5th ISRSC 2017.

Wassalaamu'alaikum Warahmatullaahi Wabarakaatuh.

Bandung, 6 Juli 2017

General Chairman

Endra Pitowarno

DAFTAR ISI

SUSUNAN PANITIA ii
KATA PENGANTAR iii
SAMBUTAN GENERAL CHAIRMAN iv
DAFTAR ISI v
SUSUNAN ACARA x

Deteksi pergerakan berbasis kontur dan warna pada Robot krsbi beroda menggunakan OpenCV 1
R. H. Taher & S. Saputra

Sistem pendeteksian multi objek menggunakan aplikasi MATLAB pada robot 10-D (Ten-De) 4
Fanrani, A. Thamrin, N. R. Wahab, & Rudi

Deteksi Bola Multipola Pada Robot Krakatau FC 10
D. Setiawan, I. S. Rosandi, M. P. K. Putra, & S. Darmawan

Rancang Bangun Robot Berkaki Empat Pada Robot "Lizard" 14
M. E. Fajarwati, R. Rianto, A. R. Mirta, Nurhidayat, M. Rifki, & M. R. Fadillah

Penerapan Kontrol Berbasis Laptop Pada Robot Sepak Bola Beroda Indonesia (KRSBI) 19
T. Rahmadani, Rahmawati, A. Thamrin, R. R. Larengi, A. Kurniawan,& J. Rahman

Penggunaan Bahasa Pemrograman Python Sebagai Pusat Kendali Pada Robot 10-D 23
Fitri, K. Reski , A. Rahmansyah, & W. Darwin

Rancang Bangun Sistem Tertanam Pada Humanoid Robot Soccer Tim GLADIATOS UI 27
G. N. Putra, F. A. Wahab, D. S. Asih, M.. Surya, & A. K. Nugraha

Implementasi Sistem Kendali PD (Proportional Derivative) Pada Navigasi Wall Follower Robot Berkaki Enam (Hexapod) 33
Y. R. Wahyudi, M. S. Hadi, A. N. Handayani, & S. Sendari

Design and Implementation Software in Indonesian Traditional Dancing Humanoid Robot Development Platform 39
D. D. Nurchalifah, K. Mutjiarsa, & . Adiprawita

Implementasi Filter Morfologi Untuk Menghilangkan Noise Objek Pada Robot Sepak Bola 44

M. Fadlullah, D. A. Darussalam, M. Maulana, M. Rovicky, M. E. Hadi, Y. Syafarinda, & Yogiswara

Sistem Persepsi Berbasis Transformasi Hough pada Robot Sepakbola Humanoid Dago Hoogeschool 49
M. Isnain , A. Ekadinata, H. Lazuardi , Y. Sanyoto, A. Habbie , Tjia, J. Setiawan, F. Ananda , A. Fahrurrozi, R. D. Putra, A. Mi'raj, & F. Husaini

Pengembangan Platform Robot Sepak Bola Beroda 55
E. Febrianto, A. Faqih, A. Gumarilang, A. Wahrudin, A. Zakiyyatuddin, B. Edwiranda, B. Maulana, G. Wahyu, G. Surya, M. H. Asyrofi, M. I. Hadi, M. K. Nadjieb, R. Arindadewi, & Y. Yoga S.

Dribbling and Kicking Mechanism of A Soccer Robot for "KRSBI Beroda" 61
A. Sugianto, M. T. Kantori, & M. I. Nugraha

Pendeteksian Bola Orange dengan Metode Contour pada Robot R2C-WARRIOR 64
A. Cornelia

Sistem Deteksi Robot Lawan Berdasarkan Warna Identitas Robot Pada Robot Barelang 7.1 68
D. G. Tinambunan, S. Prayoga, I. A. Riandi, & R. A. Fatekha

Sistem Pendeteksian Bola Secara Real-Time Pada Robot Barelang FC 73
G. B. A. Dewasakti & Susanto

Rancang Bangun Omni Vision pada Robot ENSPARTAN KRSBI Beroda 78
Esa Ilham Akbar, Muhram Muis, & Leonard D. Prakoso*

Pergerakan Otomatis Robot Sepak Bola Beroda Melalui Komunikasi dengan Referee Box Menggunakan Base Station 82
T. Ardhiansyah, I. Syarifuddin, M. R. Naufal, Y. Pramono, O. T. Hartatik, & I. Mustaqim

Disain Omnidirectional Vision pada ERSOW 87
I. K. Wibowo, M. M. Bachtiar, R. T. Ulil, Z. Tri, M. Jauharul, & M. Khoirul

Sistem Pendeteksian Gawang menggunakan Metode Houghlines dan Cornerharris pada Robot Fire-X UAD 92
T. Nurochman, Y. P. Aji, & R. D. Puriyanto

Perancangan dan Realisasi Robot Underwater Autonomous Penjejak Garis dengan Metoda Thresholding dan Moments 97
Muliady & J. Sucipto

Application of Navigation Compass Android on Robot Soccer Based Arduino 103

M. S. Maulana, M. A. Fikri, & A. Triwibowo

Sistem Deteksi Bola Menggunakan Pustaka OPENCV dengan Kombinasi Dua Kamera (Logitech 720p & Kinnect XBOX 360) di Robot Sepak Bola Beroda 108
D. Prasetyo & Z. Abidin

Motion Planning in Dynamic Environment for Middle Size League using Theta* and Polynomial Trajectory Generator 113
F. A. Rahman, R. T. Hartanto, R. I. Chairistian, Y. Abdurrahman, O. F. Suryani, M. V. Astrid, R. D. H. A. Fahsi, & I. Ardiyanto

Development of Middle-Sized Mobile Robot FUKURO for Robot Soccer Competition 118
I. Ardiyanto, R. T. Hartanto, F. A. Rahman, R. I. Chairistian, M. V. Astrid, O. F. Suryani, Y. Abdurrahman, R. D. H. A. Fahsi, M. H. Afandi, D. Hendra, M. Shidiq, L. Enrico, O. P. Yudi, D. A. Wisesa, I. M. Arif, D. A. Dewangga, & E. Febrinanda

Sistem Penendang Bola Mekanis pada Robot Sepakbola Beroda 123
D. A. Dewangga, D. A. Wisesa, E. Febrinanda, I. M. Arif, & O. P. Yudi

Desain Solenoida Sebagai Mekanisme Penendang Robot Sepak Bola Beroda 127
M. H. Afandi, M. Shidiq, L. Enrico, & D. Hendra

Rancang Bangun Lengan Robot Pendeteksi Api Menggunakan Metode Pengolahan Citra 131
T. A. Agus, R. O. Wiyagi, A. N. N. Chamim, & Danardono

PID Controller-based Object Tracking and Speed Control System for Wheeled Soccer Robot 135
A. Maulana, W. Wibisono, R. A. Pamungkas, I. Riyanto, A. Musafa, & Sujono

Segmentasi dan Estimasi Jarak Bola dengan Robot Menggunakan Stereo Vision 140
I. Marzuqi, G. P. Arinata, Z.M.A Putra, & A. Khumaidi

Desain Robot Sepak Bola Beroda Tim Abimanyu 2017 145
J. Sahertian, A. Sanjaya, M. D. Yusuf, A. Sutrisno, M. Malik, & A. Makinun

Rancang Bangun Robot Sepak Bola Beroda 149
R. W. Pratama, G. R. Ananda, Y. D. Putra, P. Mahadika, & Y. Maulana

Image Normalization for Reducing Light Effect Based on HOG Object Detection 152
A. A. Rofiq, A. Athallah, M. I. Fikri, I. N. Ahmad, M. R. Fuadin, H. Baskoro, T. N. Winantyo, I. Sabri, A. N. Muhamad, F. H. Raswa, M. H. Akbar, T. N. Iman, M. H. Mukhlis, G. S. Mahardika, B., A . A. Sumbodo, & D. N. Annisa

Algoritma Multi Planning Pada Robot Soccer 156
I. P. Suryaningsih, A. Risal, & H. Jaya

Pembuatan Dataset Pelatihan dengan Optimalisasi Pseudocoloring Berbasis Depth untuk Faster Region-Proposed Convolutional Network Network 160
Y. Pratama, W. Adiprawita, & K. Mutijarsa

Strategi Pergerakan Penjaga Gawang pada Tim Rimau FC 166
B. O. Napitupulu & B. Y. Suprapto

Artificial Neural Network Untuk Pengukuran Posisi Bola Menggunakan Kamera Omnidireksional Pada Robot Sepak Bola Beroda 169
Y. Prakoso, A. Rachmawan, M. A. Multazam, A. A. R. Stone, A. A. Rachman, R. Dikairono, M. A. Ismail, H. D. Ardianika, & K. Arief

Implementasi Model Inverted Pendulum pada Proses Akselerasi dan Deselerasi Gerakan Berjalan Robot Sepak Bola Humanoid 174
A. M. Nurrochman, M. Arifin, D. Purwanto, & Muhtadin

Implementasi Sistem Handling dan Dribbling pada ERSOW (EEPIS Robot Soccer on Wheels) 181
M. I. Mas'udi, F. Ardilla, I. K. Wibowo, R. T. Ulil A, B. Mahardika,& M. Jauharul

Deteksi Bola Putih Menggunakan OpenCV pada Robot Penyerang dengan Mini PC Odroid XU4 186
Y. S. Pradana, D. Christanto, T. M. Ary, & S. S. Akdam

Dual-camera Perception System on Soccer Robot 189
D. B. Kusumawardhana,& K. Mutijarsa

ICHIRO TEAM – Team Desciption Paper 194
Muhtadin, S. Hafizhuddin, M. Arifin, M. R. A. Razi, D. S. Wicaksono, T. Pratama, S. Ali, D. Alfitra Ramazhoni, N. I. Revandhika, A. P. Adwitya, B. M. Putri, & A. M. Nurrochman

Rancang Bangun Robot DEVATA NEW GENERATION 2017 198
A. Susanto, J. Chayadi, Danieal,& A. B. Dharmawan

Implementasi Pusat Tekanan (CoP) Untuk Kontrol Keseimbangan Postur Pada Robot Humanoid 203
M. Arifin, D. Purwanto, Muhtadin,& A. M. Nurrochman

Increasing Stabiliy of Shooting Motion On EROS (EEPIS Robot Soccer) Using Joint Trajectory Controller 208
A. Sucipto, A. Subhan Kh., A. Husein, D. Pristovani, I. Fajar, A. K. Rizgi, R. S. Wijaya, K. A. Syahputra, I. F. Arifin, M.A. Basthomi, A. H. Almutawakkil,& A. Risnumawan

Pengembangan Pola Berjalan dan Stabilisasi pada Humanoid Robot Soccer DARDROID (DARWIN- ODROID) 212
Yohanes Fridolin Hestrio, Nicholas, Athina Maria Angelica,& M. Arifin Julian

Sistem Koordinasi Berbasis Perilaku pada Robot Sepak Bola 219
M. I. Hartanto, K. Mutijarsa, & W. Adiprawita

Rancang Bangun Sistem Lokomosi, Penggiring, dan Penendang pada Robot Sepak Bola 224
F. A. Widodo,& K. Mutijarsa

URT-ROSO 230
Daniyah,& L. O. M. Idris

Rancang Bangun Robot KRSBI Beroda dengan Raspberry Pi Pada Tim ISOLA_F 234
M. Renaldy, D. Mardiansyah, B. Afwan , & D. Wahyudin

Rancang Bangun Development Platform Robot Humanoid Penari Tarian Tradisional Indonesia 238
A. Z. H. Tuqo, & K. Mutjiarsa

Thermal Vision pada Manusia dengan Pengaruh Terhadap Warna Pakaian 243
M. Arif, R. Alandani, & G. M. Arya

SUSUNAN ACARA

5th Indonesian Symposium on Robotic Systems and Control (ISRSC 2017)

Universitas Pendidikan Indonesia

Kamis, 6 Juli 2017

Waktu	*Kegiatan*
07.00–08.30	Registrasi Peserta
08.30–08.40	Sambutan General Chairman ISRSC 2017 (Dr. Ir. Endra Pitowarno, M.Eng.)
08.40–08.50	Sambutan Rektor UPI (Prof. Dr. R. Asep Kadarohman, M.Si.)
08.50–09.20	Keynote Speech*
09.20–09.30	Sesi Tanya Jawab
09.30–09.45	Coffee Break
09.45–12.00	Sesi Paralel
12.00–13.00	Makan siang dilanjutkan acara penutupan simposium

***KEYNOTE SPEAKER**

Prof. Dr. Ir. Bambang Riyanto Trilaksono, M.Eng. (Institut Teknologi Bandung)

ATTENDANCE

Pemakalah wajib:	Tim KRSBI Beroda dan Humanoid yang lolos ke KRI Tingkat Nasional (tema makalah terkait dengan pengembangan robot tim yang bersangkutan)
Pemakalah reguler:	Kelompok peneliti (mahasiswa, dosen, praktisi)

VENUE

ISOLA Resort
Universitas Pendidikan Indonesia
Jl. Dr Setiabudhi No. 229, Isola, Sukasari, Kota Bandung, Jawa Barat 40154,
Phone: (022) 2001979

Deteksi Pergerakan Berbasis Kontur Dan Warna Pada Robot KRSBI Beroda Menggunakan OpenCV

Rifaldi Hi. Taher* dan Sofyan Saputra
STMIK Adhi Guna
Palu, Sulawesi Tengah
Vhaldy.taher@gmail.com

***Abstrak*— Pengolahan citra atau *image processing* saat ini menjadi primadona tersendiri bagi para pengembang robotika di Indonesia, sangat banyak librari untuk mendukung beberapa kegiatan pengolahan gambar diantaranya adalah *OpenCV* yang dapat diterapkan pada Bahasa pemrograman python sebagai basis Bahasa pemrograman standar untuk perangkat *Raspberry Pi,* pada penelitian ini peneliti mencoba untuk mengkombinasikan *Raspberry Pi* dengan librari *OpenCV* dimana secara standar telah disediakan algoritma sobel sebagai basis deteksi tepinya dan terbukti bahwa algoritma tersebut mampu mendeteksi seluruh tepi objek target, pada tahap implementasinya seluruh pergerakan objek dalam hal ini bola dapat terdeteksi dengan baik sesuai dengan kontur dan pola warna objeknya dalam hal ini adalah sebuah bola dan lapangannya, selain itu penggunaan *OpenCV* pada raspberry pi juga dapat menghemat biaya pengembangan karena kecilnya sumber daya yang digunakan.**

Kata kunci—Raspberry; OpenCV; python; image; processing;

I. PENDAHULUAN

Pengolahan citra (*image processing*) saat ini menjadi primadona tersendiri bagi seluruh pegiat informatika di Indonesia, bahkan hingga saat ini. Kontes Robot Sepak Bola Indonsia (KRSBI) merupakan suatu kegiatan rutin yang diadakan untuk melatih mahasiswa dalam merancang sebuah teknologi dimana salah satu teknologi yang digunakan adalah pengolahan citra gambar yang dimanfaatkan pada robot untuk mendeteksi gerakan terhadap bola.

Dalam implementasinya salah satu teknik dalam mendeteksi gerakan bola tersebut adalah dengan memanfaatkan pola warna dan kontur bola tersebut sehingga robot mampu untuk terus bergerak mencari bola teknologi ini biasa disebut dengan komputer visi (*computer vision*) salah satunya adalah memanfaatkan librari yang bernama *OpenCV,* dalam majalah ilmiah ini kami akan mengupas bagaiaman robot dapat mencari bola berdasarkan kontur dan pola warna bola yang diimplementasikan dengan menggunakan librari *OpenCV* pada Bahasa pemrograman *python*.

Raspberry Pi merupakan sebuah mikrokomputer berbasis *arm,* saat ini *Raspberry Pi* sangat handal dalam proses pengembangannya pada robot karena modelnya yang kecil dan mampu mengolah proses pengolahan yang cukup besar, pada implementasi peneliti menggunakan Raspberry Pi Sebagai pemrosesan gambar dalam bentuk *video*.

II. TINJAUAN PUSTAKA

A. *OpenCV*

OpenCV adalah perpustakaan visi komputer *open source*. Perpustakaan ditulis dalam bahasa C dan C ++ dan Berjalan di bawah Linux, Windows dan menyediakan antar muka untuk Python, Ruby, Matlab dan bahasa lainnya. Perpustakaan *OpenCV* berisi banyak fungsi matematika lanjutan, pemrosesan fungsi gambar, dan fungsi penglihatan komputer yang menjangkau banyak area dalam jangkauannya[1].



Gambar 1. Logo *OpenCV*

B. *Python*

Bahasa pemrograman *python* merupakan Bahasa komputer tingkat tinggi yang menggunakan *interpreter* sebagai penerjemah sehingga seluruh proses dilakukan di*interpreter* tersebut, Bahasa pemrogeaman *python* saat telah mempunyai sangat banyak librari pendukung sehingga membuat bahasa pemrograman ini tetap eksis dengan skalabilitas yang tinggi hingga sekarang ini[2].



Gambar 2. Logo Python

C. *Raspberry Pi*

Raspberry pi merupakan sebuah mikroprosesor berbasis ARM yang dikembangkan oleh *Raspberry Pi Foundation* di

Inggris, hingga saat ini *Raspberry* telah mempunyai 4 versi yakni versi 3 merupakan yang terakhir[3]. Raspberry Pi terkenal dengan *General Purpose Input Ouput* (GPIO) dalam mengontrol peralatan baik dibidang robotika ataupun perangkat elektronik lainnya. Raspberry pi menggunakan sistem operasi basis GNU linux sebagai pusat pengontrolnya dan python sebagai Bahasa pemrogrman standarnya.



Gambar 3. Raspberry Pi

III. METODE

A. *Blok Diagram*

Dalam mendeteksi gerakan bola pada suatu robot peneliti mencoba mendeteksi gerakan bola berdasarkan pola warna dan contour, adapun blok diagram proses pendeteksian dapat dilihat pada Gambar 4. Berikut ini.



Gambar 4. Blok Diagram Proses

Proses pertama yang dilakukan adalah dengan mengecek kamera, jika kamera telah siap maka proses selanjutnya adalah dengan mendeteksi tepi setiap objek *target* dengan menggunakan algoritma sobel.

Proses selanjutnya adalah dengan mencari pola warna yang pertama akan diubah kedalam bentuk *histogram* kemudian akand direduksi dan disaring menjadi sebuah gambar *greyscale* sehingga *noise* pada gambar dapat dihilangkan.

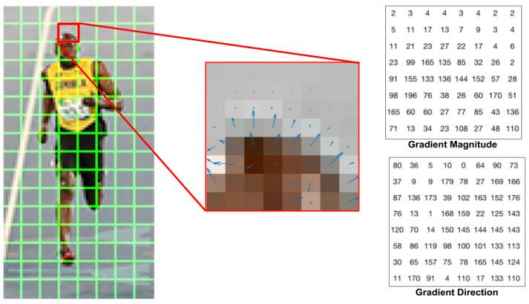
Setelah *noise* di kurangi langkah selanjutnya adalah pengambilan pola warna yang diharapkan pada kasus ini peneliti menggunakan warna hijau yang sesuai dengan rentan setelah itu kunci objek sehingga proses deteksi akan lebih baik lagi.

B. *Sobel Edge Detection*

Dalam menghitung kontur tepi pada *OpenCV* peneliti menggunakan algoritma sobel yang diterapkan secar *default* proses perhitungan adalah dengan mengambil *gradient* horizontal dan *gradient* vertical.

$$g = \sqrt{g_x^2 + g_y^2}$$
$$\theta = \arctan \frac{g_y}{g_x} \quad (1)$$

Adapun contoh pengambilan *gradient* dapat dilihat pada Gambar 5.



Gambar 5. Contoh Penentuan Pola *Gradient*

IV. HASIL DAN PEMBAHASAN

A. *Alat Dan Bahan*

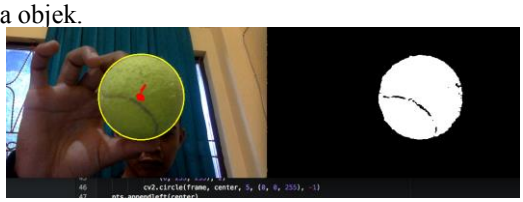
Dalam merancang Robot yang akan dibuat maka peneliti memerlukan bahan dan alat berikut yang dirincikan secara umum yaitu penggunaan perangkat keras dan lunak.

1. Perangkat Keras
   a. *Raspberry Pi* digunakan sebagai pemrosesan
   b. *Pi Camera*/*Microsoft HD Cam* digunakan untuk mendeteksi objek.
   c. *Laptop* digunakan untuk mengolah dokumen besar yang tidak mampu diolah oleh *Raspberry Pi*

2. Perangkat Lunak
   a. *Python* merupakan Bahasa standar dari *Raspberry pi*.
   b. *OpenCV* sebagai librari pendukung pengolahan visi komputer.
   c. *Numpy Array* untuk mengolah seluruh matriks dalam proses pengolahan citra.
   d. *Scikit* digunakan untuk membangun proses kecerdasan pada robot.
   e. *ImgUtils* untuk mengolah seluruh pengaturan gambar.

B. *Deteksi Kontur Pada Bola*

Deteksi kontur pada robot yaitu dengan mendeteksi tepi terlebih dahulu. Dalam pengujian peneliti menggunakan bola yang berkontur bundar dan berwarna hijau sehingga pada implementasinya *OpenCV* akan mendeteksi objek bulat kemudian objek dikunci yang ditandai dengan garis berwarna kuni pada objek.



Gambar 6. Deteksi Kontur Bulat Pada Bola

Dalam proses percobaannya peneliti mencoba untuk menggunakan beberapa objek yang bentuknya tidak bulat namun karena kontur sebelumya telah di atur sesuai dengan

objek penangkapan pada Robot yaitu bulat. Maka penanda kontur akan selalu dalam bentuk bulat.

C. Deteksi Pola Warna

Untuk mendeteksi pola warna, pada penelitian peneliti mengkonversi hasil konversi kontur agar pola warna yang direkam tidak terpecah kecuali terdapat pola warna yang hamper mirip dan kontur yang juga sama.



Gambar 7. Hasil Deteksi Pola Warna

Setelah pola warna terdeteksi maka warna tersebut akan di kunci pada kontur yang telah didapat sebelumnya agar warna tidak lepas dari tangkapan objek kamera. Jika terjadi pola warna tersebut keluar dari jangkauan kamera maka kamera akan melakukan penyegaran kemudian melakukan pemindaian ulang.

D. Tabel Percobaan Pola Warna

Dalam proses pengujian pendeteksian pengujian dilakukan oleh peneliti dengan melakukan beberapa kali pemindaian dalam hal ini pengujian dapat dilihat pada Tabel I. Berikut ini.

TABEL I : PENGUJIAN RENTAN POLA WARNA

No	Pengujian Pola Warna (RGB)		
	Pola Warna	***Berhasil***	***Tidak***
1	(12,255,0)	1	0
2	(31,239,138)	1	0
3	(71,244,71)	0	1
4	(44,243,53)	1	0

E. Kamera Dan Raspberry Pi

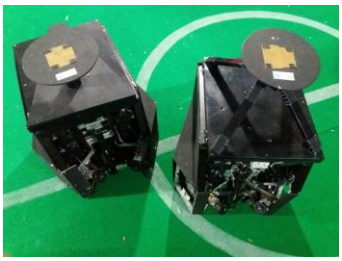
Kamera yang digunakan pada penelitian ini menggunakan kamera *Microsoft life came studio* yang dikombinasikan dengan raspberry pi sebagai proses pengolahan gambarnya.



Gambar 8. Kamera Dan Raspberry Pi

F. Implementasi Robot KRSBI

Setelah melakukan pengujian pada sistem biasa maka peneliti menerapkan konsep tersebut kedalam Robot KRSBI, adapun Robot dapat dilihat pada Gambar 9.



Gambar 9. Implementasi Robot KRSBI Beroda

Robot yang diimplementasikan terdiri dari 3 yakni 2 sebagai penyerang dan 1 sebagai penjaga gawang, robot penyerang dilengkapi dengan Raspberry Pi dan Kamera yang bergerak dari segala arah, sedangkan robot gawang terdiri dari kamera yang hanya berarah kedepan dengan pola bergerak kiri dan kanan sebagai strategi untuk menghalau bola yang digiring oleh lawan.

V. KESIMPULAN

Pengolahan citra saat ini menjadi primadona tersendiri bagi para pengembang robotika di Indonesia, sangat banyak librari untuk mendukung beberapa kegiatan pengolahan gambar diantaranya adalah *OpenCV* yang dapat diterapkan pada Bahasa pemrograman python sebagai *basis* Bahasa pemrograman standar untuk perangkat *Raspberry Pi*.

Pada penelitian ini peneliti mencoba untuk mengkombinasikan *Raspberry Pi* dengan librari *OpenCV* dimana secara standar telah disediakan algoritma sobel sebagai basis deteksi tepinya dan terbukti bahwa algoritma tersebut mampu mendeteksi seluruh tepi objek *target*, pada tahap implementasinya seluruh pergerakan objek dalam hal ini bola dapat terdeteksi dengan baik sesuai dengan kontur dan pola warna objeknya dalam hal ini adalah sebuah bola dan lapangannya, selain itu penggunaan *OpenCV* pada raspberry pi juga dapat menghemat biaya pengembangan karena kecilnya sumber daya yang digunakan.

REFERENSI

[1] Nidhi, “Image Processing and Object Detection,” Dept. of Computer Applications, NIT, Kurukshetra, vol. 1(9) , pp. 1-2, June 2017.

[2] Foundation Python,The Python Papers., vol. 4 No. 2. Python Foundation, 2009, pp.3-6.

[3] Adams, Raspberry Pi Compute Module IO Board Electrical Schematic Diagram, Raspberry Pi Foundation, 2014.

[4] Jain dkk, Raspberry Pi based Interactive Home Automation System through E-mail, International Conference on Reliability, Optimization and Information Technology-ICROIT, 2014, India.

[5] Fairhurst, Michael and Smith, Stephen L. and Mitchell, John . 1995. “Automated ImageAnalysis in Visuo-Motor Testing for the Specification of an Integrated Evaluation and Terapy Suport Toll for Rehabilitation”. IEEE Transactions on Rehabilitation Engineering.

Sistem Pendeteksian Multi Objek Menggunakan Aplikasi *MATLAB* Pada Robot 10 - D (*Ten - De*)

Fanrani*, Adrin Thamrin, Nasrul R Wahab, Rudi
Program Studi Teknik Informatika
STMIK Adhi Guna Palu
Palu, Indonesia
fanranigokill@gmail.com

***Abstrak*—Salah satu syarat robot sepakbola untuk tampil adalah dengan menempelkan kostum pada robot. Kostum yang terdapat pada robot yakni memiliki warna *cyan* dan *magenta*. Robot dituntut memiliki kecerdasan dalam mendeteksi warna tersebut, agar dapat mengenali lawan dan kawan serta bola. Diperlukan kemampuan robot agar mampu mendeteksi multi bjek. Metode yang digunakan dalam proses pengambilan data yakni : Pertama, melakukan pengambilan gambar. Kedua, mencari nilai RGB *Image*. Ketiga, menentukan nilai RGB *Image*. Nilai RGB tersebut akan di konversi ke *black white* sehingga akan menjadi data yang akan diproses dalam sebuah *range* nilai, kemudian nilai tersebut, akan menjadi data karakter untuk menjalankan mobile robot. Didapatkan hasil yang baik dalam mendeteksi multi objek pada robot dalam mengambil keputusan pada permainan robot sepak bola beroda.**

Kata kunci : cyan, magenta, multi objek, bola, RGB, black white

I. PENDAHULUAN

Pengolahan citra digital menggunakan teknologi *computer vision* saat ini sangat berkembang penggunaannya. Dimana citra yang diolah dan menghasilkan informasi dijadikan parameter untuk mengambil keputusan. Sebagai salah satu contoh bentuk pengaplikasian *computer vision* dalam dunia kompetisi yakni Kontes Robot Sepak Bola Indonesia (KRSBI). Pengolahan citra merupakan bagian dari *computer vision,* dimana data citra yang ditangkap oleh kamera di olah untuk mengetahui parameter tertentu dari objek yang dideteksi. Salah satu parameter yang dijadikan acuan dalam kontes robot pemain bola adalah melalui warna yang ada pada objek. Sementara dalam lingkungan robot pemain bola tersebut terdapat beberapa banyak objek baik yang ada dalam lingkup lapangan permainan maupun di luar lapangan.

Merujuk pada peraturan Kontes Robot Indonesia (KRI) Divisi Robot Sepak Bola Beroda (KRSBI), bahwa setiap tim harus memiliki tanda warna yang menempel pada robot. Maka berdasarkan aturan tersebut dapat dilakukan pendeteksian robot melalui pengenalan warna secara akurat yang menempel pada robot tersebut baik robot kawan maupun robot lawan. Sebagaimana prinsip dasar permainan bola, dimana bola yang menjadi objek utama untuk diperebutkan baik tim lawan maupun tim kawan. Sehingga dalam lingkungan lapangan robot harus melakukan pendeteksi minimal tiga dari objek yang ada baik robot lawan maupun robot kawan dan terlebih bola itu sendiri.

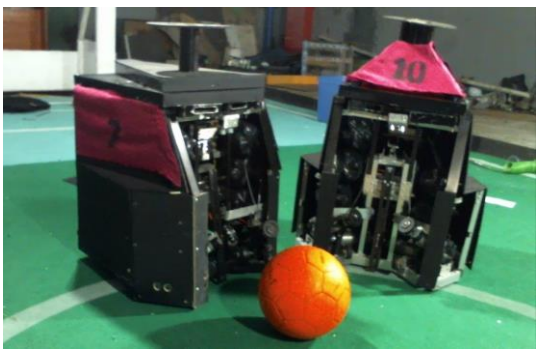
Pada penelitian ini akan difokuskan pada sistem pendeteksian multi objek pada lapangan permainan robot sepak bola beroda (KRSBI Beroda 2017) menggunakan aplikasi Matlab untuk melakukan pengolahan citra oleh robot.

II. DASAR TEORI

A. *Robot Sepak Bola Beroda*

Robosoccer adalah salah satu jenis *mobile* robot sederhana yang dapat bergerak sesuai perintah yang diberikan dari *remote* kontrol dapat berupa maju, mundur, belok kiri, dan belok kanan.[1]

Ide awal menggunakan robot dalam permainan sepak bola dicetuskan oleh Profesor Alan Mcworth dari Universitas British Columbia, Kanada dalam papernya yang *berjudul "On Seeing Robot"* yang dipresentasikan padaVI- 92, 1992 dan dipublikasikan di buku *Computer Vision: System, Theory, and Applications* halaman 1-13, *World scientific Press*, Singapore, 1993. Beberapa seri dari paper yang diterbitkan oleh kelompoknya adalah tentang motor robot untuk sepak bola.[1]



Gambar 1 : *Robot Sepak Bola Beroda 10 – D (Tende)*

B. *Pengolahan Citra Digital*

Menurut Ahmad (2005) pengolahan citra (*image processing*) merupakan suatu sistem *visual* yang mengolah data citra dengan hasil pengolahan berbentuk citra lain yang

mengandung atau memperkuat informasi khusus pada citra hasil pengolahan sesuai dengan tujuan pengolahannya. Pengertian pengolahan citra (*image processing*) sedikit berbeda dengan pengertian mesin visual (*machine vision*), meskipun keduanya seolah-olah dapat digunakan dengan maksud yang sama. Sedangkan terminologi mesin visual digunakan bila data hasil pengolahan citra langsung diterjemahkan dalam bentuk lain, misalnya grafik yang siap diinterpretasikan untuk tujuan tertentu, gerak peralatan atau bagian dari peralatan mekanis, atau aksi lainnya yang berarti bukan merupakan citra lain.[2]

Citra *digital* dapat diperoleh secara otomatis dari sistem penangkap citra *digital* (*digital image acquisition system atau digitizen*) yang melakukan penjelajahan citra dan membentuk suatu matriks dimana elemen-elemennya menyatakan nilai intensitas cahaya pada suatu himpunan diskrit dari titik-titik. Sistem tersebut merupakan bagian terdepan dari suatu sistem pengolah citra. Sistem penangkap citra digital sendiri terdiri dari tiga komponen dasar, yaitu: sensor citra yang bekerja sebagai pengukur intensitas cahaya, perangkat penjelajah yang bertugas merekam hasil pengukuran intensitas pada seluruh bagian citra, dan pengubah *analog*-ke-*digital* yang mengubah harga kontinu menjadi harga diskrit sehingga dapat diproses dengan komputer (Arymurti dan Setiawan 1992).[2]

C. *Computer Vision*

Computer vision merupakan proses otomatis yang mengintegrasikan sejumlah besar proses untuk persepsi *visual*, seperti akuisisi citra, pengolahan citra, pengenalan dan membuat keputusan. *Computer vision* mencoba meniru cara kerja sistem visual manusia (*human vision*) yang sesungguhnya sangat kompleks. Untuk itu, *computer vision* diharapkan memiliki kemmpuan tingkat tinggi sebagaiman *human visual*. Kemampuan itu diantaranya adalah:

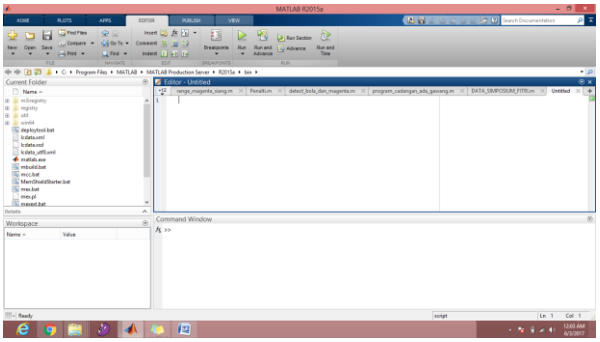
1. *Object detection* → Apakah sebuah objek ada pada *scene*? Jika begitu, dimana batasan-batasannya..?
2. *Recognation* → Menempatkan label pada objek.
3. *Description* → Menugaskan properti kepada objek.
4. 3D *Inference* → Menafsirkan adegan 3D dari 2D yang dilihat.
5. *Interpreting motion* → Menafsirkan gerakan.

Computer Vision sering didefinisikan sebagai salah satu cabang ilmu pengetahuan yang mempelajari bagaimana komputer dapat mengenali obyek yang diamati/ diobservasi. Cabang ilmu ini bersama intelijensia semu (*Artificial Intelligence*) akan mampu menghasilkan sistem intelijen *visual (Visual Intelligence System).*[3]

D. *Aplikasi MATLAB*

MATLAB *(Matrix Laboratory)* adalah sebuah program untuk analisis dan komputasi numerik dan merupakan suatu bahasa pemrograman matematika lanjutan yang dibentuk dengan dasar pemikiran menggunkan sifat dan bentuk matriks. Pada awalnya, program ini merupakan interface untuk koleksi rutin-rutin numerik dari proyek LINPACK dan EISPACK, dan dikembangkan menggunkan bahasa FORTRAN namun sekarang merupakan produk komersial dari perusahaan Mathworks, Inc.yang dalam perkembangan selanjutnya dikembangkan menggunakan bahasa C++ dan assembler (utamanya untuk fungsi-fungsi dasar MATLAB). [4]

MATLAB telah berkembang menjadi sebuah *environment* pemrograman yang canggih yang berisi fungsi-fungsi *built-in* untuk melakukan tugas pengolahan sinyal, aljabar linier, dan kalkulasi matematis lainnya. MATLAB juga berisi *toolbox* yang berisi fungsi- fungsi tambahan untuk aplikasi khusus . MATLAB bersifat *extensible*, dalam arti bahwa seorang pengguna dapat menulis fungsi baru untuk ditambahkan pada *library* ketika fungsi-fungsi *built-in* yang tersedia tidak dapat melakukan tugas tertentu. Kemampuan pemrograman yang dibutuhkan tidak terlalu sulit bila Anda telah memiliki pengalaman dalam pemrograman bahasa lain seperti C, PASCAL, atau FORTRAN.[4]



Gambar 2. *Tampilan MATLAB*

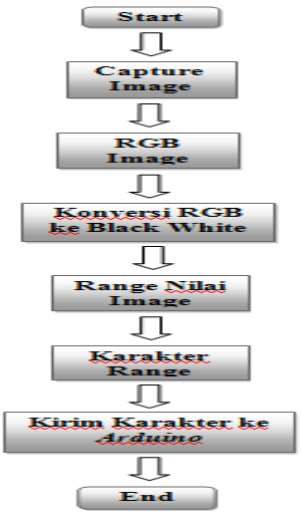
III. METODE PENELITIAN

Adapun metode penelitian yang digunakan dalam sistem pendeteksian *multi* objek pada robot pemain bola ini adalah :

1. Pengolahan Citra Pada Matlab
2. Pengolahan *Multi Image*
3. Pengambilan *Data/Scanning*
4. Penentuan karakter hasil olah data citra

Metode penelitian dapat dilihat dari beberapa diagram blok dan *flowchart* dengan beberapa model digunakan.

A. Pengolahan Citra Pada Matlab



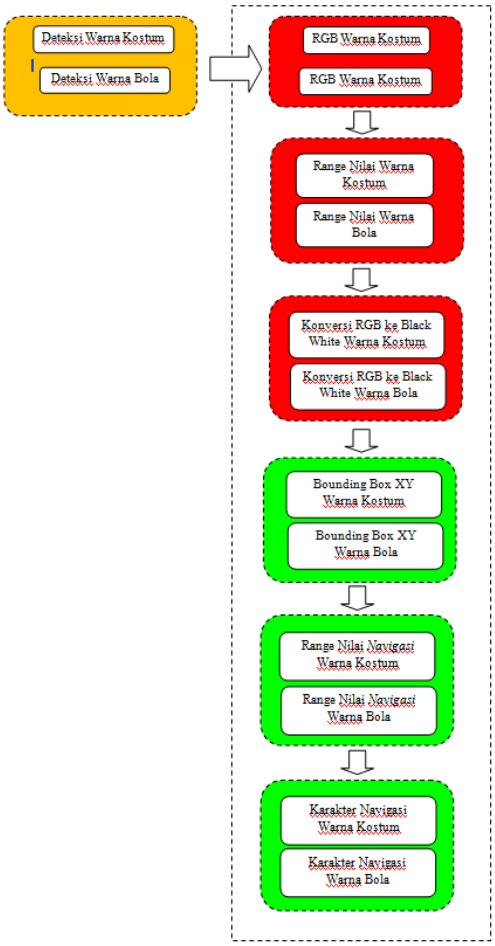
Gambar 3. *Alur Pengolahan Citra Pada Matlab*

Keterangan :

1. Tahap pertama yakni mengambil gambar atau *snapshot image*, nantinya hasil tersebut akan dikirim ke laptop.
2. Kemudian hasil yang diperoleh akan diproses lebih lanjut pada laptop melalui aplikasi *MATLAB* sebagai pemroses data. Data RGB adalah data awal yang akan diolah untuk menjadi *data real image processing*.
3. Selanjutnya data tersebut akan dikonversi ke *Black White* agar menjadi data *real* yang siap untuk dijadikan parameter eksekusi *mobile robot*.
4. Hasil data *real* yang diperoleh akan dikemas menjadi *range* nilai.
5. Selanjutnya *range* nilai tersebut akan merepresentasikan data dalam bentuk karakter angka, kemudian karakter tersebut akan dikirim ke *board* arduino melalui jalur data *serial*.

B. pengolahan Multi Image

Proses pengolahan image dilakukan pada aplikasi *MATLAB* yang merujuk pada 2 (dua) objek warna yakni objek warna kostum dan objek warna bola. Terdapat dua *focus* dalam penyusunan program pada matlab yakni pada *pengolahan image processing* dan menentukan *bounding box pada objek*. Untuk lebih jelasnya dapat dilihat pada gambar 4.



Gambar 4. *Alur Proses Pengambilan Data Citra*

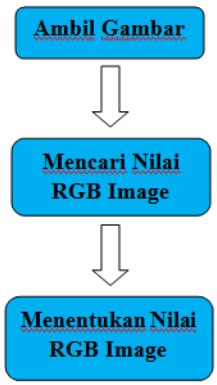
Keterangan :

1. Tahap 1 (Merah) : *Pengolahan Citra Digital*
   a. Data RGB ysng dihasilkan dari *capture image* kamera menjadi data awal pada pengolahan citra.
   b. Pengambilan data dilakukan dengan menentukan masing-masing nilai R=*Red* G=*Green* B=*Blue*. setelah pengambilan data selanjutnya data tersebut dikemas menjadi sebuah range nilai R, G dan B.
   c. Hasil nilai RGB yang didapatkan selanjutnya akan dikonversi ke *Black White* atau mengubah menjadi nilai 0 dan 1.

2. Tahap 2 (Hijau) : *Penentuan Bonding Box Citra*
   a. Memberikan *bounding box* untuk koordinat XY pada masing-masing kostum.
   b. Setelah koordinat XY, maka selanjutnya adalah menentukan *range* nilai navigasi.
   c. Selanjutnya *range* nilai akan diberikan karakter, yang mana masing-masing karakter mewakili navigasi maupun bentuk akhir dari keputusan arah gerak robot.

C. *Pengambilan data/scanning*

Adapun proses yang digunakan dalam melakukan pengambilan data atau *scanning* citra adalah sebagai berikut :



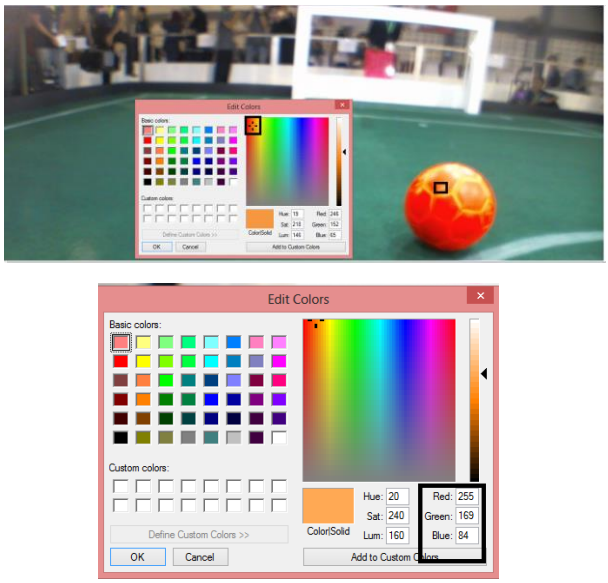
Gambar 5. Proses Pengambilan Data

1. *Ambil Gambar*

Dalam melakukan *scanning image*, tahapan pertama yang dilakukan adalah melakukan pengambilan gambar, dalam kontes robot sepakbola beroda, pengambilan gambar dilakukan dengan menscanning kostum robot yang menempel pada badan robot lawan dan pengambilan gambar bola yang digunakan dalam permainan.

2. *Mencari Nilai RGB Image*

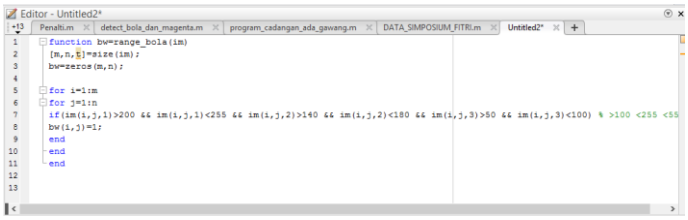
Tahapan selanjutanya adalah mencari nilai RGB menggunakan aplikasi paint.



Gambar 6. Mencari Nilai RGB Image

3. *Menentukan Nilai RGB Image*

Hasil nilai RGB yang diperolah akan dijadikan nilai acuan pada program yang dibuat dalam aplikasi Matlab.



Gambar 7. Menentukan Nilai RGB Image

IV. HASIL PENELITIAN

A. *Range Nilai RGB Image*

Range nilai RGB Image merupakan kemasan paket nilai yang nantinya menghasilkan sebuah data. Dapat dilihat pada tabel I, range nilai RGB Image mewakili masing-masing warna. Adapun tabel yang dimaksud adalah sebagai berikut :

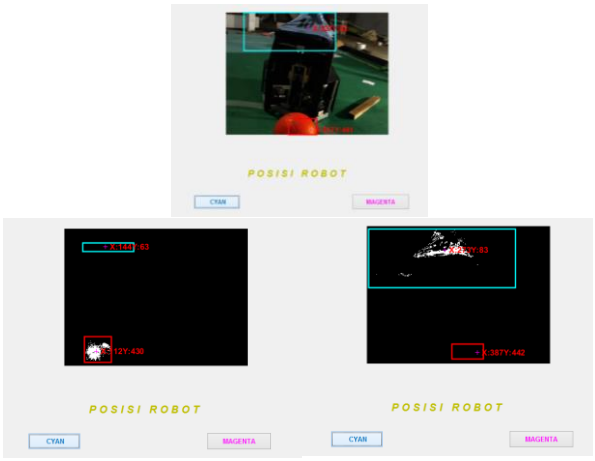
TABEL I. *RANGE* NILAI *RGB IMAGE*

Warna Image	Range Nilai		
	R	**G**	**B**
Warna Bola	Red = R>200 && R<250	Green = G>70 && G<120	Blue = B>1 && B<50
Warna Cyan	Red = R>70 && R<90	Green = G>80 && G<110	Blue = B>100 && B<125
Warna Magenta	Red = R>200 && R<220	Green = G>20 && G<70	Blue = B>50 && B<120

B. *Deteksi Multi Objek*

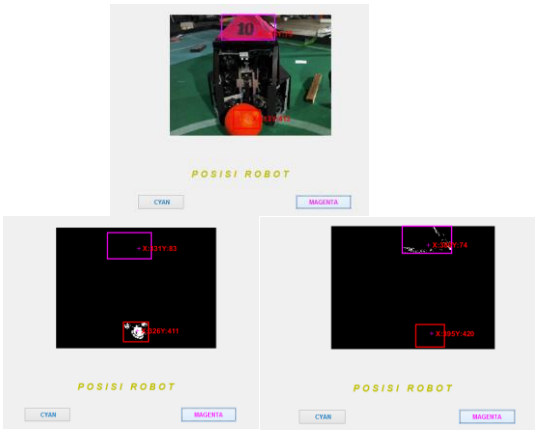
Dapat dilihat pada gambar bahwa deteksi multi objek antara bola dan warna kostum secara bersamaan, terlihat tampilan *RGB* dan *Black White*

1. Deteksi Warna Bola dan Warna Cyan



Gambar 8. Deteksi Warna Bola dan Warna Cyan

2. Deteksi Warna Bola dan Warna Magenta



Gambar 9. Deteksi Warna Bola dan Warna Magenta

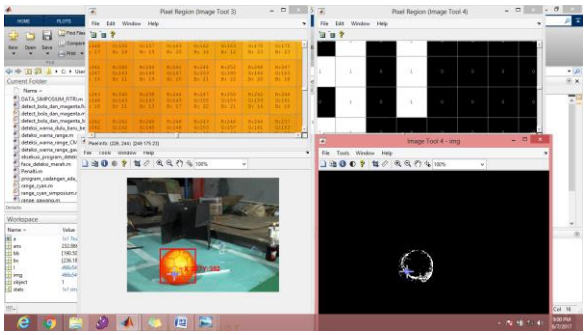
C. Data Masing-Masing Warna

Data yang diperlihatkan merupakan data real RGB dan konversi hasil *RGB* ke dalam bentuk citra *Black White*. Data RGB mempunyai nilai data *byte* sedangkan untuk *data black white* mempunyai nilai data 0 dan 1.

Data yang digunakan pada data RGB adalah data yang mencakup ruang lingkup dalam *range* yang ditentukan sebelumnya. Kemudian data yang didapatkan pada nilai *range* data RGB akan menghasilkan nilai 1 dari hasil konversi ke *black white*, sedangkan data yang tidak digunakan adalah data yang bernilai 0.

1. Warna Bola

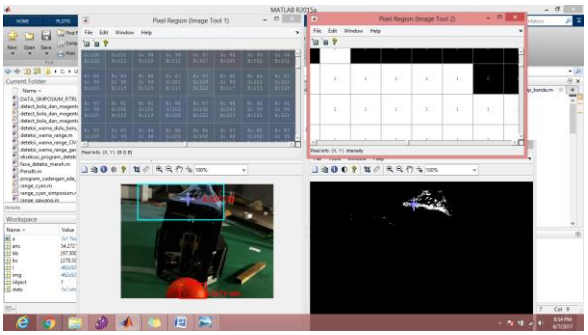
Nilai bola yang dihasilkan pada pendeteksian warna bola seperti yang diperlihatkan pada gambar 10 :



Gambar 10. Hasil Konversi Warna Bola

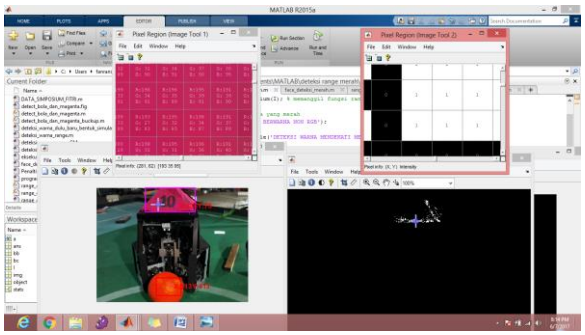
Sedangkan citra dari objek lain yang dideteksi yang merupakan warna kostum lawan baik pada posisi *cyan* maupun *magenta* masing-masing data citra diperlihatkan seperti pada gambar 11 dan gambar 12.

2. Warna Cyan



Gambar 11. Hasil Konversi Warna Cyan

3. Warna Magenta



Gambar 12. Hasil Konversi Warna Magenta

V. KESIMPULAN

Berdasarkan proses perancangan dan pembuatan serta pengujian sistem pendeteksian *multi* objek pada robot sepak bola beroda (tim TEN-DE) diperoleh hasil yang sesuai dengan yang diharapkan. Dimana kemampuan robot dalam mendeteksi citra lebih dari satu dapat dilakukan. Robot dapat melakukan pergerakan dalam mendeteksi, mengejar dan memegang objek bola melalui olah citra yang diterapkan. Selain pendeteksian objek berupa bola, robot juga dapat mendeteksi objek berupa robot lawan yang menggunakan kostum baik cyan atau magenta yang berguna bagi robot dalam melakukan manuver maupun mengambil keputusan dalam menendang bola ke arah gawang lawan.

Namun disadari dalam proses penelitian yang dilakukan masih terdapat beberapa kekurangan diantaranya adalah *noise* yang terdapat pada hasil olah citra. Maka untuk penelitian selanjutnya diperlukan algoritma yang lebih baik dalam menghasilkan data citra yang lebih akurat, khususnya untuk studi kasus dalam permainan robot sepak bola beroda.

DAFTAR PUSTAKA

[1] Tenday Agus Setya Wibawa, Akuwan Saleh, Arifin,"Aplikasi Pengolahan Citra Berbasis Gerakan Tangan untuk Pengendali Robot Soccer Wireless",PENS-ITS, Surabaya

[2] Irriwad Putri, "Pengembangan Program Pengolahan Citra Real-Time untuk Deteksi Rintangan pada Traktor tanpa Awak", Institut Pertanian Bogor, Bogor 2013.

[3] Priskanta Tarigan . Pengenalan Computer Vision https://freezcha.wordpress.com/2010/11/16/computer-vision/

[4] Anonymous, "Pengantar untuk Pemrograman MATLAB", Institut Pertanian Bogor, Bogor 2013.

[5] Adi Pamungkas. Source Code GUI. https://pemrogramanmatlab.wordpress.com/source-code-gui-matlab-download/.

[6] Panitia KRI, "Buku Panduan Kontes Robot Sepak Bola Indonesia Beroda", RISTEDIKTI, 2017

Deteksi Bola Multipola Pada Robot Krakatau FC

Dedi Setiawan, Indra Setiawan Rosandi, Muhammad Pajar Kharisma Putra*, Satria Darmawan
Robotic Teknokrat
pajarkharisma@hotmail.com

***Abstrak* – Kontes Robot Sepakbola Indonesia 2017 yang mengacu pada rule robocup 2016 mengalami peningkatan tantangan yang signifikan yaitu bola yang 50% berwarna putih dan sisanya random serta pola yang beragam. Bola yang memiliki warna cenderung sama dengan gawang dan garis lapangan tentunya akan menyulitkan robot jika melakukan deteksi berbasis warna. Begitupun halnya dengan deteksi berbasis bentuk, akan sama sulitnya dilakukan karena kondisi bola yang benar – benar bulat akan jarang ditemui saat kontes. Kami menggunakan metode learning untuk melakukan deteksi bola. Metode ini dinilai efektif untuk mengenali pola bola yang beragam. Dari hasil pengujian, kondisi eksternal robot didapat melalui *sensor passive* berupa digital kamera Logitech c920 dan di otaki oleh komputer dengan processor Intel Cherry Trail yang bekerja pada clock 1.8 GHz, didapat bahwa strategi yang diajukan dapat melokalisasi bola sekaligus mengeliminasi *noise* dalam waktu 30 ms, sehingga robot masih dapat merespon perubahan kondisi lingkungan secara *real-time*.**

Kata kunci – Deteksi Bola, learning

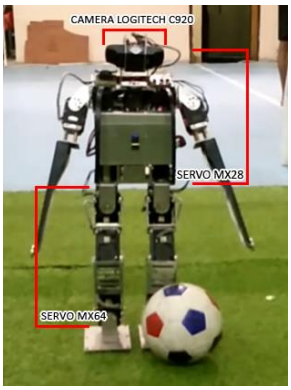
I. PENDAHULUAN

Dalam kontes robot sepak bola indonesia, setiap robot diharuskan memiliki kemampuan mandiri untuk bereaksi terhadap lingkungan sekitar, diantaranya adalah melakukan lokalisasi bola. Proses lokalisasi ini hanya diijinkan melalui kamera *digital* yang dianggap mewakili mata pada manusia. Bola yang digunakan dalam kontes robot adalah bola standar FIFA 1 dengan diameter 13cm berwarna 50% putih dan sisanya acak serta pola yang beragam. Kondisi lapangan sendiri dibuat dari rumput sintetis berwarna hijau dengan garis putih diatasnya serta gawang yang berwarna putih. Banyaknya objek yang memiliki warna serupa dengan bola membuat metode deteksi berbasis warna tidak mungkin digunakan. Begitupun dengan deteksi berbasis bentuk, Kondisi bola yang benar – benar bulat akan sulit ditemui saat kontes. Berdasarkan alasan diatas, Kami mengembangkan sebuah strategi lokalisasi bola dengan memanfaatkan fitur bola yang didapat dari proses ekstraksi.

II. METODOLOGI

A. *Platform Robot*

Robot sepak bola Krakatau dibangun dengan referensi desain DARWIN-OP. Robot memiliki 18 DOF, dimana 10 sendi tubuh bagian bawah digerakkan oleh *servo* MX-64T dan 8 sendi tubuh bagian atas dimotori oleh *servo* MX-28T. Dua sendi tubuh bagian atas digunakan sebagai basis peletakan kamera dengan fitur angguk (*Tilt*) dan geleng (*Pan*). Kondisi ekternal robot didapat melalui *sensor passive* berupa *digital* kamera Logitech c920. Robot ini sendiri di otaki oleh komputer dengan *processor* Intel Cherry Trail yang bekerja pada *clock* 1.8 Ghz.



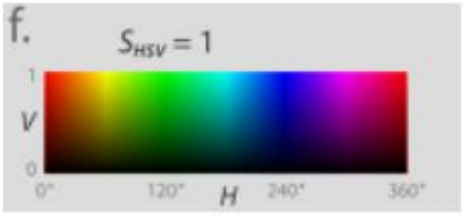
Gambar 1. Platform Robot Krakatau FC 2017

B. *Strategi Lokalisasi Bola*

Untuk menemukan bola dan secara bersamaan mengeliminasi *noise* dari objek yang memiliki kemiripan dengan bola, kami menggunakan strategi sebagai berikut :

1) Filter Warna

Meskipun deteksi bola tidak berbasis warna namun filter warna tetap harus dilakukan untuk memfilter area hijau yaitu lapangan dan mengeliminasi semua objek yang berada di luar lapangan. Selanjutkan proses deteksi akan terfokus di dalam area hasil filter. Proses ini menggunakan ruang warna HSV (*Hue*, *Saturation*, *Value*). Ruang warna ini dipilih karena dapat memisahkan antara warna dan cahaya. Seperti yang sudah kita tau, kondisi lingkungan dalam permainan, meskipun diatur sedemikian rupa, tetap menyisakan ruang yang dapat menyebabkan perbedaan warna yang diakibatkan oleh pengaruh pencahayaan.



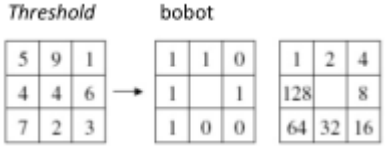
Gambar 2. Hue, Saturation, Value (HSV)

2) Deteksi Bola Multi Pola

Beragamnya pola bola yang digunakan pada KRSBI 2017 mengharuskan robot bisa mendeteksi bola dengan pola yang beragam. Untuk mengatasi masalah tersebut, kami menggunakan metode *learning*. Ada dua algoritma yang digunakan dalam metode ini yaitu :

1) Local Binary Pattern (LBP)

Local Binary Pattern (LBP) digunakan untuk ekstraksi fitur dari citra bola yang diperoleh dari sampel yang diberikan. Nantinya fitur ini akan digunakan saat proses *training*. LBP pertama kali diusulkan oleh Ojala et al. pada tahun 2002 untuk klasifikasi tekstur yang bersifat *rotation invariant* [1]. LBP didefinisikan sebagai perbandingan nilai biner piksel pada pusat citra dengan 8 nilai piksel disekelilingnya. Misal pada sebuah citra berukuran 3x3, nilai biner pada pusat citra dibandingkan dengan nilai sekelilingnya. Nilai sekelilingnya akan bernilai 1, jika nilai piksel pusat lebih kecil dan bernilai 0 jika nilai biner pusat lebih besar. Setelah itu, menyusun 8 nilai biner searah jarum jam atau sebaliknya dan merubah 8 bit biner kedalam nilai *decimal* untuk menggantikan nilai piksel pada pusat [2]. Perhitungan nilai LBP diilustrasikan dalam gambar berikut.



Gambar 3. Ilustrasi LBP

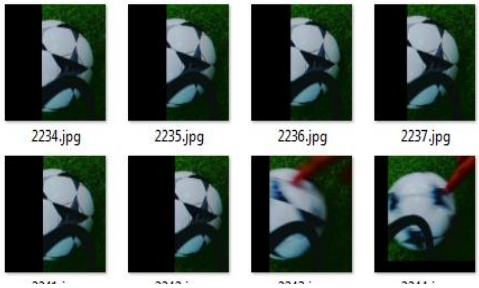
Secara matematis LBP dapat diformulasikan sebagai berikut :

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \quad (1)$$

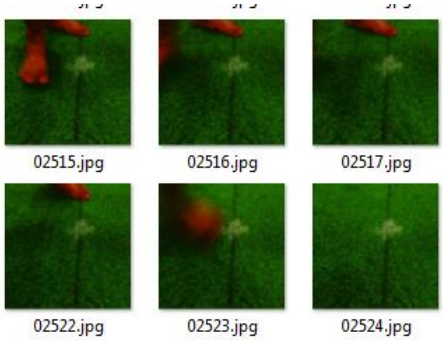
Dimana,

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (2)$$

dengan x c dan y c adalah koordinat pixel pusat, p adalah *circular sampling points*, P adalah jumlah *sampling points* atau *pixel* tetangga, gp adalah nilai *grayscale* dari p, gc adalah *pixel* pusat, dan s atau *sign* adalah fungsi *threshold*. Untuk keperluan klasifikasi nilai - nilai LBP selanjutnya direpresentasikan dalam bentuk *histogram*[1].



Gambar 4. Sampel Positif Citra Bola



Gambar 5. Sampel Negatif Citra Bola

2) AdaBoost

Setelah didapatkan fitur tekstur dari masing-masing citra sampel bola dengan alogoritma LBP, selanjutnya dilakukan proses pembelajaran untuk mendapatkan parameter yang dapat digunakan untuk membedakan objek bola dan bukan bola. Kami menggunakan algoritma adaboost untuk proses pembelajaran ini. AdaBoost singkatan dari *Adaptive Boosting*, merupakan algoritma machine learning yang tercipta dari kumpulan weak classifier yang dibentuk menjadi menjadi *strong classifier* [3]. Adaboost diperkenalkan oleh Freund dan Schapire pada tahun 1995 [4]. Dinamakan *adaptive* karena algoritma ini dapat menyesuaikan diri dengan data dan metode *classifier* yang lainnya. Dinamakan *boosting* karena algortima ini dapat mengurangi

kesalahan dari *weak classifier* dan meningkatkan akurasi dari setiap algoritma pembelajaran yang diberikan. Hal ini dikarenakan pada dasarnya AdaBoost dimunculkan untuk mampu mengurangi kesalahan dalam proses pembelajaran [5]. Untuk menentukan satu *classifier* terbaik diperlukan beberapa iterasi perhitungan seperti menghitung error dan update bobot. Hasil perolehan fitur yang memiliki *fitness function* paling optimal atau error yang paling

$$H(x) = sign \sum_{t=1}^{T} \alpha_t h_t(x) \quad (3)$$

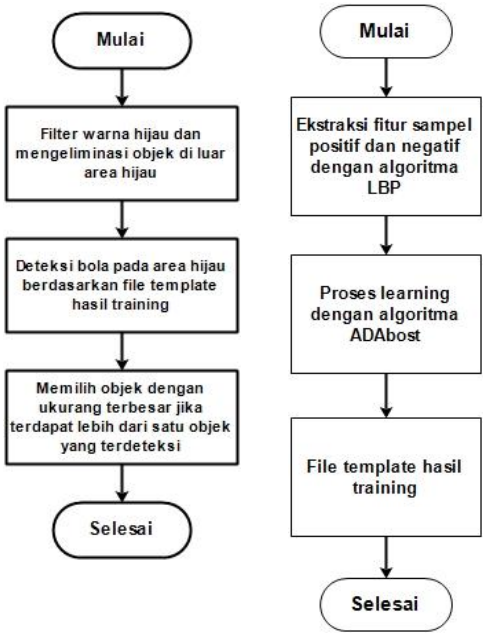
rendah maka dia yang akan masuk ke *classifier* pertama.

Hal ini dilakukan secara berulang selama iterasi tertentu hingga didapatkan *weak classifier* terpilih untuk menentukan keberadaan objek yang ditempatkan pada masing-masing stage yang telah terbentuk dengan metode AdaBoost.

3) *Confidence Scoring*

Kandidat yang layak dinyatakan sebagai bola adalah objek yang berada dalam area hijau dan memiliki fitur serupa dengan sampel yang diberikan saat *proses training*. Untuk menghindari deteksi objek lebih dari satu, kami memilih objek dengan ukuran terbesar yang dianggap paling dekat dengan posisi robot. Objek tersebut yang akan dianggap sebagai bola.

Secara umum, proses *training* sampel bola dan strategi deteksi bola dapat dilihat pada diagram blok berikut :



Gambar 6. Diagram blok proses training dan deteksi bola

III. HASIL DAN PEMBAHASAN

Strategi lokalisasi bola seperti dijelaskan diatas diuji secara *real-time* pada *platform* robot Krakatau. Pengujian dilakukan dengan kondisi pencahayaan dan lapangan yang beragam.Seperti yang ditunjukan pada gambar 6 jika robot dapat mendeteksi bola dan memilih bola yang terdekat dengan posisi robot. Selain menggunakan 2 bola, kami juga melakukan pengujian dengan meletakan bola di depan objek yang memiliki warna serupa.



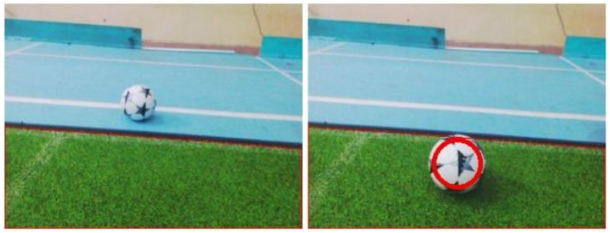
Gambar 7. Hasil deteksi lebih dari 1 bola

Gambar 7 menunjukan jika robot masih bisa mengenali bola meskipun bola berada di depan objek lain dengan warna serupa, hal ini menunjukan jika metode deteksi yang kami ajukan tidak terganggu objek bukan bola yang memiliki warna sama dengan bola.



Gambar 8. Hasil deteksi bola di depan objek dengan warna serupa

Terakhir kami melakukan pengujian dengan meletakan bola di luar *area* lapangan



Gambar 9. Robot tidak mengenali bola yang berada di luar lapangan

Secara efektif robot hanya mengenali bola yang berada di dalam lapangan. Dengan demikian robot secara otomatis akan mengeliminasi objek lain yang berada di luar lapangan meskipun objek tersebut memiliki fitur yang sama dengan bola.

IV. KESIMPULAN

Berdasarkan hasil pengujian dan pembahasan, strategi yang diajukan terbukti mampu mendeteksi bola, mengeliminasi gangguan, serta menentukan bola yang paling baik (jika terdapat lebih dari satu) untuk dieksekusi. Hasil tersebut juga dapat diraih dalam waktu yang cukup singkat sehingga robot tetap dapat merespon secara *real-time*.

DAFTAR PUSTAKA

[1] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns," pp. 1–35, 2002.

[2] T. Ahonen, S. Member, A. Hadid, and M. Pietik, "Face Description with Local Binary Patterns : Application to Face Recognition," no. June 2006, pp. 1–15, 2006.

[3] and X. L. C. Ruan, Q. Ruan, "Real Adaboost feature selection for Object Recognition," *IEEE 10th Int. Conf. Signal Process. Proceeding*, 2010.

[4] Yoav Freund and Robert E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, 1997.

[5] Yoav Freund and Robert E. Schapire, "A Short Introduction to Boosting," *J. Japanese Soc. Artif. Intell.*, 1999.

Rancang Bangun Robot Berkaki Empat Pada Robot "*Lizard*"

Meryana Eka Fajarwati*, Retno Rianto, Ahmad Rusyaid Mirta, Nurhidayat, Moh Rifki, dan Muh Rizki Fadillah
STMIK Adhi Guna Palu
Palu, Indonesia
meryanaeka96@gmail.com

***Abstrak*— Salah satu kekurangan yang dimiliki oleh robot berkaki enam adalah banyaknya servo yang digunakan. Semakin banyak servo yang digunakan maka semakin panjang algoritma dan semakin besar ukuran robot tersebut. Selain itu robot berkaki enam memiliki desain menyerupai laba-laba, sehingga robot seringkali mengalami *stuck* dan menyebabkan robot tidak bisa keluar dari ruangan. Pola gerak yang digunakan pada robot berkaki empat adalah *trot gait*. *Trot* adalah pola gerak yang menggunakan dua kaki pada posisi diagonal sebagai kaki transisi, sedangkan dua kaki yang lainnya sebagai kaki penopang, untuk menjaga keseimbangan selagi kaki transisi sedang melangkah. Dari hasil penelitian yang telah dilakukan menunjukkan bahwa sistem gerak robot berkaki empat telah mampu menyeimbangkan badan pada saat maju, belok kanan dan belok kiri.**

Kata kunci—component, Trot gait, servo, robot.

I. PENDAHULUAN

Kontes Robot Indonesia (KRI) merupakan suatu ajang kompetisi yang diselenggarakan setiap tahun oleh Direktorat Jenderal Pembelajaran dan Kemahasiswaan, Kementrian Riset, Teknologi dan Pendidikan Tinggi (DIKTI). Dalam kontes itu terdapat salah satu devisi yaitu devisi Kontes Robot Pemadam Api Indonesia (KRPAI). Dalam devisi tersebut terdapat tiga jenis robot berkaki, yaitu robot dengan jumlah kaki dua (*bipedal*), robot berkaki empat (*quadpod*), dan robot berkaki enam (*hexapod*)[1].

Dari berbagai macam desain, salah satunya yang paling menarik adalah robot berkaki enam (*hexapod*) karena menyerupai laba-laba tetapi mempunyai kekurangan yaitu banyaknya servo yang digunakan. Semakin banyak servo yang digunakan maka semakin panjang algoritma dan semakin besar ukuran robot tersebut. Robot berkaki enam sering kali mengalami *stuck* dan menyebabkan robot tidak bisa keluar dari ruangan.

Maka dari itu dirancanglah sebuah desain yang mudah dan sederhana yaitu robot kaki empat (*quadpod*). Robot berkaki empat memiliki bentuk segiempat sama sisi sehingga jalannya lebih akurat. Dengan menggunakan desain ini jika robot berhadapan dengan dinding, robot tak perlu lagi memutar badan seperti robot berkaki enam. Jumlah servo yang digunakan robot "Lizard" berjumlah delapan servo. Jumlah ini termasuk lebih ekonomis karna mengingat harga servo yang cukup mahal.

II. DASAR TEORI

A. *Robot Pemadam Api*

Robot pemadam api adalah robot yang mempunyai misi mencari dan memadamkan api di arena lapangan berbentuk simulasi interior suatu rumah. Ada dua jenis robot pemadam api yaitu robot pemadam api divisi berkaki dan pemadam api divisi beroda. Pada kontes robot pemadam api indonesia (KRPAI) divisi berkaki, robot yang diperlombakan harus mampu bergerak menyusuri dinding arena perlombaan yang berbentuk lorong dan ruangan untuk melakukan tugasnya memadamkan api dari sebuah lilin yang diletakkan pada salah satu ruang[1].

B. *Robot berkaki empat (quadruped)*

Jumlah servo yang dipakai oleh robot Lizard berjumlah delapan servo yang masing masing kakinya memiliki dua servo. Servo atas berfungsi sebagai penggerak pangkal kaki untuk menggeserkan kaki kedepan dan kebelakang. Sedangkan servo bawah berfungsi untuk menggerakkan lutut atau ruas bawah robot dimana kaki robot akan naik dan turun untuk berpijak pada landasannya[2].

C. *Servo*

Salah satu contoh penggerak robot adalah servo. Motor servo adalah motor dengan sistem *closed feedback* di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor, serangkaian *gear*, *potensiometer*, dan rangkaian kontrol[3].



Gambar 1. Bagian motor servo

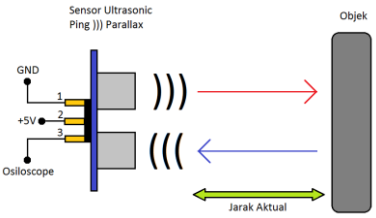
D. Sensor Jarak

SRF04 adalah modul yang berisi *transmitter* dan *receiver* ultrasonik. SRF04 ini dapat memberikan informasi jarak dari kisaran 3 cm hingga 3 m, serta dapat mengukur benda dengan diameter 3 cm pada jarak kurang dari 2 m[3].



Gambar 2. Sensor Jarak SRF04

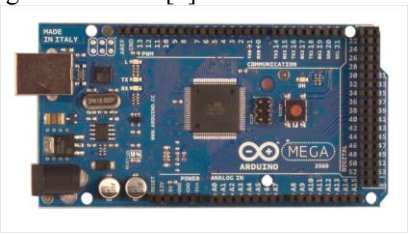
Sensor SRF04 mendekteksi jarak objek dengan cara memancarkan gelombang ultrasonik (40kHz) selama waktu pemancaran kemudian mendekteksi pantulannya. Sensor PING memancarkan gelombang ultrasonik sesuai dengan pulsa triger dari mikrokontroler sebagai pengendali. Gambar 2 menunjukkan timing diagram dari sensor jarak.



Gambar 3. Ilustrasi cara kerja sensor jarak

E. Arduino

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*. Arduino di rancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardwarenya*-nya memiliki prosessor atmel AVR dan *software*-nya memiliki bahasa pemograman sendiri[4].

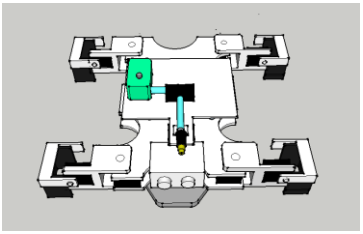


Gambar 5. *Arduino Mega*

III. DESAIN SISTEM

A. Desain Robot

Desain robot berkaki empat "Lizard" ditunjukkan pada gambar 4 Model yang dimiliki lizard adalah segiempat sama sisi sehingga empat kaki yang dimiliki lizard berada di setiap sisi robot. Sensor jarak SRF04 di letakkan di setiap sudut robot, sensor ini berfungsi sebagai mata untuk melihat dinding pada arena.

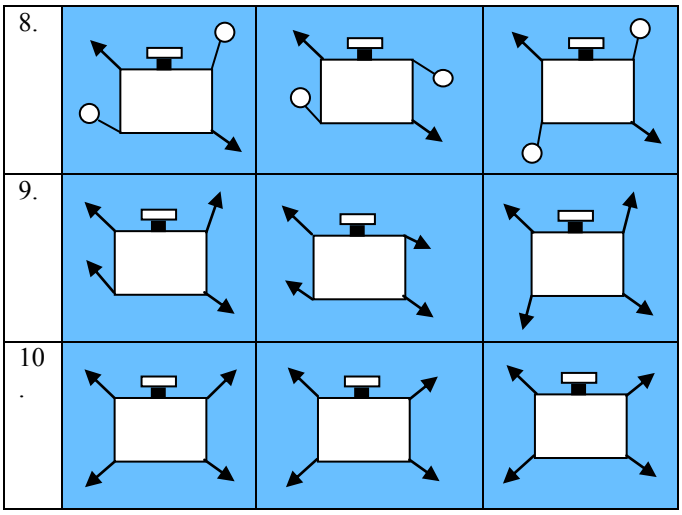


Gambar 4. Desain kaki robot Lizard

B. Kinematik gerak robot Lizard

TABEL I PERGERAKKAN ROBOT BERKAKI EMPAT

No	Maju	Rotasi Belok Kanan	Rotasi Belok Kiri
1.	Model robot maju	Model robot belok kanan	Model robot belok kiri
2.	Posisi awal		
3.			
4.			
5.			
6.			
7.			



Keterangan :
- Kaki menempel ke lantai
- Kaki digeser
- Kaki diangkat

Penjelasan :

1. Maju
   a. Perilaku 1 (Merah) : kanan belakang dan kiri depan ***aktif***. Kaki kanan belakang diangkat dan kaki kiri depan diangkat.
   b. perilaku 2 (Merah) : kanan belakang dan kiri depan ***aktif***. Kaki kanan belakang digeser ke depan dan kaki kiri depan diserong depan.
   c. perilaku 3 (Merah) : kanan belakang dan kiri depan ***aktif***. Setelah kaki kanan digeser, maka akan diletakkan dilantai.
   d. perilaku 4 (Merah) : kanan belakang dan kaki kiri ***aktif***. setelah kaki menempel dilantai, secara spontan kedua kaki di gerakkan kebelakang sampai membentuk ke posisi awal.
   e. Perilaku 5 (Biru) : kiri belakang dan kanan depan ***aktif***. Kaki kiri belakang diangkat dan kaki kanan depan diangkat.
   f. perilaku 6 (Biru) : kiri belakang dan kanan depan ***aktif***. Kaki kiri belakang digeser ke depan dan kaki kanan depan digeser ke depan.
   g. perilaku 7 (Biru) : kiri belakang dan kanan depan ***aktif***. setelah kaki digeser, maka akan diletakkan dilantai.
   h. perilaku 8 (Biru) : kiri belakang dan kaki kanan ***aktif***. setelah kaki menempel dilantai, secara spontan kedua kaki di gerakkan kebelakang sampai membentuk ke posisi awal.

2. Rotasi Belok Kanan
   a. Perilaku 1 (Merah) : kanan belakang dan kiri depan ***aktif***. Kaki kanan belakang diangkat dan kaki kiri depan diangkat.
   b. perilaku 2 (Merah) : kanan belakang dan kiri depan ***aktif***. Kaki kanan belakang digeser ke belakang dan kaki kiri depan diserong ke depan.
   c. perilaku 3 (Merah) : kanan belakang dan kiri depan ***aktif***. Setelah kaki digeser, maka akan diletakkan dilantai.
   d. perilaku 4 (Merah) : kanan belakang dan kaki kiri ***aktif***. setelah kaki menempel dilantai, secara spontan kedua kaki di gerakkan kebelakang sampai membentuk ke posisi awal.
   e. Perilaku 5 (Biru) : kiri belakang dan kanan depan ***aktif***. Kaki kiri belakang diangkat dan kaki kanan depan diangkat.
   f. perilaku 6 (Biru) : kiri belakang dan kanan depan ***aktif***. Kaki kiri belakang digeser ke belakang dan kaki kanan depan diserong ke depan.
   g. perilaku 7 (Biru) : kiri belakang dan kanan depan ***aktif***. setelah kaki digeser, maka akan diletakkan dilantai.
   h. perilaku 8 (Biru) : kiri belakang dan kaki kanan ***aktif***. setelah kaki menempel dilantai, secara spontan kedua kaki di gerakkan kebelakang sampai membentuk ke posisi awal.

3. Rotasi Belok Kiri
   a. Perilaku 1 (Merah) : kanan belakang dan kiri depan ***aktif***. Kaki kanan belakang diangkat dan kaki kiri depan diangkat.
   b. perilaku 2 (Merah) : kanan belakang dan kiri depan ***aktif***. Kaki kanan belakang digeser ke depan dan kaki kiri depan diserong belakang.
   c. perilaku 3 (Merah) : kanan belakang dan kiri depan ***aktif***. Setelah kaki kanan digeser, maka akan diletakkan dilantai.
   d. perilaku 4 (Merah) : kanan belakang dan kaki kiri ***aktif***. setelah kaki menempel dilantai, secara spontan kedua kaki di gerakkan kebelakang sampai membentuk ke posisi awal.
   e. Perilaku 5 (Biru) : kiri belakang dan kanan depan ***aktif***. Kaki kiri belakang diangkat dan kaki kanan depan diangkat.
   f. perilaku 6 (Biru) : kiri belakang dan kanan depan ***aktif***. Kaki kiri belakang digeser ke belakang dan kaki kanan depan digeser ke depan.
   g. perilaku 7 (Biru) : kiri belakang dan kanan depan ***aktif***. setelah kaki digeser, maka akan diletakkan dilantai.
   h. perilaku 8 (Biru) : kiri belakang dan kaki kanan ***aktif***. setelah kaki menempel dilantai, secara spontan kedua kaki di gerakkan kebelakang sampai membentuk ke posisi awal.

IV. HASIL

A. Hasil Pengujian Gerak Robot

Pengujian gerak robot mengacu pada hitungan derajat masing-masing servo dengan uraian posisi yang berubah-ubah sesuai dengan derajat yang ditentukan. Berikut adalah hasil percobaan dengan menghitung derajat masing – masing servo.

TABEL II. HASIL PENGUJIAN DERAJAT SERVO SAAT ROBOT MAJU

No	Uraian	Derajat Pergerakan Servo							
		S1	S2	S3	S4	S5	S6	S7	S8
1.	Posisi Awal	95°	140°	130°	110°	95°	50°	150°	60°
2.	Kaki depan kiri & belakang kanan terangkat	125°	140°	130°	110°	95°	50°	120°	60°
3.	Kaki depan kiri & belakang kanan geser kedepan	125°	176°	130°	110°	95°	50°	120°	20°
4.	Kaki depan kiri & belakang kanan turun	95°	176°	130°	110°	95°	50°	150°	20°
5.	Kaki depan kiri & belakang kanan geser kebelakang	95°	140°	130°	110°	95°	50°	150°	60°
6.	Kaki kiri belakang & kanan depan terangkat	95°	140°	160°	110°	68°	50°	150°	60°
7.	Kaki kiri belakang & kanan depan geser kedepan	95°	140°	160°	152°	68°	5°	150°	60°
8.	Kaki kiri belakang & kanan depan turun	95°	140°	130°	152°	95°	5°	150°	60°
9.	Kaki kiri belakang & kanan depan geser ke belakang	95°	140°	130°	110°	95°	50°	150°	60°

TABEL III. HASIL PENGUJIAN DERAJAT SERVO SAAT ROBOT BELOK KANAN

No	Uraian	S1	S2	S3	S4	S5	S6	S7	S8
1.	Posisi Awal	95°	140°	130°	110°	95°	50°	150°	60°
2.	Kaki depan kiri & belakang kanan terangkat	125°	140°	130°	110°	95°	50°	120°	60°
3.	Kaki depan kiri & belakang kanan geser kedepan	125°	167°	130°	110°	95°	50°	120°	60°
4.	Kaki depan kiri & belakang kanan turun	95°	167°	130°	110°	95°	50°	150°	60°
5.	Kaki depan kiri & belakang kanan geser kebelakang	95°	140°	130°	110°	95°	50°	150°	22°
6.	Kaki kiri belakang & kanan depan terangkat	95°	140°	160°	110°	68°	50°	150°	22°
7.	Kaki kiri belakang & kanan depan geser kedepan	95°	140°	160°	150°	68°	50°	150°	22°
8.	Kaki kiri belakang & kanan depan turun	95°	140°	130°	150°	95°	50°	150°	22°
9.	Kaki kiri belakang & kanan depan geser ke belakang	95°	140°	130°	110°	95°	50°	150°	22°

TABEL IV. HASIL PENGUJIAN DERAJAT SERVO SAAT ROBOT BELOK KIRI

No	Uraian	S1	S2	S3	S4	S5	S6	S7	S8
1.	Posisi Awal	95°	140°	130°	110°	95°	50°	150°	60°
2.	Kaki depan kiri & belakang kanan terangkat	125°	140°	130°	110°	95°	50°	120°	60°
3.	Kaki depan kiri & belakang kanan geser kedepan	125°	140°	130°	110°	95°	50°	120°	22°
4.	Kaki depan kiri & belakang kanan turun	95°	167°	130°	110°	95°	50°	150°	60°
5.	Kaki depan kiri & belakang kanan geser kebelakang	95°	167°	130°	110°	95°	50°	150°	60°
6.	Kaki kiri belakang & kanan depan terangkat	95°	140°	160°	110°	68°	50°	150°	22°
7.	Kaki kiri belakang & kanan depan geser kedepan	95°	140°	160°	110°	68°	5°	150°	22°
8.	Kaki kiri belakang & kanan depan turun	95°	140°	130°	150°	95°	50°	150°	22°
9.	Kaki kiri belakang & kanan depan geser ke belakang	95°	140°	130°	150°	95°	50°	150°	22°

B. Bentuk Fisik Robot Lizard

Setelah setiap komponen sudah diuji, dilakukan integrasi dari sistem. Pengujian dilakukan dengan memberikan aksi pada sensor yang telah terpasang pada robot. Hasil yang diperoleh adalah robot dapat berbelok kanan dan kiri dengan baik, untuk aksi robot dapat dilihat pada gambar 7 dan 8 .



(a) (b)

Gambar 6. Robot Lizard Tampak (a) Samping Kanan Dan (b) Samping Kiri

TABEL V. HASIL PENGUJIAN ROBOT BERDASARKAN PROGRAM YANG TELAH DIBERIKAN

NO	GAMBAR	CODE PROGRAM

1	 Robot maju	void maju ()
		{
		S1.write(125); //naik
		S7.write(120);
		delay(100);
		
		S2.write(176); //geser kedepan 174 // ka
		S8.write(20); //21 // kaki bagian kanan
		delay(70);
		
		S1.write(95); //Turun
		S7.write(150);
		delay(100);

2	 Robot belok kanan	`void belokkanan()`
		`{`
		`S1.write(125); //naik`
		`S7.write(120);`
		`delay(100);`
		
		`S2.write(167); //geser kedepan`
		`S8.write(60); //20 // kaki bagi`
		`delay(70);`
		
		`S1.write(95); //Turun`
		`S7.write(150);`
		`delay(100);`

3	 Robot belok kiri	`void belokkiri()`
		`{`
		`S1.write(125); //naik`
		`S7.write(120);`
		`delay(100);`
		
		`S2.write(140); //geser kedepan 174 // k`
		`S8.write(22); //20 // kaki bagian kanar`
		`delay(70);`
		
		`S1.write(95); //Turun`
		`S7.write(150);`
		`delay(100);`



Gambar 7. Robot Belok Kanan



Gambar 8. Robot Belok Kiri

V. PENUTUP

Berdasarkan hasil perancangan, pengujian dan analisis yang dilakukan dapat diambil kesimpulan dengan Menggunakan teknik *trot gait* yang hanya menggunakan empat kaki dimana tiap kakinya menggunakan dua servo maka di dapatkan hasil yang cukup maksimal. Robot dapat berbelok kanan dan belok kiri dengan seimbang sesuai dengan algoritma program yang telah dibuat. Sistem gerak robot yang menggunakan teknik *trot gait* dapat menyeimbangkan diri dan mampu bernavigasi ke ruangan-ruangan yang telah disediakan dan dapat menyelesaikan misi yang diberikan.

REFERENSI

[1] Wawan Purnama, Regina Puspitasari, Erik Haritman, "Perancangan algoritma dan program robot cerdas pemadam api 2013 divisi berkaki", Universitas Pendidikan Indonesia, 2014.

[2] Dwiant Ramady Priantono, 2011, Algoritma robot kaki empat, https://designautomation.wordpress.com/2011/02/18/algoritma-robot-kaki-empat/, diakses tgl 6 juni 2017.

[3] Arif Ainur Rafiq, Ary Sulistyo Utomo, Syuryadin Baskoro, "rancang bangun robot berkaki 6 menelusuri dinding menggunakan sensor ultrasonik SRF 04 berbasis MIKROKONTROLER Atmega128", Politeknik Cilacap, 2012. 1

[4] Musa Payung, Bayu Saputera "Robot sepakbola dengan pencarian dari pengolahan citra openCV dan arduino", 2016. 2

[5] Dwi Indro Widodo, Iwan Setiawan, S.T, M, Budi Setiyono, S.T,M.T, "Robot berkaki empat pendekteksi cahaya dan penghindar rintangan", Universitas Diponegoro, 2011. 3

Penerapan Kontrol Berbasis Laptop Pada Robot Sepak Bola Beroda Indonesia (KRSBI)

Tria Rahmadani*, Rahmawati, Adrin Thamrin, Rama Rifaldi Larengi, Agung Kurniawan, dan Jabal Rahman
Program Studi Teknik Informatika
STMIK Adhi Guna Palu
Palu, Indonesia
Rhanyongk39@gmail.com

***Abstrak*—Makalah ini berisi tentang robot sepak bola beroda yang menggunakan laptop sebagai basis kontrol robot. Terdapat kelemahan jika pada robot hanya mengandalkan mikorontroler sebagai pemroses utama. Apalagi jika diperlukan pengolahan data citra pada robot, yang akhirnya berdampak pada lambatnya dan tidak stabilnya pergerakan robot karna tidak mampu mengolah data yang banyak, salah satunya ialah digunakannya aplikasi pengelolahan citra. Sehingga laptop menjadi salah satu solusi yang di gunakan di robot sepak bola beroda untuk pengolahan data citra. Dalam robot sepak bola beroda metode yang digunakan yaitu *high level kontrol* dan *low level kontrol*. Dari proses penelitian yang dilakukan didapatkan hasil yang baik dimana laptop sebagai pemroses pada *high level control* dan dapat berkomunikasi dengan baik ke mikrokontroller di bagian *low level kontrol*.**

Kata kunci—high level kontrol, low level kontrol, laptop, mikrokontroller.

I. PENDAHULUAN

Teknologi robotika yang perkembangannya semakin pesat menuntut setiap negara berkembang untuk bisa mengikuti arus perkembangan tersebut. Di Indonesia khusunya robot sudah sangat populer dan dalam kalangan pelajar dan mahasiswa menjadi kegiatan tahunan yang dikompetisikan. Contohnya adalah Kontes Robot Sepak Bola (KRSBI). Pada robot sepak bola beroda, robot di tuntut untuk bisa bermain layaknya seperti manusia. Dimana dalam permainan bola, robot dituntut agar memiliki respon cepat terhadap perubahan dilingkungan bermainnya. Sehingga diperlukan pengolahan atau pemrosesan data yang cepat dan akurat.

Laptop sebagai basis kontrol robot menjadi menarik untuk dikembangkan. Hal ini dikarenakan besarnya program pada suatu robot sehingga mempengaruhi kinerja robot. Jika hanya mengandalkan mikrokontroller sebagai basis pengolahan pada robot dan dituntut untuk mengolah data citra sehingga fitur yang digunakan kurang lengkap karena tidak didukung adanya aplikasi. Maka dibutuhkan sebuah laptop yang memiliki kemudahan kepraktisan dan fitur yang lengkap.

Dengan menggunakan sebuah laptop, algoritma yang kompleks sepenuhnya akan diatasi oleh laptop, sedangkan untuk navigasinya (pergerakkan robot) dapat menggunakan sebuah mikrokontroler beserta *driver* motor. Dalam hal ini, mikrokontroler hanya akan menerima sinyal-sinyal navigasi saja, tanpa harus terbebani dengan algoritma pengolahan yang kompleks [1].

Pada makalah ini, akan menguraikan sebuah robot sederhana berbasiskan *high level kontrol* dan *low level kontrol*. Dimana terdapat dua bagian dalam kontrol robot tersebut. Pada *high level* kontrol akan digunakan *laptop* sebagai basis pengolahan data citra terhadap lingkunagan (lapangan) dimana robot bermain dan pada *low level* kontrol akan digunakan mikrokontroler sebagai kontrol terhadap *level hardware*. Digunakan MATLAB sebagai *software* utama dalam pengolahan citra. Sebuah sensor citra, yaitu webcam akan digunakan sebagai 'mata' robot. Gambar yang ditangkap oleh webcam selanjutnya akan diolah menggunakan MATLAB. Hasil dari pengolahan data citra akan diumpankan ke Arduino melalui komunikasi *serial*. Arduino selanjutnya akan mengontrol *driver motor* untuk menghasilkan pergerakkan robot sesuai perintah [1].

II. LANDASAN TEORI

A. *High Level Kontrol Dan Low Level Kontrol*

High *level* kontrol merupakan teknologi meliputi perangkat keras ataupun algoritma tingkat tinggi sedangkan *low level* kontrol merupakan teknologi meliputi perangkat keras ataupun algoritma tingkat rendah. Dalam lingkungan kerja dengan contoh robot, tingkat komplesitas *high level* kontrol bisa dikatakan rumit karena tingkat pengenalan akan lingkungan dapat dikatakan menantang. Sedangkan *low level* kontrol cakupan kerjanya tidak begitu rumit karena tingkat pengenalan akan lingkungan tidak begitu sulit.

B. *Webcam*

Webcam adalah sebuah periferal yang berupa seperti *camera* dan memiliki citra/gambar serta bisa menangkap suara/audio dan dikendalikan oleh komputer atau sebuah jaringan. Gambar yang ditampilkan lewat webcam akan muncul dilayar monitor serta terhubung menggunakan *port* webcam ke komputer atau jaringan [2].



Gambar 1. webcam

C. Laptop

Laptop adalah sebuah komputer yang dengan mudah bisa kita bawa kemana-mana karena ukurannya lebih kecil dari PC (Personal Computer). Laptop termasuk layar, keyboard, dan *trackpad* atau *trackball*, yang berfungsi sebagai mouse. Laptop mempunyai baterai yang bisa beroperasi tanpa terhubung ke listrik. Laptop juga termasuk *adaptor* daya yang menggunakan daya dari stop-kontak dan mengisi kembali baterai. Kebanyakan laptop juga mencakup beberapa *I/O port*, seperti *USB port*, yang memungkinkan *keyboard standar* dan *mouse* untuk digunakan dengan laptop. Laptop *modern* sering termasuk *adaptor* jaringan *nirkabel/wireless* [3].



Gambar 2. NetBook

D. Arduino

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari *Wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware*nya memiliki prosesor Atmel AVR dan *software*nya memiliki bahasa pemrograman sendiri. Saat ini Arduino sangat populer di seluruh dunia. Banyak pemula yang belajar mengenal robotika dan elektronika lewat Arduino karena mudah dipelajari. Tapi tidak hanya pemula, para *hobbyist* atau profesional pun ikut senang mengembangkan aplikasi elektronik menggunakan Arduino. Bahasa yang dipakai dalam Arduino bukan assembler yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (libraries) Arduino [4].



Gambar 3. Arduino

E. Motor Dc

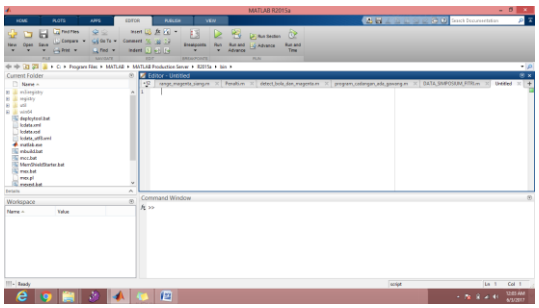
Motor DC (*Direct Current*) adalah peralatan elektromagnetik dasar yang berfungsi untuk mengubah tenaga listrik menjadi tenaga mekanik yang desain awalnya diperkenalkan oleh Michael faraday lebih dari seabad yang lalu (E. Pitowarno, 2006). Motor DC dikendalikan dengan menentukan arah dan kecepatan putarnya. Arah putaran motor DC adalah searah dengan arah putaran jarum jam (*Clock Wise*/CW) atau berlawanan arah dengan arah putaran jarum jam (*Counter Clock Wise*/CCW), yang bergantung dari hubungan kutub yang diberikan pada motor DC. Kecepatan putar motor DC diatur dengan besarnya arus yang diberikan [5].

F. Pengolahan citra

Pengolahan citra (*image processing*) sedikit berbeda dengan pengertian mesin visual (*machine vision*), meskipun keduanya seolah-olah dapat digunakan dengan maksud yang sama. Sedangkan terminologi mesin *visual* digunakan bila data hasil pengolahan citra langsung diterjemahkan dalam bentuk lain, misalnya grafik yang siap interpretasikan untuk tujuan tertentu, gerak peralatan atau bagian dari peralatan mekanis, atau aksi lainnya yang berarti bukan merupakan citra lain [6].

G. Aplikasi matlab sebagai pengolah citra

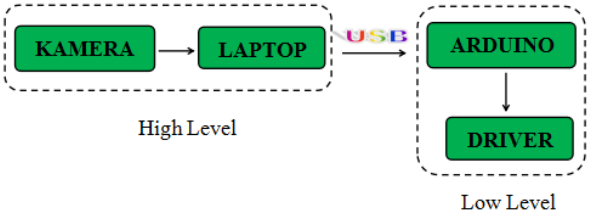
MATLAB (*Matrix Laboratory*) adalah sebuah program untuk analisis dan komputasi numerik dan merupakan suatu bahasa pemrograman matematika lanjutan yang dibentuk dengan dasar pemikiran menggunkan sifat dan bentuk matriks. Pada awalnya, program ini merupakan *interface* untuk koleksi rutin-rutin numerik dari proyek LINPACK dan EISPACK, dan dikembangkan menggunkan bahasa FORTRAN namun sekarang merupakan produk komersial dari perusahaan Mathworks, Inc.yang dalam perkembangan selanjutnya dikembangkan menggunakan bahasa C++ dan assembler (utamanya untuk fungsi-fungsi dasar MATLAB) [7].



Gambar 4. Matlab

III. METODE PENELITIAN

Pada proses ini, akan dilakukan berdasarkan skema diagram sesuai gambar 5. Dimana terdapat dua bagian yang akan di teliti yakni pada bagian *high level kontrol* dan *low level kontrol*. Pada *high level kontrol*, kamera sebagai sensor eksternal yang berfungsi menangkap objek dilingkungan permainan robot, selanjutnya diolah oleh laptop. Sedangkan pada *low level kontrol*, untuk menerjemahkan perintah ke robot dalam bentuk aksi robot maka digunakan mikrokontroler.



Gambar 5. Diagram *High Level Kontrol* dan *Low Level Kontrol*

Untuk fokus penelitian ini akan di arahkan pada bagaimana perpaduan *high level* dan *low level* kontrol yang melibatkan dua pemrosesan yakni laptop dan mikrokontroler. Selain sebagai satu kesatuan dalam kontrol robot, tinjauan dalam penlitian ini juga akan dititik beratkan pada bagaimana kedua pemroses ini berkomunikasi dalam mengirimkan data dari *high level* ke *low level* dan pada akhirnya akan diketahui respon terhadap aksi robot dilapangan.

A. High Level Kontrol

1. Kamera
   Kamera pada robot untuk menangkap gambar seperti objek bola ataupun robot lawan dan hasil tersebut akan di kirim ke laptop

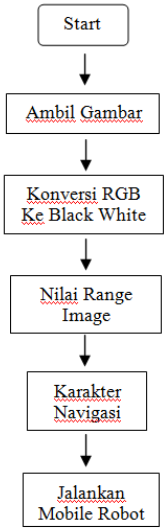
2. Laptop
   Laptop sebagai kontrol *image processing* pada data gambar yang diterima dari kamera *webcam* kemudian mengirimkan data melalui USB ke arduino

B. Low Level Kontrol

1. Arduino
   Arduino pada robot sebagai master dari *image processing* yang mana arduino akan menerima karakter dari laptop tersebut

2. Driver
   Pada robot driver sebagai navigasi penggerak motor DC yang mana data tersebut akan di transfer lewat arduino

C. Flowchart dengan sistem kerja



Gambar 6. Sistem kerja *High Level Kontrol* dan *Low Level Konrol*

Berdasarkan gambar di atas tentang flowchart dapat dijelaskan sebagai berikut :

1. Ambil gambar .

   Mengambil *data image* yang nantinya hasil tersebut akan dikirim ke laptop.

2. Konversi *RGB* ke *black white*.

   Setelah data dihasilkan dari kamera maka data tersebut akan diproses dilaptop.

3. Nilai range image.

   Nilai range image merupakan nilai rentang yang telah ditetapkan atau nilai acuan untuk agar dapat robot bergerak, pergerakan yang dihasilkan sesuai dengan nilai range. Hasil nilai range yang telah di tetapkan akan dikirim ke arduino.

4. Karakter navigasi.

   Karekter navigsi ialah hasil karakter yang di dapatkan dari perintah nilai range tersebut.

5. Jalankan mobile robot.

   Robot akan bergerak sesuai dengan perintah arduino.

IV. HASIL PENELITIAN

Pada hasil penelitian diperoleh dua bagian dari focus utama penelitian yang dilakukan yakni *high level kontrol* yang menggunakan laptop sebagai pemroses dan bagian *low level kontrol* yang menggunakan mikrokontroler.

Merek laptop yang digunakan ialah SONY VAIO model SVF142C1WW dan adapun spesifikasi laptop tersebut adalah sebagai berikut :

a. Processor : intel(2) core (TM) 13-3217U CPU@ 1,80GHz 1,80GHz

b. RAM : 6.00GB (5.88 GB usable)

c. System type : 64-bit operating system, x64-based processor

Pada tahap pengujian diperoleh hasil melalui proses yang dilakukan yakni menscaning objek melalui kamera webcam untuk menentukan nilai koordinat XY. Dari gambar yang ditangkap berupa objek bola, didapatkan titik koordinat untuk penentuan masing-masing pergerakan robot. Berikut gambar scan objek berupa bola.

a. Laptop (High Level)



Gambar 7. Nilai Koordinat XY Bola untuk Maju



Gambar 8. Nilai Koordinat XY Bola Belok Kiri



Gambar 9. Nilai Koordinat XY Bola Belok Kanan

b. Arduino (Low Level)

Pada *low level control*, mikrokontroler sebagai pemroses akan menerima data-data dari hasil pemrosesan laptop dalam mengolah citra berupa karakter yang sudah ditentukan pada pemograman. Karakter yang terkirim ke mikrokontroler didasarkan nilai koordinat citra bola pada titik X. Berdasarakan perbedaan nilai koordinat inilah yang membedakan karakter yang dikirim ke mikrokontroler.

TABEL 1. LOW LEVEL (ARDUINO)

No	Nilai Range	Karakter	Navigasi
1	X < 169	'3'	Belok Kiri
2	X > 170 && X < 550	'1'	Maju
3	X > 551	'2'	Belok Kanan

Berdasarkan tabel 1 di atas dapat dijelaskan sebagai berikut:

1. Nilai *range* yang dimaksud ialah sebagai penentuan posisi bola dimana nilai X < 169 dan untuk karakternya sendiri yakni karakter 3 dimana nilai range tersebut menunjukkan algoritma navigasi untuk bergerak ke arak belok kiri.
2. Penentuan posisi bola nilai range X > 170 && X < 550 sebagai nilai tengah untuk karakter ke 1 sehingga navigasinya akan bergerak arah maju
3. Pada nilai range X > 551 di mana posisi bola berada disebelah kanan dan karater 2 sebagai algoritma navigasinya bergerak ke arah belok kanan

V. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan diperoleh, bahwa *high level kontrol* menggunakan laptop sebagai pemrosesan utama khusus sebagai pengolah data citra dapat bekerja dengan baik, yang mana *low level kontrol* dapat mengontrol navigasi sesuai dengan intruksi dari high level kontrol melalui karakter-karakter yang dikirim. Penggunaan laptop dapat menjadi pilihan tepat dalam membangun robot sepak bola beroda untuk mengolah citra dari hasil tangkapan kamera.

REFERENSI

[1] Kusuma wardana, "membuat laptop based robot menggunakan matlab dan arduino,".

[2] http://purmaiyasadeopy.blogspot.co.id/2016/11/pengertian-web-camera-kamera-web-secara.html, Deopy purmaiyasa, "pengertian wab camera(kamera web)secara spesifik".

[3] Anonymous, "Pengertian laptop", Politeknik Negeri Sriwijaya, Palembang

[4] https://ariefeeiiggeennblog.wordpress.com/2014/02/07/pengertian-fungsi-dan-kegunaan-arduino/, Ariefeeiiggeenn "pengertian fungsi dan kegunaan arduino"

[5] Qory Hidayati, "Pengaturan Kecepatan Motor DC dengan Menggunakan Mikrokontroler Atmega 8535"

[6] Irriwad Putri, "Pengembangan program pengolahan citra real-time untuk deteksi rintangan pada traktor tanpa awak", institut pertanian", Bogor, Bogor 2013.

[7] Anonymous, "Pengantar untuk Pemrograman MATLAB", institut pertanian Bogor, Bogor 2013.

Penggunaan Bahasa Pemrograman Python Sebagai Pusat Kendali Pada Robot 10-D

Fitri*, Kiki Reski R, Ady Rahmansyah, dan Wahyuni Darwin
STMIK Adhi Guna
Palu, Indonesia
Email : fitrilad01@gmail.com

***Abstrak*— *Referee Box* merupakan sebuah aplikasi yang digunakan dalam pertandingan Kontes Robot Sepak Bola Indonesia (KRSBI). Aplikasi ini berfungsi untuk memberi aba-aba kepada pemain layaknya seorang wasit dalam pertandingan sepak bola. Tetapi banyak tim yang masih mengalami kesulitan dalam menggunakan referee box. Untuk menggunakan *referee box* dibutuhkan sebuah *base station* yang berfungsi sebagai jembatan untuk meneruskan data ke robot. Banyak team KRSBI yang masih mengalami kesulitan dalam mengintegrasi antara perintah yang dikirim oleh referee box dan pergerakan robot dilapangan. Hal ini dapat diatasi dengan menggunakan python sebagai base station. Tidak hanya dapat menerima perintah dari *referee box, python* juga dapat memilih robot mana yang akan melakukan pergerakan mengambil bola dll. Agar tidak terjadi tabrakan antar robot sesama team.**

Kata kunci*—*Referee Box, Base Station, Bahasa Pemrograman Python.

I. PENDAHULUAN

Kontes robot indonesia merupakan sebuah kegiatan yang di ikuti oleh semua penggiat robotika dari kalangan mahasiswa seluruh indonesia. Dalam kontes ini terdapat beberapa divisi lomba, salah satu diantaranya Kontes Robot Sepak Bola Indonesia (KRSBI). Pada kontes robot tersebut, terdapat *referee box* yang digunakan untuk mengatur jalannya pertandingan. *Referee box* merupakan sebuah *game controller* yang berfungsi seperti halnya seorang wasit dalam sebuah pertandingan sepak bola. *Referee Box* dapat memberikan perintah mulainya pertandingan, berakhirnya pertandingan, bahkan pelanggaran dan tendangan bebas yang harus dilakukan.

Untuk menerima perintah dari *referee box*, robot membutuhkan bantuan sebuah *base station*. *Base station* merupakan suatu program yang dibuat dan di desain untuk meneruskan perintah dari *referee box* ke robot. Dengan kata lain, *base station* merupakan seorang pelatih yang memberikan arahan kepada pemainnya agar para pemain dapat bermain dengan maksimal dan tidak melanggar peraturan yang telah ditetapkan. Umumnya sebuah *base station* dibuat oleh masing-masing team dengan cara yang berbeda-beda. Pada penelitian kali ini, peneliti memilih *python* sebagai bahasa pemrograman yang akan digunakan untuk membuat *base station* karena dinilai cukup mudah digunakan bahkan untuk para pemula dalam bahasa pemrorgaman.

Python menyediakan sebuah library *socket programming* yang dapat menghubungkan antara *base station* dengan *referee box* dan meneruskan data yang diterima ke robot. *Base station* berfungsi sebagai penerjemah data dari *referee box* sekaligus pemberi perintah pada robot untuk menentukan pergerakan yang akan dilakukan.

II. LANDASAN TEORI

A. *Referee Box*

Referee Box adalah sebuah aplikasi *game controller* yang digunakan dalam Kontes Robot Sepak Bola Indonesia. *Referee box* dikendalikan oleh juri pertandingan untuk memandu jalannya pertandingan.

Dalam *referee box* terdapat sebanyak 22 *button* yang memberikan perintah yang berbeda-beda. Perintah tersebut antara lain : *Kick Off, Free Kick, Goal Kick*, dll. Sehingga robot bisa membedakan perintah satu dengan yang lain.[1] Tampilan *Referee Box* dapat dilihat pada gambar 1

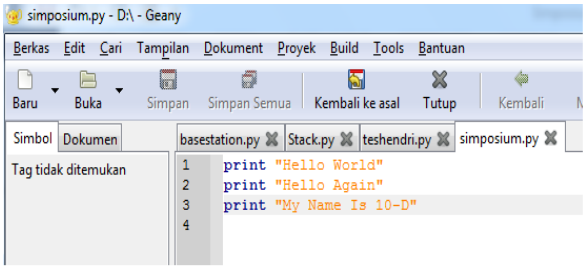


Gambar 1. *Referee Box*

B. *Phyton*

Python merupakan bahasa pemrograman yang berorientasi obyek dinamis, dapat digunakan untuk bermacam macam

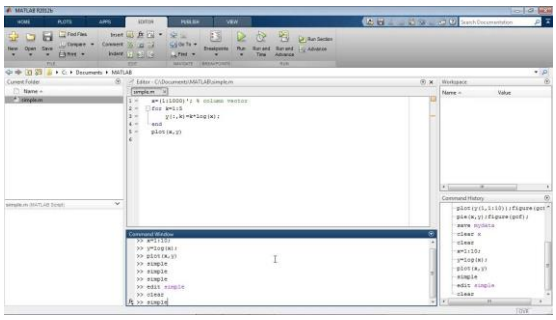
pengembangan perangkat lunak. *Python* menyediakan dukungan yang kuat untuk integrasi dengan bahasa pemrograman lain dan alat-alat bantu lainnya. *Python* hadir dengan pustaka-pustaka standar yang dapat diperluas serta dapat dipelajari hanya dalam beberapa hari. Bahasa pemrograman yang interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. *Python* diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif.[2] Contoh kode program python dapat dilihat pada gambar 2



Gambar 2. contoh kode program *python*

C. Matlab

Matlab adalah sebuah platform dengan kemampuan tinggi untuk komputasi teknis. Ia menggabungkan komputasi, visualisasi, dan pemrograman dalam satu kesatuan yang mudah digunakan di mana masalah dan penyelesaiannya diekspresikan dalam notasi matematik yang sudah dikenal. Tidak hanya itu, *matlab* juga mampu melakukan komputasi simbolik yang biasa dilakukan oleh *Maple*. Kegunaan *Matlab* meliputi : Matematika dan komputasi, Pengembangan algoritma, Akuisisi data, Pemodelan, simulasi dan *prototype*, Grafik saintifik dan engineering, Perluasan pemakaian, seperti *graphical user interface* (GUI). [3] Dalam penelitian ini, matlab difungsikan sebagai penerima data dari *base station* yang dipasangkan pada robot 10-D. Tampilan Matlab dapat dilihat pada gambar 3



Gambar 3. Tampilan Matlab

III. METODOLOGI PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang digunakan dalam penelitian ini adalah jenis penelitian metode penelitian percobaan (*experimental Research*).

Metode *eksperimental* ini dilakukan bertujuan untuk melakukan uji coba dari setiap perintah yang dikirim dari *referee box* robot mulai dari *Kick Off*, *Start*, *Stop*, *Goal Kick* dll. Hal ini dimaksud untuk memantau data yang diterima oleh *base station*.

B. Alat Dan Bahan

Base Station akan digunakan sebagai jembatan antara robot dan *referee box*. Maka, alat dan bahan yang digunakan adalah sebagai berikut :

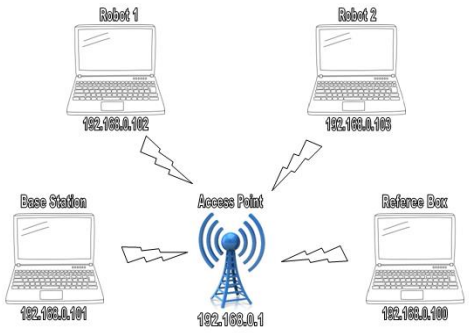
1. 3 Buah *Laptop*
2. *Access Point*
3. Aplikasi *Referee Box*
4. Aplikasi Matlab versi 2015b
5. Aplikasi Geany (untuk menjalankan python)

C. Tahapan Penelitian

Dari alat dan bahan yang telah ditetapkan oleh peneliti, maka peneliti menerapkan tahapan pengerjaan sebagai berikut:

1. Laptop A digunakan sebagai *referee box*
2. Laptop B digunakan sebagai *base station*
3. Laptop C digunakan sebagai robot
4. Semua *laptop* terhubung jaringan dengan *access point* yang sama.
5. IP *base station* ditambahkan ke daftar IP *referee box*.

Dalam penelitian ini digunakan topologi *ring* dengan IP yang telah ditentukan oleh peneliti, topologi *ring* digunakan untuk pengujian *data referee box, base station*, dan robot.

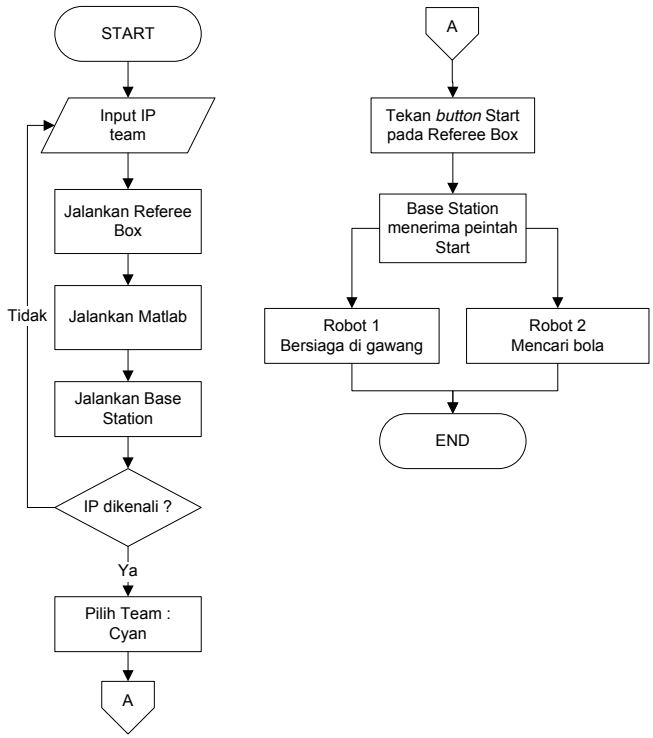


Gambar 4. Topologi Ring

Terdapat dua robot dalam pengujian sistem yang dilakukan peneliti, dengan pemisalan robot 1 adalah robot kiper dan robot 2 adalah robot penyerang.

D. *Flowchart*

Berikut merupakan *flowchart* prosedur pengujian pengiriman data dari *referee box* ke *base station* dan aksi robot yang dilakukan ketika mendapatkan sebuah perintah.

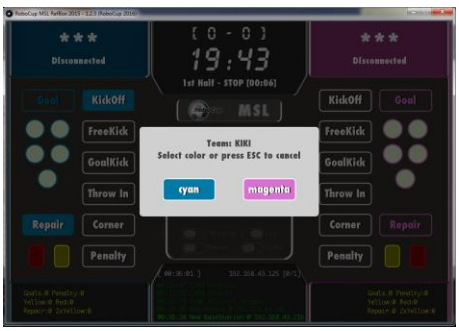


Gambar 5. *Flowchart*

IV. HASIL PENELITIAN

A. *Login Team*

Pengujian ini dimaksud untuk memastikan bahwa IP *base station* yang terdaftar di *referee box* telah benar. Jika IP yang terdaftar benar, maka akan muncul form dialog untuk menanyakan team yang akan dipilih. Hasil pengujian ini ditunjukkan dalam gambar 5.

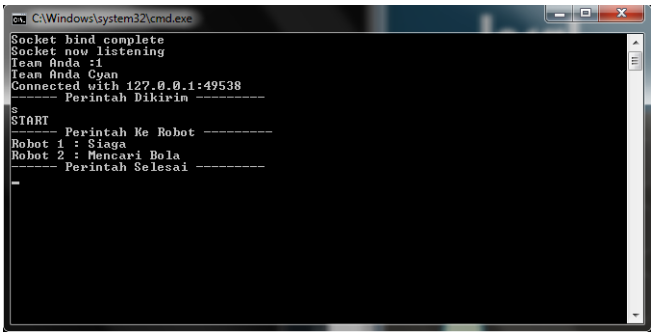


Gambar 6. Pengujian IP *Base Station* di *Referee Box*

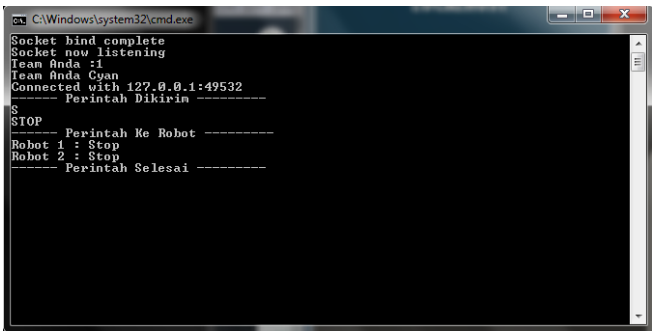
Dari hasil pengujian diatas dapat disimpulkan bahwa *base station* dapat *login* ke *referee box*. Hal ini dibuktikan dengan pilihan team yang tersedia di *referee box*.

B. *Menerima dan Mengirim Data*

Pengujian ini dilakukan untuk memastikan bahwa *base station* menerima data yang benar dari *referee box* dan dapat meneruskannya ke robot. Pengujian ini hanya dilakukan pada tombol *Start* dan *Stop*. Hasil penerimaan data pada *base station* dapat dilihat pada gambar 7 dan gambar 8.



Gambar 7. Respon *base station* saat ditekan tombol start



Gambar 8. Respon *base station* saat ditekan tombol stop

C. *Pergerakan Robot*

Berikut ini adalah tabel yang menunjukkan respon robot terhadap perintah yang diterima dari *base station*. Dalam hal ini peneliti menetapkan team yang digunakan adalah *cyan* dan secara otomatis tim lawan adalah team *magenta*.

TABEL I. PERGERAKAN ROBOT

Perintah yang dikirim (*Base Station*)	Aksi Robot 1	Aksi Robot 2
Start	Bergerak ke arah Bola (kanan-kiri)	Mencari Bola
Stop	Berhenti	Berhenti
Dropball	Kembali ke posisi awal	Berhenti
Park	Berhenti	Berhenti
End Game	Berhenti	Berhenti

End Part	Berhenti	Berhenti
Reset	Berhenti	Berhenti
Cyan Goal	Kembali ke posisi awal	Berhenti
Cyan Kick Off	Kembali ke posisi awal	Melakukan Positioning
Cyan Free Kick	Kembali ke posisi awal	Berhenti
Cyan Goal Kick	Kembali ke posisi awal	Berhenti
Cyan Throw In	Kembali ke posisi awal	Berhenti
Cyan Corner	Kembali ke posisi awal	Berhenti
Cyan Penalty	Kembali ke posisi awal	Berhenti
Magenta Goal	Kembali ke posisi awal	Berhenti
Magenta Kick Off	Kembali ke posisi awal	Melakukan Positioning
Magenta Free Kick	Kembali ke posisi awal	Berhenti
Magenta Goal Kick	Kembali ke posisi awal	Berhenti
Magenta Throw In	Kembali ke posisi awal	Berhenti
Magenta Corner	Kembali ke posisi awal	Berhenti
Magenta Penalty	Kembali ke posisi awal	Berhenti

Dari hasil pengujian data pergerakan robot yang diperoleh diatas dapat disimpulkan bahwa *base station* dapat menerima dan mengirim data dengan baik. Data yang dikirim juga dapat dilaksanakan dengan benar pada robot.

V. KESIMPULAN

Dari penelitian ini dapat diperoleh hasil sebagai berikut:

1. *Base Station* dapat login ke *referee box*.
2. *Base Station* dapat menerima data yang dikirim *referee box* dengan baik.
3. *Base Station* dapat mengirim perintah ke robot
4. Robot dapat melaksanakan dengan baik perintah yang didapatkan dari base station

REFERENSI

[1] Fahriza Azwar Muhammad, Rizky Arif Windiator,Yuridi Bintang Pratama, "pemrograman socket untuk koneksi Abtara Raspberry Pi dengan Referee Box",Universitas Islam Indonesia, 2016.
[2] Noprianto, "Python dan Pemrograman Linux", Yogyakarta, 2002.
[3] Muhamma Dahria dkk, Jurnal SAINTIKOM Vol. 11, No.3, September 2012.
[4] Referee Box 2017, https://github.com/RoboCup MSL/RefBox2015, diakses pada 12 Januari, 2017.
[5] Panduan KRSBI beroda 2017, http://kontesrobot indonesia.org/datakontes/2017/PanduanKRSBIBeroda2017.pdf, diakses pada 12 Januari, 2017.
[6] Yuliza, IncomTech, Jurnal Telekomunikasi dan Komputer, vol.4, no.1, 2013.
[7] Andi Adriansyah, Jurnal TICOM Vol.1 No.1 September, 2012.
[8] Rikki Fitria, Jurnal Ilmiah Poli Rekayasa Volume 3, Nomor 2, Maret 2008.

Rancang Bangun Sistem Tertanam Pada Humanoid Robot Soccer Tim GLADIATOS UI

Ghana Nazala Putra, Fatah Abdul Wahab, Dyah Sekar Asih*, Mutiara Surya, Achmad Kripton Nugraha
Universitas Indonesia
Depok
dyahsekarasih@yahoo.co.id

***Abstrak*— Humanoid Robot Soccer is a robot with human-like form that could play soccer autonomously. To create Humanoid Robot Soccer, a robust, compact and rich of feature embedded system is needed caused by the need of many sensors data sampled continously, weight of running process and complex calculation espesially for its control system and image processing. This paper focus on designing and implementing Odroid XU-4 Single Board Computer as Main Controller supported by Inertial Measurement Unit sensors and Dynamixel MX-28T and MX-64T actuators for Humanoid Robot Soccer. The result of this reasearch indicate that the purposed device is qualified to be an embedded system in Humanoid Robot Soccer.**

Kata kunci— Humanoid Robot Soccer, embedded system, Odroid XU-4, Inertial Measurement Unit, Dynamixel Actuators

I. PENDAHULUAN

Universitas Indonesia setiap tahunnya selalu ikut berpartisipasi dalam Kontes Robot Sepak Bola Indonesia sejak tahun 2011. Tahun ini Universitas Indonesia diwakili oleh Tim GLADIATOS UI memiliki 4 buah *Humanoid Robot Soccer* yang terdiri dari tipe BIOLOID, Darwin-OP dan dua robot buatan sendiri yang masih dalam tahap pengembangan yang mana salah satunya adalah yang sedang penulis kembangkan.

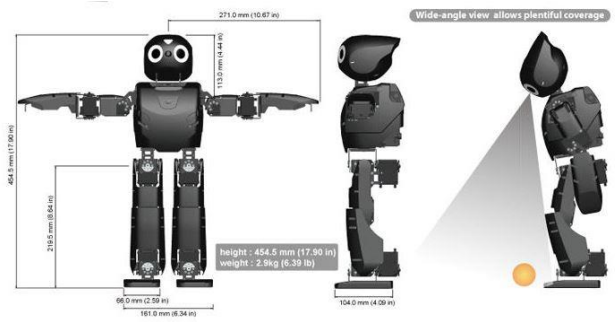
Dalam mengembangkan *Humanoid Robot Soccer* yang kami buat sendiri terdapat beberapa tantangan yang cukup besar, salah satunya adalah bagaimana mendesain arsitektur perangkat keras dan perangkat lunak yang mumpuni, efisien serta dapat mengakomodasi perhitungan-perhitungan kompleks yang dibutuhkan dalam membangun kecerdasan buatan pada *Humanoid Robot Soccer*.

Oleh sebab itu, fokus penelitian ini akan membahas mengenai pengembangan sistem tertanam pada *Humanoid Robot Soccer* yang meliputi perangkat keras dan perangkat lunak. Sistem tertanam adalah sistem elektronik yang terdiri dari mikroprosesor atau mikrokontroler, namun kita tidak menganggapnya sebagai komputer, melainkan komputer tersebut tertanam di dalam sistem [1]. Adapun pada penelitian ini penulis melakukan pengembangan terhadap robot yang dimiliki oleh Tim GLADIATOS UI dengan basis *platform* robot Darwin-OP yang merupakan sumber terbuka.

II. TINJAUAN PUSTAKA

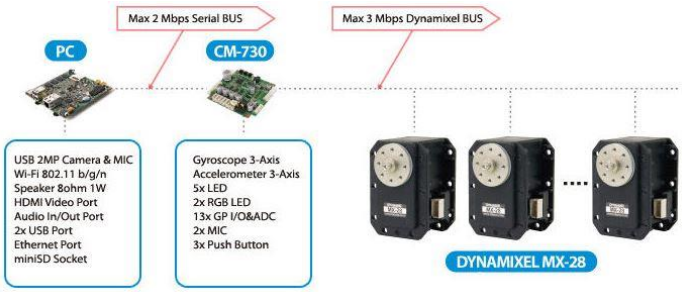
A. *Darwin-OP [2]*

Darwin-OP atau *Dynamic Anthropomorphic Robot with Intelligence-Open Plaltform* adalah salah satu *platform* terbuka dari *Humanoid Robot Soccer*. Gambar 1 memberi gambaran bentuk dari Darwin-OP.



Gambar 1. *Darwin-OP*

1. Arsitektur Perangkat Keras Darwin-OP



Gambar 2. Arsitektur Perangkat Keras Darwin-OP

Karena merupakan sebuah platform terbuka, kita dapat mengunduh seluruh berkas dari Darwin-OP yang disediakan di https://sourceforge.net/projects/darwinop. Mulai dari desain mekanika, elektrik hingga kode sumber dari Darwin-OP. Oleh sebab itu penulis mengambil Darwin-OP sebagai robot acuan dalam pengembangan *Humanoid Robot Soccer* bagi Tim GLADIATOS UI.

2. Framework Darwin-OP

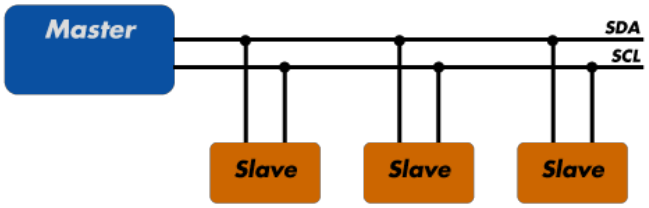
Framework Darwin-OP adalah kumpulan kode sumber yang menjadi kecerdasan dari Darwin-OP. Pada penelitian ini

penulis akan menggunakan Framework Darwin-OP sebagai Program Utama namun dengan beberapa penyesuaian disebabkan oleh perbedaan Arsitektur Perangkat Keras yang digunakan.

B. *Inertia Measurement Unit (IMU) [3]*

IMU atau *Inertial Measurement Unit* merupakan sebuah perangkat elektronik yang mengukur kecepatan sudut, gaya gravitasi dan orientasi. IMU biasanya menggunakan gabungan dari *Gyroscope* dan *Accelerometer* dan terkadang juga menggunakan *Magnetometer*. Pada dasarnya dengan menggunakan IMU sebuah perangkat dapat diatur bagaimana harus memosisikan diri, orientasi dan juga pergerakannya. Oleh sebab itu IMU dapat diimplementasikan pada pergerakan dan keseimbangan *Humanoid Robot Soccer*.

C. *Protokol Komunikasi I2C [4]*

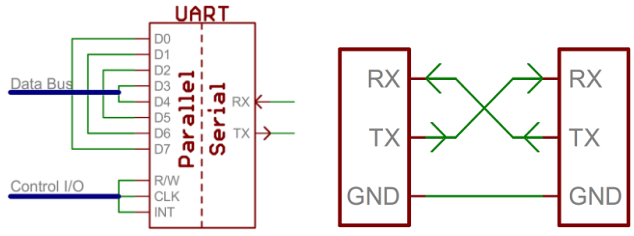


Gambar 3. Diagram I2C

Protokol Komunikasi I2C (dibaca *I-two-C* atau *I-square-C*) merupakan salah satu jenis protokol komunikasi yang mengizinkan lebih dari satu perangkat untuk terhubung pada bus yang sama. Protokol ini menggunakan dua sinyal kabel untuk berkomunikasi yaitu SDA yang bertugas sebagai data sinyal dan SCL yang bertugas sebagai *clock* sinyal. Salah satu keistimewaan dari protokol I2C adalah dapat menampung hingga 1008 perangkat. Untuk membedakan satu perangkat dengan perangkat lainnya, masing-masing perangkat mempunyai alamat masing-masing.

D. *Protokol Komunikasi UART*

UART atau *Universal Asynchronous Receiver/Transmitter* merupakan sebuah rangkaian yang menjadi penengah dan berfungsi sebagai pengubah dari serial ke paralel pada penerimaan data dari perangkat periferal dan pengubah dari paralel ke serial pada penerimaan data dari CPU [5].



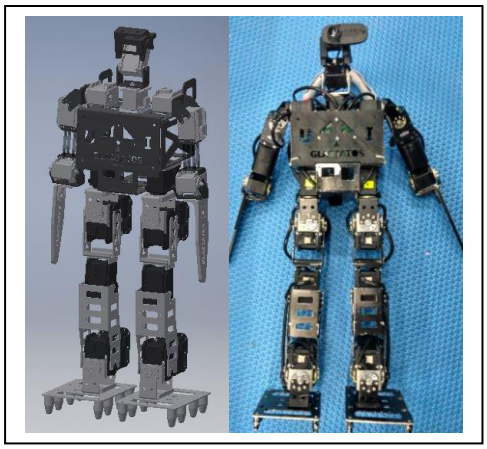
Gambar 4. Ilustrasi Sederhana UART (Kiri) dan UART Wiring (Kanan)

Untuk menghubungkan dua perangkat menggunakan UART, Komunikasi Serial bus penerima dari perangkat satu ke bus pengirim dari perangkat dua dan bus pengirim dari perangkat satu ke bus penerima dari perangkat dua. Pada komunikasi UART bolak-balik, level tegangan dari kedua perangkat terhubung haruslah sama. Hal ini disebabkan oleh Komunikasi Serial UART mengirimkan data pada suatu waktu dengan suatu kecepatan tertentu (*baudrate*).

E. *Perangkat Pendukung*

Dalam membangun sistem tertanam tentunya dibutuhkan komponen-komponen pendukung lain agar menjadi sebuah Humanoid Robot Soccer. Pada penelitian ini penulis menggunakan sistem mekanika yang telah dibuat sebelumnya oleh Tim GLADIATOS UI sebagai kerangka dari robot, begitu juga dengan servo Dynamixel yang digunakan terdapat dua jenis yaitu servo Dynamixel MX-28 dan servo Dynamixel MX-64.

1. Desain Mekanika Robot yang Dikembangkan Tim GLADIATOS UI



Gambar 5. Desain Mekanika Darnet

Gambar 5 merupakan desain mekanika dari salah satu *Humanoid Robot Soccer* yang dikembangkan Tim GLADIATOS UI yang akan menjadi kerangka bagi sistem tertanam yang dibuat pada penelitian paper ini. Robot tersebut diberi nama Darnet. Desain mekanika Darnet berbeda dengan Darwin-OP, terutama pada bagian kaki. Darnet menggunakan *servo* Dynamixel MX-64 pada ke - dua belas *servo* kakinya. Dimensi kaki Darnet adalah 30cm terukur dari ujung paling bawah kaki (pul kaki) hingga pangkal paha. Sedangkan untuk total tinggi robot adalah 60.5cm. Pada kedua tangan dan kepala Darnet menggunakan *servo* Dynamixel MX-28. Namun secara struktur penempatan dan jumlah dari derajat kebebasan tetap mengikuti Darwin-OP.

2. *Servo* Dynamixel MX-28T dan *Servo* Dynamixel MX-64T

Servo Dynamixel MX-28T dan MX-64T merupakan produksi dari ROBOTIS yang termasuk dalam kategori servo cerdas. Servo ini dikatakan cerdas karena di dalamnya terdapat sensor-sensor dan mikrokontroler yang dapat memberikan umpan balik berupa posisi, kecepatan, temperatur, dan tegangan. Servo ini menggunakan mikrokontroler ST Cortex-M3 32bit 72MHz dengan protokol komunikasi *Half-Duplex* TTL UART. Servo Dynamixel MX-28 menggunakan *contactless absolute encoder* dan *full-metal-gear* yang membuatnya dapat mencapai parameter dengan akurat.



Gambar 6. *Servo* Dynamixel MX-28T dan MX-64T

III. RANCANG BANGUN SISTEM

Pada bagian ini akan dijelaskan bagaimana perancangan dari sistem tertanam yang akan dibuat. Penjelasan meliputi tinjauan pada penelitian lain yang sudah ada, *system requrements*, perancangan perangkat keras dan perangkat lunak serta perencanaan skenario pengujian dan pengukuran performa dari sistem yang dibuat.

A. System Requirements

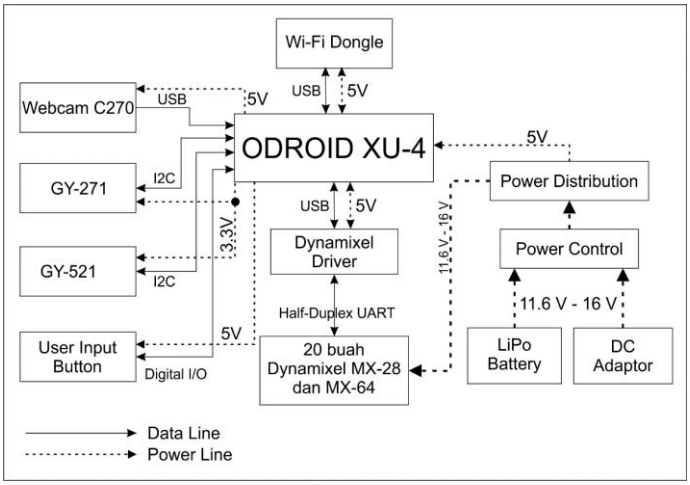
Berdasarkan peraturan lomba, robot diharapkan dapat melakukan fungsi-fungsi dasar bermain sepak bola seperti berjalan, mendeteksi bola, menentukan arah gawang, menendang bola ke arah gawang dan berkomunikasi dengan *Game Controller* yang telah disediakan oleh juri [6].

Dari penjelasan tersebut penulis menspesifikkan lagi kebutuhan dari sistem tertanam yang dirancang untuk paper ini ialah sebagai berikut:

- Sistem tertanam yang dibuat harus dapat menerima dan mengolah data dari sensor keseimbangan untuk mendukung robot dalam berjalan.
- Sistem tertanam yang dibuat harus dapat menerima dan mengolah data dari sensor arah untuk mendukung robot dalam membedakan arah gawang lawan dan gawang kawan.
- Sistem tertanam yang dibuat harus dapat mengakomodasi kebutuhan dari Darwin-OP *Framework* sebagai program utama kecerdasan robot yang akan dibuat.
- Sistem tertanam yang dibuat harus dapat berkomunikasi dengan penggerak robot yaitu motor servo Dynamixel MX-28 dan MX-64.

B. Perancangan Perangkat Keras

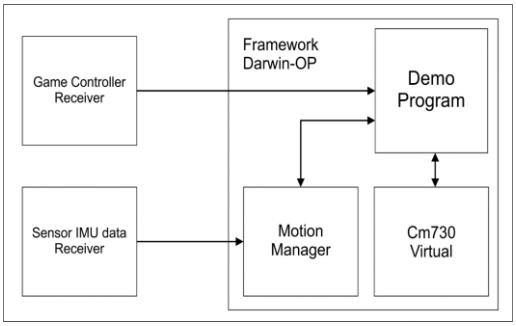
Berdasarkan daftar kebutuhan yang telah dijelaskan pada sub bab sebelumnya, maka sistem tertanam yang dibutuhkan adalah seperti yang ditampilkan pada Gambar 7. Berbeda dengan robot acuan yaitu Darwin-OP yang menggunakan dua kontroler yang terbagi menjadi *Main Controller* dan *Sub Controller*, pada rancangan ini hanya menggunakan satu kontroler sebagai *Main Controller* yaitu Odroid XU-4.



Gambar 7. Block Diagram Perangkat Keras yang Dirancang

Data dari sensor *Magnetometer*, *Accelerometer* dan *Gyroscope* akan dikirim ke Odroid XU-4 menggunakan protokol komunikasi I2C, sedangkan *user input* adalah berupa *push button* dan *led output* yang dihubungkan ke Odroid XU-4 secara digital. Data yang diterima oleh Odroid XU-4 kemudian akan diolah oleh program yang telah dibuat oleh penulis dan kemudian data hasil olahan akan dikirimkan ke program utama yang mana menggunakan Program *Framework Darwin Open Platform*. Pada bagian *output* yaitu kendali Odroid XU-4 ke *servo* Dynamixel akan dibantu oleh sebuah *Dynamixel Driver* yang telah dibuat penulis. *Driver* tersebut terhubung ke Odroid XU-4 dengan USB dan memiliki keluaran Dynamixel Bus yang dapat langsung dihubungkan ke *servo* Dynamixel.

C. Perancangan Perangkat Lunak



Gambar 8. *Block Diagram* Perangkat Lunak

Pada sub bab ini akan dijelaskan mengenai perancangan sistem dari sisi perangkat lunak yang dibutuhkan oleh Odroid XU-4 dalam mengambil data dari sensor, mengolahnya dan meneruskannya ke program utama yaitu *Framework Darwin-OP*.

Dengan kode yang dibuat untuk menerima data dari sensor yang menggunakan protokol I2C, nilai dari sensor kemudian akan diteruskan ke *Framework Darwin-OP* yang nantinya akan dijadikan parameter oleh algoritme keseimbangan dari Darwin, selain itu akan dilakukan beberapa perubahan pada *Framework Darwin-OP* yaitu pada *class* CM730 agar dapat menggunakan *Dynamixel Driver* yang telah dirancang. CM730 sendiri adalah sebuah perangkat mikrokontroler yang berperan sebagai *Sub Controller* pada robot Darwin-OP. Perubahan ini dilakukan

karena penulis tidak menggunakan CM730 sehingga perlu dibuat sebuah nilai *virtual* sebagai penggantinya. Perubahan lain juga dilakukan pada *class Kinematics* di *Framework Darwin-OP*, hal tersebut dilakukan agar *Framework Darwin-OP* dapat diimplementasikan pada desain mekanika yang digunakan. Dan yang terakhir perubahan juga dilakukan pada demo program yang mana demo program akan meminta *input* terlebih dahulu dari *user* dan juga dari *Game Controller*.

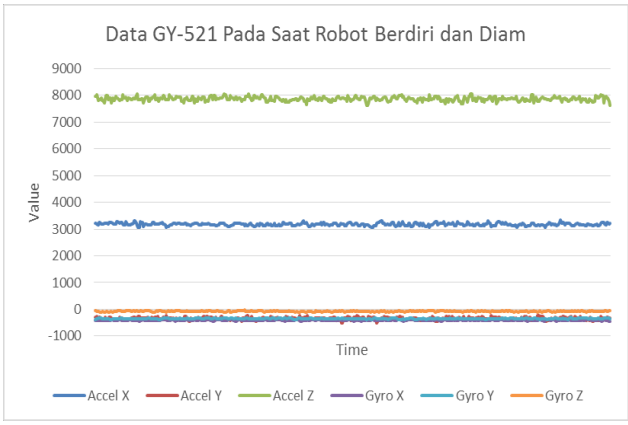
D. Skenario Pengujian dan Pengukuran Performa Sistem Tertanam yang Dibuat

Eksperimen pengujian akan dilakukan dengan melakukan pembacaan data pada sensor secara berulang – ulang dengan Odroid XU-4 untuk kemudian dianalisis data yang telah didapatkan dan keterkaitannya dengan sistem yang telah dibuat. Selain itu juga akan dilakukan uji terhadap komunikasi antara Odroid XU-4 dan *servo* Dynamixel dengan melakukan *read* dan *write* data kepada *servo* dan kemudian dicatat hasilnya untuk dianalisis.

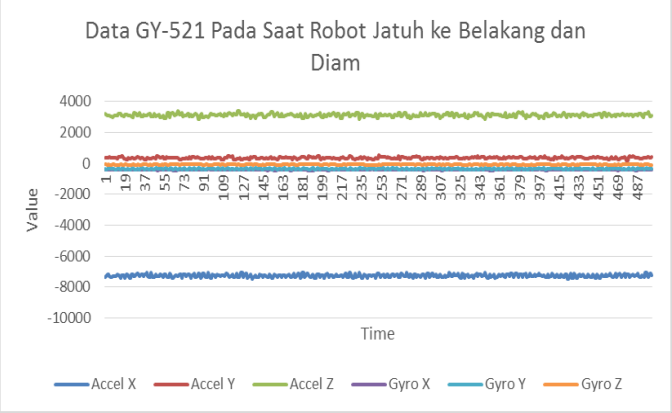
IV. HASIL IMPLEMENTASI DAN EVALUASI

1) Hasil Pengujian dan Evaluasi Pembacaan Sensor Gyroscope, Accelerometer dan Magnetometer oleh Odroid XU-4

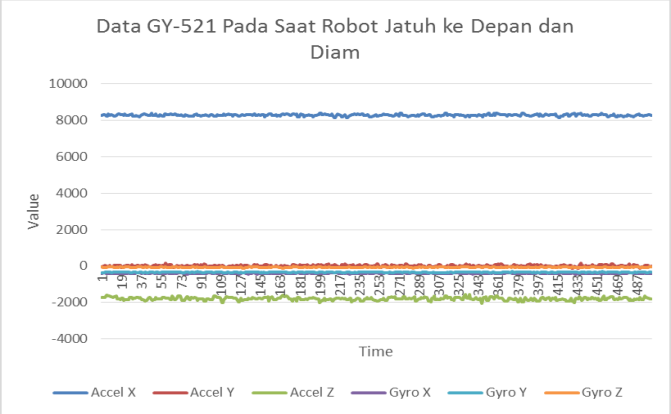
Berikut adalah grafik hasil pengujian dari sensor *Gyroscope*, *Accelerometer* dan *Magnetometer*.



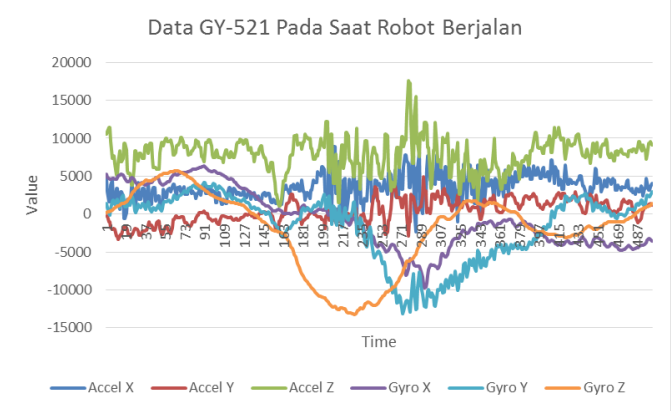
Gambar 9. Grafik Data GY-521 Pada Saat Robot Berdiri dan Diam



Gambar 10. Grafik Data GY-521 Pada Saat Robot Jatuh Belakang dan Diam



Gambar 11. Grafik Data GY-521 Pada Saat Robot Jatuh Depan dan Diam



Gambar 12. Grafik Data GY-521 Pada Saat Robot Berjalan



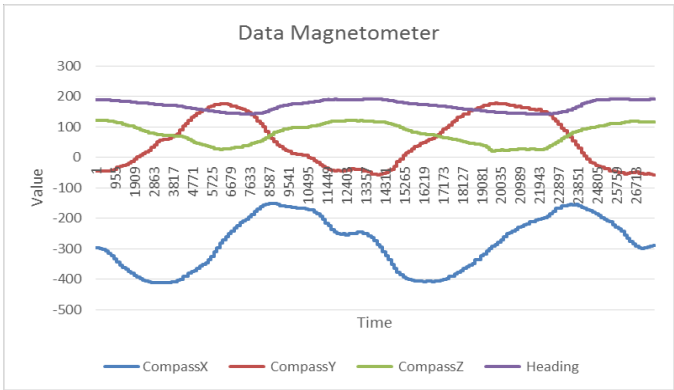
Gambar 13. Robot pada Posisi Berdiri (Kiri), Jatuh Belakang (Kanan Atas), dan Jatuh Depan (Kanan Bawah)

Grafik yang ditunjukkan oleh Gambar 9, 10, 11, dan 12 menunjukkan data *raw* atau data mentah hasil pembacaan sensor oleh Odroid XU-4. Terdapat 500 data dengan waktu sampling 1 detik. Pada Grafik 9, 10, 11 menunjukkan Kondisi Robot Diam, terlihat pada grafik tersebut nilai sensor *Accelerometer* dan *Gyroscope* cenderung stabil. Dari ketiga grafik tersebut bisa dilihat bahwa nilai *Accelerometer* berubah-ubah tergantung pada kondisi robot. Sedangkan untuk nilai *Gyroscope* cenderung stabil pada kondisi diam namun berubah-ubah pada saat robot berjalan yang ditunjukkan oleh Grafik pada Gambar 12. Nilai *Acelerometer* dan *Gyroscope* tersebut akan diteruskan ke *Framework* Darwin-OP, diolah dan kemudian dijadikan acuan untuk menentukan apakah

Coba Ke	Jml Status Gagal	Jml Status Total	Nilai *Goal Position*	Nilai *Present Position* Akhir ID 19	Nilai *Present Position* Akhir ID 20	Persentase Keberhasilan (%)
1	1	101	2500	2494	2490	99
2	3	63	2000	2006	2008	95.23
3	2	92	2500	2494	2490	97.82
4	3	79	2000	2007	2010	96.2
5	4	78	2500	2494	2491	94.87
6	2	87	2000	2003	2010	97.7
7	2	93	2500	2494	2490	97.85
8	5	72	2000	2004	2007	93.06
9	2	93	2500	2494	2490	97.85
10	3	77	2000	2007	2010	96.1
Rata-rata						**96.57**

robot sedang berdiri, jatuh ke belakang atau jatuh ke depan serta dijadikan parameter untuk membangun algoritme keseimbangan pada *Humanoid Robot Soccer* Darwin-OP.

Sedangkan untuk sensor *Magnetometer* terdapat masalah yang mana untuk I2C pada Odroid XU-4 tidak dapat menerima data dari kedua modul apabila dihubungkan pada I2C Bus yang sama. Namun berdasarkan hasil studi pustaka yang penulis baca pada *Datasheet* MPU6050 terdapat solusi yang dapat digunakan yaitu memanfaatkan fungsi *slave device* pada modul GY-521 yang mana sensor *Magnetometer* dapat diteruskan ke Odroid-XU4 melalui pin Eksternal I2C pada GY-521 yang dihubungkan dengan sensor *Magnetometer*.



Gambar 14. Grafik Data Magnetometer

Gambar 14 menunjukkan data dari sensor *Magnetometer*. Data diambil dengan cara memutar robot secara 360 derajat sebanyak 2 kali. Dari grafik tersebut bisa disimpulkan bahwa data dari sensor sesuai dan dapat dijadikan acuan untuk menentukan arah bagi robot.

2) Hasil Pengujian dan Evaluasi Komunikasi antara Odroid XU-4 dengan Servo Dynamixel MX-28T dan MX-64T

Untuk pengujian komunikasi dengan *servo* Dynamixel dilakukan *read and write* data posisi pada *servo* dengan menggunakan program yang telah dibuat menggunakan *library Dynamixel_SDK*. Pengujian dilakukan pada 2 buah *servo* Dynamixel yang terhubung pada kepala robot yaitu *servo* Dynamixel MX-28 dengan ID 19 dan ID 20.

Pada *servo* dilakukan perintah untuk bergerak dari posisi 2500 ke 2000. Pembacaan yang dilakukan berhenti apabila *servo* mencapai posisi yang kurang lebih 10 poin dari posisi yang diperintahkan. Pembacaan gagal ditandai dengan tulisan "*There is no status packet*". Tabel 1. menunjukkan pengujian di atas dilakukan sebanyak 10 kali dengan kondisi yang sama.

TABEL I. TABEL HASIL UJI KOMUNIKASI *SERVO* DYNAMIXEL

Berdasarkan data di atas, kegagalan *Status Packet* terjadi secara acak. Berdasarkan studi pustaka yang penulis lakukan, pada *Datasheet USB port*, dijelaskan bahwa *port USB* memiliki *time framing* sebesar 1ms untuk mengirimkan dan menerima data, sedangkan pada *datasheet servo* Dynamixel MX, *servo* Dynamixel MX *Intruction Packet* maupun *Status Packet* membutuhkan 2ms dan akan bertambah pada setiap *servo* yang dihubungkan. Sedangkan pada *Framework* Darwin-Op maupun Dynamixel SDK *control loop* membutuhkan 8ms sehingga apa bila dilakukan transmisi data secara terus menerus, pasti akan didapatkan beberapa kali kegagalan transmisi seperti yang terlihat pada Tabel 1.

Solusi yanng dapat dilakukan yaitu menggunakan Pin UART yang disertai *Hardware Flow Control* sebagai pengatur pada *Half-Duplex UART* yang mana tidak dimiliki oleh Odroid XU-4 sehingga harus menambahkan *controller* tambahan yang memiliki fungsi tersebut. Meskipun begitu, Sistem Tertanam yang dirancang tetap dapat berkomunikasi dengan *servo* Dynamixel MX dengan tingkat keberhasilan yang cukup tinggi yaitu sebesar 96.57% dan meskipun terdapat kegagalan *servo* tetap dapat mencapai *Goal Position*.

V. KESIMPULAN

Dari keseluruhan pembahasan dalam penelitian paper ini dapat disimpulkan beberapa hal, yaitu:

- Pengambilan data dari sensor IMU dengan komunikasi I2C pada Odroid XU-4 berhasil. Namun dari segi keakuratan tetap bergantung pada spesifikasi modul sensor dan juga dapat ditingkatkan dengan menggunakan filter seperti Kalman Filter.
- Komunikasi dari Odroid XU-4 dengan servo Dynamixel menggunakan driver yang dibuat berhasil.
- Jumlah servo berpengaruh terhadap waktu yang dibutuhkan dalam komunikasi dengan servo Dynamixel. Salah satu cara untuk meningkatkan kecepatannya adalah dengan meningkatkan baudrate yang digunakan.
- Tetap dibutuhkan pengaturan parameter pada saat mengeksekusi jalan dari robot.

- Sistem Tertanam yang dikembangkan berhasil diuji keandalannya dan dapat digunakan dalam kompetisi KSRBI-H.

REFERENSI

[1] T. D. Morton, Embedded Microcontrollers, New Jersey: Prentice Hall, 2001

[2] I. Ha, Y. Tamura, H. Asama, J. Han dan D. W. Hong, "Development of Open Humanoid Platform DARwIn-OP," dalam *SICE Annual Conference*, Tokyo, 2011.

[3] Starlino Electronics, "A Guide to Using IMU in Embedded Aplication," 29 Desember 2009. [Online]. Available: http://www.starlino.com/imu_guide/html. [Diakses 15 Maret 2017].

[4] Sparkfun, "I2C," Sparkfun, [Online]. Available: https://learn.sparkfun.com/tutorials/i2c#introduction. [Diakses 15 Maret 2017].

[5] Sparkfun, "Serial Communication," Sparkfun, [Online]. Availbale: https://learn.sparkfun.com/tutorials/serial-communication#uarts. [Diakses 15 Maret 2017].

[6] Panitia Pusat Kontes Robot Indonesia, "Tema Divisi dan Rule Book," 2017. [Online]. Available: https://www/kontesrobotindondesia.org/kri-2017/ [Diakses 27 Januari 2017].

Implementasi Sistem Kendali PD (Proportional Derivative) Pada Navigasi Wall Follower Robot Berkaki Enam (Hexapod)

Y. R. Wahyudi*, M. S. Hadi, A. N. Handayani, dan S. Sendari
Jurusan Teknik Elektro
Fakultas Teknik, Universitas Negeri Malang
Malang, Indonesia
Email : yudarismawahyudi@gmail.com

Abstrak—Wall follower **merupakan suatu metode yang digunakan pada navigasi mobile robot. Metode ini bertujuan agar robot dapat bergerak dengan menelusuri dinding. Pada penelitian ini navigasi *wall follower* dikembangkan menggunakan sistem kendali PD yang diimplementasikan pada robot berkaki enam (*hexapod*). Penggunaan kendali PD bertujuan agar robot dapat berjalan dengan halus (*smooth*) dalam mengikuti dinding dan menghindari halangan. Robot dilengkapi dengan 7 buah sensor ultrasonik yang digunakan untuk mengukur jarak robot terhadap dinding. Ada beberapa gerakan dasar yang harus dapat dilakukan oleh robot, antara lain berjalan, menghindari halangan, dan mengikuti dinding atau koridor. Berdasarkan percobaan yang telah dilakukan menunjukkan bahwa dengan mengimplementasikan sistem kendali PD robot dapat berjalan menelusuri dinding dengan tingkat keberhasilan 90% pada mode telusur dinding kanan dan 95% pada mode telusur dinding kiri.**

Kata kunci—Wall Follower, Hexapod Robot, Kendali PD

I. PENDAHULUAN

Mobile robot merupakan robot yang dirancang untuk bergerak secara otomatis atau dikenal robot bergerak otonom, robot yang bergerak bebas pada area yang tidak terstruktur dan dapat menentukan sendiri jalur pergerakanya [1]. Berdasarkan alat geraknya, mobile robot diklasifikasikan menjadi dua jenis, yaitu robot beroda dan robot berkaki. Dalam aplikasinya, robot beroda lebih banyak digunakan karena kecepatanya dalam melintasi bidang yang rata, dan kemudahan dalam desain dan implementasi [2]. Namun robot jenis beroda akan mengalami kesulitan jika digunakan di bidang yang tidak rata. Sedangkan robot berkaki dapat digunakan dengan lebih fleksibel. Robot ini mampu berjalan di bidang yang rata maupun di bidang yang tidak rata.

Salah satu jenis robot berkaki yang paling banyak digunakan adalah jenis robot berkaki enam (hexapod). Hexapod memiliki banyak keuntungan, diantaranya adalah fleksibilitas dalam pola gerakan, karena robot memiliki kestabilan statis jika tiga atau lebih kakinya ada di tanah [2]. Saat ini robot berkaki telah banyak digunakan dalam berbagai aplikasi dan riset. Salah satunya adalah pada Kontes Robot Pemadam Api Indonesia (KRPAI) yang diselenggarakan oleh DIKTI [3]. Masalah yang sering dihadapi pada *wall follower* robot adalah masalah penelusuran dinding. Pada saat berjalan robot masih sering menyentuh atau menabrak dinding dan robot juga kesulitan untuk meluruskan posisi (sejajar) dengan dinding. Akibatnya robot tidak akan dapat berjalan dengan baik, mengurangi efisiensi waktu, dan dapat menyebabkan kerusakan pada komponen dan mekanik robot. Oleh karena itu dibutuhkan suatu metode agar robot dapat berjalan menelusuri dinding dengan baik yaitu dengan menggunakan metode *wall follower*. *Wall follower* merupakan salah satu metode yang digunakan untuk navigasi robot [4].

Pada penelitian ini, metode *wall follower* akan diimplementasikan menggunakan sistem kendali PD. Sistem kendali ini terdiri dari kontroler proporsional dan derivatif. Elemen-elemen kontroler P bertujuan untuk mempercepat reaksi sebuah sistem (*rise time* yang cepat), sedangkan kontrol D bertujuan untuk mengurangi *overshoot* akibat aksi kontrol P. Sehingga akan didapatkan respon sistem yang baik, yaitu *rise time* yang cepat dan *overshoot* yang miniminal [5]. Penerapan algortima PD pada robot *wall follower* akan membuat robot dapat bernavigasi mengikuti dinding dengan lebih halus (*smooth*) [6]. Algortima ini akan digunakan pada robot berkaki enam (*hexapod)* yang dilengkapi dengan 7 buah sensor ultrasonik.

II. LANDASAN TEORI

A. Sistem Navigasi Wall Follower

Wall follower merupakan salah satu metode navigasi yang digunkan untuk menelusuri permukaan dinding. Penggunaan algoritma ini bertujuan untuk menjaga jarak robot terhadap dinding agar robot tetap berada di tengah-tengah lintasan dan menjaga agar posisi robot tetap sejajar dengan dinding selama robot bergerak maju.

Pada dasarnya ada dua metode telusur dinding pada navigasi *wall follower*, yaitu metode telusur dinding kianan dan telusur dinding kiri [7]. Jika menggunakan metode telusur dinding kiri, maka fungsi masukanya adalah membaca data jarak sensor yang ada di bagian kiri robot. Sebaliknya, jika menggunakan metode telusur dinding kanan, maka fungsi masukan yang dibaca adalah jarak sensor yang ada di bagian

kanan robot. Dalam penelitian ini, kedua metode tersebut akan diimplementasikan dengan menerapkan sistem kendali PD (*Proportional Derivative*).

B. *Sistem Kendali PD (Proportional Derivative)*

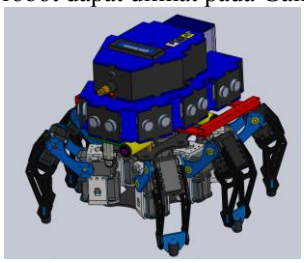
Gabungan aksi kontrol proporsional dan diferensial mempunyai keuntungan dibandingkan dengan masing-masing kontrol yang menyusunya. Keunggulan utamanya adalah diperolehnya keuntungan dari masing-masing aksi kontrol dan kekurangan dari masing-masing aksi kontrol yang satu dapat diatasi dengan kontrol yang lain. Kontroler P berfungsi untuk mempercepat reaksi sistem, sedangkan kontroler D berfungsi untuk menghilangkan *offset* [5]. Persamaan kontroler PD ini dapat dinyatakan pada persamaan berikut,

$$m(t) = Kp\ x\ e(t) + Kd\ \frac{de(t)}{dt}$$

III. PERANCANGAN SISTEM

A. *Perancangan Robot Hexapod*

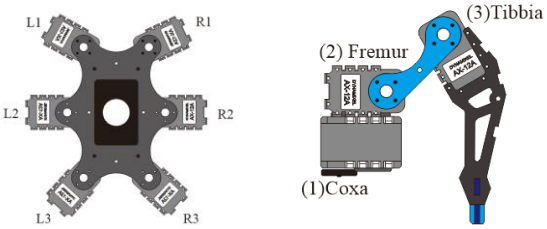
Dalam penelitian ini, desain robot yang dibuat adalah robot berkaki enam (*hexapod*) dimana pada masing-masing kaki didesain dengan tiga derajat kebebasan (3DOF). DOF merupakan sendi robot yang digabungkan dengan motor servo dan dibangun menyerupai kaki binatang [8]. Secara keseluruhan, desain robot dapat dilihat pada Gambar 1 berikut.



Gambar 1. Desain Robot Hexapod

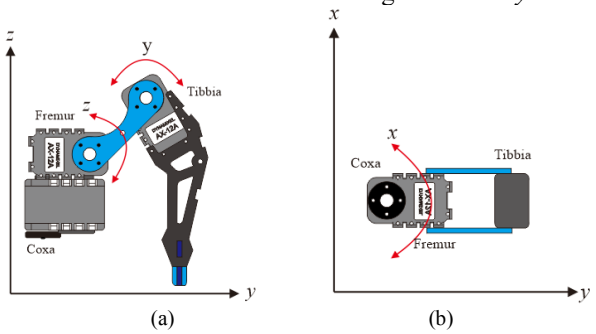
1. *Desain Kaki Robot*

Pada perancangan ini, kaki robot tersusun dari 18 buah motor servo yang dibagi menjadi 6 kaki. Masing-masing kaki terdiri dari 3 buah motor servo yang diberi nama (1) Coxa (2) Fremur (3) Tibbia. Desain mekanik kaki robot dapat dilihat pada Gambar 2. Dalam desain ini, jenis servo yang digunakan adalah Dynamixel AX-12A. Servo ini mampu berputar hingga sudut 300^0 [11].



Gambar 2. Mekanik Kaki Robot Hexapod Dengan 3 DOF (a) Desain Kaki (b) Susunan Servo Pada Kaki Robot

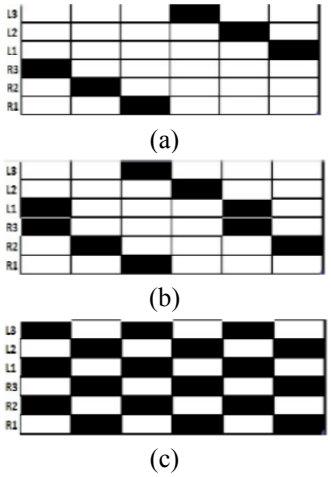
Arah gerakan servo *coxa, fremur* dan *tibbia* dalam diagram *Cartesian* dapat dilihat pada Gambar 3. Servo *coxa* akan membuat robot bergerak maju atau mundur, mereprentasikan sumbu "*x*" dari diagram Cartesian. Servo *fremur* akan membuat robot bergerak naik atau turun, merepresentasikan sumbu "*z*". Sedangkan Servo *tibbia* akan membuat robot bergerak ke kiri atau ke kanan sesuai dengan sumbu "*y*".



Gambar 3. Arah Gerakan Masing-Maing Servo (a) Tampak Samping (b) Tampak Atas

2. *Model Pergerakan Robot*

Model pergerakan dari robot berkaki disebut dengan *gait*, yaitu urutan kaki robot yang menyentuh lantai. Ada tiga macam tipe gait, yaitu (1) *metachronal gait*, (2) *ripple gait*, dan (3) *tripod gait* [8]. Gambar dari ketiga pola gait tersebut dapat dilihat pada Gambar 4. Warna putih mengindikasikan kaki yang menyentuh lantai.

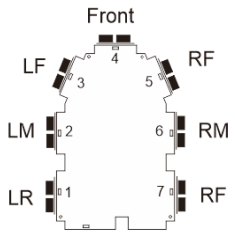


Gambar 4. Gait Diagram (a) Metachonal Gait (b) Ripple Gait (c) Tripod Gait

Mekanisme gerak dari *hexapod* dalam paper ini menggunakan *tripod gait*. *Gait* ini bekerja berdasarkan kestabilan robot pada tiga titik segitiga berupa kaki-kaki robot yang berada di tanah dalam satu waktu [2]. *Tripod gait* menghasilkan speed robot yang paling cepat [9].

3. *Desain Sensor*

Sensor ultrasonik yang digunakan sebanyak 7 buah. Jenis sensor ultrasonik yang digunakan menggunakan sensor ultrasonik SRF-04 [10]. Konfigurasi penempatan sensor ultrasonik dapat dilihat pada Gambar 5 berikut.



Gambar 5. Penempatan Sensor Ultrasonik

Desain penempatan sensor yang dibuat terdiri dari 7 buah, dimana dua sensor ditempatkan di depan dan di belakang robot, kemudian tiga sensor disamping kiri dan tiga sensor di samping kanan robot.

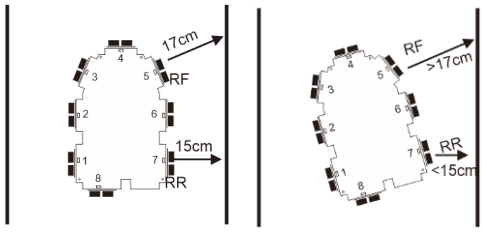
B. Perancangan Kendali PD

Penggunaan kendali PD ini bertujuan agar robot dapat meluruskan posisi sehingga tetap sejajar dengan dinding selama robot bergerak maju. Hal ini diperlukan agar ketika robot bergerak maju, sensor depan (*front*) tidak terhalang dinding sehingga robot tidak menabrak atau menyentuh dinding.

Ada dua mode yang digunakan dalam meluruskan dinding, yaitu mode scan kanan dan mode scan kiri.

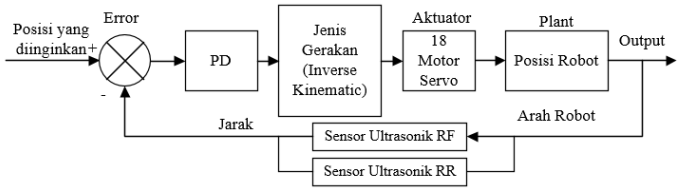
1. Mode Scan Kanan

Tujuan pada mode scan kanan ini adalah agar robot dapat berjalan dengan tetap menjaga jarak dengan dinding sejauh 15 cm.Ilustrasi robot meluruskan dinding kanan dapat dilihat pada Gambar 6 berikut.



Gambar 6. Ilustrasi
(a) Robot Lurus Terhadap Dinding
(b) Robot Tidak Lurus Terhadap Dinding

Pada kondisi ini diagram sistem PD untuk mode scan kanan digambarkan sebagai berikut :



Konstruksi robot yang didesain memberikan beda pembacaan sensor antara RF dan RR sebagaimana ditunjukkan pada Gambar 6. Pada Gambar 6a menggambarkan kondisi lurus kanan yang terjadi ketika jarak sensor RF = 17 cm dan jarak sensor RR = 15 cm. Dalam kondisi ini nilai error (e) harus bernilai 0, karena ketika kondisi lurus atau sejajar tidak ada error yang terjadi. Nilai *error (e)* diperoleh dari selisih antara jarak RF dengan jarak RR, sehingga pada kondisi ini *error (e)* adalah :

$$e = RF - RR = 17 - 15 = 2 \text{ cm.}$$

Karena pada kondisi sejajar nilai *e* harus sama dengan 0, maka nilai sensor RR harus ditambahkan dengan nilai *offset* agar setara denga nilai sensor RF. Nilai *offset* didapatkan dari selisih sensor RF dan sensor RR, yaitu sebesar 2cm. Sehingga nilai *error(e)* untuk mode scan kanan dapat dihitung dengan rumus :

$$e_{kanan} = RF - (RR + offset)$$
$$e_{kanan} = RF - (RR + 2)$$

Agar dapat diimplementasikan pada mikrokontroler, persamaan PD *analog* harus diubah kedalam bentuk *digital*. Persamaan PD dalam bentuk digital dituliskan sebagai berikut :

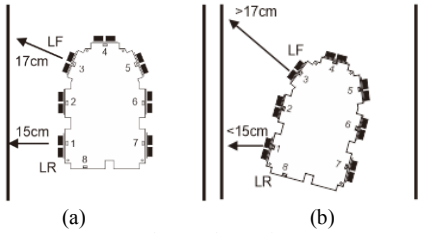
$$PD_{lurus_kanan} = (Kp \,.\, e_{kanan}) + \frac{Kd}{Ts}(e_{kanan} - last_e_{kanan})$$

Keterangan :

PD_{lurus_kanan}= nilai kendali PD
e_{kanan} = nilai error scan kanan
Kp = Konstanta Proportional
Kd = Konstanta Derivative
Ts = Time sampling (ms)

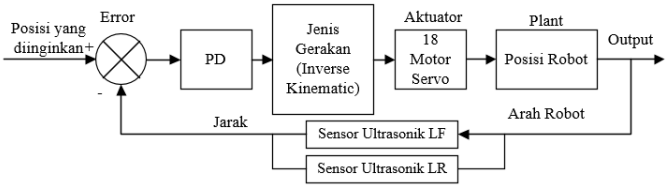
2. Mode Scan Kiri

Ilustrasi robot meluruskan dinding kiri dapat dilihat pada Gambar 7 berikut.



Gambar 7. Ilustrasi
(a) Robot Lurus Terhadap Dinding
(b) Robot Tidak Lurus Terhadap Dinding

Pada dasarnya pada mode scan kiri, metode yang digunakan sama dengan mode scan kanan. Perbedaanya adalah pada input sensor yang digunakan. Pada mode scan kiri jarak yang digunakan sebagai input sistem PD adalah jarak LF dan LR. Diagram sistem PD untuk mode scan kiri digambarkan sebagai berikut :



Nilai error (e) diperoleh dari selisih antara jarak LF dengan jarak LR. Dalam kondisi lurus nilai jarak LF dan jarak LR memiliki selisih sebesar 2 cm. Nilai selisih ini digunakan sebagai nilai *offset.* Perhitungan nilai error (e) dituliskan sebagai berikut :

$$e_{kiri} = \text{Jarak LF} - (\text{Jarak LR} + offset)$$
$$e_{kiri} = \text{Jarak LF} - (\text{Jarak LR} + 2)$$

Persamaan PD_{lurus_kiri} dapat dituliskan sebagai berikut :

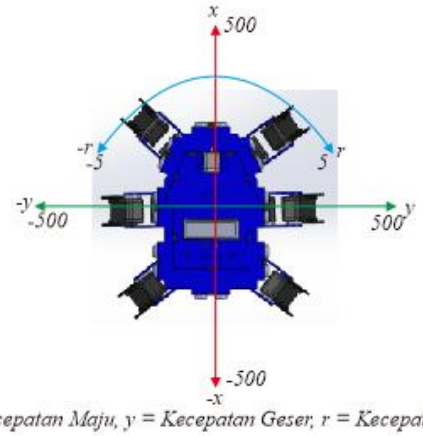
$$PD_{lurus_kiri} = (Kp \,.\, e_{kiri}) + \frac{Kd}{Ts}(e_{kiri} - last_e_{kiri})$$

Keterangan :

PD_{lurus_kiri}= nilai kendali PD
e_{kiri} = nilai error scan kiri
Kp = Konstanta *Proportional*
Kd = Konstanta *Derivative*
Ts = *Time sampling* (ms)

C. *Perancangan Output*

Hasil dari perhitungan nilai PD akan digunakan sebagai acuan nilai untuk menentukan jenis pergerakan robot. Pergerakan robot diatur dengan menggunakan rumus *inverse* kinematik. *Inverse* kinematik merupakan metode yang digunakan untuk mengatur gerak kaki dengan menghitung nilai sudut pada masing-masing servo. Pola pergerakan robot ditentukan oleh jenis gait yang digunakan. Pada perancangan ini, pola pergerakan yang dipilih menggunakan *Tripod Gait*. Hasil perhitungan *gait* inilah yang kemudian akan menentukan pola pergerakan robot. Output yang dihasilkan dari perhitungan gait ini berupa tiga parameter *x, y,* dan *r*. Paramater *x* merupakan parameter yang digunakan untuk mengatur kecepatan maju, parameter *y* digunakan untuk mengatur kecepatan geser, dan parameter *r* digunakan untuk mengatur kecepatan rotasi robot.



Gambar 8. Arah Pergerakan Robot

Pada perancangan ini, nilai *x* dan *y* merupakan nilai konstan yang digunakan sebagai set point awal kecepatan robot. Namun karena dalam perancangan ini gerakan geser tidak digunakan, maka nilai $y = 0$. Nilai *r* didapatkan dari hasil perhitungan PD. Secara matematis persamaan output sistem dapat dituliskan sebagai berikut :

(1) Penentuan nilai parameter x (mm/s):
$x = n$
Dimana :
n = set point kecepatan maju (mm/s)

(2) Penentuan nilai parameter y (mm/s):
$y = 0$

(3) Penentuan nilai parameter r (rad/s):
Karena rentang nilai *r* adalah dari -5 – 5 rad/s, maka hasil output PD harus dibagi dengan 10, sehingga:
$r = PD / 10$

Karena terdapat dua mode yang digunakan dalam scanning dinding, maka untuk mode scan kanan, nilai *r* diperoleh hasil dari PD_{lurus_kanan}, sehingga :

$$r = PD_{lurus_kanan} / 10$$

Sedangkan untuk mode scan kiri, nilai r diperoleh dari hasiil PD_{lurus_kiri}, sehingga:

$$r = PD_{lurus_kiri} / 10$$

IV. PENGUJIAN DAN ANALISIS DATA

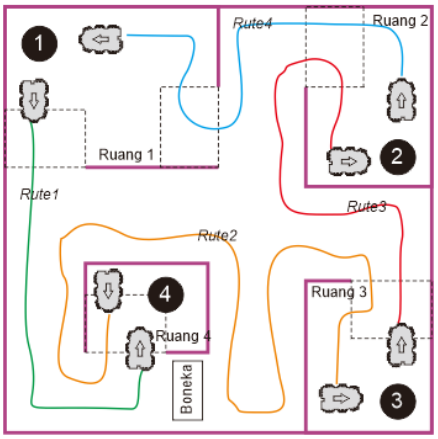
Pada tahapan ini, pengujian dilakukan menjadi tiga bagian yaitu (1) Tuning (2) pengujian robot dalam mode scan kanan dan (2) pengujian robot dalam mode scan kiri. Arena yang digunakan dalam pengujian merupakan arena yang digunakan pada Kontes Robot Pemadam Api Indonesia. Prosedur pengujian dilakukan dengan meletakkan robot pada tempat start yang telah ditentukan. Kemudian robot dijalankan untuk memutari arena dengan mode scan kanan atau scan kiri hingga robot kembali ke titik start.

A. *Tuning Paremater PD*

Penentuan nilai parameter PD diperoleh dengan melakukan *trial and error* agar repon yang diperoleh sesuai dengan harapan. Metode *tuning* ini dibutuhan untuk memberikan nilai Kp dan Kd pada kendali PD. Berdasarkan hasil *tuning*, diperoleh kondisi stabil robot dengan nilai Kp = 30 dan Kd = 1.

B. *Pengujian Robot Mengikuti Dinding Kanan*

Pengujian ini bertujuan untuk mengetahui performa sistem kontrol PD dari parameter Kp dan Kd yang telah ditentukan pada proses *tuning*. Arena pengujian dan jalur lintasan pengujian pada mode scan kanan ditunjukkan pada Gambar 9.



Gambar 9. Lintasan Robot Pada Mode Scan Kanan



Gambar 10. Penempatan Halangan Boneka

Pengujian robot dilakukan sebanyak 20 kali pada masing-masing konfigurasi *start* dan *finish*. Pada mode scan kanan, ada 4 rute yang telah disediakan seperti ditunjukkan pada Gambar 8. Hasil pengujian robot dengan mode *scan* kanan dapat dilihat pada Tabel 1.

Tabel I. Hasil Pengujian Robot Mengikuti Dinding Kanan

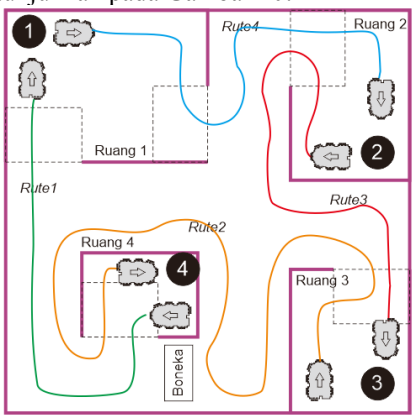
Start	Finish	Berhasil	Rata-Rata Benturan	Rata-Rata Waktu (s)
1	4	17 kali	0,4	21,4
4	3	15 kali	1,35	49,5
3	2	20 kali	0,35	28,1
2	1	20 kali	0,45	33,2

Berdasarkan pengujian yang telah dilakukan, tingkat keberhasilan dan jumlah benturan pada masing-masing konfigurasi ruang start dan finish diperoleh hasil yang berbeda-beda. Dengan menerapkan kendali PD, untuk konfigurasi start dan finish dari ruang 3 ke ruang 2, dan ruang 2 ke ruang 1 tingkat keberhasilanya mencapai 100% dengan beberapa kali benturan dengan dinding. Untuk konfigurasi *start* ruang 1 dan *finish* ruang 4, robot mengalami 3 kali kegagalan dengan rata-rata benturan 0,4. Sedangkan untuk konfigurasi start ruang 4 dan finish ruang 3, robot mengalami 5 kali kegagalan dengan rata-rata benturan 1,3 kali dari 20 kali percobaan.

Berdasarkan analisa yang dilakukan, faktor utama penyebab kegagalan robot adalah adanya halangan berupa boneka yang ditempatkan dilintasan. Adanya boneka membuat pembacaan data sensor ultrasonik menjadi tidak valid dikarenakan boneka terbuat dari bahan yang tidak padat. Akibatnya data jarak yang digunakan sebagai input kendali PD tidak sesuai dengan jarak sebenarnya, sehingga menyebabkan robot tersangkut di dinding dan gagal menyelesaikan misi. Secara keseluruhan tingkat keberhasilan implementasi kendali PD pada mode scan kanan sebesar 90%.

C. *Pengujian Robot Mengikuti Dinding Kiri*

Pengujian ini bertujuan untuk mengetahui performa sistem kontrol PD jika digunakan dalam mengikuti dinding kiri (scan kiri). Arena pengujian dan jalur lintasan pengujian pada mode scan kanan ditunjukkan pada Gambar 10.



Gambar 11. Lintasan Robot Pada Mode Scan Kiri

Sama seperti pengujian pada mode scan kanan pengujian robot pada mode scan kiri dilakukan sebanyak 20 kali dengan 4 konfigurasi strat seperti ditunjukkan pada Gambar 10. Hasil pengujian robot mengikuti dinding kiri dapat dilihat pada Tabel 2.

Tabel II. Hasil Pengujian Robot Mengikuti Dinding Kiri

Start	Finish	Berhasil	Rata-Rata Benturan	Rata-Rata Waktu (s)
4	1	20 kali	0,2	19,4
3	4	17 kali	1,05	48,8
2	3	19 kali	0,45	25,3
1	2	20 kali	0,3	31,6

Pada mode *scan* kiri, robot juga mengalami beberapa kegagalan sehingga tidak dapat menuju titik finish yang telah ditentukan. Kegagalan terbanyak terjadi pada lintasan start ruang 3 dan finish ruang 4. Robot tidak dapat mendeteksi halangan boneka dengan baik sehingga tersangkut di dinding dan gagal menyelesaikan misi. Namun, pada rute1, rute 3, dan rute 4, robot sudah dapat bernavigasi dengan baik. Secara keseluruhan, tingkat keberhasilan implementasi kendali PD pada mode scan kiri sebesar 95% dengan rata-rata benturan 0,5 kali dari 80 kali percobaan.

V. KESIMPULAN

Pada penelitian ini sistem *wall follower* pada robot hexapod dirancang menggunakan kendali PD. Berdasarkan hasil pengujian terlihat bahwa dengan mengimplementasikan sistem kendali PD, robot berhasil mengikuti dinding dengan presentase keberhasilan 90% pada mode *scan* kanan, dan 95% pada mode scan kiri.

Berdasarkan hal tersebut dapat disimpulkan bahwa penggunaan kendali PD cukup efektif digunakan pada sistem navigasi wall follower robot *hexapod*. Kondisi dinding dan halangan yang ada dilintasan sangat mempengaruhi *performance* sensor yang berpengaruh pada gerakan robot yang dihasilkan. Pengujian dan percobaan masih harus tetap dilakukan, terutama dalam menentukan *tuning* kontrol PD nya, sehingga nantinya robot *hexapod* dapat bernavigasi tanpa menyentuh atau menabrak dinding.

DAFTAR PUSTAKA

[1] Rendyansyah, K. Exaudi, A. P. P. Prasetyo, "Navigasi Berbasis Behavior dan Fuzzy Logic Pada Simulasi Robot Bergerak Otonom", Jurnal Nasional Teknik Elektro, Vol. 5, No. 1, hal 136 – 144, 2016.

[2] H. Wicaksono, Prihastono, K. Anam, E. Rusdhianto, I. A. Sulistijono, S. Kuswadi, A. Jazidie, M. Sampei, "Perancangan Sistem Navigasi Otonom Pada Behavior Based Hexapod Robot", Jurnal Teknik Elektro, Vol. 8, No. 2, hal 70 – 78, 2008.

[3] DIKTI. 2017. *Panduan Kontes Robot Pemadam Api Indonesia 2017*. Jakarta : DIKTI.

[4] Hasri Awal, "Algoritma Fuzzy Logic dan Wall Flollower Pada Sistem Navigasi Hexapod Berbasis Mikrokontroler AVR". Jurnal Teknologi Informasi dan Pendidikan, Vol. 9, No. 1, hal 102 – 107, 2016.

[5] M. D. Nurmansyah, Supriyono, "Autotuning Parameter Kendali PD dengan Tsukamoto Fuzzy Menggunakan Bahasa C", Seminar Nasional Aplikasi Teknologi Informasi (SNATI), hal 172 – 177, 2011.

[6] U. D. Fatmawati, W. Hidayat, D. Lelono, "A Novel Method of Mobile Robot Fire Detection and Tracing Using Proportional Derivative (PD) Algorithm", Applied Mechanics and Materials, Vol. 771, hal 68 – 71, 2015.

[7] S. B. Suharto, Purwanto, G. D. Nusantoro, "Sistem Navigasi Wall Following Robot KRPAI Divisi Berkaki Menggunakan Kontroler PID". Makalah Seminar Hasil.
[8] S. Sendari, H. Elmunsyah, Y. D. Mahandi, "Six-Legged Animaloid Robot as a Trainer for Robotic Course". Technical an Vocational Education and Training (TVET), hal 199 – 202, 2015.
[9] R. Campos, Matos, Vitor; Oliviera, Miguel; and Santos, Cristina. "Gait Generation for A Simulated Hexapod Robot: A Nonlinier Dynamical System Approach". Research Report of The Portuguese Science Foundation (grant PTDC/EEA-CRO100655/2008).
[10] *Datasheet SRF-04 Ultrasonic Ranger*, https://www.robot-electronics.co.uk/htm/srf04tech.htm, diakses 12 Juni, 2017.
[11] *Datasheet Dynamixel AX-12A Smart Serial Servo,* http://www.robotshop.com/content/PDF/datasheet-902-0003-001.pdf, diakses 12 Juni, 2017.

Design and Implementation Software in Indonesian Traditional Dancing Humanoid Robot Development Platform

Desiana Dien Nurchalifah*, Kusprasapta Mutjiarsa, dan Widyawardana Adiprawita
Electrical Engineering Departement, School of Electrical Engineering and Informatics
Bandung Institute of Technology
Bandung, Indonesia
desiana.nurchalifah@gmail.com

***Abstract* – This paper discussed the results of design and implementation software and user interface used in building dance movements of humanoid robot. Sequences of dancing movements are created from several basic set of movements which had been planted in this software. Basic dance movements including hand, legs, neck, and hip in the humanoid robot with 25 degree of freedom (DOF). This development platform software of dancing movements built in low level and high level. On the side of low level, it is designed ability to control every servo motor in humanoid robot using forward kinematic. On the high-level side, graphical user interface is built as a composer of dancing movements set practically. Set of dancing movements in which had created in the software platform on the computer, then sent to robot by communication module that had been made. Subsequently, humanoid robot would dance based on sequences recieved. In the implementation, dancing movements created are two traditional dances in Indonesia, which are Topeng Betawi dance and Gending Sriwijaya dance. Based on results, development software platform created could compose and control six sets of Gending Sriwijaya dancing movements to dance in the duration of four minutes. This software platform created to alleviate humanoid robot dances developer to build new dance movements. Dance movements is not to be programmed based on text.**

Keywords—Software development platform, humanoid robot, traditional dancer robot, GUI, low level controller, high level controller, interface, communication.

I. INTRODUCTION

Tourism is one of the most important sectors in many countries in general, and Indonesia in particular. Divided into 17,504 islands and 34 provinces, making Indonesia has unique and variety of art and culture that shapes into an Indonesian characteristic. The manifestation of copyrights and works is shaping the identity of the Indonesian nation. Culture can be viewed as an individual identity in a nation that reflects the diversity of the nation by providing insight and meaning to national development. With the development of technology in Indonesia, the next generation prefer to leave the conservation of local culture, so the culture in Indonesia is extinct. One form of technological development that is renewable humanoid robot. Humanoid robot is a robot designed to resemble humans. Humanoid robot is made with a design to perform certain functions, one of which is dancing. Based on BPS data in August 2015, the tourism sector in Indonesia continues to increase [1]. This challenges researchers to combine technology and the culture value of Indonesia in developing especially in the tourism sector.

Programming on the humanoid robot that has been used until now has run well in controlling the robot, but the complexity of programming caused limitations to whom using the humanoid robot. This could decrease the development of education about humanoid robots in certain communities. The complex system of humanoid robots requires in-depth knowledge to operate the robot, so it requires an intermediary between the humanoid robot and the user. One alternative is to create humanoid robot control software to perform traditional Indonesian dance movements.

Software designed using secure shell connection between user and robot. With software, the complexity of robotic programming that previously needed to understand low-level language is represented as a button containing the basic movements of traditional Indonesian dance. Software created has a class that controls the framework of the robot in order for the robot to execute movements or the intended action in accordance with commands sent by the user. The robot framework runs forward kinematics so that the robot can move 25 servo pieces in accordance with traditional dance moves. In addition, coordination between dancers is required by communication between two dance robots. For example, to synchronize movements between robots, communication process of waiting the other robot is needed to make the movements created in harmony with each other.

II. LITERATURE STUDIES

A. Humanoid Robot

The development of humanoid robots such as ASIMO [2] and HUBO [3] has various research in the field of robotics for example motion planning, communication, manipulation, and walking. Various platforms are developed to achieve the ideal state of research, which concludes the manufacture of humanoid robots to focus on modifiable systems, high performance, easy repairs, and also has a common

environment at an affordable price. One of the platforms that can be used is the DARwIn-OP platform.

DARwIN-OP (Dynamic Anthomorphic Robot with Intelligence - Open Platform) is a platform developed by Virginia Tech's RoMeLa (Robotics & Mechanism Laboratory) laboratory in collaboration with University of Pennsylvania, Purdue University, and South Korean company called Robotis. The developed platform has high computing capabilities. Due to the development of dynamic robot capabilities, this platform is well suited for research and education.

This platform has a total of 20 degrees of freedom, with a distribution of 12 pieces on the foot, 6 on the hand, and 2 on the neck and head. This platform has much in common with the Bioloid GP platform, but it has more DOFs and has a more robust frame with aluminum usage as a whole frame material. This platform uses single board computer technology (SBC) as the main data processor, so this system has the capability of qualified to perform data processing with good performance.

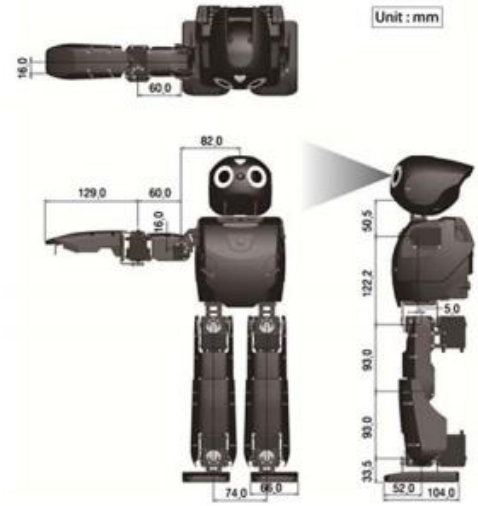


Figure 1. DARwIn-OP Platform[4]

B. Object Oriented Programming

Object-oriented programming is organizing program to combine data and functionality into an object. In a development platform that requires a program with complex structures, it is necessary to use object oriented programming techniques. In object-oriented there are two important aspects, which are class and object. The class creates a new type with the object as an instance. Classes have attributes and methods (function that exist in the class). Methods within the class can be private, public or protected. [5] While objects can store data using common variables. Objects can have functionality by using functions in a class (method). Objects in a class can be used as an input or output parameters from the use of library. [6] In the development platform, the main program able to use the library in variety amount, so that objects in the library could be related to each other in accordance with the main purpose of programming. Examples of object-oriented programming can be done using C ++ and Python.

C. *TCP/IP Communication Protocol*

TCP provides services with establishing connection. A logical connection is established between the original station and transmitter. Features on TCP connections such as error and flow control make TCP a connection with the preference of data transfer. TCP receives data from the application layer, adds the data needed to get a reliable operation, then sends it to the IP. TCP connects port numbers with specific applications, generates the number of each byte in the data stream, calculates the timeout to ensure undetected frames, and uses certain messages to create and stop duplex communications. TCP IP is a point-to-point connection with communication from a computer on a network to another computer. Connection using TCP is a non-state connection due to a client's request at a given time has no correlation with any other time. The process stack using TCP/IP described as follows.

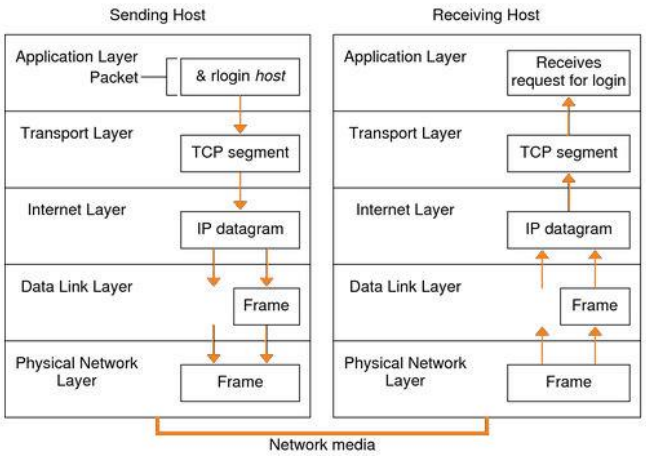


Figure 2. TCP/IP Protocol Diagram[7]

III. IMPLEMENTATION AND DESIGN

The design and implementation consisted of Graphical User Interface and communication module which results as below

- Implementation and Design of Graphical User Interface (GUI)

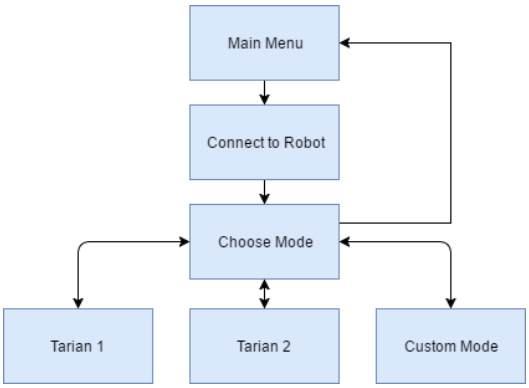


Figure 3. GUI Block Diagram

The created application has six windows that are main menu, connect to robot, selection of desired mode, dance example 1, dance 2 example, and free mode.

• Design and Implementation of communication modules

TCP communication system is used in communication design between two humanoid robots that are used. One humanoid robot sends data packets using IP addresses. One of the humanoid robot is made into a server of TCP connection and the other robot become the client on the same connection. The design procedure performed by the humanoid robot on server is shown in the diagram as follows

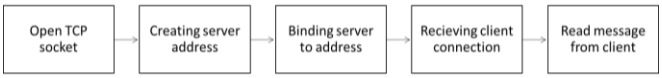


Figure 4. Server Procedure

While the procedure design performed on client is shown in the diagram as follows



Figure 5. Client Procedure

IV. RESULTS

Graphical user interface robot test results are as follows

• Secure Shell Connection Testing

In this test is performed performance testing of the software by measuring the speed of the connection through the router under normal conditions. The results of this test can be a reference to evaluate the software. Test results are as follows.

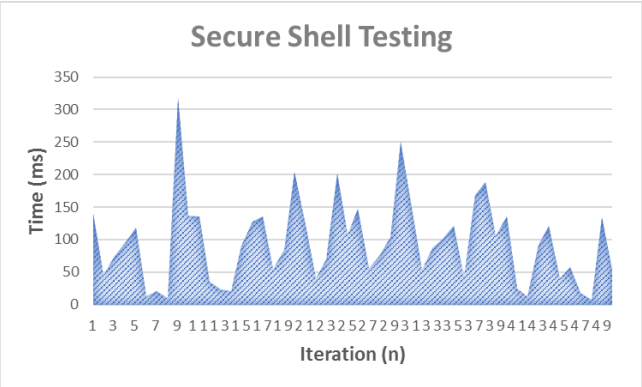


Figure 6. Secure Shell Connection Testing

On connection test performed has been successful with the average length of time required in the connection is 95.54ms

• Secure Shell Speed Test

In this test, performance testing of the software is done by measuring the speed opening a Secure Shell connection. The results of this test could be a reference to evaluate the software. Test results are as follows.

```
================ RESTART: D:\apps\buatTA\results\ref\ssh_3.py ================
Connecting to server on ip 192.168.124.4.
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 3.10.103 armv7l)

 * Documentation:  https://help.ubuntu.com/

Last login: Tue Jan  1 00:19:36 2013 from 192.168.124.8
odroid@odroid:~$
$
 Waktu yang diperlukan untuk koneksi adalah :
1.8348820209503174
detik
```

Figure 7. SSH Speed Testing 2

From the test results with Secure Shell, the user interface was able to successfully connect with the duration of 1-2 seconds after several data logging to test the connection.

• Testing Low Level Program Executions

This test is measuring the software performance by testing the speed of executed low level program in humanoid robot using Secure Shell. The results of the test could be reference to the software evaluation. The results are as follows.

```
 Waktu yang diperlukan untuk eksekusi program adalah :
7.033968925476074
detik

>>> [H[J
[Action Editor for DARwIn v1.00]

 *Terminal screen size must be 80(col)x30(row).
 *Current terminal has 80 columns and 24 rows.


Press any key to start program...
[H[JID: 1(R_SHO_PITCH)[    ]

ID: 2(L_SHO_PITCH)[    ]                              Page Number:
ID: 3(R_SHO_ROLL) [    ]                               Address:
ID: 4(L_SHO_ROLL) [    ]                                 Play Count:
ID: 5(R_ELBOW)    [    ]                                  Page Step:
ID: 6(L_ELBOW)    [    ]                                 Page Speed:
ID: 7(R_HIP_YAW)  [    ]                                 Accel Time:
ID: 8(L_HIP_YAW)  [    ]                               Link to Next:
ID: 9(R_HIP_ROLL) [    ]                               Link to Exit:
```

Figure 8. Low Level 1 Testing

```
 Waktu yang diperlukan untuk eksekusi program adalah :
7.036706924438477
detik

>>> [H[J
[Action Editor for DARwIn v1.00]

 *Terminal screen size must be 80(col)x30(row).
 *Current terminal has 80 columns and 24 rows.


Press any key to start program...
[H[JID: 1(R_SHO_PITCH)[    ]

ID: 2(L_SHO_PITCH)[    ]                              Page Number:
ID: 3(R_SHO_ROLL) [    ]                               Address:
ID: 4(L_SHO_ROLL) [    ]                                 Play Count:
ID: 5(R_ELBOW)    [    ]                                  Page Step:
ID: 6(L_ELBOW)    [    ]                                 Page Speed:
ID: 7(R_HIP_YAW)  [    ]                                 Accel Time:
ID: 8(L_HIP_YAW)  [    ]                               Link to Next:
ID: 9(R_HIP_ROLL) [    ]                               Link to Exit:
ID:10(L_HIP_ROLL) [    ]
ID:11(R_HIP_PITCH)[    ]
```

Figure 9. Low Level 1 Testing

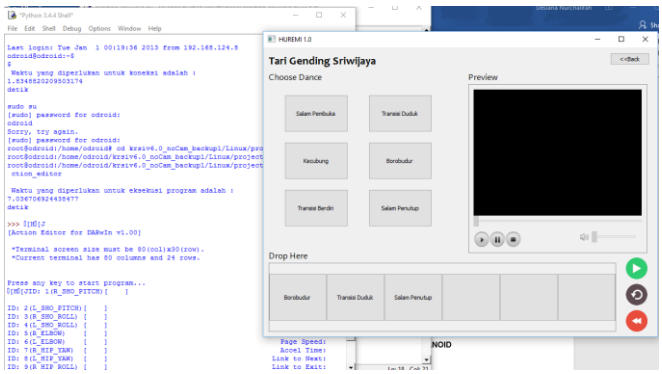


Figure 10. Command Testing

From the test results, the user interface was able to successfully measure the speed of program execution on the humanoid robot with the required length of time is in the range 7-8 seconds. This execution causes time delays on commands sent by the software.

• Testing Every Button on The Interface

Testing each software button produces as follows

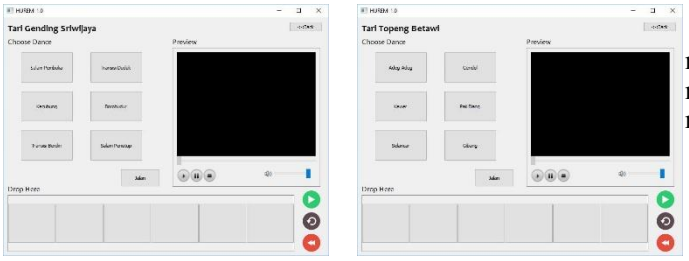


Figure 11. Gending Sriwijaya (left) and Topeng Betawi (right)

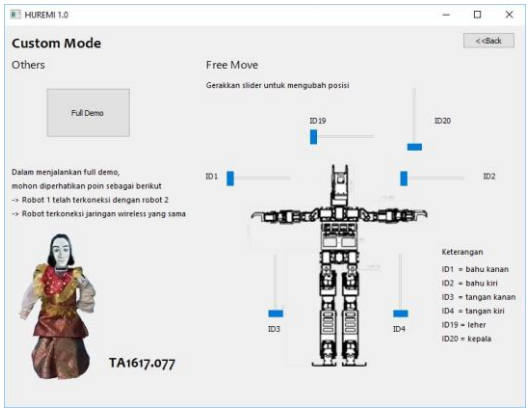


Figure 12. Custom Window



Figure 13. Robot Dancing Results

From the test results, the software can perform its function in giving commands to the robot.

In addition to testing the software, communication module testing is also done with the following results

• Packet loss testing

The test results of the humanoid robot communication module is the long-lasting wireless connection using wireless module and the time required for communication between robots with the following results

```
64 bytes from 192.168.124.4: icmp_seq=88 ttl=64 time=1314 ms
64 bytes from 192.168.124.4: icmp_seq=89 ttl=64 time=395 ms
64 bytes from 192.168.124.4: icmp_seq=90 ttl=64 time=30.8 ms
64 bytes from 192.168.124.4: icmp_seq=91 ttl=64 time=18.5 ms
64 bytes from 192.168.124.4: icmp_seq=92 ttl=64 time=28.1 ms
64 bytes from 192.168.124.4: icmp_seq=93 ttl=64 time=24.5 ms
64 bytes from 192.168.124.4: icmp_seq=94 ttl=64 time=20.9 ms
64 bytes from 192.168.124.4: icmp_seq=95 ttl=64 time=1033 ms
64 bytes from 192.168.124.4: icmp_seq=96 ttl=64 time=42.3 ms
64 bytes from 192.168.124.4: icmp_seq=97 ttl=64 time=33.6 ms
64 bytes from 192.168.124.4: icmp_seq=98 ttl=64 time=1013 ms
64 bytes from 192.168.124.4: icmp_seq=99 ttl=64 time=21.3 ms
64 bytes from 192.168.124.4: icmp_seq=100 ttl=64 time=19.8 ms

--- 192.168.124.4 ping statistics ---
100 packets transmitted, 98 received, 2% packet loss, time 99209ms
rtt min/avg/max/mdev = 17.513/320.427/2678.101/538.230 ms, pipe 3
```

Figure 14.7 Robot 1 Testing Communication Data

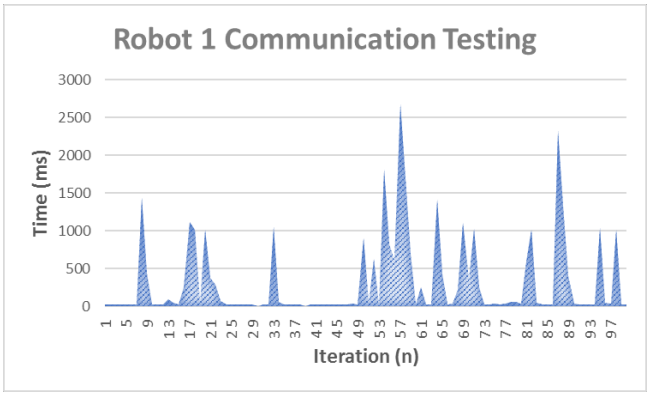


Figure 15. Robot 1 Testing Communication Graph

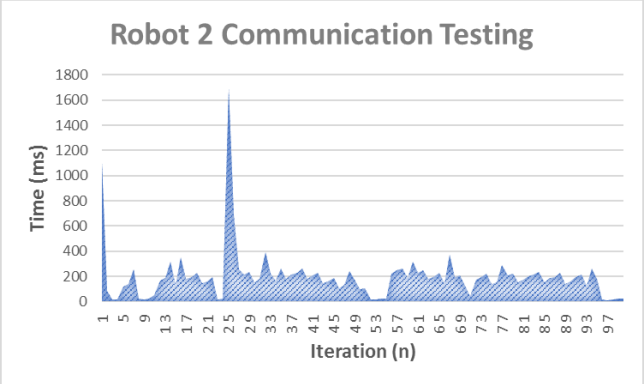


Figure 16. Robot 2 Testing Communication Graph

On successful testing the communication between the two robots with the average time required is 257.5 ms in a single command.

• Testing communication between robots

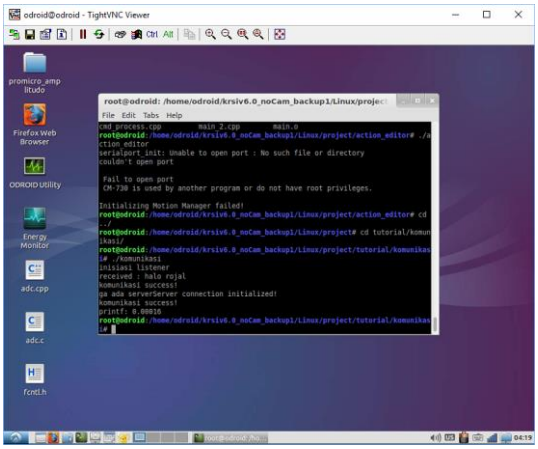


Figure 17. Communication Command Speed 1

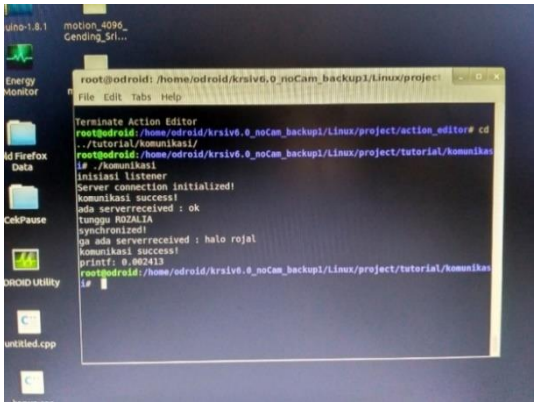


Figure 18. Communication Command Speed 2

From the test results, communication between robots directly in giving orders or receipts generate time in the range 0-2 seconds.

V. CONCLUSION

This paper discussed the design and implementation software to develop dance moves on traditional Indonesian humanoid robot dancer. Some conclusions that can be drawn are as follows:

1) Software development platform in humanoid robot dancer had designed and implemented using GUI. The interface had transformed from text-based into GUI which has succeeded in reducing complexity in developing humanoid robot moves, specifically in creating traditional Indonesian dance moves.

2) Software is created using PyQt which is cross-platformed, so that the software can be widely used on computer with different background of operating systems

3) Software is created to control movements of the robot on the sub-controller by using forward kinematics

4) Communication protocol between humanoid robot can be done with TCP protocol. The total delay time on the measured TCP communication protocol is 2 seconds

ACKNOWLEDGEMENTS

This paper is created solely for the author's Final Documentation. Acknowledgments are conveyed by the authors to Team TA 161701077 and Team Dago Concordia who participated in the manufacture and development of the traditional dancing humanoid robot platform used in this paper.

REFERENCES

[1] https://www.bps.go.id/linkTabelStatis/view/id/1387, (diakses 24 April 2017)

[2] M. Hirose and K. Ogawa, *Honda Humanoid Robots Development*, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 365, issue 1850, pp. 11-19, 2007.

[3] I.W. Park, J. Y. Kim, J. Lee, and J. H. Oh, Mechanical Design of Humanoid Robot Platform KHR-3 (KAIST Humanoid Robot – 3:HUBO),International Conference on Humanoid Robots, pp. 321-326, 2005

[4] https://support.robotis.com/en/product/darwin-op.htm, (diakses 24 April 2017)

[5] J. Farrell, Object-Oriented Programming Using C++, 4th ed. Boston: Course Technology, Cengage Learning, 2009, pp.455–467.

[6] M. Lutz, Learning Python, 4th ed. Boston: O'Reilly Media, 2009, pp.533-540.

[7] Sun Microsystems, *System Administration Guide: IP Services*, California : Network Circle Santa Clara, 2002, p.40

Implementasi Filter Morfologi Untuk Menghilangkan Noise Objek Pada Robot Sepak Bola

Muhammad Fadlullah*, Dony Ainur Darussalam, Moh Maulana, M. Rovicky, Masita Erfina Hadi, Yuni Syafarinda, Yogiswara
Jurusan Teknologi Informasi, Jurusan Kesehatan Politeknik Negeri Jember
Jember, Indonesia
muhammadfadlullah9@gmail.com

***Abstrak* - Robot sepak bola mendeteksi bola berdasarkan bentuk atau warna bola. Agar dapat bermain dengan benar,sebuah robot soccer harus dilengkapi dengan sensor deteksi bola yang baik. Sensor yang dibutuhkan adalah salah satunya adalah kamera Webcam. Untuk mendeteksi bola berwarna orange, sensor kamera harus melakukan beberapa tahapan image processing, dari segementasi HSV hingga filtering morfologi. Operasi morfologi merupakan operasi yang umum dikenakan pada citra biner (hitam-putih) untuk mengubah struktur bentuk objek yang terkandung dalam citra. Filter morfologi dapat menghilangkan noise yang dapat mengganggu sistem deteksi bola.**

***Katakunci* - *Filter* Morfologi, *Image Processing*, HSV**

I. PENDAHULUAN

Kontes Robot Sepak Bola Indonesia (KRSBI) merupakan salah satu program kreatifitas mahasiswa di bawah kegiatan induk Kontes Robot Indonesia (KRI). Sedangkan Robot Sepak Bola (robot*soccer*) adalah salah satu bentuk implementasi teknologi dalam bidang robotika yang memiliki kemampuan menirukan salah satu atau beberapa kegiatan manusia.

Robot *soccer* mendeteksi sebuah bola berdasarkan bentuk dan warna bola tersebut. Sensor yang dibutuhkan adalah kamera web (*webcam)* sehingga dibutuhkan suatu sistem kendali secara visual. Pada prinsipnya tujuan dari kendali visual pada robot soccer ini diterapkan agar robot memiliki kecerdasan dalam mengenali objek berupa bola, pemrosesan gambar yang dilakukan yaitu mendeteksi objek menggunakan metode *filter morfologi* yang mana merupakan metode mencocokkan warna yang sesuai dengan bentuk benda apapun serta menentukan titik tengah dari objek yang terdeteksi. Informasi lokasi titik tengah pada objek merupakan informasi navigasi sehingga menghasilkan kondisi-kondisi yang direspon oleh robot dengan cara bergerak ke arah objek tersebut.

II. PEMBAHASAN

A. *Image Processing*

Menurut Murni (1992), *image processing* (pengolahan citra digital) merupakan peningkatan mutu citra yang bertujuan untuk memperbaiki mutu citra untuk memperoleh keindahan gambar, untuk kepentingan analisa citra, dan untuk mengkoreksi citra dari segala gangguan yang tejadi sewaktu perekaman data. Pengolahan Citra bertujuan memperbaiki kualitas citra agar mudah diinterpretasioleh manusia atau mesin (dalam hal ini komputer). Citra dalam perwujudannya dapat bermacam-macam, mulai dari gambar hitam putih dalam sebuah foto (yang tidak bergerak) sampai pada gambar berwarna yang bergerak. Proses transformasi dari bentuk tiga dimensi ke bentuk dua dimensi untuk menghasilkan citra akan dipengaruhi oleh bermacam-macam faktor yang mengakibatkan penampilan citra suatu benda tidak sama persis seperti bentuk fisik nyatanya.

B. *Citra warna RGB*

Pemodelan warna digunakan untuk menggambarkan sifat-sifat warna secara matematis, sehingga dengan demikian dapat dilakukan pengolahan citra digital dengan cara memanipulasi nilai-nilai warna dari citra tersebut secara matematis. Model warna yang paling banyak digunakan di bidang computer maupun elektronika adalah model RGB(red-green-blue). Model RGB ini cocok digunakan dalam implementasi perangkat keras elektronik dan penyimpanan data secara digital. (Anymous,2005).

Di dalam model warna RGB, setiap warna dapat didefinisikan dalam tigakomponen warna yaitu komponen R (merah), G (hijau), dan B (biru). Warna-warnayang lain dapat diperoleh dengan melakukan kombinasi dari masing-masingkomponen R, G, dan B dengan nilainya masing-masing. Dalam penerapannya dibidang komputer, harga dari tiap-tiap komponen R, G, dan B berkisar dari 0 sampai255

C. *Model warna HSV*

Model warna HSV mendefinisikan warna dalam *terminology Hue*, *Saturation* dan *Value*. *Hue* menyatakan warna sebenarnya, seperti merah, violet, dan kuning dan digunakan menentukan kemerahan (redness), kehijauan (greeness), dsb. *Saturation* kadang disebut chroma, adalah kemurnian atau kekuatan warna. *Value* merupakan kecerahan dari warna. Nilainya berkisar antara 0-100 %. Apabila nilainya 0 maka warnanya akan menjadi hitam, semakin besar nilai maka semakin cerah dan muncul variasi-variasi baru dari warna tersebut. Keuntungan HSV adalah terdapat warna-warna yang sama dengan yang ditangkap oleh indra manusia.

D. *Color Filtering*

Color Filtering adalah suatu teknik pengolahan citra yang yang dipakai untuk memanipulasi suatu citra

berdasarkan warna spesifik. Cara kerjanya adalah dengan membandingkan komponen warna setiap piksel citra dengan warna spesifik. Pada dasarnya pencarian ini menggunakan kombinasi dari komponen *Red*, *Green*, dan *Blue* yang terdapat pada gambar. Nilai dari masing-masing komponen ini didapat dari hasil beberapa kali eksperimen. Hasil dari eksperimen ini adalah sebuah kombinasi dari komponen warna *Red*, *Green*, dan *Blue*. Kombinasi warna ini yang kemudian dijadikan filter yang merupakan penentu sebuah warna diloloskan atau tidak. Keluaran dari metode ini langsung akan menghasilkan sebuah gambar biner. Gambar biner sendiri adalah sebuah gambar yang hanya memiliki2 derajat keabuan yaitu hitam dan putih.

E. Thresholding

Thresholding merupakan proses konversi citra *grayscale* ke dalam bentuk citra biner yaitu citra dalam warna hitam dan putih. *Tresholding* digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra. Tiap-tiap piksel dalam citra levelnya dirubah melalui suatu thresholding tertentu apabila piksel tersebut nilainya diatas nilai thresholding maka piksel tersebut akan diubah ke warna putih, dan apabila nilai piksel tersebut berada pada level di bawah nilai *thresholding* maka piksel tersebut di ubah ke warna hitam**.**

Dengan menggunakan tresholding maka derajat keabuan bisa diubah sesuai keinginan, misalkan diinginkan menggunakan derajat keabuan 16, maka tinggal membagi nilai derajat keabuan dengan 16. Proses tresholding ini pada dasarnya adalah proses pengubahan kuantisasi pada citra, sehingga untuk melakukan tresholding dengan derajat keabuan dapat digunakan persamaan 2.1 :

$$x = w / b \quad \text{.....} \quad (2.1)$$

dimana :

x = nilai derajat keabuan setelah tresholding
w = nilai derajat keabuan sebelum tresholding
b = jumlah derajat keabuan yang diinginkan

F. Masking

Jika pada *point processing* kita hanya melakukan operasi terhadap masing-masing piksel, maka pada *mask processing* kita melakukan operas ierhadap suatu jendela ketetanggaan pada citra. Kemudian kita menerapkan (mengkonvolusikan) suatu mask terhadap jendela tersebut. *Mask* sering juga disebut filter.

G. Contour

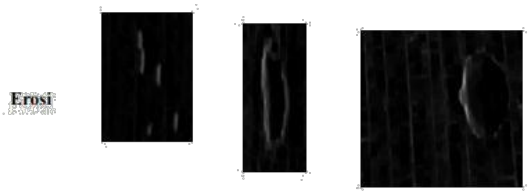
Contour dapat dijelaskan hanya sebagai kurva yang menggabungkan semua titik kontinu (sepanjang batas), memiliki warna atau intensitas yang sama. *Contour* adalah alat yang berguna untuk analisis bentuk dan deteksi dan pengenalan objek.Atau akurasi yang lebih baik, gunakan gambar biner. Jadi sebelum menemukan kontur, gunakan *threshold* atau *canny edge detection*. Fungsi *findContours* memodifikasi citra sumber. Jadi jika Anda menginginkan gambar sumber bahkan setelah menemukan kontur, sudah menyimpannya ke beberapa variabel lainnya. Dalam *OpenCV*, menemukan kontur seperti menemukan benda putih dari latar belakang hitam. Jadi objek yang harus ditemukan harus berwarna putih dan *background*nya harus hitam. Fungsi *drawContours* digunakan untuk menggambar kontur. Ini juga bisa digunakan untuk menggambar bentuk apapun jika Anda memiliki titik batasnya.

H. Morfologi Filterisasi

Operasi morfologi merupakan operasi yang umum dikenakan pada citra biner (hitam-putih) untuk mengubah struktur bentuk objek yang terkandung dalam citra. Pendekatan morfologi mengubah pandangan suatu citra sebagai himpunan. Morfologi merupakan suatu teknik dari pengolahan citra yang didasarkan pada pengolahan bentuk. Nilai dari tiap piksel citra yang diolah merupakan bentuk perbandingan antara piksel yang bersesuaian dari citra masukkan dengan nilai piksel tetangganya. Dengan memilih ukuran dan bentuk dari lingkungan, kita dapat membangun operasi analisis untuk mengolah citra yang dimasukkan agar lebih spesifik. Beberapa contoh operasi morfologi adalah erosi dan dilasi

1) Erosi

Erosi adalah dua operasi pokok yang ada pada Operasi Morfologi. Proses ini akan membuat ukuran sebuah citra menjadi lebih kecil. Berbeda dengan dilasi, apabila erosi dilakukan maka yang dikerjakan adalah memindahkan piksel pada batasan-batasan objek yang akan di erosi. Jumlah dari piksel yang ditambah atau dihilangkan bergantung pada ukuran dan bentuk dari structuring element yang digunakan untuk memproses image tersebut. Pada



operasi ini, ukuran objek diperkecil dengan mengikis sekeliling objek.

Gambar 2.1 Contoh Erosi

2) Dilasi

Dilasi adalah suatu proses menambahkan piksel pada batasan dari objek dalam suatu image sehingga nantinya apabila dilakukan operasi ini maka *image* hasilnya lebih besar ukurannya dibandingkan dengan *image* aslinya. Operasi dilasi dilakukan untuk memperbesar ukuran



segmen objek dengan menambah lapisan di sekeliling objek

Gambar 2.2 Contoh Dialasi

3) Operasi Opening

Operasi opening adalah operasi erosi yang diikuti dengan dilasi dengan menggunakan elemen penstruktur yang sama. Operasi ini berguna untuk menghaluskan kontur objek dan menghilangkan seluruh piksel di area yang terlalu kecil untuk ditempati oleh elemen penstruktur. Operasi opening sering dikatakan sebagai idempotent. Artinya, jika suatu citra telah dikenai operasi opening, pengenaan opening dengan elemen penstruktur yang sama tidak membawa efek apapun.

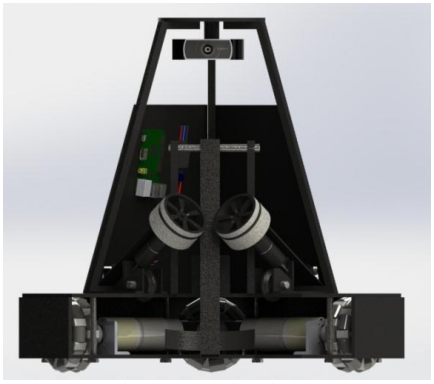
4) Operasi Closing

Operasi *closing* berguna untuk menghaluskan kontur dan menghilangkan lubang-lubang kecil.operasi closing dilaksanakan dengan melakukan operasi dilasi terlebih dahulu dan kemudian diikuti dengan operasi erosi.

III. METODE PENELITIAN

A. Sistem Mekanik

Desain robot menggunakan tigabuah roda sebagai penggerak, dengan kamera *Logitech C922 Pro Stream 1080P Webcam* sebagai sensor seperti pada gambar 3.1 dan gambar 3.2

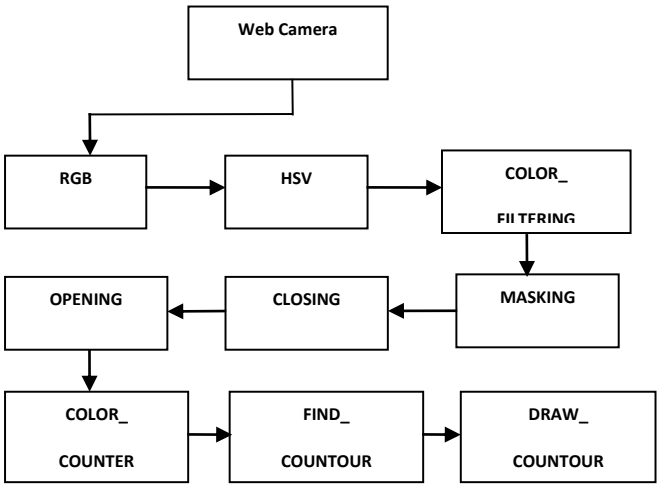


Gambar 3.1 Desain robot



Gambar 3.2 Kamera *Logitech C922 Pro Stream 1080P Webcam*

B. Proses Morfologi Filterisasi



Gambar 3.3 Diagram blok sistem deteksi bola *orange*

Penggambilan gambar dari kamera merupakan proses awal. Setelah itu sistem deteksi mendeteksi objek berdasarkan warna yaitu mengubah citra RGB menjadi HSV. Kemudian citra di filter terhadap warna bola dan warna lapangan yaitu merupakan proses dari binarisasi. Citra hasil dari ekstraksi binarisasi selanjutnya perlu dilakukan filterisasi morfologi. Dengan tujuan untuk menghilangkan noise dari gambar dilakukan erosi. Proses penekanan area objek yang ingin dideteksi merupakan proses filter dari dilasi.

IV. HASIL PENELITIAN

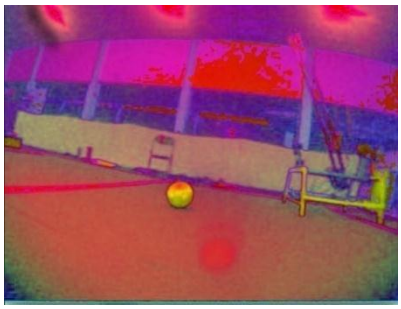


A. Pengujian menghilangkan noise

Untuk proses pengujian menghilangkan noise saat mendeteksi bola. Bola yang digunakan adalah Bola FutsalNike tipe 4 berwarna orange dengan ukuran diameter 68cm. Seperti pada gambar berikut:

Gambar 4.1 Bola futsal nike tipe 4

Untuk pengujian diletakkan benda berwarna orange didaerah bola. Gambar ini memiliki citra RGB untuk proses selanjutkan akan di konversi menjadi HSV seperti gambar berikut:



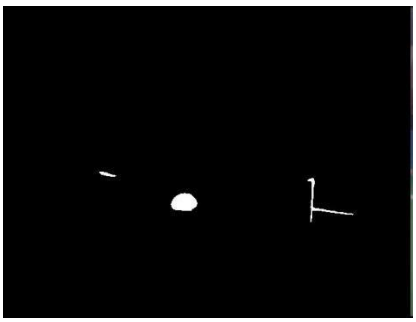
Gambar 4.2 Konversi bola futsal berwarna *orange* menjadi HSV

Proses selanjutnya adalah melakukan *filter* warna sesuai dengan warna bola yaitu *orange* dengan *range* data warna HSV bola. Setelah itu proses *masking* yaitu merubah menjadi *tresholding* dimana gambar yang berwarna orange akan dirubah menjadi putih dan selain itu akan berwarna hitam. Hasilnya seperti berikut:



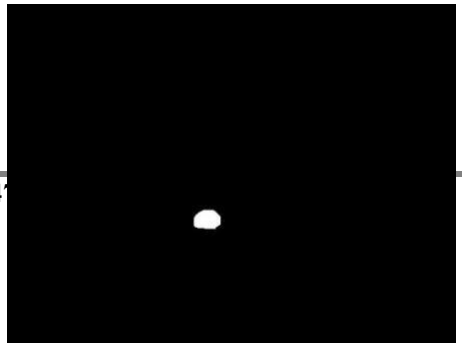
Gambar 4.3 Hasil *thresholding* bola *orange*

Pada gambar diatas terdapat noice. Noise tersebut akandihilangkan dengan metode filter morphology closing dan opening. Proses Closing adalah menjadikan daerah hitam yang berada dalam putih akan menjadi putih.



Gambar 4.4 Proses *closing* bola futsal *orange*

Setelah itu proses selanjutnya adalah opening untuk menghilangkan warna putih yang bukan bola dengan morphology opening. Untuk gambarnya seperti berikut:



Gambar 4.5 Proses *opening* bola futsal *orange*

Setelah semua *noise* hilang proses selanjutnya adalah mengambil koordinat daerah bola yang terdeteksi dengan *countour*. Untuk gambarnya seperti berikut:



Gambar 4.6 Hasil countour bola futsal orange

V. KESIMPULAN

Robot soccer mendeteksi sebuah bola berdasarkan warna bola. Sensor yang dibutuhkan adalah kameraLogitech C922 Pro Stream 1080P Webcam**,** sehingga dibutuhkan suatu sistem kendali secara visual, agar robot memiliki kecerdasan dalam mengenali objek berupa bola, pemrosesan gambar yang dilakukan yaitu mendeteksi objek menggunakan metodel *filtering* morfologi yang mana merupakan operasi yang umum dikenakan pada citra biner (hitam-putih) untuk mengubah struktur bentuk objek yang terkandung dalam citra. Nilai dari tiap piksel citra yang diolah merupakan bentuk perbandingan antara piksel yang bersesuaian dari citra masukkan dengan nilai piksel tetangganya. Dengan memilih ukuran dan bentuk dari lingkungan, kita dapat membangun operasi analisis untuk mengolah citra yang dimasukan agar lebih spesifik.

DAFTAR PUSTAKA

[1] Thome, Brian, “Introduction to Computer Vision in Phyton.”, University Of Canterbury, 2009.

[2] Aronof, Mark, “Morphology and Morphological Analysys.”, Blackwell, 2010.

[3] Nahla, G.S. 2010. Tracking Bola Menggunakan Robotino[pdf], (*http://digilib.its.ac.id/public/ITS-Undergraduate-14629-paperpdf.pdf,* diakses pada 31 Mei 2017).

[4] Kadir, Abdul, “*Teori dan Aplikasi Pengolahan Citra.”*

[5] Andi, Yogyakarta,2013. Universitas Telkom. Morfologi untuk Pengolahan Citra. Bandung, 2015.

[6] Morphological Transformations OpenCV Pyton Tutorial, (*https://pythonprogramming.net/morphological-transformation-python-opencv-tutorial/, diakses pada 1 Juni 2017)*

Sistem Persepsi Berbasis Transformasi Hough pada Robot Sepakbola Humanoid Dago Hoogeschool

Muhammad Isnain H.*, Andreas Ekadinata W., Hisham Lazuardi Y., Yoga Sanyoto, Ahmad Habbie T., Tjia, Johan Setiawan, Fildah Ananda A., Achmad Fahrurrozi M., Rio Dwi Putra P., Alif Mi'raj J., dan Fachriyan Husaini
Institut Teknologi Bandung
Bandung, Indonesia
mihisnain14@gmail.com

***Abstrak*— Robot sepak bola humanoid merupakan bidang riset yang populer. Robot harus mengenali dan membedakan objek yang terdapat di lapangan. Makalah ini membahas hasil rancang bangun sistem persepsi robot humanoid yang menggunakan kamera. Sistem persepsi terdiri dari dua, yaitu deteksi bola dan deteksi gawang. Metode yang dikembangkan memanfaatkan fitur bentuk dari bola dan gawang. Untuk dapat melakukan pengenalan fitur bentuk objek, digunakan metode transformasi Hough. Pada implementasinya, sistem persepsi ini menggunakan library OpenCV. Transformasi Hough merupakan teknik yang digunakan untuk melakukan pengenalan fitur objek berupa garis dan lengkungan. Sebelum dilakukan transformasi, citra juga mengalami proses enhancement citra untuk melakukan penajaman fitur garis pada citra. Sistem persepsi robot ini telah diimplementasikan pada robot humanoid tim Dago Hoogeschool. Hasil pengujian di lapangan, menunjukan robot mampu membedakan objek bola dan gawang. Kemampuan robot dalam mengenali bola berjarak maksimal 3 m dan dalam mengenali gawang maksimal 3 m dengan error perhitungan jarak gawang sebesar 8 cm.**

***Katakunci* – robot *humanoid*, sistem persepsi, transformasi *Hough*, *enhancement* citra.**

I. PENDAHULUAN

Robot *humanoid* adalah robot dengan penampakan dan lokomosinya berdasarkan tubuh manusia. Secara umum, robot humanoid memiliki aktuator pada kepala, dua tangan, dan dua kaki. Robot *humanoid* didesain untuk suatu fungsi tertentu, salah satunya pada cabang olah raga sepak bola. Robot *humanoid* yang digunakan pada permainan sepakbola harus memiliki kemampuan untuk menemukan letak bola dan memasukkannya ke gawang.

Robocup sebagai organisasi yang menaungi sepak bola robot dunia, mencanangkan bahwa pada tahun 2050 akan terdapat tim robot *humanoid* yang dapat bermain dan menang melawan tim robot manusia secara *autonomous*. Berdasarkan tujuan ini, peraturan yang digunakan pada permainan sepak bola robot humanoid terus diperbarui agar menyerupai permainan sepak bola manusia. Salah satu peraturan yang mengalami pembaruan adalah penggunaan lapangan rumput, gawang berwarna putih, dan bola FIFA *size* 1 berwarna 50% putih [1].



(a) (b)

Gambar 1 (a) Bola FIFA *size* 1 (b) Gawang Robocup

Untuk beradaptasi dengan peraturan baru, salah satu kemampuan dasar yang harus dimiliki robot adalah sistem persepsi robot dalam mengenali objek di lapangan. Sistem persepsi ini yang menjadikan robot dapat mencari bola dan mengarahkannya ke gawang.

Dalam riset deteksi bola dan gawang, terdapat dua metode yang umum digunakan, yaitu menggunakan metode transformasi *Hough* dan metode *machine learning*. Metode transformasi *Hough* melakukan proses deteksi dengan melakukan pemetaan gambar dari domain *image space* menjadi suatu domain *parameter space*. Sehingga hasil transformasi ini dapat dilakukan parameterisasi fitur objek berupa garis dan lengkungan (*curve*). Metode *machine learning* melakukan proses deteksi dengan melakukan *training* terhadap suatu objek untuk memperoleh *weight* dan *bias* dari objek tersebut. Hasil *training* ini dilakukan pencocokan dengan citra yang diterima untuk menentukan objek yang terdeteksi.

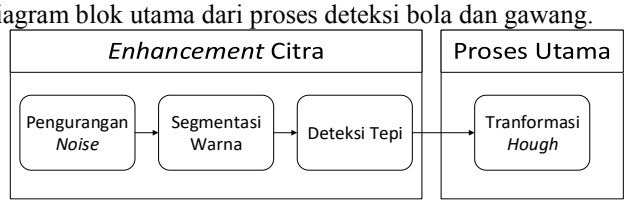
Pada riset ini, digunakan metode transformasi *Hough*. Hal ini disebabkan keterbatasan *hardware* pemroses yang digunakan oleh robot. Metode *machine learning* memiliki akurasi yang tinggi bergantung pada jumlah *training* yang dilakukan. Tetapi pemrosesan yang digunakan cukup berat. Sedangkan metode transformasi *Hough* memiliki akurasi tinggi apabila parameter-parameter yang digunakan sesuai [2]. Pemrosesan yang digunakan juga lebih ringan.

Sebelum dilakukan transformasi, citra yang diperoleh akan melalui proses *enhancement* citra. Proses *enhancement* bertujuan untuk melakukan aksentuasi dan penajaman fitur tertentu dari citra, agar citra dapat diproses secara lebih baik [3]. Adapun proses *enhancement* citra yang dilakukan terdiri dari penghilangan derau, segmentasi warna, dan deteksi tepi.

Sistem persepsi yang telah dirancang ini diimplementasikan pada robot sepak bola *humanoid* Dago Hoogeschool. Implementasi dilakukan menggunakan *library OpenCV* dengan basis *platform* program DARwIn-OP. Makalah ini memiliki garis besar penulisan sebagai berikut. Bab II mendeskripsikan desain sistem persepsi yang dirancang. Bab III membahas implementasi sistem persepsi untuk bola dan gawang pada robot. Bab IV membahas hasil pengujian sistem persepsi pada robot sepakbola humanoid Dago Hoogeschool. Terakhir bab V merupakan kesimpulan yang diperoleh.

II. Desain Sistem Persepsi Robot

Sistem persepsi robot dibagi menjadi dua bagian, yaitu deteksi bola dan deteksi gawang. Masing-masing deteksi ini dilakukan secara terpisah. Namun masing-masing deteksi memiliki proses *Enhancement* citra yang sama. Berikut diagram blok utama dari proses deteksi bola dan gawang.

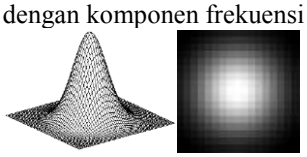


Proses Enhancement citra yang dilakukan berupa pengurangan *noise*, segmentasi warna, dan deteksi tepi. Masing-masing proses memiliki fungsi masing-masing sebagai berikut.

1. Pengurangan *Noise*

Bola yang berwarna 50% putih, serta gawang yang juga berwarna putih, menjadi masalah dalam deteksi objek. Karena beberapa objek yang ada di sekitar lapangan juga berwarna putih, maka terdapat kemungkinan adanya *noise* yang dapat mengganggu proses deteksi. Oleh karena itu, proses pertama yang dilakukan adalah penghilangan *noise* menggunakan *Gaussian blur*.

Gaussian blur merupakan teknik untuk melakukan pengaburan citra. Proses dilakukan dengan melakukan operasi matriks *Gaussian* terhadap setiap piksel citra. Karena menggunakan operator matriks *Gaussian*, maka hasil citra menjadi blur dengan distribusi yang merata. Fungsi *Gaussian* merupakan jenis filter *lowpass*, sehingga *noise* yang hilang merupakan *noise* dengan komponen frekuensi tinggi.



Gambar 2 Fungsi Operator *Gaussian*

2. Segmentasi Warna

Segmentasi dilakukan menggunakan model warna HSV (*Hue Saturation Value*) karena model warna ini memisahkan warna menggunakan informasi intensitas citra. Sehingga segmentasi dapat menyesuaikan kondisi perubahan cahaya dan kemampuan menghilangkan bayangan. Model warna HSV sudah diimplementasikan pada beberapa riset ([4][5]) serta sudah dievaluasi pada ([6][7]).

$$g(x,y) = \begin{cases} 1 \; if \; Hmin < H(x,y) < Hmax \\ \quad and \; Smin < S(x,y) < Smax \\ \quad and \; Vmin < V(x,y) < Vmax \\ 0 \; otherwise \end{cases} \quad \text{Eq 1}$$

Segmentasi dilakukan dengan melakukan pengecekan nilai *hue*, *saturation*, dan *value* dari citra. Setiap piksel yang memiliki nilai HSV pada rentang nilai yang telah ditentukan, maka akan bernilai 1. Sebaliknya, piksel yang nilai HSV di luar rentang tersebut akan bernilai 0.

3. Deteksi Tepi

Setelah citra dipisahkan antara objek yang akan dideteksi dengan *background*, dilakukan ekstraksi tepi dari citra *binary* hasil segmentasi. Proses ini dilakukan untuk mengurangi jumlah garis yang akan dimasukkan ke proses transformasi *Hough*. Keluaran dari proses ini adalah piksel tepi yang membentuk garis tepian objek.

Piksel tepi diperoleh dengan metode operator *Canny*. Penggunaan operator *Canny* merupakan metode deteksi tepi yang paling banyak digunakan karena hasil deteksi yang cukup *robust* [8]. Adapun proses deteksi tepi *Canny* membutuhkan beberapa tahap pemrosesan citra [3].

Langkah pertama dilakukan pemfilteran citra dengan *derivative Gaussian*. Dari citra yang telah difilter, dilakukan penghitungan magnitude dan sudut dari gradien tiap piksel. Dari hasil perhitungan ini, dicari *non maxima suppression* untuk memperoleh lebar piksel tunggal (tidak saling bertumpuk). Setelah diperoleh piksel tunggal, dilakukan *linking* (penghubungan) dan *thresholding* (pembatasan) untuk menghubungkan antar piksel tunggal.

4. Transformasi *Hough*

Setelah tepi dapat diekstrak dari citra yang telah tersegmentasi, proses selanjutnya adalah transformasi *Hough*. Transformasi *Hough* adalah metode yang popular digunakan untuk melakukan pengenalan pola kompleks pada citra. Transformasi *Hough* menggunakan transformasi dari domain *image space* menjadi domain *parameter space*. Transformasi ini dapat mendeteksi lengkungan dengan parameter tertentu, seperti garis lurus, polynomial, dan lingkaran.

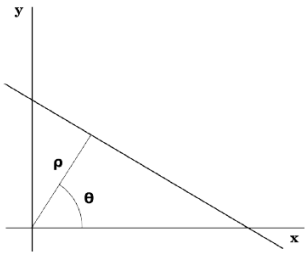
Transformasi *Hough* pertama kali digunakan oleh Paul Hough pada 1962 [2]. Ketika pertama kali digunakan, transformasi yang digunakan untuk parameterisasi garis dalam bentuk persamaan berikut.

$$y = mx + b \quad \text{Eq 2}$$

Parameterisasi ini mempunyai masalah dalam mendeteksi garis vetikal dimana nilai gradien bernilai tak terhingga. Sehingga parameterisasi didefinisikan dalam bentuk persamaan berikut.

$$\rho = x\cos(\theta) + y\sin(\theta) \quad \text{Eq 3}$$

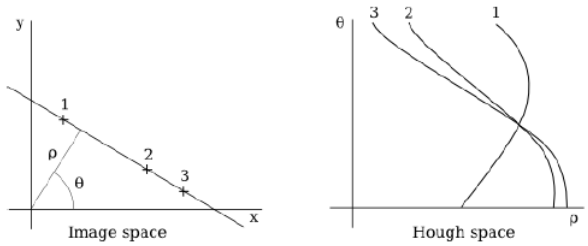
Dalam bentuk ini, θ merupakan sudut kemiringan dari garis normal dari garis dan ρ adalah panjang dari garis normal tersebut. Adapun garis normal sendiri didefinisikan sebagai garis terpendek antara garis dan titik origin.



Gambar 3 Parameterisasi garis dalam bentuk garis normal

Transformasi *Hough* bekerja dengan melakukan perhitungan nilai (ρ, θ) dari semua titik yang ada di garis. Kemudian dilakukan pemetaan titik-titik tersebut dalam domain *bidimensional* yang disebut domain *Hough*.

Prosedur ini menghasilkan akumulator terisi oleh *path sinusoidal* dari setiap piksel hasil deteksi tepi. Setiap *path* terakumulasi dan titik perpotongan antar *path* mempunyai *voting* yang lebih tinggi. Koordinat dari titik-titik perpotongan dengan *voting* terbesar korespon terhadap parameter suatu garis.



Gambar 4 Pemetaan dari tiga titik pada garis menjadi domain *Hough*

Pada implementasinya, parametisasi garis dilakukan dalam domain diskrit. Sehingga nilai parameter θ dibatasi hanya sampai 360 dengan resolusi satu derajat. Nilai parameter ρ juga dibulatkan menjadi nilai bilangan bulat terdekat untuk memudahkan perhitungan. Adapun maksimal nilai ρ adalah jarak maksimum dari dua titik yang dapat terjadi di citra. Sehingga nilai jarak maksimum adalah $\sqrt{Width^2 + Height^2}$. Nilai *Width* dan *Height* di atas merupakan nilai lebar dan tinggi dari dimensi citra. Dari perhitungan ini, diperoleh bahwa nilai sebuah garis dengan parameter (-ρ, θ) sama dengan parameter (ρ, θ+180).

III. Ekstraksi Informasi Bola dan Gawang

1. Ekstraksi Bola

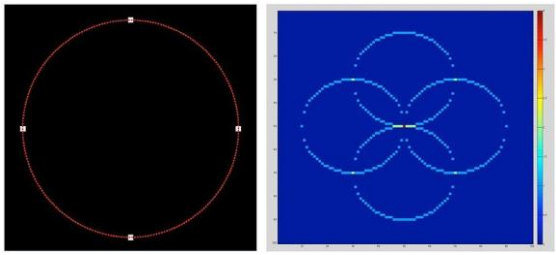
Untuk melakukan deteksi lingkaran pada bola, dilakukan modifikasi Transformasi *Hough*. Transformasi *Hough* yang digunakan berupa teknik ekstraksi fitur untuk deteksi lingkaran. Adapun metode yang digunakan untuk memilih kandidat lingkaran sama, yaitu dengna melakukan *voting* terhadap kandidat pada domain *parameter space* dan memilih *local maxima*.

Adapun bentuk persamaan yang digunakan untuk melakukan transformasi sebagai berikut.

$$(x-a)^2 + (y-b)^2 = r^2 \qquad \text{Eq 4}$$

Dimana nilai (a, b) adalah titik tengah lingkaran dan r adalah radius. Sehingga parameter space yang dihasilkan merupakan tiga dimensi (a, b, r)

Pada deteksi lingkaran yang digunakan pada riset ini, nilai radius dari lingkaran telah ditentukan di awal sebagai nilai radius dari bola. Sehingga *parameter space* dapat dikurangi menjadi dua dimensi (a, b). Untuk setiap titik (x, y) pada suatu lingkaran berjari-jari r, dapat didefinisikan menggunakan persamaan Eq 4. Titik perpotongan antar lingkaran pada parameter space yang dihasilkan merupakan titik tengah dari lingkaran original.



Gambar 5 Transformasi *Hough* untuk 4 titik di lingkaran

Dengan menggunakan persamaan Eq 4, untuk 4 titik di lingkaran pada citra original, dapat didefinisikan lingkaran-lingkaran pada domain *Hough* dengan pusat (x, y) dan radius r. Sebuah matriks akumulator diunakan untuk melakukan *tracking* titik perpotongan antar lingkaran. Pada matriks ini, setiap lingkaran yang saling berpotongan menambahkan nilai *voting* pada titik potong tersebut. Kemudian titik *local maxima* dapat diperoleh untuk nilai *voting* paling besar. Posisi (a, b) dari *local maxima* ini diambil sebagai pusat lingkaran original.

2. Ekstraksi Gawang

a. Pemrosesan garis

Pada deteksi gawang, setelah diperoleh garis-garis hasil transformasi *Hough*, terdapat proses tambahan berupa pemrosesan garis. Teknik ini digunakan untuk menentukan struktur dari gawang menggunakan informasi garis yang telah diperoleh. Kunci utama dari teknik pemrosesan garis adalah klasifikasi garis menjadi garis vertikal dan horizontal.

Pada riset ini, diasumsikan bahwa kamera robot tidak akan mengalami kemiringan lebih dari 45^0. Klasifikasi dilakukan berdasarkan parameter θ dari garis. Adapun klasifikasi dilakukan sebagai berikut.

$$Line = \begin{cases} horizontal, & 45^0 \le \theta_L \le 135^0 \\ & or\ 225^0 \le \theta_L \le 315^0 \\ vertical, & otherwise \end{cases}$$

Garis horizontal didefinisikan sebagai mistar dari gawang, sedangkan garis vertikal didefinisikan sebagai tiang dari gawang. Untuk garis vertikal, dilakukan pemrosesan nilai parameter ρ untuk menentukan apakah terdapat satu atau dua tiang dengan melihat nilai variansinya. Apabila variansi nilai ρ tinggi, maka diputuskan terdapat dua tiang pada citra untuk kemudian diklasifikasikan menjadi tiang kanan dan kiri. Jika nilai ρ bernilai kurang dari *mean* ρ, maka tiang tersebut merupakan tiang kiri. Sebaliknya, jika nilai ρ lebih dari nilai *mean* ρ, maka tiang tersebut merupakan tiang kanan.

b. Ekstraksi Struktur Gawang

Jenis struktur gawang yang sedang terlihat oleh robot dapat diidentifikasi dengan menggunakan nilai variansi garis. Terdapat enam struktur gawang, struktur pertama C2P

(*Crossbar with Two Posts*) yaitu kedua tiang (*post*) dan mistar gawang terdeteksi seluruhnya atau nilai variansi vertikal lebih besar dari nilai batas (*treshold*) yaitu 5000. Kedua, C1P (*Crossbar with One Post*) yaitu salah satu tiang baik itu tiang kanan atau kiri dan mistar gawang terdeteksi atau nilai variansi vertikal kurang dari nilai batas. Ketiga, NC2P (*No Crossbar with Two Posts*) yaitu kedua tiang gawang terdeteksi, namun tidak ada garis horizontal berupa mistar gawang yang terdeteksi. Keempat, NC1P (*No Crossbar with One Post*) yaitu ketika salah satu tiang gawang terdeteksi dan tidak ada garis horizontal yang terdeteksi atau mistar gawang yang terlihat. Kelima, CO (*Crossbar Only*) yaitu hanya mistar gawang saja yang terdeteksi. Struktur keenam adalah NL (*No Line*), yaitu ketika tidak ada tiang dan mistar gawang yang terdeteksi oleh robot atau tidak ada garis dominan yang terdeteksi oleh robot.

c. Perhitungan Jarak Gawang

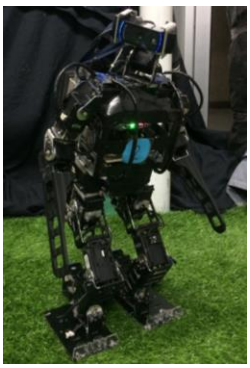
Perhitungan terhadap jarak gawang dari posisi robot dapat dilakukan dengan pencarian nilai *focal length* kamera pada robot. Data *focal length* tersebut kemudian digunakan untuk menghitung jarak antara robot dan gawang, dengan menggunakan konsep *triangle similarity*. Konsep ini menggunakan lebar gawang sesungguhnya serta lebar gawang yang dideteksi oleh kamera robot dengan memanfaatkan data resolusi kamera. Adapun persamaan yang digunakan dalam perhitungan jarak gawang adalah sebagai berikut.

$$goaldistance = \frac{focallength \times goalwidth}{abs(GoalRightFoot.x - GoalLeftFoot.x)}$$

IV. Konfigurasi dan Hasil Pengujian

A. Kondisi Pengujian

Pengujian dilakukan pada robot *Humanoid Kid Size* Dago Hoogeschool yang sesuai strandar yang ditentukan oleh peraturan *Robocup*. Robot yang digunakan merupakan pengembangan dari *platform* robot DARwIn-OP yang dinamakan platform Alfassary. Robot mempunyai 20 DOF servo degan tinggi 58 cm. Aktuator robot terdiri dari dua jenis motor servo, yaitu 8 servo Dynamixel MX-28 untuk tubuh bagian atas dan 12 servo Dynamixel MX-64 untuk tubuh bagian bawah. Semua modul seperti *compass, gyroscope, accelerometer,* dan *servo controller* terpasang pada *Single Board Computer* robot berupa Odroid-XU4 dengan *processor* Samsung Exynos 2 GHz dan RAM 2 GHz.



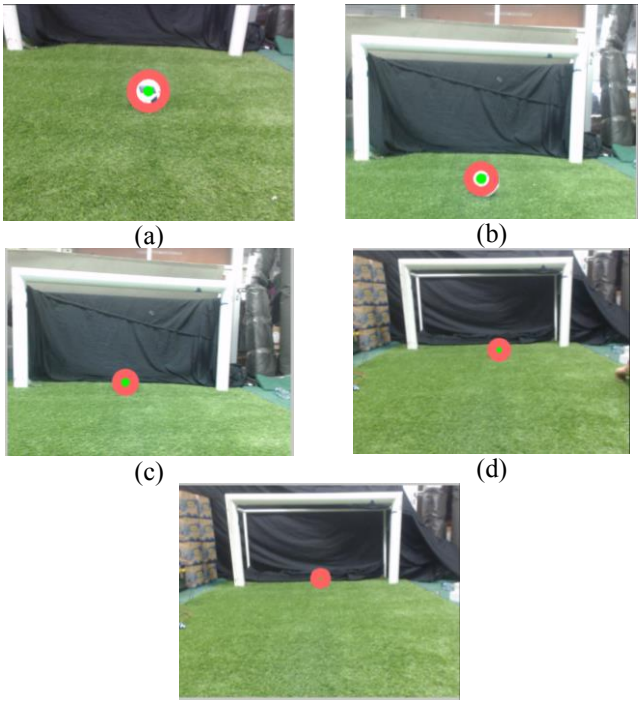
Gambar 6 *Platform* Robot Alfassary

Untuk melakukan pengujian sistem persepsi robot, digunakan kamera Logitech C920. Kamera ini memiliki lensa Carl Zeiss dengan sensor CMOS. Kamera ini juga memiliki 20-*step autofocus* dan koreksi rendah cahaya otomatis. Adapun resolusi citra yang dipakai adalah 320 x 240 piksel. Pengujian juga dilakukan di rumput sintetis standar Robocup.

B. Hasil Pengujian

Pengujian sistem persepsi dibagi menjadi dua, yaitu deteksi bola dan deteksi gawang. Setiap deteksi dilakukan untuk kondisi beberapa jarak bola dan gawang terhadap robot. *Sampling* jarak dilakukan setiap setengah meter. Berikut hasil pengujian yang diperoleh.

1. Pengujian deteksi bola



(a) (b) (c) (d) (e)

Gambar 7 Pengujian deteksi bola pada jarak (a) 1 meter (b) 1.5 meter (c) 2 meter (d) 2.5 meter (e) 3 meter

Tabel I. Hasil pengujian deteksi bola

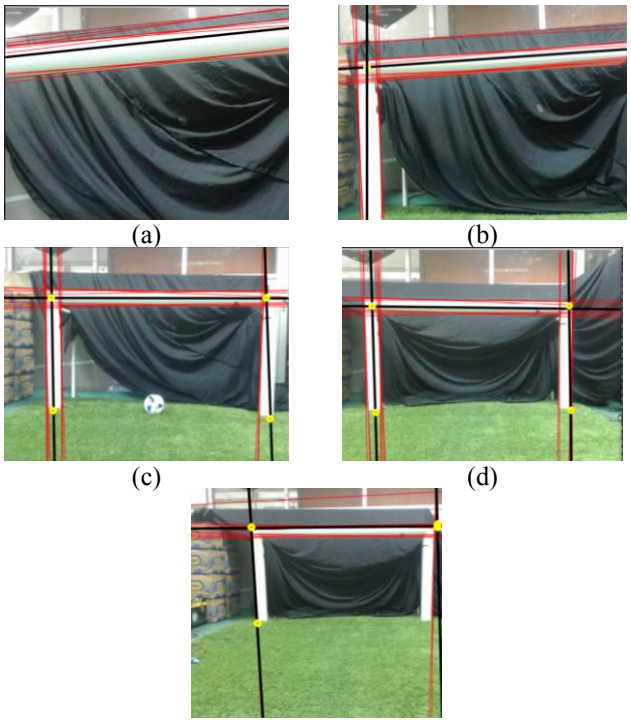
Jarak Bola	Terdeteksi
1 meter	Ya
1.5 meter	Ya
2 meter	Ya
2.5 meter	Ya
3 meter	Ya

Pengujian deteksi bola dilakukan untuk jarak dari 1 meter sampai batas maksimum bola terdeteksi. Dari gambar hasil citra yang ditangkap kamera di atas, terlihat bahwa bola selalu dapat terdeteksi sampai jarak maksimal 3 meter. Setelah jarak tersebut, deteksi bola cenderung tidak stabil. Pada implementasinya, ketika bola terdeteksi maka dilakukan penandaan bola dengan ditandai pada titik hijau dan lingkaran merah.

Dari hasil pengujian, juga diperoleh bahwa robot dapat membedakan objek bola dengan objek lain berwarna identik.

Hal ini terbukti bahwa meskipun terdapat objek gawang yang juga berwarna putih, robot dapat membedakan objek bola dari fitur lingkaran bola.

2. Pengujian deteksi gawang



(a) (b) (c) (d) (e)

Gambar 8 Pengujian deteksi gawang pada jarak (a) 1 meter (b) 1.5 meter (c) 2 meter (d) 2.5 meter (e) 3 meter

Tabel II. Hasil pengujian deteksi gawang

Jarak Gawang	Terdeteksi	Status	Jarak Perhitungan
1 meter	Ya	*Crossbar Only* (CO)	Tidak terdeteksi
1.5 meter	Ya	*Crossbar* 1 Tiang Kiri (C1P)	Tidak terdeteksi
2 meter	Ya	*Crossbar* 2 Tiang (C2P)	197 cm
2.5 meter	Ya	*Crossbar* 2 Tiang (C2P)	245 cm
3 meter	Ya	*Crossbar* 2 Tiang (C2P)	308 cm

Pengujian deteksi gawang dilakukan untuk jarak dari 1 meter sampai 3 meter. Jarak 3 meter merupakan jarak daerah serang, yaitu setengah dari panjang lapangan. Dari gambar hasil citra yang ditangkap kamera di atas, terlihat bahwa gawang selalu dapat terdeteksi sampai jarak 3 meter. Pada implementasinya, ketika gawang terdeteksi maka dilakukan penandaan gawang. Garis merah menandakan garis hasil deteksi garis dari proses *transformasi Hough*. Garis hitam menandakan nilai mean dari tiang dan mistar yang terdeteksi. Adapun titik kuning merupakan titik hasil perpotongan mistar dengan tiang dan titik piksel putih paling bawah.

Dari hasil pengujian, juga diperoleh bahwa robot dapat membedakan objek gawang dengan objek lain berwarna identik. Hal ini terbukti bahwa meskipun terdapat objek bola dan beberapa garis yang juga berwarna putih, robot dapat membedakan objek gawang dari fitur struktur gawang yang telah didefinisikan sebelumnya.

Selain fungsionalitas deteksi gawang, juga dilakukan pengujian perhitunga jarak gawang. Dari hasil jarak yang diperoleh, nilai error maksimum perhitungan jarak sebesar 8 cm untuk jarak gawang 3 meter. Nilai error ini semakin besar untuk jarak gawang yang semakin jauh.

V. KESIMPULAN

PADA RISET INI, SISTEM PERSEPSI BERBASIS TRANSFORMASI HOUGH TELAH BERHASIL DIIMPLEMENTASIKAN PADA ROBOT SEPAK BOLA HUMANOID DAGO HOOGESCHOOL. SISTEM PERSEPSI BERUPA DETEKSI BOLA DAN DETEKSI GAWANG. METODE YANG DIRANCANG MULAI DARI ENHANCEMENT CITRA YANG TERDIRI DARI GAUSSIAN BLUR, SEGMENTASI WARNA, DAN DETEKSI TEPI HINGGA PROSES UTAMA BERUPA TRANSFORMASI HOUGH.

UNTUK DETEKSI BOLA, ROBOT DAPAT MENDETEKSI BOLA PADA JARAK MAKSIMUM 3 METER. SEDANGKAN UNTUK DETEKSI GAWANG, ROBOT DAPAT MENDETEKSI BOLA SAMPAI JARAK 3 METER DENGAN ERROR MAKSIMUM PERHITUNGAN JARAK GAWANG SEBESAR 8 CM. PADA RISET YANG DILAKUKAN JUGA MEMILIKI KONSTRAIN YANG MEMBATASI YAITU KONDISI PENCAHAYAAN YANG STABIL. UNTUK MEMPEROLEH HASIL DETEKSI YANG OPTIMAL, DIPERLUKAN KALIBRASI ULANG UNTUK KONDISI PENCAHAYAAN YANG BERBEDA.

Ucapan Terima Kasih

Penulis menyampaikan terima kasih kepada tim Dago Hoogeschool ITB yang telah bekerja sama dalam pengembangan algoritma sistem persepsi robot.

REFERENSI

[1] Reinhard Gerndt, Sean Luke, Jacky Baltes, and Soroush Sadeghnejad, "RoboCup Soccer Humanoid League Laws of the Game 2016/2017", 2017.

[2] J. Illingworth, J. Kittler, “A Survey of the Hough Transform,” Computer Vision, Graphics, and Image Processing 44, Academic Press, 1988.

[3] Andriyan Suksmono, Astri Handayani, Tati Mengko, "Peningkatan Kualitas Citra (*Image Enhancement*)", Program Studi Teknik Elektro Institut Teknologi Bandung, Bandung, 2016.

[4] J.M. Canas, D. Puig, E. Perdices, T. Gonzalez, “Visual goal detection for the robocup standard platform league,” X Workshop De Agentes Fisicos, September 2009.

[5] P. Shengze, W. Yongge, “Research based on the HSV humanoid robot soccer image processing,” IEEE International Conference on Communication Systems, Networks and Applications, 2010.

[6] D. Budden, S. Fenn, A. Mendes, S. Chalup, “Evaluation of colour models for computer vision using cluster validation techniques,” unpublished.

[7] H. K. Kelda, P. Kaur, “A review : color models in image processing,” International Journal Computer Technology and Applications, vol.5, pp. 319-322, April 2014.

[8] N. Abdullah Al, K. Yinan, H. Md Nazmul, "Performance Analysis of Canny’s Edge Detection Method for Modified Threshold Algorithms", in Proc. ICEEE, 2015, India.

Pengembangan Platform Robot Sepak Bola Beroda

Evan Febrianto*, Abdullah Faqih A. M., Andi Gumarilang C. A., Ahmad Wahrudin, Azka Zakiyyatuddin, Bangsa Edwiranda, Beny Maulana A., Genesia Wahyu S., Ghovindo Surya T. S., M. Hilmi Asyrofi, M. Ihsan Hadi, M. Kamal Nadjieb, Reghina Arindadewi, Yudhistira Yoga S.
Tim Dagozilla, Unit Robotika
ITB (Institut Teknologi Bandung)
Bandung, Indonesia
van.evanfebrianto@gmail.com

***Abstrak*-- Robot sepak bola beroda adalah topik yang banyak diminati di kalangan peneliti dan akademisi. Hingga saat ini, platform yang dikembangkan baik di Indonesia dan di dunia sangat beragam. Pada tulisan ini akan dibahas tentang pengembangan platform robot sepak bola beroda. Platform dikembangkan dengan lokomosi menggunakan 3 roda dan 4 roda, dua buah kamera, dan kontroler. Dari segi lokomosi, robot dapat bergerak dengan maksimal error sebesar 5%. Pengembangan platform menggunakan dua kamera yang digunakan meliputi kamera atas dan kamera depan. Kamera atas digunakan untuk mendeteksi bola saat bola berada di dekat robot, sedangkan kamera depan digunakan saat bola berada jauh dari robot.**

Katakunci—kamera; lokomosi; platform; robot

I. PENDAHULUAN

ROBOT sepak bola beroda banyak diminati oleh berbagai kalangan baik dari peneliti, akademisi, maupun penggiat robotika. Hal ini disebabkan karena pengembangan platform robot sepak bola beroda jauh lebih murah jika dibandingkan robot humanoid. Selain itu, tingkat kesulitan kontrolnya juga lebih mudah dibandingkan robot humanoid. Pada platform robot humanoid, pengembang harus memiliki kemampuan untuk melakukan kontrol keseimbangan robot yang cukup kompleks. Dari segi desain, robot humanoid juga memiliki keterbatasan pada robot yang menyerupai manusia, sedangkan pada robot sepakbola beroda memiliki fleksibilitas desain yang lebih tinggi jika dibandingkan dengan robot humanoid. Akibatnya, platform robot sepak bola beroda yang dikembangkan juga sangat beragam. Pengembangan robot sepak bola yang beragam ini menyebabkan robot memiliki kelebihan dan kekurangan masing-masing. Pada tulisan ini akan dipaparkan tentang platform robot sepak bola beroda dengan menggunakan lokomosi 3 dan 4 roda serta 2 kamera sebagai vision pada robot yang dikembangkan.

II. SPESIFIKASI PIRANTI KERAS

A. Dimensi robot

Robot ini dirancang dengan spesifikasi dimensi seperti Tabel 1 [1].

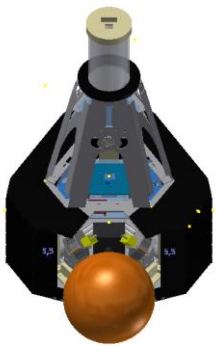
Tabel 1. Spesifikasi dimensi robot

Parameter	Ketentuan
Panjang x lebar	52 cm x 52 cm (maksimal) dengan diameter maksimal pada ketinggian 60 cm adalah 25 cm
Tinggi	80 cm (maksimal)
Berat	40 kg (maksimal)
Warna Robot	Hitam
Spesifikasi Cat Kerangka	Tidak memantulkan cahaya (*matte*)

Kerangka robot didominasi dari bahan alumunium yang kuat dan ringan. Hal ini bertujuan karena pada permainan sepakbola kontak fisik antarrobot tidak dapat dihindari. Sehingga kerangka robot yang digunakan harus memiliki reliabilitas dan stabilitas yang tinggi. Adapun proses perancangan kerangka robot dilakukan dengan menggunakan software Autodesk Inventor sehingga diperoleh gambar teknik dari desain robot yang akan diproduksi.

B. Desain Robot

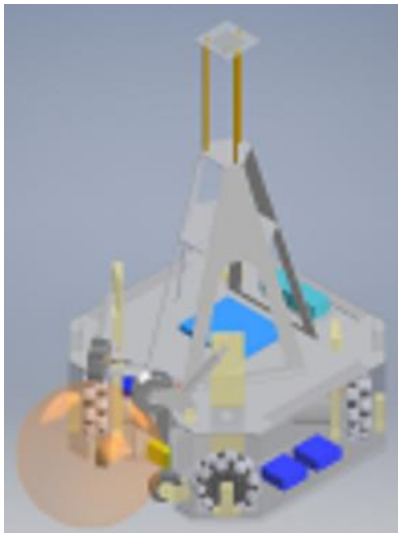
Berikut desain robot dengan menggunakan 3 roda yang dikembangkan.



Gambar 1. Desain Robot Tampak Depan Atas

Robot ini juga memiliki 3 motor lokomosi. Sehingga bentuk *base* yang digunakan menyerupai bentuk segitiga sama sisi. Motor lokomosi berjarak sebesar 120^0 antara satu motor dengan motor lain. Hal ini bertujuan untuk memudahkan perhitungan kinematika gerakan robot.

Sedangkan berikut ini robot dengan desain 4 roda yang dikembangkan.



Gambar 2. Desain Robot Tampak Samping Atas

Robot ini juga memiliki 4 motor lokomosi. Sehingga bentuk *base* yang digunakan menyerupai bentuk persegi. Motor lokomosi berjarak sebesar 90^0 antara satu motor dengan motor lain. Hal ini bertujuan untuk memudahkan perhitungan kinematika gerakan robot [2].

C. Main Controller



Gambar 3. Blok Diagram Subsistem *Single Board Computer*

Main *controller* yang digunakan berupa SBC. SBC ini berfungsi sebagai perangkat utama yang mengatur kerja sistem dari menerima *output* sensor, melakukan pengolahan informasi, membuat keputusan, dan memberi perintah pada modul kendali. *Single board computer* yang digunakan adalah ODROID XU4 dengan spesifikasi pada Tabel 2.

Tabel 2. Spesifikasi Odroid XU-4

Prosesor	Samsung Exynos5422 Cortex™ – A15 2GHz Cortex™ – A7 Octa Core CPUs
Memori	2GB LPDDR3 RAM

D. Sub Controller

Modul *sub controller* merupakan modul yang menghubungkan modul SBC ke modul lokomosi, modul penggiring, dan modul penendang. Modul *sub controller* akan menerima perintah-perintah dari modul *single board computer,* menerjemahkan perintah-perintah yang diterima, kemudian mengirim sinyal kontrol hasil komputasi ke motor lokomosi, sistem penggiring, dan sistem penendang. Selain itu modul *main board* juga mengirim sinyal pembacaan *encoder* motor lokomosi ke *single-board-computer*.

Modul *sub controller* menggunakan *microcontroller* Nucleo-64 STM32F446RE berbasis ARM® mbed 32 bit. Nucleo-64 menggunakan mikroprosesor ARM® Cortex®-M4 CPU dengan frekuensi mencapai 180 MHz. Nucleo-64 dapat menampung memori sebesar 512 KB untuk Flash dan 128 KB untuk SRAM. Mikrokontroler ini memiliki *timer* berjumlah hingga 17 buah.

III. SPESIFIKASI PIRANTI LUNAK

A. Sistem Citra Robot

Modul pengolahan citra digunakan untuk mengambil data *visual* dari lingkungan dan memprosesnya menjadi masukan pada algoritma pemecahan masalah pada robot. Platform robot dikembangkan dengan menggunakan dua buah kamera, yaitu kamera depan dan kamera atas. Untuk kamera depan digunakan kamera Logitech C170 dengan spesifikasi pada Tabel 3.

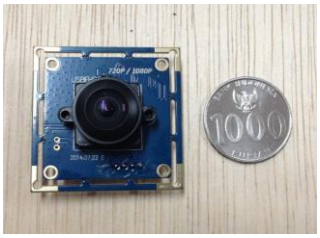
Tabel 3. Spesifikasi Kamera Logitech C170

Parameter	Spesifikasi
Optional models	Logitech C170
Max. Resolution	Supports VGA 640 x 480 Video Calling XVGA Video Capture Up to 1024 x 768
Lens size	Take 5MP Photos (Software Enhanced)
Sensor	Logitech Fluid Crystal Technology
USB Protocol	Hi-Speed USB 2.0 Certified
Additional features	Built-In Microphone with Noise Reduction Universal Clip for Laptops & Monitors



Gambar 4. Kamera Logitech C170

Sedangkan untuk kamera atas, digunakan kamera dengan lensa *fish eye*, yaitu lensa dengan dengan sudut pandang 180°. Sensor ini dipilih karena dapat menangkap citra lingkungan dengan cakupan yang lebih lebar, sehingga dapat meningkatkan efisiensi sistem [3]. Dilakukan proses *filtering* untuk membedakan objek pada lapangan, sehingga robot dapat menentukan lokasi bola dan gawang. Sensor kamera ini dihubungkan ke *main controller* dengan menggunakan USB. Perangkat yang digunakan memiliki resolusi maksimal sebesar 2 megapiksel. Namun, pada pemrosesan citra, hasil tangkapan dari sensor kamera akan dikompresi menjadi resolusi yang lebih kecil agar pemrosesan data menjadi lebih cepat. Jenis sensor kamera yang digunakan adalah Ominivision OV2710 dengan seri ELP-USBFHD01M-L180 seperti pada gambar 5.



Gambar 5. Sensor kamera ELP-USBFHD01M-L180

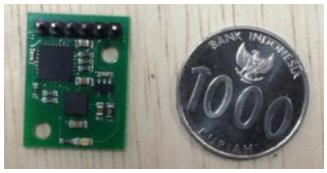
Pada robot sepak bola, pengolahan citra dibutuhkan untuk mendeteksi lingkungan yang berupa garis-garis lapangan, gawang, serta bola. Pertama, digunakan metode segmentasi warna pada citra objek yang dideteksi serta penentuan koordinat objek. Proses segmentasi warna sangat bergantung pada pencahayaan lingkungan, sehingga memiliki reliabilitas yang kurang baik. Untuk mengatasi hal tersebut, digunakan pula transformasi Hough untuk mengenali fitur atau bentuk dari objek yang dideteksi, sehingga hasil yang didapatkan menjadi lebih akurat. Diagram blok dari sistem pengolah citra dapat dilihat pada gambar 6.



Gambar 6. Diagram blok proses pengolahan citra

B. *Sistem Navigasi Robot*

Navigasi memberikan petunjuk arah yang berfungsi untuk pergerakan robot. Navigasi pada *mobile soccer robot* menggunakan sensor kompas yang dapat mengetahui posisi sudut. Sensor yang dipakai adalah sensor magnetik kompas CMPS11. Sensor magnetik kompas CMPS11 ini akan memberikan referensi bagi robot dimana posisi gawang tim dan lawan. Bentuk fisik dari sensor yang digunakan dapat dilihat pada gambar 7.



Gambar 7. Sensor Magnetik Kompas CMPS11

IV. HASIL PENGUJIAN

A. *Dimensi Robot*

Berdasarkan hasil pengukuran, robot yang dikembangkan memiliki dimensi dan berat pada tabel 4.

Tabel 4. Hasil Pengujian Kerangka Robot

Parameter	Robot 1	Robot 2
Panjang (cm)	50	40
Lebar (cm)	48	48
Tinggi (cm)	79	65
Massa (kg)	16.3	14.7

Robot 1 adalah robot dengan desain 3 roda, sedangkan robot 2 adalah robot dengan desain 4 roda. Dari hasil pengukuran, dapat disimpulkan bahwa dimensi dan berat dari kedua robot yang telah diimplementasikan sesuai dengan spesifikasi.

Kemudian dibuat algoritma kontrol yang diintegrasikan dengan sistem penggerak robot secara keseluruhan untuk menguji kemampuan robot melakukan gerakan dasar [4]. Hasilnya robot dapat bergerak sesuai dengan jalur yang sudah ditentukan sebelumnya dengan galat yang variatif dan masih dalam batas kewajaran. Galat tersebut terjadi dikarenakan adanya *slippage* yang terjadi antara roda dan karpet yang digunakan. Hasil dari pengujian implementasi gerakan dasar dapat dilihat pada tabel 5 dan tabel 6.

Tabel 5. Hasil Pengujian Gerakan Dasar untuk Robot dengan 3 Roda

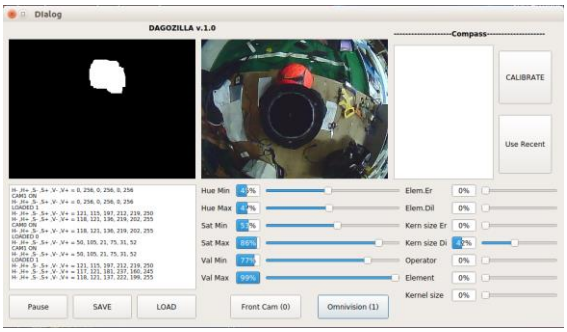
Gerakan	Durasi (detik)	Kecepatan (PWM)	Jarak Ditempuh (m)	Galat Miring (cm)
Maju	4	0.5	4.10	4
Mundur	4	0.5	4.10	6
Kanan	4	0.5	4.09	4
Kiri	4	0.5	4.13	3

Tabel 6. Hasil Pengujian Gerakan Dasar untuk Robot dengan 4 Roda

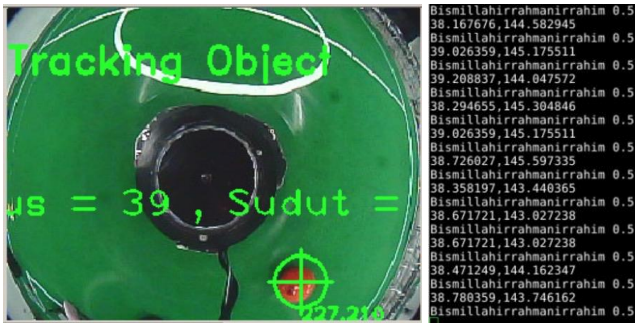
Gerakan	Durasi (detik)	Kecepatan (PWM)	Jarak Ditempuh (m)	Galat Miring (cm)
Maju	4	0.5	4.70	8
Mundur	4	0.5	4.72	5
Kanan	4	0.5	4.70	10
Kiri	4	0.5	4.69	4

B. *Sistem Citra Robot*

Untuk mengetahui informasi yang mampu diekstrak oleh kamera omnidireksional, dilakukan pengamatan pada hasil pengolahan citra digital yang pengamatannya dilakukan melalui *interface* yang sudah dibuat. Pengujian dilakukan di empat titik di sekitar robot dengan mencatat jarak dan sudut bola terhadap robot. Jarak diberikan dalam satuan centimeter, sedangkan sudut diberikan dalam satuan derajat dengan perjanjian 0 derajat berada tepat di depan robot, sudut dengan arah searah putaran jarum jam bernilai positif dan sudut dengan arah berlawanan arah jarum jam bernilai negatif.



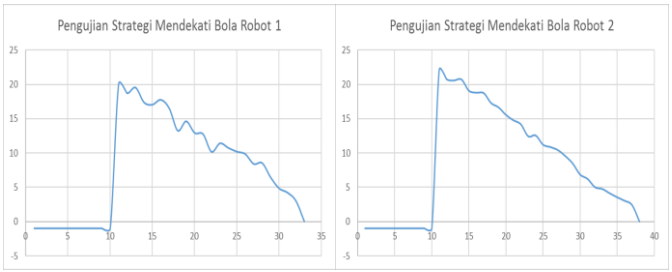
Gambar 8. Antarmuka aplikasi ketika program utama dijalankan



Gambar 9. Ekstraksi fitur jarak dan sudut bola pada kamera atas

Ekstraksi sudut dari citra kamera atas memberikan hasil yang tidak sepenuhnya akurat. *Distorsi* yang muncul pada citra akibat lensa cembung pada kamera berdampak pada pembacaan bola pada koordinat *cartesian* di citra. *Distorsi* pada lensa cembung menggunakan koordinat polar (arah radial), sedangkan penentuan sudut menggunakan koordinat *cartesian*. Akibatnya, semakin jauh jarak bola dari pusat robot (pusat kamera), semakin besar *error* yang dihasilkan.

Untuk melakukan verifikasi performa, dilakukan plot nilai bacaan sudut kamera atas. Nilai sudut ini dapat menunjukkan kestabilan algoritma dalam mengompensasi delay program. Gambar 10 menunjukkan hasil plot yang diperoleh.



Gambar 10. Grafik nilai *error* sudut dari kamera atas pada robot

Berbeda dari kamera atas yang dimaksudkan untuk mendeteksi bola di dekat robot, kamera depan dimaksudkan untuk mencari bola yang berjarak jauh dari robot. Fitur yang dihasilkan adalah koordinat bola terhadap kamera dalam satuan pixel, dengan sumbu original nya terletak di pojok kiri atas. Sumbu x positif mengarah ke kanan dan sumbu y positif mengarah ke bawah.

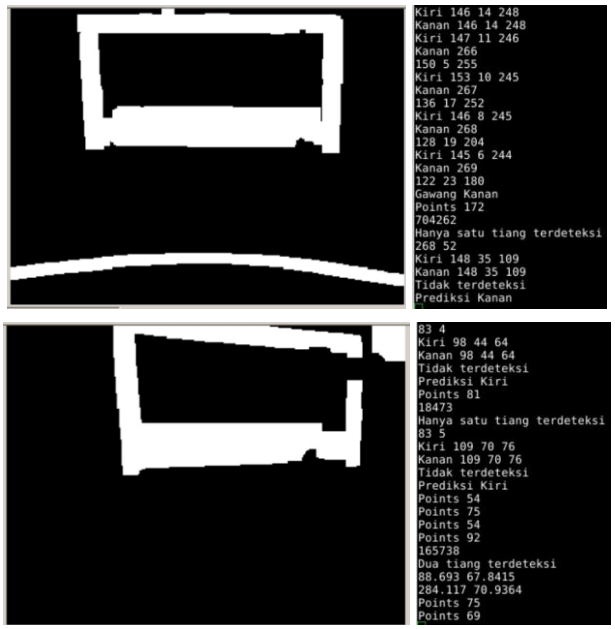


Gambar 11. Ekstraksi fitur posisi bola terhadap kamera depan

Citra yang mengalami blur memberikan efek yang baik terhadap algoritma *hough circle* untuk mendeteksi bentuk lingkaran bola. Akibat dari *blurring* yang dilakukan, noise yang ada di sekitar objek bola menjadi berkurang sehingga

proses deteksi bola menjadi lebih mudah. Bola yang berjarak 7,2 meter sesungguhnya terlihat sangat kecil pada citra, namun dengan algoritma *hough circle*, objek lingkaran kecil sekalipun dapat didteksi sebagai bola. Dengan begitu robot mampu mendeteksi bola meskipun bola berada di ujung lapangan yang berbeda dengan robot.

Untuk mendeteksi gawang, digunakan algoritma *hough line* yang bekerja dengan mencari garis untuk diklasifikasikan sebagai objek gawang. Hasil algoritma *hough line* sangat dipengaruhi oleh sudut pengambilan citra sebab perbedaan ukuran proyeksi tiang gawang memengaruhi proses seleksi citra [5].



Gambar 12. Ekstraksi fitur posisi gawang

Proses deteksi gawang untuk dua sudut pengambilan gambar yang berbeda memberikan hasil yang sama, yaitu mampu memberikan informasi posisi gawang terhadap kamera. Pada tahap *color thresholding*, semua objek dengan warna putih dideteksi. Namun pada tahap kedua, algoritma *hough line* menyeleksi semua objek putih dan menentukan objek yang diyakini sebagai gawang dengan melakukan deteksi tiang gawang. Pada prosesnya, tidak selamanya kedua tiang gawang langsung terbaca, namun adakalanya dilakukan prediksi bilamana hanya ada satu tiang yang terdeteksi. Selama ada tiang yang terdeteksi, baik hanya satu maupun dua, program selalu dapat memberikan informasi posisi tiang gawang.

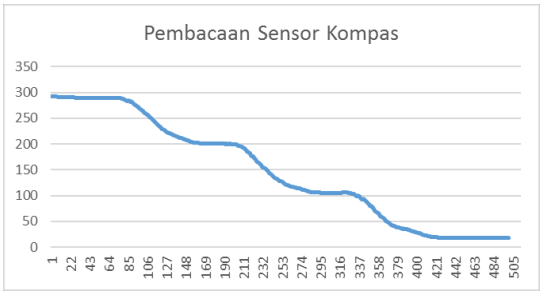
C. *Sistem Navigasi Robot*

Kalibrasi dilakukan dengan linierisasi pembacaan sensor kompas. Kalibrasi dilakukan dengan memutar robot sebesar satu putaran penuh agar robot mampu membaca medan magnet dari seluruh arah dan mendeteksi adanya ketidaklinieran. Hasil akhir dari kalibrasi adalah pembacaan kompas yang linier, sehingga *noise* yang diakibatkan oleh medan magnet gangguan dianulirkan dampaknya. Berikut merupakan informasi yang ditampilkan pada layar terminal setelah proses kalibrasi selesai dilakukan.

```
data : 1884
Delta X    : -527
Delta Y    : -1129
R    : 5243.21
Q    : 4225.11
Sigma      : 0.805824
Angle      : -1.2625
```

Data tersebut menunjukkan proses kalibrasi menggunakan 1884 data pembacaan sensor, yang dengan kata lain proses kalibrasi seleisai dilakukan setelah robot melakukan 1884 *cycle* program. Secara teori, sensor yang telah dikalibrasi mampu memberikan nilai bacaan sensor yang akurat. Pembuktian keberhasilan kalibrasi ini ditunjukkan pada pengujian selanjutnya, yaitu pembacaan sensor di empat arah.

Pengujian pembacaan sensor kompas dilakukan dengan menghadapkan robot ke arah gawang lawan (depan), 90 derajat dari gawang lawan (kanan), gawang sendiri (belakang), dan 270 derajat dari gawang lawan (kiri). Pembacaan orientasi dilakukan dalam 500 *cycle* dengan satu proses pengambilan nilai orientasi yang berarti didapatkan 500 data bacaan sensor.



Gambar 13. Pembacaan Sensor Kompas untuk Empat Arah Lapangan

Tabel 7. Pembacaan Sensor Kompas untuk Empat Arah Lapangan

Arah	Cycle	Bacaan Sensor
Depan	1-76	289
Kanan	162-202	200
Kiri	286-324	106
Belakang	423-500	17

Tabel 7 menunjukkan pembacaan sensor untuk empat arah di lapangan. Selisih bacaan sensor antara masing-masing arah merupakan representasi perbedaan sudut. Idealnya, perbedaan sudut antara depan dan kanan, kanan dan kiri, kiri dan belakang, serta belakang dan depan adalah 90 derajat. Menurut data pada tabel, selisih sudut antara keempat pasang arah tersebut dalam derajat adalah 89, 94, 89, dan 88. Selisih sudut disebabkan oleh penempatan arah robot yang tidak tepat 90 derajat pada saat pengujian. Meskipun begitu, data tersebut sudah membuktikan bahwa sensor mampu memberikan informasi yang cukup. Pada kenyataannya, untuk membuat suatu keputusan berdasar nilai bacaan sensor kompas, robot tidak membutuhkan nilai eksak, melainkan suatu rentang nilai.

Dengan kata lain, sub sistem navigasi memang tidak dituntut memberikan nilai yang pasti.

V. KESIMPULAN

Berdasarkan pengujian dan hasil analisis dari pengembangan platform robot sepak bola beroda ini diperoleh kesimpulan sebagai berikut:

1. Maksimal error yang diperoleh robot untuk melakukan gerakan dasar adalah 5% dari referensi.
2. Robot mampu untuk mendekati bola pada jarak 7.2 meter dengan gerakan yang halus dalam waktu rata-rata 13.2 detik.
3. Kedua kamera yang digunakan pada robot dapat berfungsi dengan baik dengan kamera atas untuk mendeteksi bola dekat sebagai prioritas utama dan kamera depan untuk mendeteksi bola jauh.

REFERENCES

[1] M. Asada, T. Balch, A. Bonarini, A. Bredenfeld, and S. Gutmann, "Middle Size Robot League Rules and Regulation", in International MSL Workshop, Kassel, Germany, 2016.

[2] Helder P. Oliveira, Armando J. Sousa, A. Paulo Moreira, Paulo J. Costa. “Modeling and Assessing of Omni-directional Robots with Three and Four Wheels”. Universidade do Porto.

[3] Y. Huashan, S. Guocan, W. Lina, Z. Min, "A new method based on Hough Transform for quick line and circle detection", Hubei University of Science and Technology, 2015.

[4] Kiam Heong Ang, G. Chong, Yun Li. “PID Control System Analysis, Design, And Technology”. IEEE Transactions on Control Systems Technology. 2005

[5] A. N. Fitriana, K. Mutijarsa, W. Adiprawita, "Color-Based Segmentation and Feature Detection for Ball and Goal Post on Mobile Soccer Robot Game Field", in Proc. ICITSI, 2015.

Dribbling and Kicking Mechanism of A Soccer Robot for "KRSBI Beroda"

Zaldi Sugianto*, Muhammad Teddy Kantori, Muhammad Iqbal Nugraha
Electronic Engineering
Politeknik Manufaktur Negeri Bangka Belitung (Polmanbabel)
Sungailiat-Bangka, Indonesia
zaldisugianto96@gmail.com

***Abstrak*—This paper discusses dribbling and kicking mechanism of a soccer robot applied in KRSBI Beroda or Robocup MSL. In the soccer robot competition, dribbling and kicking mechanism play important role for winning the match beside the robot mapping system. Dribbling system used in this paper has a little bit modification compared to most of popular dribbling system in which only used rubber wheels instead of omniwheels. After applying this dribbling system, the performance of robot on handling the ball was very similar to the one who used omniwheels. In term of kicking system, a single power window motor was used as the main actuator of the kicker. By using this actuator, the robot was able to kick the ball with a maximum high of the ball from the ground at approximately 50cm and then the ball rolled for around 15m.**

Katakunci*—soccer robot; dribbling; kicking; rubber wheels; actuator

I. INTRODUCTION

Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda is a national robot competition in the field of soccer game competition held in Indonesia. This competition refers to the international soccer robot competition called Robocup-Medium Size League (MSL) and the winner of KRSBI Beroda will compete in Robocup-MSL as Indonesia's representation. This paper discusses specifically about dribbling and kicking mechanism of a soccer robot. In [1], kicker systems were categorized into three kinds of mechanism which are stroke mechanism through spring, stroke mechanism through pneumatic, and stroke mechanism through solenoid. In term of power, price, robustness and dimension, solenoid was considered as the best option for the kicking system[1]. This kicker system has also been adopted by most of robot teams[1,2,3,4]. According to [2], the spring based kicker system used a worm gear before the rack and pinion. In this case, the spring gave less benefit to the kicker because of the worm gear. As a result, the kicker system did not perform well. This paper implemented a new kicker system in which mechanically without spring, pneumatic or solenoid. The kicker system only applied a DC motor with a worm gear. This kind of kicker is very simple and easy to control as it has a stable torque and constant speed. For dribbling system, this paper shows a modified design of [1,2,3] which it does not use springs and ball joints. Instead, the dribbling system only implemented simple rotational joints and relied on gravity.

II. DRIBBLING MECHANISM

Dribbling mechanism is important for handling the ball during the match. If the robot is not equipped with a dribble system, the robot will difficult to control or take the ball to opponent's area or another area depending on needs or strategy. Hence, dribbling system has been considered as one of the most important thing in the soccer robot.

According to Robocup-MSL rule, the maximum area of the ball which can be touched during dribbling is one third of the ball size. It means that a specific design of ball handling must be carefully designed and built.

A. Mechanical Construction

Basically, the mechanical design of dribbling system is similar to [2] where it consists of some wheels for touching the ball. Actually, there are four wheels in which two of them are omniwheels and the rest are rubber wheels. The latter are driven by DC Motor to catch and hold the ball. In [2], all of wheels are omniwheels.

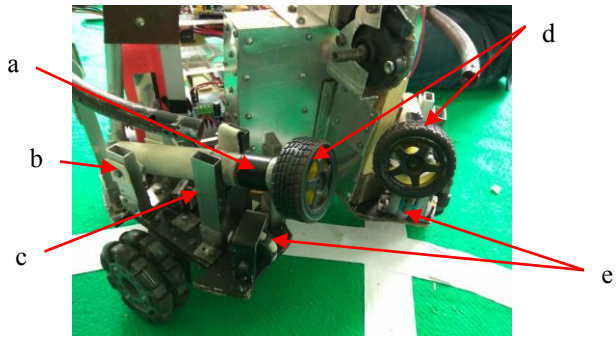


Fig.1 Dribbling System; (a) DC Motor, (b) Rotational joint, (c)Stopper, (d) Wheels, (e) Omniwheels

Rubber wheels are more simple and cheaper than omniwheels. Besides, the performance in controlling the ball is very similar with the one who used the omniwheels.



Fig. 2. Dribbling a ball

B. *Actuator Specifications*

For dribbling system, Maxon DC motor 471949 was used. This DC motor has a maximum speed of 2000 rpm and torque up to 1.6 Nm. The voltage applied was 12 VDC with the speed of 450 rpm.

Table 1. Dribble Motor Specifications

Motor	
Diameter	25 mm
Power	20 W
Nominal Voltage	15 V
Idle Speed	9660 rpm
Torque (max)	20.5 mNm
Gear	
Diameter	32 mm
Reduction	5.3 : 1
Max. Intermittent Torque	1.6 Nm

Without load or ball, this DC motor consumed current at around 1.7A. If the ball was attached to the load, the current flew through the motor increased which was up to 2.2A. When the robot was moved whether it was forward, backward or rotate, the current consumption did not change or remained stable.

III. KICKING MECHANISM

A soccer robot must have a kicker. How strong and accurate of the kicking result very depends on the kicking mechanism. In the KRSBI beroda, a Futsal ball is used during the competition which is around 410 gram of weight and about 62mm in diameter. The kicker was designed so that this Futsal ball can be kicked as fast as possible.

A. *Mechanical Construction*

In the kicker system, a power window motor had been used. This motor was chosen as it produces a high torque which is up to 10Nm. Therefore, the arm of kicker can be attached directly to its shaft.

However, the motor speed is quite low which is around 100 rpm at 12 V. This level of speed is not fast enough to kick the ball. Hence, the speed must be increased. To speed up the movement of end-effector of the kicker, the length of kicker's arm was made longer (220mm). As long as the angular velocity does not change, then the speed of the end-effector will increase.

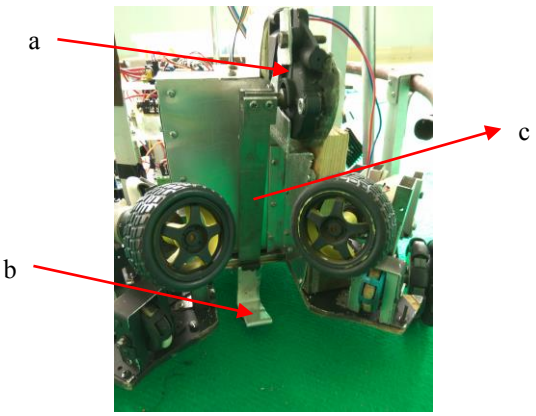


Fig. 3. Kicker System; (a) power window motor, (b) end-effector, (c) kicker's arm

B. *Actuator of Kicker*

In this paper, a 24VDC LIPO battery was used to power the power window motor. The purpose was to increase the performance of power window motor. As a result, the motor speed go up to 200 rpm and the motor torque rise up to almost double.

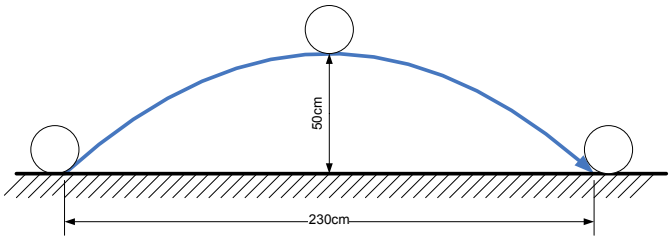
Table 2. Power Window Motor Specification

Voltage rating	12VDC
No Load Speed	100RPM
Rated Speed	75RPM
Current (No Load)	<5A
Rated Current (Load)	<15A
Stall Current	<28A at 12V
Rated Torque	30kgcm (2.9Nm)
Stall Torque	100kgcm(~10Nm)

C. *Experimental Result*

After implementing a 24 VDC power supply to the power window motor, the rated current was still below 15A, which was 10.5A. This current value did not change whether it kicked a ball or kicked without a ball.

Based on experimental testing, the robot was able to kick the ball with a maximum high of the ball from the ground at approximately 50cm and then the ball rolled for around 15m. Furthermore, it took less than one second for the ball to touch the land again after being kicked, i.e the ball flied away for less than one second.



(a)



(b)

Fig. 4. Experimental result; (a) the distance of the ball after kicking, (b) ball after being kicked

In fig.4, it can be seen that the impact of the kicker was strong and fast enough to make the ball fly away.

IV. CONCLUSION

Rubber wheels are good enough to hold the ball during dribbling. Beside that, it is quite powerfull to seize the ball from opponent. For kicking system, instead of using the spring, pneumatic or solenoid, a power window motor can be considered as a new option for the actuator. This is due to high torque and constant speed so that it can be easily controlled with only a single unit step signal.

Acknowledgment

We would like to thank very much to director and vice director of Polmanbabel who always supported us to compete in KRSBI beroda. We are also very grateful to all our sponsors.

References

[1] Kasaei, S.Hamidreza Mohades, S. Mohammadreza Mohades Kasaei, and S.Alireza Mohades Kasaei. "Design and Implementation Solenoid Based Kicking Mechanism for Soccer Robot Applied in Robocup-MSL." *Advanced Robotic Systems* vol. 7., No. 4 (2010) ISSN 1729-8806, pp.73-80

[2] Serafim, M.S. (2013). “Ball handling Mechanism for Mobile Robots”. Master’s Thesis. Tecnico Lisboa

[3] Beck D., Ferrein A., Lakemeyer G. (2007). “A Simulation Environtment for Middle-size Robots with Multi-level Abstraction”. Computer Science Department. RWTH Aachen. Germany

[4] Martinez C. L., Schoenmakers F., Naus G. (2014). “Tech United Eindhoven, Winner Robocup 2014 MSL”. Eindhoven University of Technology

Pendeteksian Bola Orange dengan Metode Contour pada Robot R2C-WARRIOR

Ane Cornelia
Fakultas Teknik Elektronika dan Komputer
Universitas Kristen Satya Wacana
Salatiga, Indonesia
ac.cornelia@gmail.com

***Abstrak*—Kontes Robot Indonesia selalu mengalami perubahan peraturan tiap tahun. Salah satu perubahannya yaitu perubahan divisi beroda yang semula merupakan divisi Kontes Robot Pemadam Api Indonesia (KRPAI) menjadi Kontes Robot Sepak Bola Indonesia (KRSBI). Dengan adanya perubahan peraturan yang signifikan ini, robot diharuskan untuk dapat mendeteksi bola. Pada KRI 2017, divisi KRSBI beroda akan menggunakan bola futsal FIFA ukuran empat berwarna orange. Dengan algoritma pendeteksian bola yang digunakan pada divisi KRBSI Humanoid R2C-R9 sebelumnya, pendeteksian menggunakan segmentasi warna, bola dapat terdeteksi. Namun algoritma ini memiliki kelemahan. Untuk mengatasi kelemahan tersebut, algoritma baru dengan menggunakan fitur contour, akan dibuat tanpa mengabaikan waktu pemrosesan dan memori yang digunakan.**

Katakunci— KRI2017; contour; KRSBI; Beroda; R2C-WARRIOR.

I. PENDAHULUAN

Kontes Robot Indonesia (KRI) adalah kontes robot nasional di Indonesia yang diselenggarakan tiap tahun. Ada lima divisi yang diperlombakan pada KRI yaitu: Kontes Robot Sepak Bola Indonesia (KRSBI) Humanoid, Kontes Robot Pemadam Api Indonesia (KRPAI) Beroda, KRPAI Berkaki, Kontes Robot Abu Indonesia (KRAI), dan Kontes Robot Seni Tari Indonesia (KRSTI). Setiap tahun selalu terjadi perubahan peraturan pada KRI. Pada KRI 2017 terdapat perubahan peraturan yang cukup signifikan. Pada KRI 2016, KRPAI Beroda bertugas untuk memadamkan api dan menyelamatkan bayi. Namun pada KRI 2017, KRPAI Beroda harus dikembangkan menjadi KRSBI Beroda. Perubahan divisi yang semula KRPAI menjadi KRSBI merupakan tantangan tersendiri bagi tim untuk membuat sistem baru, baik secara algoritma, citra, komunikasi, ataupun mekinik. Pada bagian citra, tentunya sebagai divisi KRSBI, robot diharuskan untuk dapat mendeteksi bola. Bola yang akan digunakan pada perlombaan adalah bola futsal FIFA ukuran empat.

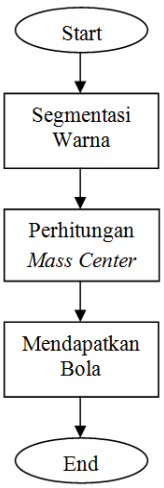
Tidak hanya mampu mendeteksi bola, namun keefektifan robot dalam menjalankan algoritma seperti kompleksitas komputasi dan memori yang digunakan juga harus diperhatikan. Semakin kompleks komputasi yang dilakukan dan semakin banyak memori yang digunakan, maka proses yang dijalankan akan semakin lambat. Pada kenyataannya, robot dituntut untuk dapat cepat dan *real-time* dalam melakukan pemrosesan.

Dengan mementingkan keakurasian pendeteksian bola tanpa mengabaikan kompleksitas komputasi dan memori yang digunakan, maka dibuatlah algoritma ini yang mengambil ide dari penelitian [2]. Algoritma ini dibuat dengan menggunakan fungsi *contour* yang ada pada *OpenCV*. *Contour* yang dijadikan dasar dari pendeteksian bola pada algoritma ini diharapkan dapat menangani permasalahan yang terjadi pada algoritma sebelumnya.

Paper ini terdiri dari 5 bagian. Bagian I merupakan bagian pengenalan, berisi latar belakang dan permasalahan kasus yang akan diselesaikan. Bagian II adalah penjelasan singkat tentang algoritma yang sebelumnya pernah dibuat. Bagian III berisi penjelasan tentang algoritma yang dibuat pada paper ini. Bagian IV merupakan hasil pengujian beserta pembahasan. Dan bagian terakhir, yaitu bagian V berisi kesimpulan dan pengembangan kedepannya.

II. DASAR TEORI

Pengolahan citra yang digunakan sebelumnya pada divisi KRSBI Humanoid R2C-R9 untuk pendeteksian bola *orange* adalah dengan menggunakan segmentasi warna. Gambaran singkat algoritma dapat dilihat pada gambar 1.



Gambar 1 *Flowchart* algoritma lama KRSBI Humanoid R2C-R9

Dengan menggunakan algoritma seperti pada gambar 1, pendeteksian bola *orange* menggunakan segmentasi warna dapat berjalan dengan baik. Algoritma ini memiliki tingkat komplesitas komputasi yang rendah. Pertama, segmentasi warna akan dilakukan pada gambar yang diinput, dalam hal ini merupakan gambar yang berasal dari kamera. Segmentasi warna dapat disebut sebagai kalibrasi. Dengan melakukan kalibrasi, input gambar akan diubah menjadi gambar biner. Kemudian akan dilakukan perhitungan untuk mendapatkan nilai titik tengah bola dengan melakukan perhitungan *mass center pixel* yang bernilai 1.

Kelemahan dari algoritma ini adalah jika terdapat latar yang berwarna sama dengan warna bola, maka hasil perhitungan untuk mencari *mass center* bola akan terganggu. Kejadian ini akan menghasilkan nilai yang tidak *valid* dan menyebabkan *false alarm*.

Dari kelemahan tersebut, algoritma baru yang dibuat ini diharapkan dapat menyelesaikan dan mengatasi permasalahan pendeteksian bola tanpa memiliki tingkat kompleksitas komputasi yang besar.

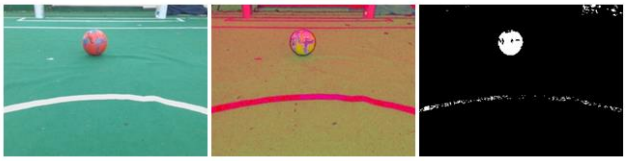
III. METODE YANG DIGUNAKAN

A. *Overview*

Pertama, input gambar akan diambil dengan menggunakan *webcam*. Segmentasi warna akan digunakan pada input gambar untuk mengubah gambar dari kanal *Red Green Blue* (RGB) menjadi gambar biner untuk mempermudah proses pengolahan selanjutnya. Gambar biner tadi kemudian akan di proses untuk dicari *contour*nya. Dengan bantuan fungsi *MinEnclosingCircle*, *contour* yang didapatkan akan dihitung untuk pendeteksian bola. Jika bola tidak dapat terdeteksi, artinya bola memang tidak terdeteksi. Ini disebabkan karena *cost* dari *missed detection* lebih kecil daripada *cost* dari *false alarm* pada pendeteksian bola.

B. *Preprocessing Section*

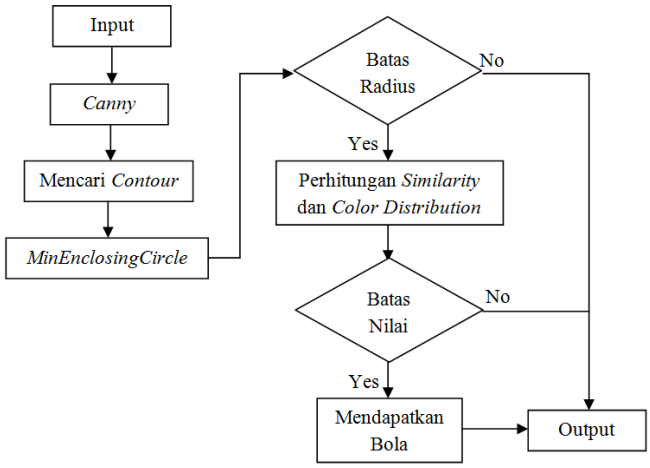
Sebelum melakukan pengambilan gambar, data untuk melakukan segmentasi warna diperlukan terlebih dahulu. Kalibrasi atau segmentasi warna merupakan hal yang sangat penting karena sangat berpengaruh terhadap keberhasilan pendeteksian bola. Gambar akan dikalibrasi dengan menggunakan kanal *Hue Saturation Value* (HSV). HSV merupakan kanal yang lebih toleran terhadap perubahan pencahayaan dibandingkan *grayscale*, RGB, dan lainnya, walaupun pencahayaan tetap berpengaruh pada kanal HSV. Kalibrasi yang dilakukan adalah dengan memilih warna bola orange jadi bernilai 1, sedangkan warna lainnya bernilai 0.



Gambar 2 Segmentasi warna pada bola *orange*. Gambar RGB akan diubah menjadi HSV , kemudian dilakukan kalibrasi untuk mengahsilkan gambar biner

C. *Contour Section*

Setelah proses segmentasi warna selesai dilakukan, fungsi *canny* akan digunakan pada gambar biner. Fungsi *canny* merupakan sebuah detektor tepian yang sampai saat ini masih dikenal sebagai detektor tepian yang paling optimal. Fungsi *canny* memiliki *error rate* yang rendah, prediksi tepian yang menyerupai tepian aslinya, dan dapat meminimalkan tepian ganda pada suatu tepian. Hasilnya akan digunakan untuk mencari *contour* pada gambar. Selanjutnya, *contour* akan diproses menggunakan fungsi *MinEnclosingCircle*. Fungsi *MinEnclosingCircle* adalah fungsi yang digunakan untuk memprediksi bola dengan memasukan *contour* ke dalam lingkaran yang disesuaikan dengan besar dan posisi *contour*. Dengan menggunakan fungsi ini, *similarity* dan *color distribution* dapat dihutung untuk pendeteksian bola. Algoritma dari metode ini dapat dilihat pada gambar 3.



Gambar 3 *Flowchart* algoritma metode *contour*

Perhitungan *similarity* akan didapatkan dengan melakukan pembagian luas *contour* dengan luas *enclosing circle*. Rumus lingkaran dapat digunakan untuk mendapatkan luas *enclosing circle* dimana jari-jari dari lingkaran bisa didapatkan dari fungsi tersebut. Untuk menghitung luas *contour*, *mass center* dari *contour* harus diketahui. Persamaan (1) digunakan untuk melakukan perhitungan *moments* dari *contour* atau dapat langsung menggunakan fungsi *moments* yang ada pada *OpenCV* [4].

$$m_{ij} = \sum_x \sum_y x^i y^j I(x,y) \quad (1)$$

dengan *I(x,y)* adalah gambar atau dalam kasus ini adalah *contour*. Sedangkan *i* dan *j* ± 0, 1, 2 … bergantung dari *moments* yang akan dihitung. Setelah mendapatkan perhitungan dari *moments* yang dibutuhkan, persamaan (2) akan digunakan untuk mencari *mass center* dari *contour*.

$$x = \frac{m_{10}}{m_{00}} \quad y = \frac{m_{01}}{m_{00}} \quad (2)$$

Luas *contour* dapat dihitung setelah memiliki *mass center* dari *contour*. Jika hasil perhitungan *similarity* mendekati ~1, artinya *contour* yang didapatkan memiliki bentuk yang mirip dengan lingkaran.

Color distribution merupakan distribusi warna yang ada pada bola, yaitu dengan menghitung jumlah *pixel* berwarna yang telah dipilih yang ada di dalam lingkaran yang telah terprediksi sebagai bola. *Color distribution* adalah perhitungan dari *ball pixel ratio*, tiap *pixel* dalam *enclosing circle* akan dicek apakah *pixel* tersebut merupakan bagian dari bola atau bukan. *Ball pixel ratio* didapatkan dengan membagi jumlah *pixel* bola berwarna *orange* dengan jumlah *pixel* pada *enclosing circle*. Jika hasil dari *color distribution* mendekati ~1, maka artinya bagian tersebut menyerupai bentuk lingkaran.

Selain *similarity* dan *color distribution*, nilai jari-jari minimum dan maksimum dapat dijadikan salah satu batasan untuk membantu proses pendeteksian bola. Dengan pemberian nilai jari-jari minimum dan maksimum bola, diharapkan dapat mengurangi komplesitas komputasi, waktu pemrosesan, dan mengurangi *noise.*

IV. ALGORITMA

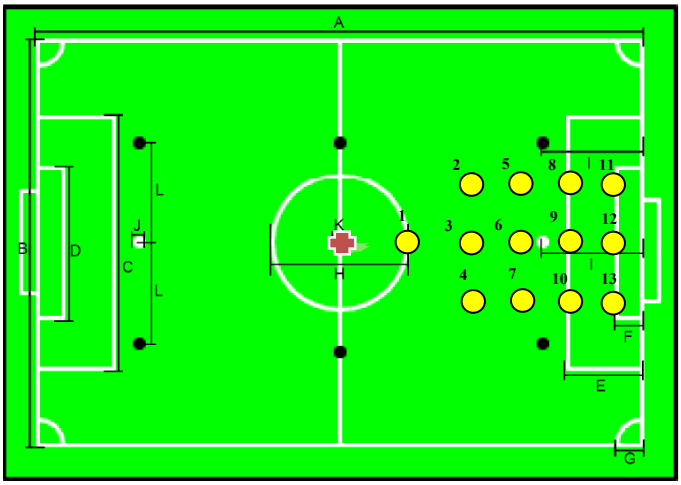
Dengan menggunakan metode *contour* yang sudah dijelaskan sebelumnya, masalah pendeteksian bola dapat diselesaikan. Pada umumnya gambar yang akan diproses menggunakan metode *contour* akan menghasilkan dua buah nilai yaitu: *similarity* dan *color distribution*. Pemberian nilai batas sangat dianjurkan untuk digunakan. Jika kedua nilai berada di atas nilai batas, maka *contour* akan dimasukan sebagai kandidat bola. Kemudian nilai *similarity* dan *color distribution* akan dijumlahkan, dan penjumlahan yang paling tinggi akan dianggap sebagai bola.Untuk mempercepat waktu pemrosesan, pemberian nilai jari-jari mimimal dan maksimal harus diberikan setelah fungsi *MinEnclosingCircle* digunakan dan sebelum perhitungan *similarity* dan *color distribution*. Jika salah satu nilai berada dibawah batas, maka akan dianggap bukan kandidat bola.

V. HASIL EKSPERIMEN

Pengujian dilakukan dengan menggunakan gambar statik. Kamera diletakan setinggi ~45cm dari permukaan lapangan. Tinggi tersebut telah disesuaikan dengan tinggi kamera pada robot R2C-WARRIOR. Pengujian ini menggunakan *webcam Logitech C922* dengan resolusi 640*480. Prosessor yang digunakan adalah *Laptop AMD A8* dengan *OS Ubuntu 14.2 LTS*. Pengujian dilakukan menggunakan bola futsal *Adidas* FIFA ukuran empat berwarna *orange*.

A. Scenario Eksperiment

Pengujian ini dilakukan dengan menggunakan 13 posisi bola yang berbeda-beda sebanyak 30 *frame* tiap posisi. Konfigurasi posisi dapat dilihat pada Gambar 4, dengan lingkaran beserta nomor menunjukan posisi bola, sedangkan di tengah lapangan atau pada tanda + merupakan posisi peletakan kamera. Pengujian ini bertujuan untuk menguji kestabilan algoritma.



A	7.2		G	0.3
B	4.8		H	1.6
C	Lebar gawang + 1.8		I	1.2
D	Lebar gawang + 0.6		J	0.04
E	0.9		K	0.06
F	0.3		L	1.2

Gambar 4 Konfigurasi lapangan KRI 2017 divisi KRSBI Beroda (dalam meter) pada *scenario experiment*

Berikut ini merupakan hasil percobaan pada *scenario experiment* dengan metode *contour*.

TABEL I KINERJA ALGORITMA LAMA DAN ALGORITMA BARU PENDETEKSIAN BOLA ORANGE PADA SCENARIO EXPERIMENT

Posisi	Jarak (cm)	*Frame*	Algoritma Lama		Algoritma Baru	
			Detected	FPS	*Detected*	FPS
1	80	30	30	58.41	30	47.28
2	150	30	30	57.5	30	50.63
3	150	30	30	55.96	30	50.21
4	150	30	30	58.16	30	46.38
5	200	30	30	54.65	30	50.24
6	200	30	30	56.06	30	47.6
7	200	30	30	55.7	30	49.62
8	270	30	30	57.58	30	50.14
9	270	30	30	56.13	30	45.31
10	270	30	30	57.7	30	47.62
11	330	30	30	55.66	30	47.81
12	330	30	30	54.52	30	50.36
13	330	30	30	54.01	30	50.11

Dalam pengujian ini, algoritma baru yang merupakan algoritma dengan metode *contour* adalah algoritma yang stabil. Dari 13 posisi dengan 30 *frame* tiap posisinya, tidak terdapat kesalahan pada pendeteksian bola. Algoritma lama dan algoritma baru memiliki tingkat keakurasian dan kestabilan yang sama pada percobaan ini. Hanya saja untuk

waktu pemrosesan, algoritma yang baru sedikit lebih lambat dari algoritma yang lama.

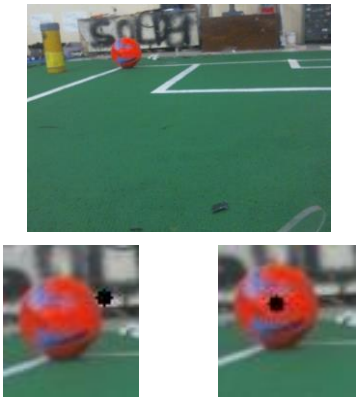
B. *Random experiment*

Pengujian ini dilakukan dengan menggunakan 60 gambar statik dimana lokasi bola berbeda tiap gambarnya. 30 gambar diantaranya diambil dengan posisi kamera yang sama, yaitu di tengah lapangan. Peletakan kamera di posisi yang sama ini diatur untuk melihat kinerja algoritma tanpa adanya pengaruh gangguan dari luar misalnya perubahan cahaya. Sedangkan 30 gambar lainnya diambil dengan posisi kamera dan bola yang berbeda-beda, yang bertujuan untuk melihat fleksibilitas algoritma dengan adanya perubahan lingkungan dan pencahayaan. Jarak maksimum antara kamera dengan bola ~360cm, sedangkan jarak minimumnya ~60cm.

TABEL II KINERJA ALGORITMA LAMA DAN ALGORITMA BARU PENDETEKSIAN BOLA ORANGE PADA RANDOM EXPERIMENT

Nomor	*Frame*	Algoritma Lama		Algoritma Baru	
		Detected	**FPS**	***Detected***	**FPS**
1-30	30	30	53.78	30	50.09
31-60	30	28	47.46	30	39.93

Pada tabel II, nomor 1-30 menunjukan kondisi pertama dimana peletakan kamera berada di satu tempat, sedangkan nomor 31-60 menunjukan kondisi kedua dimana kamera mengalami pergantian posisi. Seperti yang dapat dilihat pada hasil *Random Experiment*, pada kondisi pertama, keakurasian dan waktu pemrosesan algoritma baru sudah memuaskan. Pada kondisi kedua, terlihat keunggulan dari algoritma baru. Pada algoritma lama, terdapat dua buah *false alarm*. Ini disebabkan karena adanya objek lain yang memiliki warna sama dengan warna bola. Sedangkan pada algoritma baru, objek yang memiliki warna yang sama dengan warna bola tidak akan mempengaruhi pendeteksian bola, kecuali objek tersebut juga berbentuk bola.



(a) Algoritma Lama (b) Algoritma Baru

Gambar 5 Hasil pendeteksian pendeteksian bola pada *Random Experiment*

Gambar 5 merupakan salah satu contoh kesalahan deteksi yang terjadi pada algoritma lama dan berhasil pada algoritma baru. Jika diperhatikan pada gambar asli, terdapat benda di dekat kanan atas, berwarna hampir menyerupai bola. Prediksi titik bola meleset sedikit dari bola. Ini disebabkan karena warna benda yang memiliki warna menyerupai bola sangat kecil, sehingga tidak terlalu dominan. Jika terdapat benda yang memiliki warna sama dengan bola dan berukuran cukup besar, maka prediksi bola akan semakin tidak sesuai. Ini menyebabkan algoritma lama tidak dapat bekerja dengan baik. Namun pada algoritma baru, masalah tersebut dapat diatasi dengan baik.

VI. KESIMPULAN

Metode *contour* merupakan metode yang baik untuk digunakan untuk pendeteksian bola. Metode ini memiliki tingkat keakurasian yang tinggi dan waktu pemrosesan yang baik, bergantung pada banyaknya *contour* yang dihasil, sekitar 45-50 fps.

Untuk pengembangan selanjutnya, akan diteliti dan diuji lebih lanjut untuk memperbaiki kinerja algoritma pendeteksian bola menjadi lebih efisien terutama untuk mengatasi gangguan dari luar seperti faktor pencahayaan.

REFERENCES

[1] Buku Panduan Kontes Robot Sepakbola Indonesia Beroda (KRSBI Beroda) 2017. RISTEKDIKTI. Dec. 2016.

[2] J. Mu and Y. Li, "*A New Efficient Real-Time Arbitrary Colored Ball Recognition Method for A Humanoid Soccer Robot*," 2016 12th World Congress on Intelligent Control and Automation (WCICA), vol. 16, pp.494-499, Jun. 12-15 2016.

[3] M. K. Hu, "*Visual Pattern Recognition by Moment Invariants*", IRE Transactions on Information Theory, vol. IT-8, pp.179–187, 1962.

[4] *Structural Analysis and Shape Descriptors*. [Online]. Available: http://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html, accesed Jun. 2017.

[5] *Image Moments*. [Online]. Available: https://en.wikipedia.org/wiki/Image_moment, accessed Jun. 2017.

Sistem Deteksi Robot Lawan Berdasarkan Warna Identitas Robot Pada Robot Barelang 7.1

Devi Gusnaensi Tinambunan*, Senanjung Prayoga, Imam A. Riandi, Rifqi A. Fatekha
Politeknik Negeri Batam
Mechatronics Engineering Study Program
Batam, Kepulauan Riau, Indonesia
devigusn@gmail.com

***Abstrak*—Tahun 2017 ini, KRI kembali diadakan dan peraturan KRSBI Beroda mengacu kepada Robocup Middle Size League (MSL). Pada peraturan selama permainan, robot dilarang keras untuk menabrak atau mendorong robot lawan. Jika hal tersebut terjadi maka robot akan mendapatkan kartu kuning dan beresiko mendapatkan kartu merah dan dikeluarkan dari lapangan jika robot menabrak lawan dengan keras. Untuk mencegah hal tersebut, robot harus dilengkapi dengan sebuah sistem deteksi robot lawan. Pada penelitian ini dikembangkan sistem deteksi warna identitas robot dengan menggunakan segmentasi HSV ditambah filter dilasi, erosi, dan kontrol blob. Ukuran dari warna identitas robot dideteksi menggunakan library cvblob. Hasil dari penelitian ini, robot Barelang 7.1 mampu mendeteksi robot lain dengan baik dan tingkat keberhasilan sistem membaca warna identitas robot adalah sebesar 90.9%.**

Katakunci—HSV; kamera; dilasi; erosi; blob

I. INTRODUCTION

Kontes Robot Sepakbola Beroda Indonesia diadakan untuk meningkatkan keilmuan dan kreatifitas mahasiswa di bidang robotika. Kontes ini diselenggarakan berdasarkan aturan yang dilakukan di *RoboCup Middle Size League* (MSL). Berdasarkan peraturan permainan nomor 12 tentang *fouls*, pada peraturan nomor 12.3.2 tentang mendorong robot lain, robot harus berusaha untuk menghindari kontak fisik untuk mencegah terkena *pushing foul* [1]. Berdasarkan peraturan inilah peneliti membuat sebuah sistem deteksi robot untuk menghindari kontak fisik dengan lawan. Sistem ini menggunakan kamera yang diletakkan pada robot. Kamera berfungsi untuk mendeteksi robot lawan berdasarkan warna identitas dari robot.

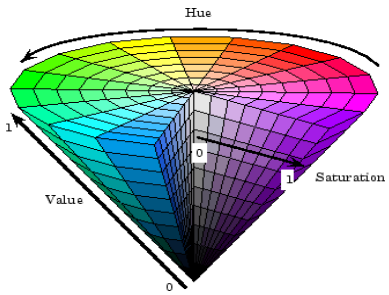
Berdasarkan penelitian Chau dan Ian dari *University of Kent* [2], mereka membuat sebuah *low-cost* robot yang dapat menghindari rintangan. Filter gambar yang digunakan adalah filter RGB. Hasil yang didapatkan dari filter gambar dapat digunakan untuk menentukan jarak diantara kedua robot tetapi karena filter gambar yang kurang baik dan karena resolusi gambar yang kurang tinggi membuat robot tidak bisa mendeteksi jarak hanya mengandalkan *vision* kamera saja. Lalu berdasarkan penelitian Calvinsan dari Politeknik Negeri Batam [3], ia menggunakan sistem deteksi objek dengan menggunakan ekstrasi citra HSV. Proses ini memiliki tingkat keberhasilan 90% jika dibandingkan dengan ekstrasi citra RGB yang tingkat keberhasilanya sebesar 66.7 %. Kelemahan metode ini adalah pengenalan objek yang hanya menggunakan ciri dari ekstrasi warna objek tanpa memperhatikan ciri lain dari objek tersebut. Objek yang dideteksi adalah bola.

Dari dua penelitian ini, penulis akan membuat sebuah sistem deteksi robot lawan pada robot Barelang 7.1. Sistem ini dibangun dari vision kamera pada robot. *Vision* pada kamera menggunakan filter warna HSV ditambah filter dilasi, erosi, dan kontrol blob. Untuk mengurangi pengaruh perubahan intensitas cahaya di lapangan, maka fitur-fitur auto dari kamera dinon-aktifkan. Ukuran panjang dan lebar warna identitas robot yang dideteksi menggunakan *library cvblob*. Dengan demikian, sistem diharapkan dapat mengurangi kesalahan deteksi sehingga proses pendeteksian warna identitas robot dapat berjalan dengan akurat dan *real time*.

II. DASAR TEORI

A. Model Warna HSV

Model HSV (*Hue, Saturation, Value*) menunjukkan ruang warna dalam bentuk tiga komponen utama yaitu *hue, saturation,* dan *value* (atau disebut juga *brightness*). *Hue* adalah sudut dari 0 sampai 360 derajat. Biasanya 0 adalah merah, 60 derajat adalah kuning, 120 derajat adalah hijau, 180 derajat adalah *cyan*, 240 derajat adalah biru dan 300 derajat adalah *magenta* seperti pada Gambar 1. *Hue* menunjukkan jenis warna (seperti merah, biru, kuning), yaitu tempat warna tersebut ditemukan dalam spectrum warna. Saturasi (*saturation*) suatu warna adalah ukuran seberapa besar kemurnian dari warna tersebut akibat pengaruh dari warna putih. Seperti warna merah, dengan pengaruh warna putih, warna merh menjadi bervariasi dari warna merah menuju merah muda, yang artinya *hue* masih tetap bernilai merah tetapi nilai saturasinya berkurang. Komponen HSV berikutnya adalah nilai *value* atau disebut juga intensitas, yaitu ukuran seberapa besar kecerahan suatu warna atau seberapa besar cahaya datang dari suatu warna. *Value* memiliki nilai dengan jangkauan 0% sampai 100% [4]. Berikut gambar model warna HSV.



Gambar 1 Model Warna HSV

B. *Erosi*

Erosi adalah proses penghapusan titik-titik batas objek menjadi bagian dari latar, berdasarkan *structuring element* yang digunakan. Pada operasi ini, ukuran objek diperkecil dengan mengikis sekeliling objek [5]. Cara yang dapat dilakukan ada 2 yaitu dengan mengubah semua titik batas menjadi titik latar, dan menetapkan semua titik di sekeliling titik latar menjadi titik latar. Berikut ini adalah contoh gambar yang menjelaskan proses erosi.



Gambar 1 Representasi Proses Erosi

C. *Dilasi*

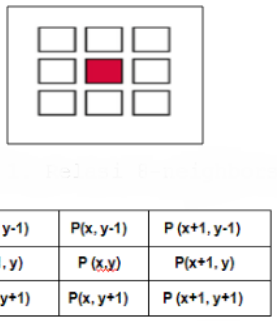
Dilasi merupakan proses penggabungan titik-titik latar menjadi bagian dari objek, berdasarkan *structuring element* yang digunakan [5]. Proses ini adalah kebalikan dari erosi, yaitu merubah latar disekeliling objek menjadi bagian dari objek tersebut. Ada 2 cara untuk melakukan dilasi. Pertama dengan mengubah semua titik latar yang bertetangga dengan titik batas menjadi titik objek, atau lebih mudahnya setting setiap titik yang tetangganya adalah titik obyek menjadi titik objek. Kedua dengan mengubah semua titik di sekeliling titik batas menjadi titik obyek, atau lebih mudahnya menetapkan semua titik tetangga sebuah titik obyek menjadi titik obyek. Berikut ini adalah contoh gambar yang menjelaskan proses dilasi.



Gambar 2 Representasi Proses Dilasi

D. *Blob*

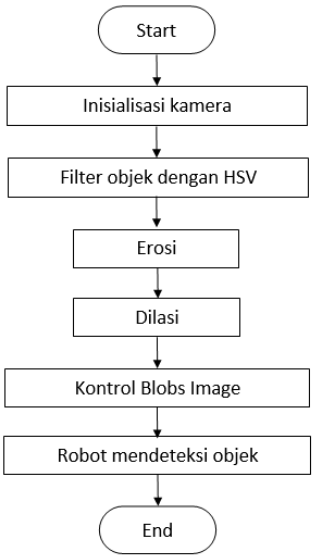
Blob merupakan sekumpulan piksel - piksel yang memiliki hubungan tetangga. Proses perhitungan blob dapat dilakukan dengan melakukan analisis piksel yang bertetangga. Piksel bertetangga pada sebuah piksel ditentukan sebagai piksel yang berjarak 1 dari piksel asal. Proses perhitungan blob akan memanfaatkan relasi piksel 8- *neighbors*. Gambar 4 merupakan gambaran sederhana dari relasi 8 – neighbors [6]. Pada filter untuk deteksi warna identitas lawan, akan dilakukan control nilai blob untuk membatasi nilai blob minimal agar dapat terdeteksi. Jika kurang dari nilai tersebut maka warna identitas robot tidak akan terdeteksi oleh kamera.



Gambar 3 Relasi 8 – *Neighbors*

III. PERANCANGAN SISTEM

Pada robot Barelang 7.1, *flowchart* sistem deteksi warna identitas robot yang digunakan adalah sebagai berikut.



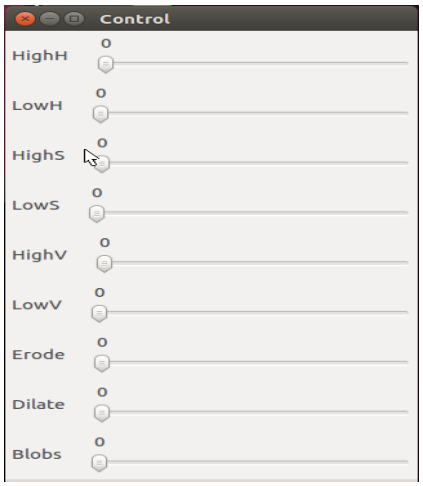
Gambar 4 *Flowchart* Sistem

Tahap pertama yang dilakukan oleh sistem adalah inisialisai kamera. Lalu tahap selanjutnya adalah pengambilan gambar oleh kamera yang langsung difilter dengan filter warna HSV. Pemfilteran dilakukan secara *real time*. Citra hasil filter HSV secara langsung dibinarisasi untuk selanjutnya difilter dengan erosi dan dilasi. Erosi dan dilasi digunakan untuk menghilangkan *noise-noise* yang masih ada dari hasil pemfilteran dengan filter HSV. Faktor penentu kapan objek harus dilakukan erosi dan dilasi adalah hasil dari filter objek dengan HSV. Jika hasil filter HSV saja sudah baik dan tidak ada *noise* pada gambar, maka erosi dan dilasi tidak perlu dilakukan. Namun sebaliknya jika hasil dari filter dengan HSV masih ada *noise* yang tersisa dapat dihilangkan dengan mudah dan cepat dengan erosi dan dilasi. Erosi dan dilasi dilakukan sampai *noise* pada gambar hilang dan objek dapat terdeteksi dengan akurat. Selain erosi dan dilasi, citra hasil filter juga di kontrol nilai *blobs*nya. Nilai *blobs* adalah jumlah piksel dari

objek pada frame kamera. Tujuan dari kontrol blob *image* adalah membatasi nilai blob minimal suatu objek untuk dideteksi dan faktor penentu batasan nilai blob adalah titik terjauh bola untuk dideteksi. Ketika semua filter telah dilakukan, sistem akan mendeteksi dan mengenali objek yaitu warna identitas robot sesuai dengan yang diinginkan.

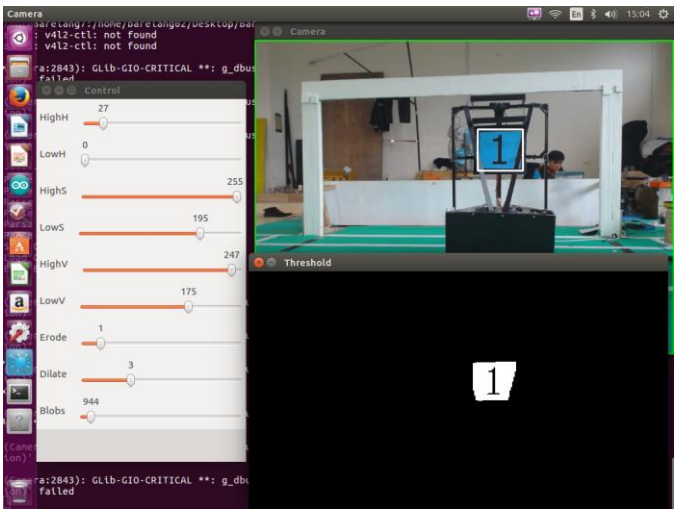
IV. HASIL DAN ANALISA

Untuk mendapatkan hasil dan analisa maka perlu dilakukan beberapa pengujian. Pengujian yang dilakukan yaitu dengan meletakkan robot di lapangan lengkap dengan warna identitas robot. Sample yang diambil adalah deteksi warna *cyan* dan *magenta* dari rentang jarak 5 cm sampai 7,5 m. Untuk filter objeknya, peneliti membuat sebuah kontrol *trackbar* dengan menggunakan pemograman C pada linux. Kontrol tersebut terdiri dari 9 buah *trackbar* yaitu HighH (High *Hue*), LowH (Low *Hue*), HighS (High Saturation), LowS (Low Saturation), HighV (High Value), LowV (Low Value), Erode, Dilate, dan *Blobs*. *Trackbar* untuk nilai *hue*, *saturation* dan *value* terbagi menjadi dua yaitu *High* dan *Low* untuk membuat rentang nilai. *Trackbar* HighH sampai LowV bernilai 0 – 255 sedangkan *trackbar erode* dan *dilate* berada pada rentang 0 – 10. Berikut tampilan kontrol *trackbar* untuk filter objek.



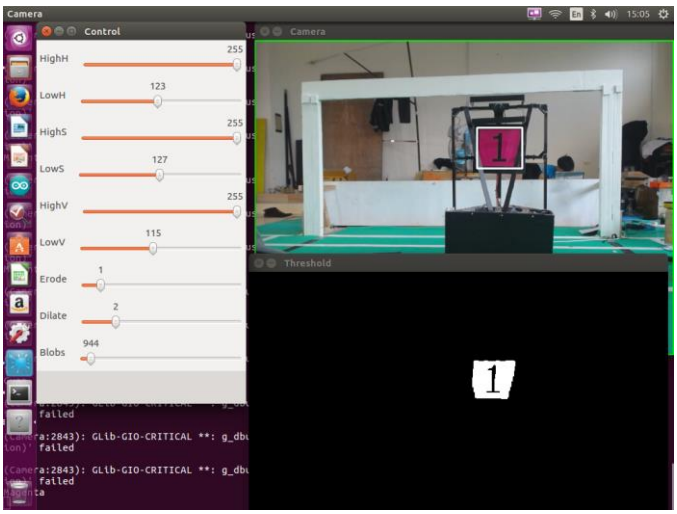
Gambar 5 Trackbar untuk Filter Objek

Pada sistem ini, objek yang telah difilter dan terdeteksi akan berwarna putih sedangkan objek lain disekitarnya berwarna hitam. Warna hitam dan putih adalah hasil dari proses binerisasi. Saat pertama merunningkan program, tampilan *trackbar* yang muncul akan sama seperti gambar 4 dan tampilan *window threshold* akan berwarna hitam. Untuk memudahkan pemfilteran objek, tampilan *window threshold* harus diputihkan terlebih dahulu dengan membuat nilai *trackbar High* menjadi 255 dan *Low* bernilai 0. Setelah dilakukan filter terhadap warna *cyan* dan *magenta*, hasil yang didapatkan dapat dilihat pada Gambar 7 dan 8.



Gambar 6 Deteksi Warna Identitas Robot *Cyan*

Dari gambar diatas, kita dapat melihat hasil dari deteksi warna identitas robot *cyan*. Untuk deteksi warna ini, nilai *hue* berada pada rentang 0 – 27, saturation 195 – 255, dan *value* 175 – 247. Untuk filter warna *cyan* ini dibutuhkan proses erosi 1 kali, dilasi 3 kali, dan minimal *blobs* nya yaitu 944. Selanjutnya untuk filter warna *magenta* dapat dilihat pada Gambar 8.



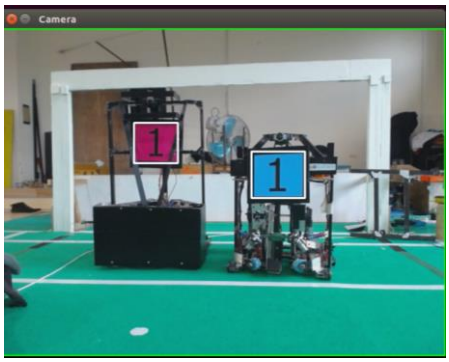
Gambar 7 Deteksi Warna Identitas Robot *Cyan*

Untuk filter warna *magenta*, nilai *hue* yag didapatkan berada pada rentang 122 – 255, saturation 127 – 255, dan value 115 – 255. Proses erosi yag dilakukan sebanyak 1 kali, dilasi 2 kali, dan minimal *blobs* nya yaitu 944. Berikut hasil pengujian sistem deteksi warna identitas robot

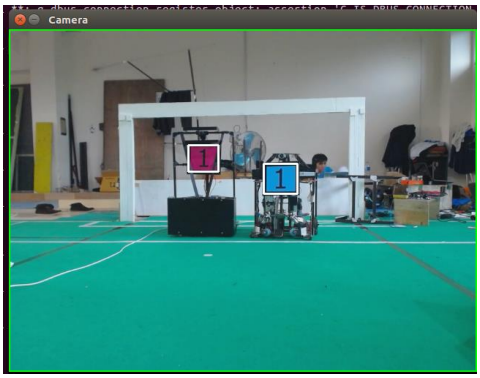
TABLE I. HASIL PENGUJIAN SISTEM DETEKSI WARNA IDENTITAS ROBOT *CYAN* DAN *MAGENTA*

Pengujian Tingkat Keberhasilan Deteksi *Cyan* dan *Magenta*			
Jarak (m)	***Warna Aktual***	***Deteksi***	***Status***
0.05	*Cyan*	Terdeteksi	Berhasil
	Magenta	Terdeteksi	Berhasil
0.8	*Cyan*	Terdeteksi	Berhasil
	Magenta	Terdeteksi	Berhasil
1.5	*Cyan*	Terdeteksi	Berhasil
	Magenta	Terdeteksi	Berhasil
2.3	*Cyan*	Terdeteksi	Berhasil
	Magenta	Terdeteksi	Berhasil
3	*Cyan*	Terdeteksi	Berhasil
	Magenta	Terdeteksi	Berhasil
3.8	*Cyan*	Terdeteksi	Berhasil
	Magenta	Terdeteksi	Berhasil
4.5	*Cyan*	Terdeteksi	Berhasil
	Magenta	Terdeteksi	Berhasil
5.3	*Cyan*	Terdeteksi	Berhasil
	Magenta	Terdeteksi	Berhasil
6	*Cyan*	Terdeteksi	Berhasil
	Magenta	Terdeteksi	Berhasil
6.8	*Cyan*	Terdeteksi	Berhasil
	Magenta	Terdeteksi	Berhasil
7.5	*Cyan*	Tidak Terdeteksi	Gagal
	Magenta	Tidak Terdeteksi	Gagal

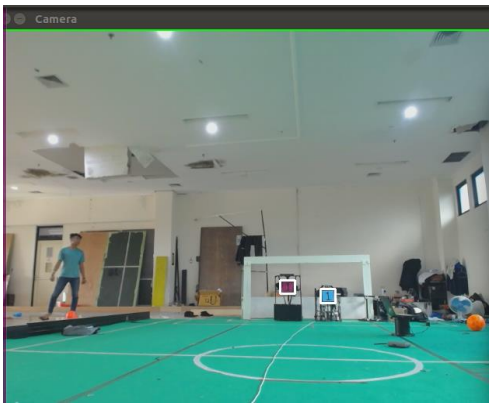
Berdasarkan TABLE I dapat dilihat bahwa persentase keberhasilan dari sistem pendeteksian warna identitas robot *cyan* dan *magenta* ini sebesar 90.9 % dari persentase keseluruhan yaitu sebesar 100%. Berikut contoh hasil deteksi kedua warna pada jarak-jarak tertentu.



Gambar 8 Deteksi Warna Identitas Robot *Cyan* dan *Magenta* pada Jarak 1.54 Meter



Gambar 9 Deteksi Warna Identitas Robot *Cyan* dan *Magenta* pada Jarak 4.52 Meter



Gambar 10 Deteksi Warna Identitas Robot *Cyan* dan *Magenta* pada Jarak 7.5 Meter

Berdasarkan gambar diatas dapat dilihat bahwa sistem pengenalan warna identitas robot dapat mengenali warna *cyan* dan *magenta* yang ada pada robot.

V. KESIMPULAN

Dari penelitian ini dapat diambil kesimpulan bahwa sistem deteksi warna identitas robot dengan model warna HSV cukup baik dengan persentase keberhasilan sebesar 90.9%. Proses deteksi pada sistem ini memiliki proses yang cepat dan *real time*. Robot dapat terdeteksi dari jarak terdekat yaitu 0.05 meter sampai jarak 7.5 meter. Saat jarak 7.5 meter, warna identitas robot tidak terdeteksi oleh kamera dikarenakan nilai kontrol *blobs* yang tinggi. Dengan mengurangi blob dapat membuat warna identitas robot terdeteksi kembali. Hasil yang telah didapatkan ini nantinya dapat digunakan untuk membuat sebuah algoritma pergerakan untuk menghindari robot lawan.

ACKNOWLEDGMENT

Kami berterima kasih kepada Tuhan Yang Maha Esa dan rekan-rekan kami dari Barelang Robotic Team Politeknik Negeri Batam yang telah memberikan dukungan dalam penelitian ini.

REFERENCES

[1] RISTEKDIKTI, "Panduan KRSBI beroda 2017," Jakarta Pusat, 2016.

[2] C.N. Viet, dan I. Marshall, "Vision-based obstacle avoidance for a small, low-cost robot," United Kingdom, 2007.

[3] C.P. Dewangga, "Sistem deteksi bola, gawang, dan kostum pada robot Barelang FC menggunakan machine vision," Buku Tugas Akhir Diploma IV Jurusan Mekatronika, Politeknik Negeri Batam, Batam, 2014.

[4] R. Rulaningtyas, A.B. Suksmono, T.L.R. Mengko, dan G.A.P. Saptawati, "Segmentasi citra berwarna dengan menggunakan metode clustering berbasis patch untuk identifikasi mycobacterium tuberculosis," Surabaya, 2015.

[5] G.S. Nahla, "*Tracking* bola menggunakan robotino," Buku Tugas Akhir Diploma IV Jurusan Teknik Informatika, Politeknik Elektronika Negeri Surabaya, Surabaya, 2010.

[6] A. Nana H, M. Ichwan, dan I.M.S. Putra, "Segmentasi citra untuk deteksi objek warna pada aplikasi pengambilan bentuk citra rectangle," Bandung, 2015.

Sistem Pendeteksian Bola Secara Real-Time Pada Robot Barelang FC

Gobar Bethara Agung Dewasakti*, Susanto
Teknik Mekatronika
Politeknik Negeri Batam
Batam, Indonesia
gobar.bethara@yahoo.com

***Abstrak*—Robot Barelang FC merupakan robot yang memiliki bentuk menyerupai manusia atau yang dikenal dengan nama robot humanoid. Robot ini dirancang untuk dapat bermain sepak bola layaknya manusia, seperti: mencari bola, mengejar bola, menggiring dan menendang bola. Sebagai pengganti indra penglihatan pada manusia, robot ini menggunakan kamera sebegai sensor untuk mendeteksi objek seperti bola. Untuk itu, robot dituntut untuk dapat mendeteksi berbagai macam jenis bola, dalam hal ini adalah bola berwarna putih. Permasalahan yang muncul adalah bagaimana robot dapat mendeteksi bola warna putih sebagai objek utama sedangkan sangat banyak objek lainnya dilapangan yang mengandung warna putih seperti garis lapangan dan warna gawang. Pada penelitian ini dikembangkan sistem deteksi bola berdasarkan warna dan bentuk dari bola. Hasil penelitian menunjukkan sistem filterisasi objek putih menggunakan ruang warna YUV mampu melihat objek putih dengan baik pada ukuran lapangan dengan panjang 6 meter dan lebar 4 meter dan pada sistem deteksi bola putih, dengan menggunakan perpaduan metode deteksi warna (cvblob) dan bentuk (hough circle transform) robot mampu mendeteksi bola putih pada jarak 0 sampai 360 cm.**

***Kata kunci* — robot *humanoid*, sistem deteksi bola, *cvblob*, *hough circle transform*.**

I. PENDAHULUAN

Dalam pertandingan sepak bola, robot *humanoid* bermain layaknya permainan sepak bola yaitu mampu menggiring dan menendang bola kearah gawang. robot ini menggunakan kamera sebegai sensor untuk mendeteksi objek seperti bola. Untuk dapat bermain sepak bola dengan baik, maka robot harus didukung dengan sistem pendeteksian bola yang baik juga.

Seiring dengan kemajuan teknologi, robot *soccer humanoid* dituntut untuk dapat mendeteksi berbagai macam jenis bola, salah satunya adalah bola berwarna putih [1]. Permasalahan yang muncul adalah bagaimana robot dapat mendeteksi bola warna putih sebagai objek utama sedangkan sangat banyak objek lainnya di lapangan yang mengandung warna putih seperti garis lapangan dan warna gawang. Permasalahan lainnya adalah pada sistem pendeteksian menggunakan kamera, warna putih merupakan warna yang rentan dipengaruhi oleh cahaya, sehingga semakin banyak derau (*noise*) warna yang menyerupai warna putih. Apabila robot tidak dapat membedakan hal tersebut, maka robot tidak akan dapat menentukan objek yang ingin dituju untuk keputusan selanjutnya.

Penelitian yang telah dilakukan untuk memecahkan permasalahan tersebut yaitu dengan menggunakan sistem deteksi bola berdasarkan warna bola dan *background* warna lapangan pada robot Barelang FC [2]. Kelemahan dari metode ini adalah sistem ini hanya mengandalkan sistem deteksi warna dari bola sehingga masih terdapat banyak kesalahan pendeteksian pada bola warna putih.

Untuk mengatasi cara deteksi bola yang tidak salah, maka pada penelitian ini akan dikembangkan sistem deteksi bola berdasarkan warna dan bentuk dari bola. Dengan demikian dapat mengurangi kesalahan deteksi dari sistem sehingga proses pendeteksian bola dapat berjalan dengan akurat dan *real-time.*

II. DASAR TEORI

A. OpenCV

OpenCV (*Intel Open Source Computer Vision Library*) adalah suatu *library* gratis yang dikembangkan oleh *developer Intel Corporation,* dapat digunakan dalam rangka komersil maupun non komersil, tanpa harus membayar lisensi ke *intel*. *Library* ini terdiri fungsi-fungsi *computer vision* untuk pengolahan citra (*Image Processing*) yang memungkinkan komputer dapat melihat seperti manusia. Dengan *vision* tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek [3].

B. Color Filtering

Color Filtering adalah suatu teknik pengolahan citra yang dipakai untuk memanipulasi suatu citra berdasarkan warna spesifik. Cara kerjanya adalah dengan membandingkan komponen warna setiap *pixel* citra dengan warna spesifik.

Warna yang digunakan dalam *color filtering* dapat direpresentasikan dalam berbagai ruang warna. Ada beberapa ruang warna yang dikenal, antara lain *RGB (Red, Green, Blue), HSV (Hue, Saturation, Value), CMY (Cyan, Magenta, Yellow), YCbCr*, *YUV,* dsb.

YUV merupakan ruang warna yang diturunkan dari ruang warna *RGB*. *YUV* sendiri terdiri dari komponen pencahayaan (*luminance*), dan dua komponen warna (*crominance*). Komponen pencahayaan dinyatakan dengan *Y*, sedangkan dua komponen warna dinyatakan dengan *U* dan *V*. Berikut merupakan persamaan transformasi ruang warna *RGB* menjadi ruang warna *YUV* [4]:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Gambar 1(a)

YUV memiliki komponen pencahayaan dan warna yang tidak saling mempengaruhi, sehingga pencahayaan dari gambar dapat diproses tanpa mempengaruhi komponen warna. Dalam hal ini *YUV* merupakan ruang warna yang cocok untuk mengidentifikasi warna-warna putih seperti bola, gawang, dan garis lapangan pertandingan [5,6,7].

C. *Binerisasi Citra*

Binerisasi citra adalah sebuah metode untuk mengubah suatu citra menjadi dua jenis intensitas yaitu 0 atau 255. Untuk melakukan proses ini digunakan ambang batas (*threshold*) yang nilainya dapat diatur sesuai keperluan. Persamaa 2.2 adalah persamaan untuk mengubah citra asli menjadi citra biner [8].

f(x,y);if f(x,y)>*Threshold then* f(x,y)=255 *else* 0

Gambar 1(b)

Jika nilai *threshold* yang dipasang adalah 150, maka semua nilai *pixel* yang lebih kecil dari nilai *threshold* akan diubah menjadi 0 (hitam), sedangkan nilai *pixel* yang lebih besar atau sama dengan nilai *threshold* akan diubah menjadi 255 (putih).

D. *Blob detection*

Blob detection adalah algoritma yang digunakan untuk mendeteksi kumpulan dari titik-titik *pixel* yang saling terhubung satu sama lain. Pada metode ini, citra harus diproses dengan metode binerisasi citra terlebih dahulu berdasarkan warna yang ingin dideteksi. Setelah melalui proses binerisasi citra, kumpulan dari nilai *pixel* yang bernilai 255 (putih) pada suatu area akan dideteksi sebagai objek.

E. *Hough Circle Detection*

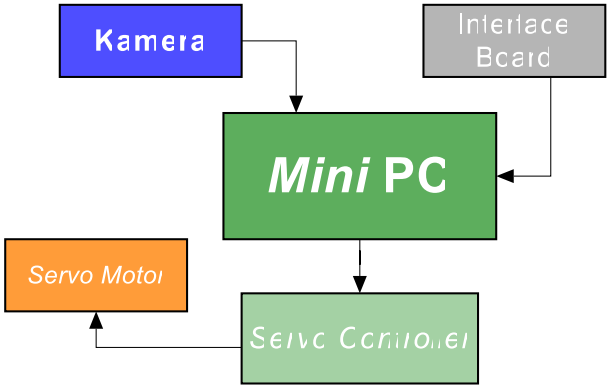
Pada *OpenCV,* sistem deteksi bentuk dari citra dikenal dengan istilah *hough transform*, paling umum digunakan untuk mendeteksi tepi objek seperti garis, lingkaran atau elips. Keuntungan dari metode ini adalah dapat mendeteksi tepian objek walaupun citra tersebut dipengaruhi oleh derau (*noise*). Dalam hal ini, untuk mengetahui bentuk dari suatu bola maka digunakan pendeteksian tepi berupa lingkaran, biasa disebut dengan metode *hough circle transform*.

III. PERANCANGAN SISTEM

Pada robot Barelang FC yang digunakan memiliki sistem kontruksi yang dibagi menjadi sistem elektrikal dan sistem deteksi.

A. *Sistem Elektrikal*

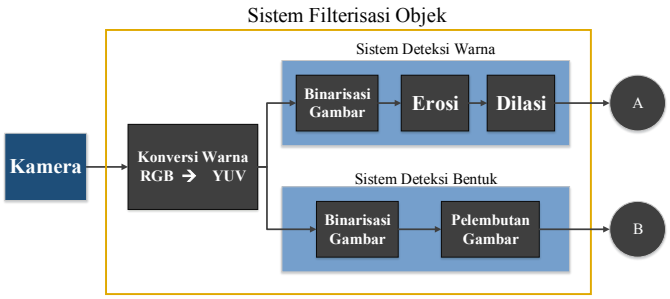
Komponen pengontrol robot ditunjukkan pada diagram blok dibawah ini:



Gambar 2 Blok Diagram Elektrikal Robot

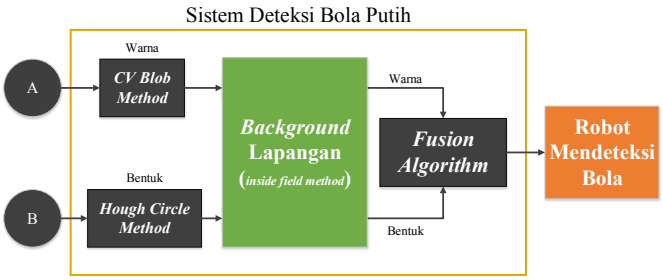
B. *Sistem Deteksi*

Sistem deteksi bola yang dirancang terbagi dalam dua proses besar yaitu sistem filterisasi objek (gambar 3) dan sistem deteksi bola putih (gambar 4).



Gambar 3 Diagram Blok Sistem Filterisasi Objek

Bola merupakan objek utama yang akan dideteksi oleh robot *soccer humanoid*. Pengambilan gambar dari kamera merupakan proses awal kemudian warna citra asli dirubah kedalam ruang warna *YUV,* yang merupakan ruang warna sesuai untuk mendeteksi objek putih. Citra hasil dari konversi ruang warna diteruskan pada proses binerisasi citra pada masing-masing sistem. Pada citra hasil binerisasi perlu dilakukan filterisasi *morphologi* yaitu erosi dan dilasi, dengan tujuan untuk menghilangkan *noise-noise* pada citra sehingga memperjelas objek yang dideteksi.



Gambar 4 Diagram Blok Sistem Deteksi Bola Putih

Setelah melalui proses filterisasi objek, pendeteksian bola warna putih dilakukan pengenalan menggunakan penggabungan dua metode, yaitu pendeteksian berdasarkan warna bola menggunakan *cvblob*, dan pendeteksian berdasarkan bentuk dari bola menggunakan *hough circle transform*.

IV. HASIL DAN ANALISA

Untuk mendapat hasil dan analisa maka perlu dilakukan beberapa pengujian. Adapun pengujian seperti dibawah ini.

A. Pengujian sistem filterisasi bola

Pada proses pengujian sistem filterisasi warna bola putih digunakan 3 buah bola dengan warna dominan oren, kuning, dan putih. Dilakukan pengambilan gambar citra asli (gambar 5.a), citra konversi *YUV* (gambar 5.b), dan citra binarisasi terhadap bola putih (gambar 5.c).



(a) (b)



(c)

Gambar 5 Bola Pengujian

Pengujian ini dilakukan dengan cara meletakan tiga bola warna berbeda secara acak, pada lapangan ukuran 6 x 4 meter. Berikut dibawah ini adalah hasil pengujian sistem filterisasi bola warna putih:

TABEL I Hasil Pengujian Sistem Filterisasi Bola Warna Putih

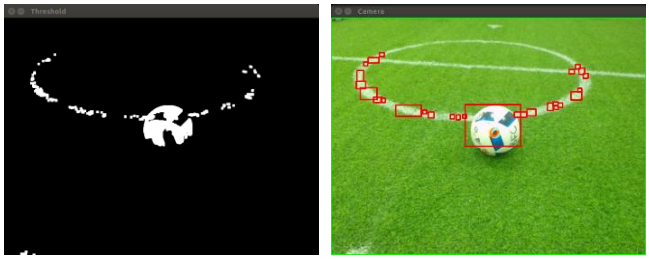
Pengujian	Warna Bola	Deteksi	Status
1	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil
	Kuning	Tidak Terdeteksi	Berhasil
2	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil
	Kuning	Tidak Terdeteksi	Berhasil
3	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil
	Kuning	Tidak Terdeteksi	Berhasil
4	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil
	Kuning	Tidak Terdeteksi	Berhasil
5	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil
	Kuning	Tidak Terdeteksi	Berhasil
6	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil
	Kuning	Tidak Terdeteksi	Berhasil
7	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil
	Kuning	Tidak Terdeteksi	Berhasil
8	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil
	Kuning	Tidak Terdeteksi	Berhasil
9	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil
	Kuning	Tidak Terdeteksi	Berhasil
10	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil
	Kuning	Tidak Terdeteksi	Berhasil
11	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil
	Kuning	Tidak Terdeteksi	Berhasil
12	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil
	Kuning	Tidak Terdeteksi	Berhasil
13	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil
	Kuning	Tidak Terdeteksi	Berhasil
14	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil
	Kuning	Tidak Terdeteksi	Berhasil
15	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil
	Kuning	Tidak Terdeteksi	Berhasil
16	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil
	Kuning	Tidak Terdeteksi	Berhasil
17	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil
	Kuning	Tidak Terdeteksi	Berhasil
18	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil

Pengujian	Warna Bola	Deteksi	Status
	Kuning	Tidak Terdeteksi	Berhasil
19	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil
	Kuning	Tidak Terdeteksi	Berhasil
20	Putih	Terdeteksi	Berhasil
	Oren	Tidak Terdeteksi	Berhasil
	Kuning	Tidak Terdeteksi	Berhasil

Berdasarkan Tabel 1 pengujian filterisasi bola warna putih terhadap bola warna lain adalah berhasil secara sempurna sehingga persentase keberhasilan pengujian sistem filterisasi bola warna putih menggunakan ruang warna *YUV* adalah 100%.

B. Pengujian sistem deteksi bola putih

Pada proses pengujian sistem deteksi bola putih dilakukan dua jenis pengujian. Metode pertama adalah pendeteksian berdasarkan warna bola menggunakan *cvblob*. Pada metode ini seluruh area putih hasil dari filterisasi objek warna putih (binarisasi) akan dideteksi, yang ditunjukkan pada gambar 6.



(a) (b)

Gambar 6 Hasil Deteksi Bola Putih Menggunakan *cvblob*

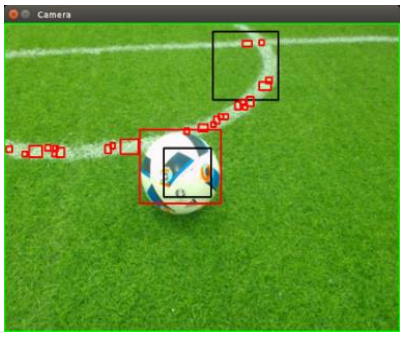
Deteksi bola putih berdasarkan metode dua adalah pendeteksian bentuk dari bola menggunakan *hough circle transform*. Hasil deteksi ditunjukkan pada gambar 7, kelemahan metode ini terletak pada garis lingkaran di tengah lapangan yang terdeteksi karena memiliki ciri lingkaran.



(a) (b)

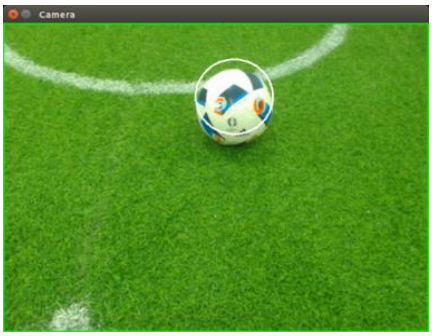
Gambar 7 Hasil Deteksi Bola Putih Menggunakan *Hough Circle Transform*

Berdasarkan kedua metode diatas, ditampilkan hasil dari deteksi warna menggunakan *cvblob* (metode 1) dan deteksi bentuk bola menggunakan *hough circle transform* (metode 2) yang ditunjukkan pada gambar 8.



Gambar 8 Deteksi Bola Putih Menggunakan *CV Blob* dan *Hough Circle Transform*

Setelah dilakukan perpaduan deteksi dari metode satu dan metode dua menghasilkan objek yang terdeteksi pada suatu *region of interest (ROI)*. Hasil dari kombinasi kedua metode ini dilakukan *scanning* terhadap ciri bola yang kemudian dianggap sebagai bola putih (gambar 9).



Gambar 9 Hasil Kombinasi Deteksi Bola Putih Menggunakan *CV Blob* & *Hough Circle Transform*

Pengujian dari perpaduan deteksi bola putih ini dilakukan dengan cara meletakan bola pada jarak 0 – 6 meter dengan berbagai posisi seperti pada saat pertandingan.



Gambar 10 Pengambilan Data Deteksi Bola Putih

Berikut dibawah ini adalah hasil rata-rata dari pengujian sistem deteksi bola putih:

TABEL II Hasil Pengujian Sistem Secara Keseluruhan

Jarak Pengujian (cm)	Status
0 - 360	Terdeteksi
380 - 600	Tidak Terdeteksi

Berdasarkan tabel 2 dapat dilihat bahwa sistem mampu mendeteksi bola putih pada jarak 0 – 360 cm.

V. KESIMPULAN

Berdasarkan hasil dan analisa sistem pendeteksian bola secara *real-time* pada robot Barelang FC dapat disimpulkan bahwa sistem filterisasi objek putih menggunakan ruang warna *YUV* mampu melihat objek putih dengan baik pada ukuran lapangan 6 x 4 meter. Pada sistem deteksi bola putih, dengan menggunakan perpaduan metode deteksi warna (*cvblob*) dan bentuk (*hough circle transform*) robot mampu mendeteksi bola putih pada jarak 0 – 360 cm.

DAFTAR PUSTAKA

[1] Reinhard Gerndt, Daniel Seifert, Jacky Hansjoerg Baltes, Soroush Sadeghnejad, and Sven Behnke, “Humanoid Robots in Soccer,” IEEE Robotics & Automation Magazine. Vol. 22, pp. 147-154, September 2015.

[2] Susanto, Hendawan Soebhakti, Eko Rudiawan, Barelang FC Team Description Paper for Humanoid Kidsize League of Robocup 2017.

[3] Kurniawan Dwi Irianto, “Pendeteksi Gerak Berbasis Kamera Menggunakan OpenCV Pada Ruangan,” KomuniTi. Indonesia, Vol. 2, No. 1, pp. 52-59, Juni 2010.

[4] Roshan Jameer Patan, Madhuparna Chakraborty and Sarungbam Sanahanbi Devi, “Blind Color Image De-Convolution in YUV Space,” Proceedings of the 2016 IEEE/RSJ, Intl. Conference on Communication and Signal Processing, India, 2016.

[5] Daniel Seifert, Lutz Freitag, Jan Draegert, FUmanoids Team Description Paper for Robocup 2015.

[6] Yasuo Hayashibara, Hideaki Minakata, Kiyoshi Irie, CIT Brains Team Description Paper for Robocup 2016.

[7] R. Fabre, H. Gimbert, L. Gondry, Rhoban Football Club Team Description Paper for Humanoid Kidsize League of Robocup 2016.

[8] Calvinsan Pratama Dewangga, “Sistem Deteksi Bola, Gawang, Dan Kostum Pada Robot *Barelang FC* Menggunakan *Machine Vision*,” Buku Tugas Akhir Diploma IV, Jurusan Mekatronika, Politeknik Negeri Batam, 2014.

Rancang Bangun Omni Vision pada Robot ENSPARTAN KRSBI Beroda

Esa Ilham Akbar*, Muhram Muis, Leonard D. Prakoso
Universitas Brawijaya
Laboratorium Mekatronika dan *Robot*ika Teknik Elektro
Malang, Indonesia
esaakbar.46@gmail.com

***Abstrak* - Makalah ini menjelaskan tentang konsep rancang bangun omnivision yang diterapkan pada robot KRSBI Beroda sesuai peraturan yang diterapkan oleh Dikti. Pada tahun kedua penyelenggaraan kompetisi ini kami berfokus untuk mengoptimalkan penggunaan omnivision sebagai lokalisasi objek berdasarkan hasil pengujian didapatkan jarak terjauh untuk deteksi bola sekitar 3m dengan cakupan sudut dari 0 derajat sampai 360 derajat.**

Katakunci- omnivision, KRSBI Beroda, tracking

I. PENDAHULUAN

ENSPARTAN adalah tim robot sepakbola beroda dari jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya Malang. ENSPARTAN mulai bergabung tahun 2016 pada Eksebisi Sepakbola Robot Beroda yang diselenggarakan oleh Kementerian Riset Teknologi dan Pendidikan Tinggi. Tim kami terdiri dari 2 robot yang bertugas sebagai penyerang dan penjaga gawang berdasarkan peraturan dari ESRB 2016 yang mengharuskan satu tim terdiri dari 2 robot. Untuk kompetisi tahun ini kami memaksimalkan penggunaan omnivision sebagai penentu keputusan robot.

Rancangan sistem *omnivision* pada tim Robot Universitas Brawijaya dibagi menjadi 3 bagian, yaitu secara mekanik, elektrik, dan *software*. Pada bagian mekanik merupakan bagian yang menompang kamera dan kaca sedangkan pada bagian *software* pengolahan citra yang dilakukan untuk mengolah data yang didapat dari kaca. Pada bagian elektrik merupakan pemilahan jenis kamera yang digunakan.

II. TINJAUAN PUSTAKA

A. Deskripsi Umum

Untuk penglihatan omnidirectional, susunan yang biasa dilakukan adalah memiliki satu kamera menuju ke arah cermin yang mencerminkan pandangan 360 derajat benda disekitar robot. Pilihan populer untuk bentuk cermin adalah bagian kerucut, seperti bola, paraboloid, *hyperboloids*, atau cermin berbentuk bulat panjang dan ini untuk alasan yang baik.

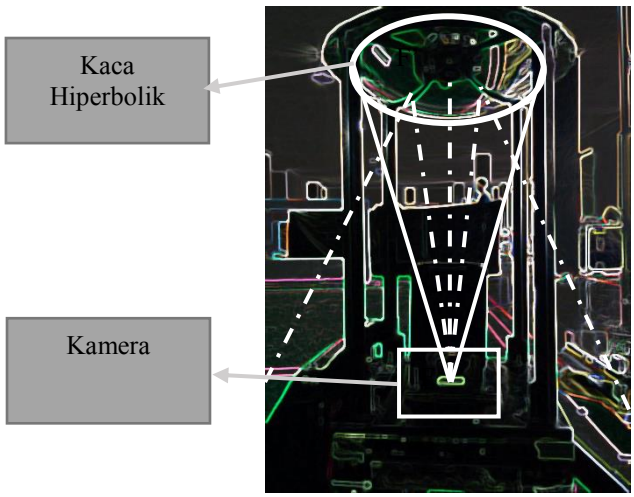
Hiperboloid memiliki kelebihan dibanding jenis lainnya bentuk. Gambar 1 menunjukkan hiperbola pada robot. Jika kita memikirkan bagian atas kaki hiperbola sebagai cermin cembung, lalu ada sinar yang keluar dari luar dan diarahkan langsung ke fokus, akan tercermin ke fokus lainnya. Kamera bisa diposisikan persis disana. Oleh karena itu, semua sinar tercakup masuk akan menjadi berkonsentrasi sempurna dalam tujuan kamera dan kamera akan memiliki tercermin gambar fokus sempurna.



Gambar 1 Robot dengan *Hiperbolic OmniVision*

B. Desain OmniVision

Desain rancangan untuk *OmniVision* robot penyerang dan keeper menggunakan 1 buah *hyperbolic mirror* yang dipasang secara tegak lurus[1][2]. Hasil gambar yang didapat mampu menghasilkan cakupan area 360°. Ilustrasi pantulan cahaya sesuai gambar 2.



Gambar 2 Titik pantul *hyperbolic mirror* dan *elliptic mirror*

C. Logitech C525

Webcam atau kamera *web* adalah sebuah kamera *video* digital kecil yang dihubungkan ke komputer melalui *port USB* ataupun *port COM* [3]. Kamera *web* juga dapat dipergunakan

untuk komunikasi *visual* sehingga sesama pengguna dapat saling bertatap muka melalui internet. Ada beberapa metode penyiaran, metode paling umum adalah perangkat keras mengubah gambar ke dalam bentuk format JPG dan mengunggahnya ke *web server* menggunakan *File Transfer Protocol* (FTP).



Gambar 3 Logitech *Webcam* C525

Spesifikasi adalah sebagai berikut :

- *Video calling*: 1280 x 720 *pixels*,
- *Video capture*: sampai 1280 x 720 *pixels*,
- Foto: sampai 8 *megapixels*,
- USB 2.0 kecepatan tinggi,
- Dimensi: 20,95 x 7,62 x 15,24 cm
- *Autofocus*

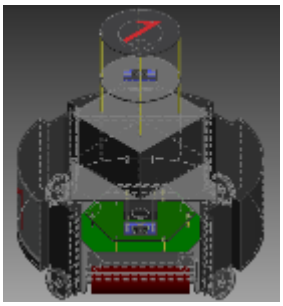
Frame rate mengindikasikan jumlah gambar sebuah perangkat lunak dapat ambil dan kirim dalam satu detik. Untuk *streaming video* membutuhkan minimal 15 fps atau idealnya 30 fps. Untuk mendapatkan *frame rate* yang tinggi, dibutuhkan koneksi internet yang tinggi kecepatannya.

III. METODE PENELITIAN

A. *Hardware*

OmniVision didesain dengan karakteristik yang disesuai kan dengan desain robot pada gambar 4 sebagai berikut.

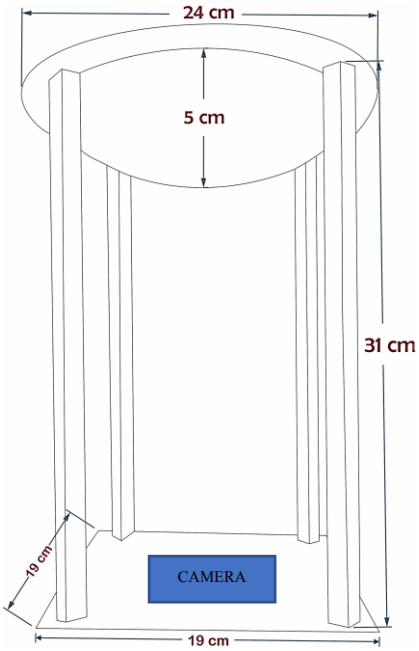
- Tinggi chasis : 77 cm
- Luas *robot* : 49 x 47 cm



Gambar 4 Desain Robot

Sehingga batas tinggi robot yang tersisa untuk robot berkisar +- 3 cm. Dengan menempatakan pada tinggi 73 cm dari *omnivision* didapatkan jarak pandang sejauh 3 m dan radius pengelihatan 360 derajat. Area jangkauan ini yang menjadi keunggulan *omnivision* dalam implementasi deteksi objek secara *autonomous*. *Omnivision* beserta kamera diletakan pada ketinggian 42 cm sesuai pada gambar 5. Bentuk sesuai dengan spesifikasi berikut:

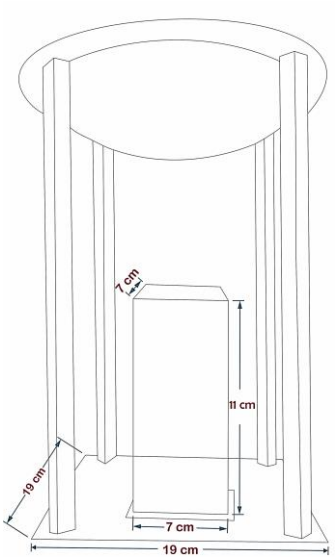
- Tinggi Penyangga *OmniVision* : 31 cm
- Lebar Alas Penyangga *OmniVision* : 19 cm
- Panjang Alas Penyangga *OmniVision* : 19 cm
- Tinggi *Convex Hull Mirror* : 5 cm



Gambar 5 Desain *OmniVision*

Pantulan cahaya dari *body omnivision* sangat mengganggu dari performansi robot. Banyaknya *noise* cahaya akibat pantulan dari *mirror* sehingga pengambilan keputusan yang kadang tidak sesuai dengan *real-time* nya. Dengan menambahkan penangkal cahaya berupa kotak yang dilapisi lapisan yang mampu menangkal cahaya sehingga deteksi objek lebih *efisien*. Desain kotak sesuai pada gambar 6 dengan spesifikasi berikut.

- Panjang : 7 cm
- Lebar : 7 cm
- Tinggi : 11 cm



Gambar 6 Desain *Anti-lightning Box*

B. *Software*

Pada bagian *programming* untuk robot ENSPARTAN dititik beratkan kepada kemampuan robot mendeteksi posisi bola,mengejar bola,serta algoritma deteksi gawang lawan dan menendang bola. Untuk mendapatkan target yang diinginkan perlu di lakukan pengolahan citra pada objek yang tertangkap pada *OmniVision*. Pengolahan citra dilakukan dengan *masking* pada area-area yang tidak diinginkan sehingga mengurangi *noise* yang disekitar. Secara diagram blok digambarkan seperti pada gambar 7.



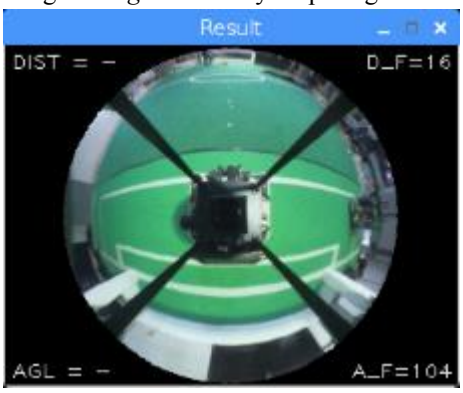
Gambar 7 *Flowchart masking area*

Tahapan yang diperlukan dengan mendapatkan tiap *frame* yang dirubah nilai *pixel* menjadi 320x240 kemudian di-*masking* dengan *frame* yang memiliki lingkaran. Setelah melakukan *masking* hasil dari *frame* berupa lingkaran yang terlihat sebagai *frame* baru yang selanjutnya akan diolah sebagai data deteksi.

IV. HASIL DAN PEMABAHASAN

A. *Deteksi Bola terhadap Jarak*

Berdasarkan hasil *flowchart* didapatkan sebuah *image* yang akan diolah sebagai acuan baru untuk mendeteksi objek. Hasil berupa *frame real-time* dengan latar lingkaran berwarna hitam ditumpuk dengan *image* sebenarnya seperti gambar 8.



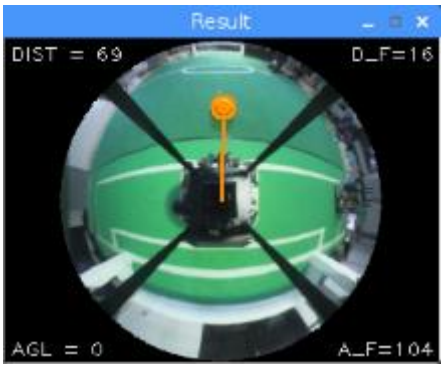
Gambar 8 *Output Merge Image*

Hasil pengujian jarak jangkauan deteksi objek dapat dilihat pada TABEL I. Berdasarkan tabel dapat diketahui bahwa jarak yang mampu dijangkau untuk mendeteksi objek adalah paling dekat = $\pm 10\ cm$ dan terjauh = $\pm 300\ cm$.

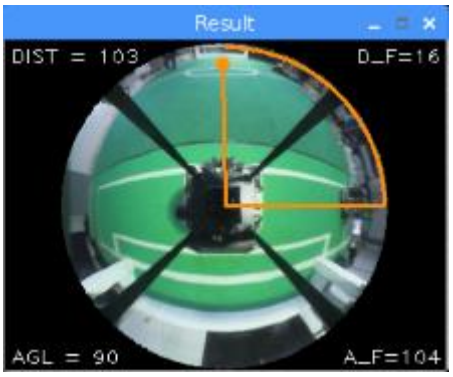
TABEL I Pengujian Jarak Jangkauan Deteksi Objek

Jarak Deteksi	
Minimal	Maksimal
$\pm 10\ cm$	$\pm 300\ cm$

. Tampilan hasil deteksi objek pada jarak minimal dan maksimal dapat dilihat dalam gambar 9 dan gambar 10.



Gambar 9 Deteksi jarak terdekat



Gambar 10 Deteksi jarak terjauh

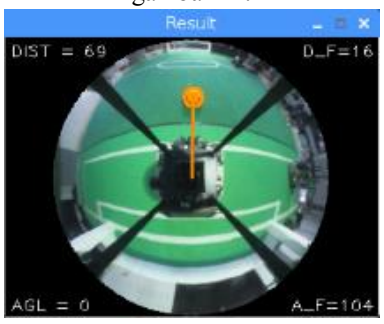
B. Deteksi Bola terhadap Sudut Visual

Dengan menggunakan *omnivision* cakupan objek dapat terdeteksi pada sudut 360 menjadikan robot lebih mudah dalam mendeteksi objek secara cepat. Cakupan area yang dapat terdeteksi sesuai dengan TABEL 2.

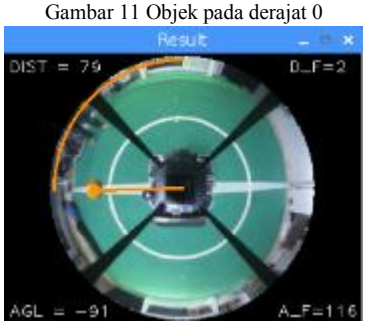
TABEL II Pengujian cakupan Deteksi Objek

Cakupan Area	
Sudut	Kondisi
0	Terdeteksi
90	Terdeteksi
180	Terdeteksi
270	Terdeteksi

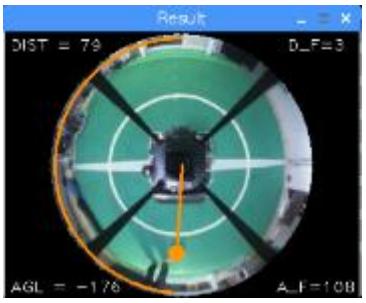
.Tampilan hasil deteksi objek pada masing-masing sudut dapat dilihat dalam gambar 11, gambar 12, gambar 13, dan gambar 14.



Gambar 11 Objek pada derajat 0



Gambar 12 Objek pada derajat 90



Gambar 13 Objek pada derajat 180



Gambar 14 Objek pada derajat 270

Berdasarkan data hasil dapat disimpulkan untuk efektivitas dalam deteksi sangat tinggi. Dapat dilihat cakupan area yang mampu ditangkap pada desain *omnivision* sebesar $0° - 360°$ sehingga robot lebih besar kemungkinan untuk deteksi.

V. KESIMPULAN

Berdasarkan hasil percobaan dapat disimpulkan sebagai berikut.

- Cakupan area untuk mendeteksi objek luas, dimulai dari 0 – 360 derajat.
- *Robot* mampu menghitung jarak dari posisi *robot* terhadap objek oleh omnivision dengan jarak pandang maksimal 3 m.

REFERENSI

[1] Asama, H., Sato, M., Bogoni, L., Kaetsu, H., Matsumoto, A., Endo, I. (1995) Development of an omni directional mobile *robot* with 3 DOF decoupling drive mechanism, Proc. IEEE Int. Conf. on *Robot*ics and Automation, pp1925-1930

[2] Yagi , Y. (1999) . Omni directional Sensing And its Applications , IEICE TRANS ,INF. & SYST . Vol. E82- D(No.3), pp. 568-579.

[3] Logitech 2011. "Logitech C525 specification".(Online).*(https://www.arp.ch/webmedias/datasheet/548708f0154fe21ea5cf720b.pdf).* Diakses Tanggal 20 Mei 2017.

Pergerakan Otomatis Robot Sepak Bola Beroda Melalui Komunikasi dengan Referee Box Menggunakan Base Station

Thoha Ardhiansyah, Ihsan Syarifuddin, Muhammad Rafii Naufal, Yuli Pramono, Ovi Tri Hartatik, Ilmawan Mustaqim
UNY *Robotic Team* (Tim Robot UNY)
Yogyakarta State University (Universitas Negeri Yogyakarta)
Yogyakarta, Indonesia
thoha.ardhiansyah@student.uny.ac.id

***Abstrak*—The newest division in Indonesian Robot Contest, Indonesian Wheeled Soccer Robot Contest, an interested division to follow because they have many newest technologies and innovations that included in the robot, the wheeled soccer robots. One of the newest innovation is the game controller named RefBox (Referee Box). This innovation helps "Law of the Game" in giving a decision or commands to the robots who played the game, it will be Kick Off, Drop Ball, Penalty, etc. The purpose of this paper is to discuss how data communication works from the RefBox to the robots, so the robots can move autonomously according to the commands from "Law of the Game".**

Katakunci—robot; soccer; wheeled; autonomous; communication; referee; box; base; station.

I. PENDAHULUAN

Kontes Robot Indonesia (KRI) adalah sebuah kontes robot bagi mahasiswa yang dilaksanakan rutin setiap tahun sekali. KRI diselenggarakan oleh Kementerian Riset, Teknologi dan Pendidikan Tinggi yang bertujuan untuk meningkatkan minat, bakat dan kemampuan mahasiswa pada pengembangan teknologi terkini khususnya di bidang robotika. Salah satu divisi terbaru dalam KRI tahun ini yakni Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda. KRSBI Beroda menggunakan banyak teknologi yang digunakan dalam menjalankan pertandingan [1].

RefBox (*Referee Box*) adalah salah satu teknologi yang digunakan dalam mengatur jalannya pertandingan dalam KRSBI. *RefBox* yang digunakan ini menganut dari peraturan *RoboCup* MSL. Fungsi *RefBox* yakni untuk membantu wasit, terutama dalam menyampaikan keputusan wasit kepada robot yang sedang bermain, sebagai penampil waktu pertandingan, dan juga untuk mencatat skor pertandingan. Secara khusus, setiap kali juri memerintahkan wasit untuk memberi sebuah isyarat, protokol komunikasi *RefBox* mengirimkan satu atau beberapa pesan ke komputer *base station* kedua tim (*magenta* dan *cyan*). Pengoperasian *RefBox* diserahkan kepada asisten wasit [2].

Panitia KRSBI menggunakan *RefBox* yang sudah disiapkan untuk mengatur jalannya pertandingan. *RefBox* merupakan *game controller* yang digunakan wasit untuk mengatur pertandingan, seperti: memulai sebuah pertandingan, menampilkan waktu pertandingan, menampilkan skor pertandingan, menampilkan beberapa tombol serta indikator perintah wasit. Indikator perintah wasit berupa perintah memperbolehkan mengangkut robot agar mendapatkan perbaikan atau biasa disebut dengan *pick up*, pelanggaran, dan penalti. *RefBox* menggunakan jaringan komunikasi yang terhubung pada komputer *base station* dari masing-masing tim robot yang melakukan pertandingan (tim *magenta* dan tim *cyan*) [3].

RefBox mengirimkan data ke komputer *base station* terlebih dahulu, selanjutnya robot akan menerjemahkan oleh robot berupa pergerakan otomatis di lapangan, sebagai perintah dari wasit. Paket data untuk mulai *kick off* akan berbeda dengan paket data perintah *pick up*, sehingga robot harus bisa membedakannya. Pada penelitian ini yang menjadi pokok pembahasan adalah bagaimana alur robot dapat menerima perintah dan membedakan perintah dari *RefBox* hingga ke robot itu sendiri.

II. DASAR TEORI

A. Autonomous Robot

Autonomous Robot adalah robot yang dapat melaksanakan pekerjaan atau tugas tingkat tinggi, pergerakannya secara otomatis tanpa bantuan manusia, seperti dalam bidang eksplorasi ruang angkasa, pemeliharaan rumah tangga (contohnya : menyapu, mengelola air limbah, mengantar barang, dan memberikan jasa lainnya).

Sebuah robot bisa dikatakan sepenuhnya *autonomous* apabila dapat :

1. Mendapatkan informasi tentang lingkungan sekitarnya.
2. Bekerja untuk jangka waktu lama tanpa campur tangan manusia.
3. Memindahkan seluruh atau sebagian dari dirinya sendiri untuk beroperasi di sebuah lingkungan tanpa bantuan manusia.
4. Menghindari situasi yang berbahaya bagi manusia, properti lain, atau dirinya sendiri [4].

B. *Robot Soccer MSL*

KRSBI Beroda adalah divisi terbaru pada KRSBI yang merupakan pengembangan dari KRPAI Beroda. Robot beroda ini dapat bermain sepak bola dan bergerak secara *autonomous* mencari bola dan gawang sendiri tanpa bantuan kendali manusia. Aturan kompetisi KRSBI Beroda mengacu pada *RoboCup Soccer Middle Size League* (MSL). Ukuran panjang dan lebar robot antara 30 hingga 52 cm, tinggi robot antara 40 hingga 80 cm. Robot bermain di lapangan pertandingan berukuran 6 m x 9 m. Bola yang digunakan merupakan bola futsal berwarna *orange* (Ukuran standar FIFA no. 4, ukuran lingkar bola 62-64 cm, berat 390-430 gram). Sensor yang digunakan harus berada pada robot. Tidak ada sensor yang diletakkan di luar robot (tidak ada kamera di atas lapangan) [5].



Gambar 1 *Robot Striker* KRSBI Beroda Tim MOBO-EVO UNY
Sumber : Dokumentasi Pribadi

C. *RefBox*

RefBox atau *Referee Box* adalah sebuah perangkat lunak untuk mengatur dan mengendalikan pertandingan. *RefBox* dioperasikan oleh wasit. *RefBox* mengirimkan instruksi ke *Base Station*, untuk diteruskan ke Robot. Instruksi mencakup memulai pertandingan *START*, menghentikan pertandingan *STOP*, instruksi *Kick Off, Free Kick, Drop Ball*, dan status pertandingan lainnya. Robot mendengarkan instruksi dari *RefBox* melalui *Base Station*. Robot beroperasi secara otomatis berdasarkan instruksi yang diterima dari *RefBox* [6]. Format instruksi yang dikirim dari *RefBox* ke *Base Station* dalam bentuk satu buah karakter (*char*), sudah sangat rinci berada pada mailing list dari panitia KRSBI Beroda.



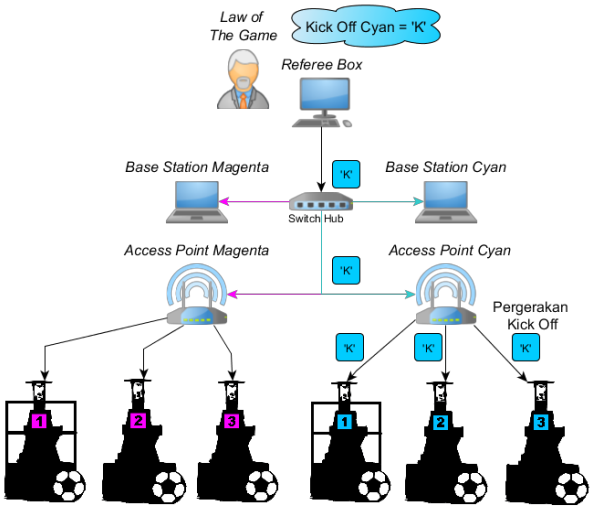
Gambar 2 Tampilan GUI Resmi *RefBox* Panitia KRSBI [6].

D. *Base Station*

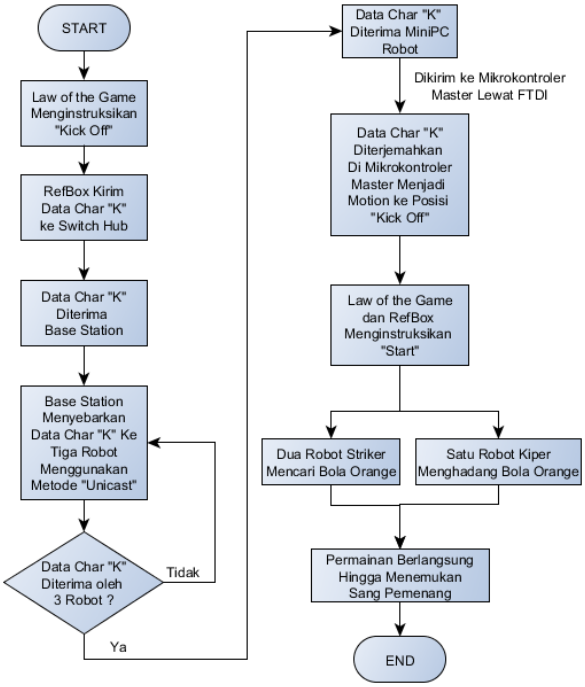
Base Station adalah komputer, *notebook*, atau laptop yang menjalankan *software* yang digunakan untuk memantau dan mengatur kerja robot secara otomatis berdasarkan instruksi yang diterima dari *RefBox*. *Base Station* melakukan koordinasi aktivitas setiap robotnya dalam pertandingan. Selama pertandingan sedang berjalan, anggota tim peserta tidak diizinkan untuk mengoperasikan *Base Station*-nya. *Base Station* menggunakan kabel *Ethernet Switch* untuk melakukan koneksi dengan *RefBox*. *Base Station* memanggil alamat IP (*IP Address*) *RefBox* untuk melakukan koneksi melalui *Port* 28097 (*default*). Alamat IP Komputer yang menjalankan *RefBox* akan diumumkan pada saat pertandingan. Cara koneksi antara *Base Station* dengan robot yakni setiap robot memiliki alamat IP (*IP address*). *Base Station* dapat terhubung dengan robot secara *wireless* melalui *access point* yang khusus disediakan panitia dalam pertandingan [6].

III. ALUR KOMUNIKASI

Berdasar ketentuan aturan main KRSBI Beroda [7], maka dari itu alur komunikasi data yang digunakan oleh tim MOBO-EVO UNY yaitu dengan metode *Unicast,* (kami contohkan salah satu perintahnya *Kick Off Cyan*), dan cara kerjanya sebagai berikut :



Gambar 3 Skema Alur Komunikasi Data Robot Tim MOBO-EVO UNY
Sumber : Dokumentasi Pribadi



Gambar 4 *Flowchart* Pengiriman Data dalam Robot
Sumber : Dokumentasi Pribadi

Komputer panitia KRI yang menjalankan *RefBox* mengirim perintah *Kick Off Cyan*, yang diwakilkan satu karakter huruf 'K', lalu data karakter 'K' ini diterima *Base Station,* kemudian karakter 'K' ini disebar dengan metode *Unicast* ke *Mini PC* Robot 1 & Robot 2. Dari *MiniPC*, karakter 'K' dikirim ke Sistem Minimum Mikrokontroler *Master* melalui *hardware* FTDI. Char 'K' kemudian diterjemahkan menjadi perintah gerak melalui program yang sudah ada di Sistem Minimum Mikrokontroler *Master*. Aktuator (Motor dan Roda Omni) berputar sesuai program, menjadikan kedua robot ini bergerak otomatis menuju mode posisi *Kick Off* pada tim sisi Cyan. Penjelasan ini menjelaskan skema dan *flowchart* yang ada di Gambar 3 dan 4.

Berdasarkan Gambar 4, alur komunikasi dari *RefBox* sampai ke Robot secara keseluruhan terdiri dari tiga bagian berikut:

1. Komunikasi *RefBox* dengan *Base Station,*
2. Komunikasi *Base Station* dengan *MiniPC* Robot,
3. Pengiriman data serial dari *MiniPC* Robot ke Sistem Minimum Mikrokontroler pada robot.

IV. PERANCANGAN SISTEM

RefBox yang telah dibuat dan digunakan selama latihan pertandingan merupakan sebuah GUI (*Graphical User Interface*) berbasis bahasa pemrograman C#. GUI *RefBox* ini dapat berjalan secara lancar dengan menggunakan sistem operasi Windows 10.



Gambar 5 Perancangan GUI *RefBox* dengan Ms. *Visual* Studio 2012 Menggunakan Bahasa C#
Sumber : Dokumentasi Pribadi

Seperti terlihat pada Gambar 5 di atas, perancangan sistem pada *RefBox ini* menggunakan perangkat lunak Microsoft *Visual* Studio 2012, untuk membuat aplikasi *Base Station* yang akan digunakan. Dalam sistem ini, protokol jaringan yang digunakan adalah protokol TCP/IP, nantinya data akan dikirim dari *RefBox* ke *Base Station,* dan dari *Base Station* ke *MiniPC* robot. Sedangkan untuk komunikasi dari *MiniPC* ke sistem minimum mikrokontroler robot menggunakan komunikasi berbasis serial menggunakan *hardware* FTDI.

V. IMPLEMENTASI

Tahapan ini dilakukan saat perancangan sistem sudah selesai dilaksanakan dan selanjutnya diimplementasikan pada bahasa pemrograman yang digunakan. Tujuan implementasi sistem adalah untuk menerapkan perancangan yang telah dilakukan terhadap perangkat lunak sehingga nantinya maksud dan tujuan penelitian dapat tercapai.

1. Proses penyambungan *Base Station* ke *RefBox*. Metode penyambungan koneksi dari *Base Station* dengan *RefBox* adalah dengan men-*setting Base Station* menjadi *client*. *Base Station* akan mencoba menyambungkan ke *RefBox* dengan ketentuan *IP Address* dan juga *port* khusus. *IP Address* yang digunakan adalah *IP Address RefBox*, sedangkan *port* yang digunakan adalah *port* 28097.
2. Proses penerimaan data dari *RefBox*. Metode yang digunakan dalam penerimaan data ini adalah dengan men-*setting* jenis data yang diterima menjadi jenis data yang sesuai. Ketidaksesuaian jenis data bisa mengakibatkan proses penerimaan data menjadi gagal. Berbagai jenis data yang digunakan dalam penelitian ini yaitu *char, string*, dan juga *byte*.
3. Proses pengiriman data dari *Base Station* ke *MiniPC Robot*. Pengiriman data dari *Base Station* ke *MiniPC* menggunakan protokol TCP/IP dengan bantuan *library "*NetworkApi.dll", dimana *Base Station* di-*setting* sebagai *Server*, sedangkan *MiniPC Robot* disetting sebagai *Client*. Untuk settingan dari *Server* dan juga *Client* digunakan *port* 6000, sedangkan untuk *IP Address Server* menggunakan *IP Address* yang terdapat pada laptop yang menjadi *Base Station*.
4. Proses pengiriman data dari *MiniPC Robot* ke sistem minimum mikrokontroler. Pengiriman data dari *MiniPC* ke sistem minimum mikrokontroler ini menggunakan metode pengiriman serial yang menggunakan *hardware* FTDI. Pengiriman data menggunakan fungsi "serial*Port*1.Write()", sedangkan untuk penerimaan data di sistem minimum mikrokontroler menggunakan fungsi "scanf()".

VI. PENGUJIAN

Proses pengujian dilakukan untuk memastikan kebenaran program yang telah diimplementasikan, sehingga pengguna dapat menggunakan program ini dengan baik sesuai fungsi program ini dibuat. Sesuai dengan tujuan dari penelitian ini yaitu menghasilkan sistem yang dapat mengkomunikasikan *RefBox* dengan *Robot* melalui *Base Station*, maka dalam penilitian ini *RefBox* yang digunakan adalah *RefBox* resmi dari *Robocup MSL*, untuk *Base Station* menggunakan laptop yang nantinya akan digunakan pada KRI 2017 Tingkat Nasional, dan untuk *MiniPC* menggunakan *MiniPC* yang sudah terpasang pada *robot*.

TABEL I Uji coba jenis data yang digunakan

Jenis data	Base Station	Keterangan
char	"Kosong"	Penerimaan Gagal
string	"Kosong"	Penerimaan Gagal
byte	75	Penerimaan Berhasil

Pada pengujian jenis data ini, hanya jenis data *byte* yang berhasil menerima perintah dari *RefBox*. Karena data yang diterima berupa *byte*, maka agar pengguna program dapat mengetahui data apa yang diterima, data *byte* yang diterima diterjemahkan ke dalam bahasa ASCII. Data "75" yang diterima setelah diterjemahkan ke dalam bahasa ASCII berhasil dirubah menjadi karakter "K", dimana karakter tersebut sesuai dengan data yang dikirim *RefBox* ketika *Kick Off Cyan*.

TABEL II Uji coba perintah dasar dari *RefBox*

Referee Box	***Base Station***	**Keterangan**
Kick Off Cyan ('K')	'K'	Penerimaan Berhasil
Dropped Ball ('N')	'N'	Penerimaan Berhasil
Start ('s')	's'	Penerimaan Berhasil
Stop ('S')	'S'	Penerimaan Berhasil

Pada pengujian ini, *RefBox* mencoba mengirimkan 4 perintah dasar yang biasa digunakan dalam KRI 2017 Regional 3, yaitu *Kick Off Cyan*, *Dropped Ball*, *Start*, dan juga *Stop*. Dari semua perintah dasar tersebut, semuanya dapat diterima oleh *Base Station* dengan baik.

TABEL III Uji coba kecepatan respon robot

Kondisi	**Perintah**	**Respon Sistem Minimum Mikrokontroler Robot**
Awal mulai pertandingan	*Kick Off Cyan* ('K')	Cepat (< 300 ms)
Disaat robot bergerak	*Stop* ('S')	Cepat (< 300 ms)
Sebelum menendang	*Stop* ('S')	Lumayan Lambat (> 700 ms)
Saat menendang	*Stop* ('S')	Lumayan Cepat (> 300 ms & < 700 ms)
Setelah menendang	*Stop* ('S')	Cepat (< 300 ms)

MiniPC berhasil menerima dari *Base Station* dengan sangat cepat. Dalam pengujian ini, *MiniPC* mencoba mendistribusikan data perintah dari *RefBox* ke sistem minimum mikrokontroler robot dalam beberapa kondisi. Dapat dilihat pada TABEL 3, terdapat tiga jenis respon yang terjadi, yaitu: Cepat, Lumayan Lambat, dan juga Lumayan Cepat. Respon ini menjadi bervariasi dikarenakan di dalam program sistem minimum mikrokontroler terdapat beberapa perulangan yang dapat mengakibatkan terkuncinya kondisi tertentu. Seperti pada kondisi sebelum menendang, program terkunci di perulangan mempersiapkan penendang.

VII. KESIMPULAN

Sesuai dengan hasil pengujian, dapat ditarik kesimpulan bahwa data yang dapat dikirim dari *RefBox* ke *Base Station* hanya berupa data *byte*, yang nantinya data *byte* tersebut mewakili satu karakter dalam bahasa ASCII, sehingga pada akhirnya akan dikirim ke *MiniPC robot* berupa satu karakter khusus dan diterjemahkan di sistem minimum mikrokontroler *robot*, dan berubah menjadi respon gerak otomatis *robot*.

DAFTAR PUSTAKA

[1] Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia. "*Tentang KRI*". (Online). (https://www.kontesrobotindonesia.org/tentang-kri/), 2017. Diakses pada tanggal 1 Juni 2017.

[2] RoboCup Federation. "*Middle Size Robot League - Rules and Regulations for 2017*". Halaman 23 (PDF). (http://www.robocup.org/system/sub_leagues/rules_papers/000/000/006/original/2017-msl-rules_v18_4.pdf?1486072480), 2016. Diakses pada tanggal 1 Juni 2017.

[3] F. Azwar Muhammad, R. Arif Windiarto, Y. Bintang Pratama. "*Pemrograman Socket Untuk Koneksi Antara Raspberry Pi Dengan Referee Box*". Surabaya : Politeknik Elektronika Negeri Surabaya - The 4th Indonesian Symposium on Robot Soccer Competition 2016, June 1th, 2016.

[4] N. Musandu Nyagudi. "*Humanitarian Algorithms : A Codified Key Safety Switch Protocol for Lethal Autonomy*". New York : arXiv.org - Cornell University Library, 2014.

[5] Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia. "*KRSBI - Kontes Robot Sepak Bola Indonesia Beroda*". (Online). (https://www.kontesrobotindonesia.org/kri-2017/), 2017. Diakses pada tanggal 1 Juni 2017.

[6] K. Mutijarsa. Mailing List - KRI Digest, Vol 31, Issue 1. "*Referee Box KRSBI Beroda 2017*". (https://groups.pens.ac.id/mailman/private/kri/attachments/20170430/00a379a5/attachment.htm), 2017.

[7] Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia. "*Aturan Main KRSBI 2017*". Halaman 12. (PDF). (http://kontesrobotindonesia.org/data-kontes/2017/PanduanKRSBIBeroda2017.pdf), 2017. Diakses pada tanggal 1 Juni 2017.

Disain Omnidirectional Vision pada ERSOW

Iwan Kurnianto Wibowo, Mochamad Mobed Bachtiar, Rizki Taufik Ulil A*,
Zanuar Tri R, M. Jauharul M, dan Muhamad Khoirul
Laboratorium Robotika
Politeknik Elektronika Negeri Surabaya
Surabaya, Indonesia
rizkitaufik@ce.student.pens.ac.id

***Abstract*—The Robocup Middle Size League competiton is a standard real-world test bed for autonomus multi-robot control, robot vision and other relative research subject. In the past decade, omnidirectional vision system has become one of the most important sensor for robot soccer in Robocup, for it can provide a 360º view of the robot's surrounding environment in a single image. Robot can use it for tracking object and self-localization which very important for robot control, strategy, and coordination. From the above explanation, in this research we create a system using omnidirectional vision for tracking ball and determine the positon of robot soccer.**

Keywords—omnidirectional; vision; tracking; middle size robot soccer

I. PENDAHULUAN

Pertandingan antara robot dan tim juara dunia dalam permainan sepakbola yang diperkirakan akan dilaksanakan pada tahun 2050 adalah cita- cita dari pengembang Robocup. Para peneliti robot soccer terus melakukan berbagai penelitian untuk meningkatkan kemampuan dari robot agar mampu melakukan permainan sepakbola seperti halnya manusia. Dalam kompetisi di Robocup, bidang robot soccer dibagi menjadi 2 divisi yaitu humanoid robot soccer dan mobile robot soccer. Divisi humanoid dikonsentrasikan dalam membangun sebuah mekanik robot pemain sepakbola layaknya seperti manusia. Sedangkan pada divisi mobile robot soccer lebih berkonsentrasi pada kecerdasan dan penerapan strategi pada robot soccer.

Pada pertandingan robot soccer setiap robot selalu mempunyai kemampuan untuk bermain sepak bola, tetapi tidak semua memiliki kecerdasan yang sama untuk menyelesaikan sebuah tugas dan membuat sebuah goal, karena banyak faktor yang mempengaruhi hal tersebut. Seperti dalam hal pergerakan robot yang kurang stabil, sistem penglihatan yang kurang bagus, mekanik yang tidak stabil atau bahkan system kecerdasan yang terlalu rumit sehingga menyebabkan lama dalam penyelesaian sebuah perintah. Aturan dan lingkungan yang digunakan sesuai dengan Peraturan Kontes Robot SepakBola Indonesia Beroda tahun 2017.

Dalam penelitian ini, difokuskan pada penggunaan omnidirectional vision pada robot soccer untuk mendeteksi objek dan juga untuk mengetahui posisi robot di lapangan. Omnidirectional vision sendiri terdiri dari sebuah kamera dan cermin yang digunakan untuk menangkap gambar yang berada di sekitar robot. Keunggulan dari omnidirectional vision adalah dapat menangkap gambar sekitar dengan sudut 360º hanya dalam satu frame. Jika menggunakan kamera biasa maka hanya dapat menangkap gambar hanya dalam satu sisi sehingga diperlukan tambahan langkah untuk mencari sebuah objek apabila objek berada di sisi yang berlainan.

II. DASAR TEORI

A. Penelitian yang Berkaitan

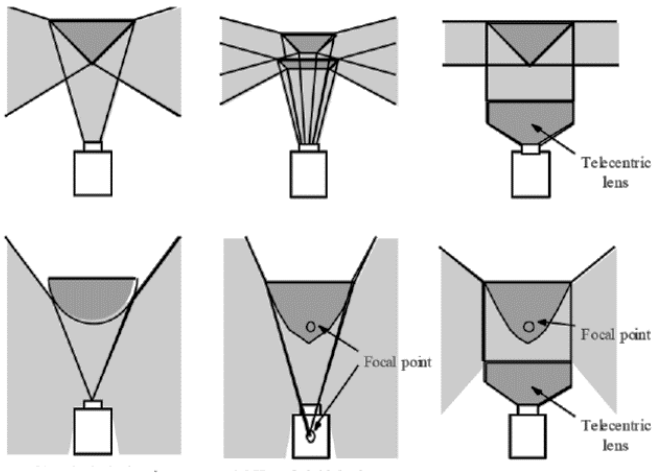
Antonio J.R. Neves (2010)[1]. Judul penelitian ini adalah "An efficient omnidirectional vision system for soccer robots : From calibration to object detection". Dalam penelitian ini diterapkan algoritma yang efisien untuk mendeteksi objek secara real time. Sistem vision digunakan untuk mendeteksi bola, halangan, dan garis dengan metode self-localization. Hasil yang diperoleh menunjukkan algoritma yang dibuat dapat meningkatkan efekifitas proses dan akurasi dalam mendeteksi objek secara real time.

Gil Lopes (2012)[2]. Judul penelitian ini adalah "Catadioptic system optimization for omnidirectional Robocup MSL robots". Penelitian ini membahas tentang cermin omnidirectional : cermin conic, spherical, hiperbola, dan parabola. Dijelaskan pula mengenai pencerminan yang dihasilkan oleh masing- masing cermin omnidirectional tersebut. Berdasarkan hasil penelitian menunjukkan bahwa agar kamera dapat menangkap objek dengan jelas dan akurat, maka perpaduan bentuk cermin digunakan, yaitu cermin berbentuk spherical yang dipadukan dengan cermin hiperbola.

B. Kamera Omnidirectional

Dalam fotografi, kamera omnidirectional adalah kamera dengan sudut pandang 360º pada bidangn horizontal, atau bidang visual yang mencangkup seluruh sphare. Kamera omnidirectional penting untuk mendapatkan cangkupan wilayah bidang visual yang luas, seperti dalam fotografi panorama dan robotika.

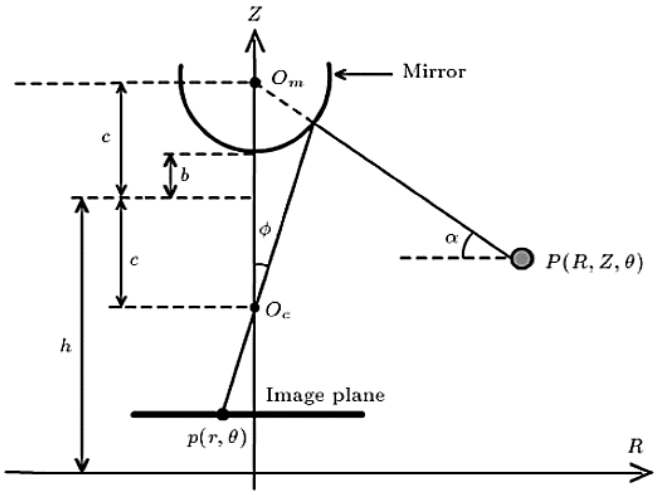
Cermin cembung/ cermin omnidirectional memiliki jenis yang berbeda- beda seperti kerucut (*canonical*), bola (*spherical*), silinder (*cylindrical*), parabola (*paraboloidal*), elips (*ellipsoidal*) dan hiperbola (*hyperboloidal*). Jenis cermin cembung pada umumnya digunakan pada kamera omnidirectional adalah bentuk bulat dan hiperbola. Namun pada beberapa kasus digunakan jenis cermin kombinasi antara cermin berbentuk kerucut, bola, dan datar. Pada Gambar 1 ditunjukkan beberapa jenis cermin yang dapat digunakan untuk omnidirectional vision.



Gambar 1. Jenis- jenis cermin omnidirectional [2].

C. *Permodelan Cermin Omnidirectional*

Bentuk dan karakteristik cermin omnidirectional mempengaruhi ukuran jarak sebenarnya dari obyek yang terdeteksi. Sehingga bentuk dan karakteristik cermin perlu dimodelkan terlebih dahulu agar hasil transformasi jarak lebih akurat. Gambar 2 menunjukkan persamaan dari cermin hyperbola [4].



Gambar 2. Model geometri cermin hyperbola [4].

$$\frac{R^2}{a^2} - \frac{(Z-h)^2}{b^2} = -1 \quad (1)$$

dimana :

$$a = \sqrt{c^2 - b^2}$$

Perhitungan untuk pemetaan cermin di atas adalah sebagai berikut:

$$Z = \mathrm{R}\tan\alpha + c + h \quad (2)$$

Semua obyek di dalam lapangan robot soccer menyetuh dengan lapangan, sehingga nilai Z=0, maka :

$$\tan\alpha = -\frac{c+h}{R} \quad (3)$$

Berdasarkan Gambar 2, persamaan dapat ditulis menjadi :

$$\tan\theta = \frac{b^2+c^2}{(b^2-c^2)}\tan\alpha + \frac{2bc}{c^2-b^2} \times \frac{1}{\cos\alpha} \quad (4)$$

$$\tan\alpha = \frac{f}{r} \quad (5)$$

Untuk menyederhanakan persamaan tersebut, maka :

$$a_1 = \frac{b^2+c^2}{b^2-c^2} \quad (6)$$

$$a_2 = \frac{2bc}{c^2-b^2} \quad (7)$$

$$a_3 = -(c+h) \quad (8)$$

Titik fokus cermin hiperbola terletak pada titik utama lensa kamera. Sehingga persamaan (4) dan (8) dapat disederhanakan menjadi :

$$\frac{f}{r} = a_1\tan\alpha + \frac{a_2}{\cos\alpha} \quad (9)$$

$$\frac{f^2}{r^2} - \frac{2a_1 f\tan\alpha}{r} + a_1(\tan\alpha)^2 = a_2{}^2(1+(\tan\alpha)^2) \quad (10)$$

Berdasarkan persamaan (3) dan (10) dapat ditulis menjadi:

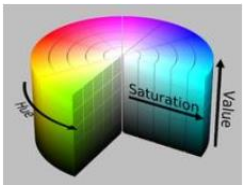
$$\frac{f^2}{r^2} - \frac{2a_1 f\frac{a_3}{R}}{r} + \frac{a_1{}^2 a_3{}^2}{R^2} = a_2{}^2\left(1+\frac{a_3{}^2}{R^2}\right), \quad (11)$$

$$(f^2 - r^2 a_2{}^2)R^2 - (2a_1 a_3 fr)R + r^2 a_3{}^2(a_1{}^2 - a_2{}^2) = 0 \quad (12)$$

D. *Segmentasi Warna HSV(Hue, Saturation, Value)*

Segmentasi warna merupakan proses segmentasi dengan pendekatan daerah yang bekerja dengan menganalisa nilai warna dari setiap piksel pada citra dan membagi citra tersebut sesuai dengan fitur yang diinginkan. Segmentasi citra dengan deteksi warna HSV menggunakan dasar seleksi warna pada model HSV degan nilai toleransi tertentu.

Hue adalah representasi warna sesungguhnya dari masing-masing piksel dan mempunyai nilai 0 sampai 2π. *Saturation* adalah komponen yang menyatakan tingkat kemurnian suatu warna, yaitu mengindikasikan seberapa banyak warna putih yang diberikan pada suatu warna. Nilai *saturation* berkisar antara 0 sampai 1. Sedangkan *value* atau juga disebut dengan *intensity* merupakan atribut yang menyatakan banyak cahaya. Dengan nilai *intensity* yang rendah, meskipun dengan nilai *saturation* yan tinggi, warna akan mendekati nilai abu- abu. *Saturation* memberikan nilai kemurnian warna tetapi mata manusia kurang sensitive dengan variasinya jika dibandingkan dengan variasi pada *Hue* dan *Intensity*. Untuk mendapatkan model warna HSV perlu dilakukan konversi dari model warna RGB karena gambar masukan dari layar perangkat keras atau kamera memiliki default model warna RGB.

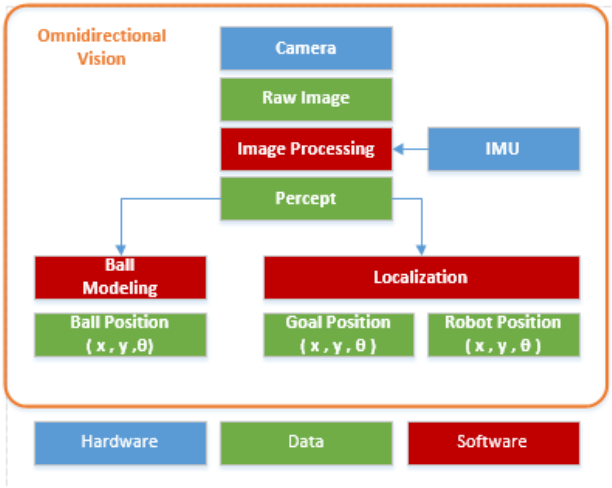


Gambar 3. Model warna HSV.

III. Desain Sistem

A. *Desain Omnidirectional Vision*

Pada bagian ini akan dijelaskan mengenai desain omnidirectional vision yang digunakan pada robot ERSOW.

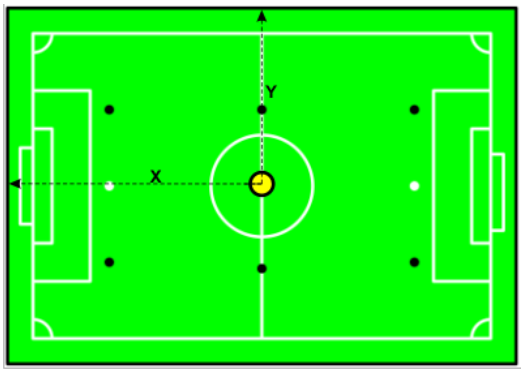


Gambar 4. Diagram Sistem Omnidirectional Vision.

Pada diagram sistem diatas merupakan gambaran proses yang dilakukan untuk mendapatkan data persepsi pada robot ERSOW. Gambar yang dihasilkan oleh kamera akan diubah, segmentasi warna gambar yang sebelumnya adalah RGB di ubah menjadi HSV untuk selanjutnya dilakukan proses ekstraksi berdasarkan nilai warna yang didapatkan pada saat proses kalibrasi.

Nilai dari IMU digunakan untuk mengetahui arah hadap dari robot dan juga untuk melakukan rotasi gambar agar tidak berubah sehingga mudah dalam melakukan pengolahan nilai obyek yang dideteksi untuk selanjutnya dijadikan nilai masukan ke dalam pergerakan robot. Untuk mengetahui posisi dari robot dilakukan dengan menggunakan pendekatan nilai jarak dengan sisi lapangan. Untuk mendapatkan data lapangan pada omnidirectional vision maka diperlukan proses *undistortion* untuk mengembalikan gambar sesuai dengan keadaan sebenarnya.

Pada Gambar 5 ditunjukkan koordinat penentuan posisi robot di lapangan. Penetuan posisi dilakukan dengan menarik garis dari hasil kalibrasi lapangan. Dengan area lapangan tersebut maka didapatkan nilai x dan y, yang merupakan koordinat posisi dari robot di lapangan.



Gambar 5. Koordinat Penentuan Posisi Robot.

Untuk melakukan kalibrasi data antara data *vision* dengan kondisi nyata maka digunakan persamaan linier. Kalibrasi dilakukan di titik tengah lapangan, hal ini dikarenakan untuk mempermudah dalam hal pengukuran dengan kondisi lapangan.

$$x_{app} = \frac{x_{input}}{x_{calibration}} \times 45$$

$$y_{app} = \frac{y_{input}}{y_{calibration}} \times 30$$

dimana :

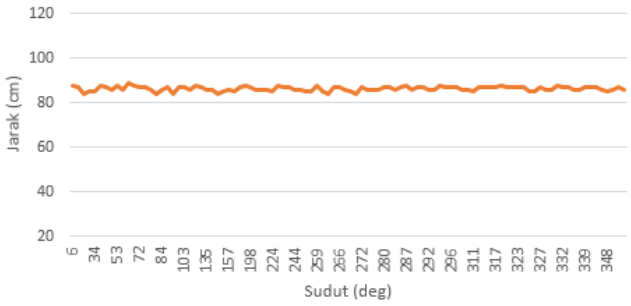
$(x,y)_{app}$:nilai (x,y)yang didapatkan
$(x,y)_{input}$:nilai (x,y)masukan dari data vision
$(x,y)_{calibration}$:nilai (x,y)yang didapatkan saat proses kalibrasi

Hasil yang didapatkan per 10 cm. Nilai 45 dan 30 merupakan data titik kalibrasi. Dalam hal ini lapangan yang digunakan memiliki ukuran 9 x 6 meter.

IV. Hasil dan Analisa

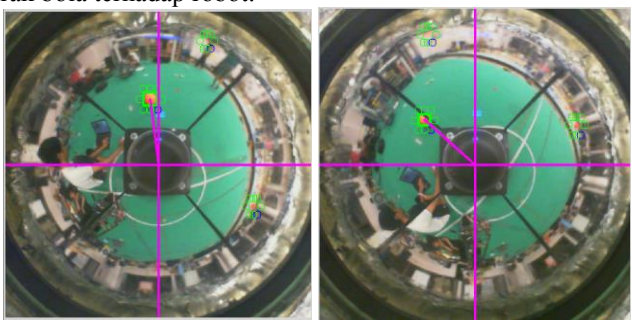
Pengujian dan analisa dimaksudkan untuk mengetahui sejauh mana sistem yang dibuat dalam penelitan ini dapat berkerja sesuai dengan yang diharapkan serta melakukan analisa pada setiap tahapan proses dan hasil secara keseluruhan.

Pada pengujian tracking terhadap bola dilakukan dengan kondisi robot berotasi dan bola diam, adapun hasil pengujian tracking bola ditunjukkan pada Gambar 6.



Gambar 6. Grafik tracking bola dengan kondisi robot berotasi dan bola diam.

Pada Gambar 6 menunjukkan bahwa tidak terjadi error deteksi bola yang ditunjukkan oleh grafik tersambung. Sehingga persentase error ketika robot berotasi dan bola diam adalah sebesar 0%. Namun keberhasilan pendeteksian tersebut juga dipengaruhi pada proses kalibrasi yang dilakukan dan jarak bola terhadap robot.



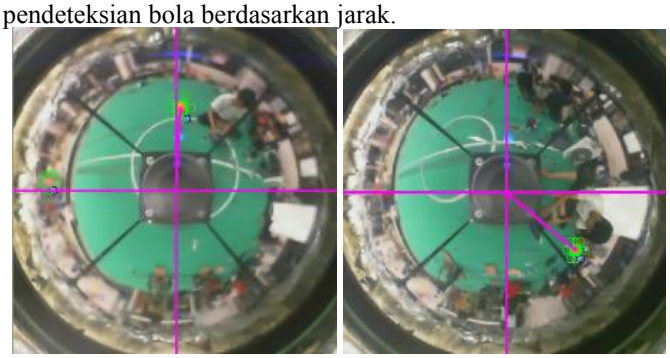
Gambar 7. Pendeteksian bola di lapangan dengan rotasi robot.

Pengujian selanjutnya adalah pengujian pendeteksian bola berdasarkan jarak. Pada TABEL I dapat dilihat bahwa dalam jarak 5 meter bola masih terdeteksi dengan baik.

TABEL I. PENGUJIAN PENDETEKSIAN BOLA

No	Jarak (cm)	Hasil	
		Terdeteksi	Tidak Terdeteksi
1	50	✓	-
2	100	✓	-
3	150	✓	-
4	200	✓	-
5	250	✓	-
6	300	✓	-
7	350	✓	-
8	400	✓	-
9	450	✓	-
10	500	✓	-

Gambar dibawah ini adalah beberapa proses pengujian pendeteksian bola berdasarkan jarak.



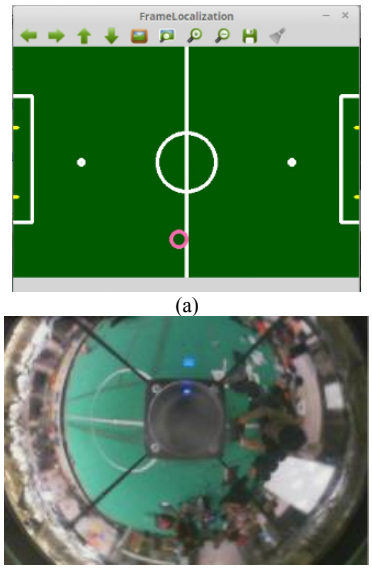
Gambar 8. Pendeteksian bola (a) 1 meter , (b) 2 meter.

Pada pengujian posisi robot dilakukan dengan membandingkan data jarak sesungguhnya dengan jarak hasil perhitungan aplikasi. Data disajikan per 10 cm dengan ukuran lapangan 9 x 6 meter.

TABEL II. DATA POSISI ROBOT

No	Real Position (X,Y)	Application (X,Y)
1	(45,30)	(45,30)
2	(30,30)	(31,28)
3	(10,20)	(13,23)
4	(15,10)	(15,9)
5	(20,50)	(19,53)

Dari data hasil pengujian yang ditampilkan pada TABEL II menunjukkan data yang dihasilkan cukup stabil dan memiliki selisih yang tidak terlalu jauh. Perbedaan yang dihasilkan dapat dipenaruhi pada hasil ekstraksi warna lapangan yang tidak sempurna sehingga menyebabkan hasil pengukuran tidak sesuai dengan kondisi sebenarnya.



(a)

(b)

Gambar 9. (a) Hasil aplikasi (b) Posisi robot sebenarnya.

V. KESIMPULAN

Berdasarkan hasil yang didapatkan mengenai penggunaan cermin omnidirectional, maka dapat disimpulkan bahwa dalam menggunakan cermin omnidirectional diperlukan resolusi kamera tinggi untuk mempermudah dalam pengenalan objek. Dan sistem penentuan posisi robot dilapangan dengan error yang kecil, sehingga mempermudah dalam penyusunan strategi permainan.

VI. PENELITIAN SELANJUTNYA

Penelitian selanjutnya adalah memperbaiki proses mapping robot di lapangan dengan menambahkan referensi data seperti dengan menggunakan data dari fitur garis dan gawang. Sehingga meningkatkan akurasi data dari hasil penentuan posisi robot di lapangan.

REFERENSI

[1] Neves, A., Pinho, A., Martins, D., et al."An efficient omnidirectional vision system for soccer robots: From calibration to object detection". In Mechatronics, Volume 21, Issue 2, March 2011,p.399-410, 2010.

[2] Lopez, G., Ribeiro, F. "Catadioptric system optimisation for omnidirectional Robocup MSL robots". In RoboCup 2011: Robot Soccer World Cup XV, 2012.

[3] F. Liu, H.Lu, and Z.Zheng, "A Modified Color Look-Up Table Segmentation Method for Robot Soccer". In Proc. Of the 4th IEEE LARS/COMRb 07, 2007.

[4] Jamzad, M., Hadjkhodabakhshi, A. Mirrokni, V, "Object Detection and Localization Using Omnidirectional Vision in the Robocup Environment". In Scientia Iranica 14(6). November. 2007.

[5] Middle Size Robot League Rules and Regulation for 2017.2016.

Sistem Pendeteksian Gawang menggunakan Metode Houghlines dan Cornerharris pada Robot Fire-X UAD

Tofik Nurochman*, Yogo Paranto Aji, dan Riky Dwi Puriyanto
Teknik Elektro
Universitas Ahmad Dahlan
Yogyakarta, Indonesia
Tofik216nurochman@gmail.com

***Abstrak*— Dalam pertandingan robot sepak bola beroda robot dituntut harus dapat bermanuver secara autonomous, serta dapat mengenali bola, gawang, serta parameter lain yang berada di lapangan yang dapat digunakan sebagai bagian dari strategi robot, selain itu robot juga harus mampu membedakan antara gawang sendiri dengan gawang lawan. Penerapan pengolahan citra digital merupakan salah satu pilihan yang tepat yang dapat diimplementasikan pada robot agar robot dapat mengenali bola, gawang maupun objek lain. Pengolahan citra menggunakan metode *houghlines* dan *cornerharris* merupakan metode pendeteksian gawang yang telah dikembangkan oleh tim FIRE-X UAD. Metode tersebut menggunakan parameter garis tepi gawang dan sudut gawang sebagai acuan untuk pendeteksian gawang, serta menggunakan parameter lain yaitu warna hijau lapangan. Sehingga dengan menggunakan metode tersebut robot hanya dapat mendeteksi gawang yang berada diatas objek yang berwarna hijau dan dapat membedakan objek yang serupa gawang yang berada di sekitarnya. Hasil pengujian menunjukan penerapan metode *houghlines* dan *cornerharris* pada robot memiliki keakuratan tinggi untuk mendeteksi gawang, hal itu dibuktikan dengan berbagai macam skema pengujian robot berhasil mendeteksi gawang dengan baik.**

Kata Kunci—Pendeteksian gawang; Houghlines; Cornerharris

I. PENDAHULUAN

Dalam pertandingan sepak bola robot beroda, robot dituntut agar dapat mendeteksi bola dan gawang serta dapat mengetahui komponen lain yang terdapat pada lapangan pertandingan sehigga robot dapat memainkan sepakbola seperti layaknya manusia bermain sepakbola. Untuk itu robot harus memiliki kemampuan dasar dalam bermain sepakbola seperti mengenali bola, mengenali gawang, menggiring bola dan menendang bola serta robot dapat berorientasi terhadap perannya dilapangan pertandingan [1].

Teknologi *computer vision* merupakan penerapan disiplin ilmu yang saat ini banyak digunakan sebagai objek penelitian pada robot sepak bola. Pengolahan citra merupakan bagian dari *computer vision* yang menggunakan analisis warna yang dapat diolah dengan beberapa model diantaranya, RGB, CMY, HSI, HSV dan normalized RGB [2]. Salah satu model yang paling banyak digunakan yaitu model RGB dan HSV karena sederhana dalam pemrogaman dan cepat dalam pemrosesan, sehingga sangat cocok untuk di terapkan dalam robot sepak bola untuk sistem deteksi bola dan gawang.

Salah satu metode untuk mendeteksi gawang yang telah di kembangkan dan diterapkan pada tim robot sepak bola beroda UAD yaitu menggunakan metode *Houghlines* dan *Cornerharris*, metode tersebut cukup sukses jika hanya digunakan untuk mendeteksi gawang, tetapi akan terjadi lambat dalam pemrosesan jika di gabungkan dengan metode yang digunakan untuk medeteksi bola. Sehingga perlu dilakukan pengembangan lebih lanjut untuk sistem metode penelitian gawang pada robot sepakbola beroda UAD.

II. DASAR TEORI

Sistem deteksi gawang yang digunakan memanfaatkan tanda visual sebuah objek yang berada pada lapangan yang kemudian di padukan dengan ciri visual yang di deteksi, sehingga dapat membedakan antara gawang dengan objek serupa yang berada di luar lapangan.

A. Filter Warna (Color Filtering).

Filter warna merupakan suatu teknik pengolahan citra untuk memanipulasi citra berdasarkan warna khusus. Cara kerjanya yaitu dengan membandingkan antara komponen warna pada setiap *pixel* citra dengan warna spesifik yang digunakan [3]. *Color Filtering* dapat mereprentasikan warna dalam berbagai model, diantaranya RGB (Red, Green, Blue), HSV (Hue, Saturation, Value), YCbCr, dsb. HSV merupakan ruang warna yang paling mudah dan sederhana sehingga cocok untuk mengidentifikasi warna-warna dasar, dimana warna dasar ini digunakan dalam penelitian untuk identifikasi gawang. Selain itu, HSV memiliki kemudahan kalibrasi terhadap perubahan intensitas cahaya [4].

B. Model HSV.

Model ruang warna HSV memiliki 3 komponen yaitu, *Hue, Saturation*, dan *Value*. Model warna HSV dapat mendeteksi ciri visual seperti yang ditangkap oleh indra manusia [5].

$$H = \begin{cases} 60\left(\frac{(G-B)}{\delta}\right) & MAX = R \\ 60\left(\frac{(B-R)}{\delta}+2\right) & MAX = G \\ 60\left(\frac{(R-G)}{\delta}+2\right) & MAX = B \\ not_defined & MAX = 0 \end{cases} \quad (1)$$

$$S = \begin{cases} \frac{\delta}{MAX} & MAX \neq 0 \\ 0 & MAX = 0 \end{cases} \quad (2)$$

$$V = MAX \quad (3)$$

C. *Tresholding.*

Tresholding merupakan sebuah konsep untuk mengubah suatu citra menjadi dua jenis intensitas yaitu 0 atau 255. Untuk melakukan proses ini digunakan ambang batas (*threshold*) yang nilainya dapat diatur sesuai kebutuhan. Model *treshholding* dapat di wakilkan seperti pada persamaan (4).

f(x,y);if f(x,y)>*Threshold then* f(x,y)=255 *else* 0 (4)

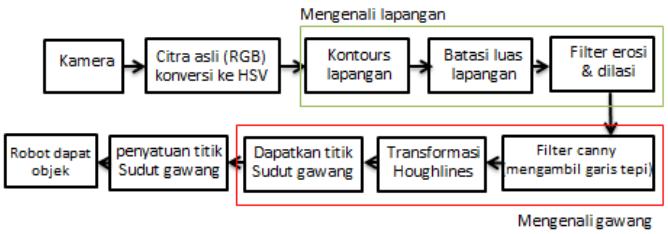
Persamaan untuk merubah citra asli menjadi citra biner Jika nilai *threshold* yang dipasang tidak sesuai dengan nilai ambang batas yang di berikan maka akan di ganti dengan nilai 0 untuk dibawah ambang dan dengan nilai 255 jika berada diatas ambang batas yang di berikan.

III. METODOLOGI PENELITIAN

Pada pendeteksian gawang menggunakan metode *Houghlines* dan *Cornerharris* ada beberapa proses tahapan yang dilalui. Dibawah ini, beberapa tahapan-tahapan dari pendeteksian gawang.

A. *Sistem Deteksi.*

Perancangan sistem untuk melakukan proses mendapatkan hasil, dapat digambarkan dalam bentuk blok diagram seperti yang diperlihatkan pada Gambar 1. Sistem terdiri atas beberapa bagian seperti berikut.



Gambar 1. Blok diagram proses deteksi gawang.

B. *Pemrogaman Visual.*

Pemrogaman visual untuk *computer vision* yang digunakan berupa software Ultra VNC, dimana software tersebut dapat di masukan library Open CV. Open CV (*Open Source Computer Vision*) merupakan sebuah *library* yang dikhususkan untuk penglihatan komputer secara *real time* yang dikembangkan oleh pusat penelitian Intel di Nizhny Novgorod, Rusia.Open CV ini dapat digunakan di berbagai bahasa pemrograman, seperti C, C++, Java, Python, dan suport dengan Windows, Linux, Mac OS, iOS dan Android. Open CV didesain untuk efisiensi komputasi dan dengan fokus yang kuat pada aplikasi *realtime* [6].

C. *Konversi Warna RGB ke HSV*

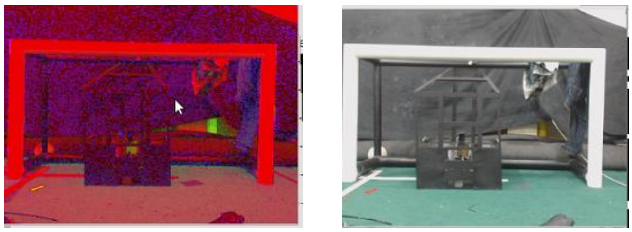
Dalam pengolahan citra digital kamera pertama kali akan menangkap citra RGB, yang digunakan sebagai dasar agar dapat mengenali objek yang akan dideteksi. Dari citra RGB tersebut kemudian dikonversi menjadi model HSV agar lebih sederhana untuk diproses ke tahap selanjutnya. Persamaan (5) merupakan perumusan sederhana yang di gunakan untuk mengkonversi dari RGB ke HSV.

$$H = \tan\left(\frac{3(G-B)}{(R-G)+(R-B)}\right)$$

$$S = 1 - \frac{\min(R,G,B)}{V}$$

$$V = \frac{R+G+B}{3} \quad (5)$$

Kemudian dari proses tersebut dilakukan proses segmentasi untuk mendapatkan objek. Hasil segmentasi yang telah dilakukan ditampilkan oleh Gambar 2.



(a) (b)

Gambar 2. (a) Model RGB, (b) Model HSV.

Dari Gambar 2 dapat dilihat model RGB merupakan campuran dari warna-warna primer yang menyatu sehingga akan sedikit lebih susah untuk dipisahkan kedalam kelas-kelas warna. Sedangkan model HSV lebih mendekati dengan citra yang dapat ditangkap indra manusia sehingga lebih mudah dipisahkan kedalam kelas warna tertentu.

D. *Kontours Lapangan.*

Pada tahap ini citra yang digunakan sebagai objek berupa warna hijau pada lapangan KRSBI beroda. Penggunaan warna lapangan sebagai parameter pendeteksian gawang bertujuan agar robot hanya dapat mendeteksi gawang yang berada diatas lapangan. Sehingga robot dapat mengurangi *noise* terhadap objek yang serupa gawang yang berada di sekitar lapangan. Setelah mendapatkan *kontours* lapangan kemudian diberikan batas luasan lapangan hal ini robot hanya dapat mendeteksi gawang diatas lapangan yang berada pada luasan tertentu. Pendeteksian *kontours* lapangan diperlihatkan oleh Gambar 3.

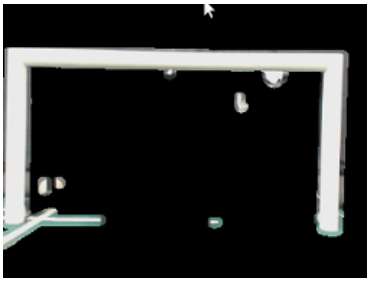


Gambar 3. Kontours lapangan.

Warna hijau lapangan yang dapat dideteksi dibatasi pada luasan tertentu dengan tujuan agar robot tidak mendeteksi objek serupa di sekitar lapangan sehingga *noise* pada saat mendeteksi lapangan dapat dikurangi. Citra lapangan kemudian di tutup dengan warna hitam agar lebih mudah untuk mendapatkan citra objek yang akan dideteksi yaitu berupa gawang.

E. *Erosi & Dilasi.*

Tahap yaitu selanjutnya dilakukan proses *Erosi* dan *Dilasi*. Proses *erosi* dilakukan pada citra gawang dengan tujuan untuk menperkecil *pixel* pada citra gawang. Sementara proses *dilasi* dilakukan pada citra lapangan dan citra lain yang berada di sekitar gawang. Untuk proses *erosi* dan *dilasi* yang telah dilakukan ditunjukan oleh Gambar 4.

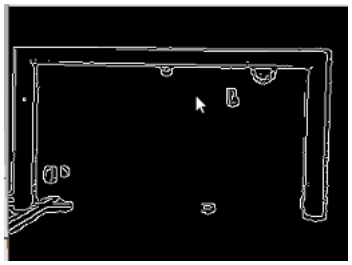


Gambar 4. Representasi citra setelah dilakukan proses *erosi* dan *dilasi*.

Dengan kedua proses tersebut robot akan lebih mudah mengenali gawang, karena hanya objek gawang yang dideteksi selain itu akan ditutup dengan warna hitam.

F. *Filter Canny.*

Penggunaan filter *canny* bertujuan untuk mendapatkan garis tepi citra gawang. Yang kemudian direpresentasikan dengan titik-titik tepi yang sangat pendek sehingga menjadi sebuah garis tepi. Filter *canny* yang telah dilakukan pada citra gawang ditunjukan pada Gambar 5.

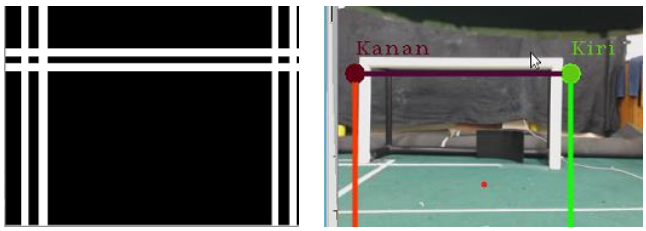


Gambar 5. Citra gawang setelah dilakukan filter *canny*.

Dari gambar tersebut dapat dilihat objek yang ditampilakan sudah berupa garis-garis yang membentuk gawang. Garis tersebut merupakan garis tepi gawang setelah mengalami proses filter *canny*.

G. *Alogaritma Houghlines dan Cornerharris.*

Alogaritma ini digunakan untuk mendeteksi garis lurus yang dihasilkan oleh filter *canny*. Hal ini dilakukan agar dapat dideteksi lebih spesifik sehingga robot akan lebih jelas dan cepat dalam mengenali gawang. Sedangkan alogaritma *Cornerharris* melupakan alogaritma yang digunakan untuk mendeteksi sudut. Gawang pada KRSBI beroda terdiri dari dua garis yaitu vertikal dan horisontal, pada titik temu kedua garis tersebut terdapat sudut-sudut singgung, sudut tersebutlah yang digunakan sebagai acuan dalam mendeteksi gawang. Pada garis tepi yang ditemukan oleh pada alogaritma *houghlines* juga akan menghasilkan sudut-sudut singgung antar garis. sehingga Dengan kedua proses robot sudah dapat mengenali gawang. Hasil penggunaan alogaritma houghlines dan cornerharis ditampilkan pada Gambar 6.



(a) (b)

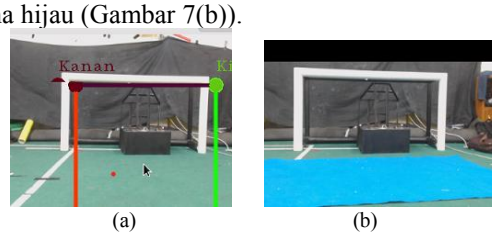
Gambar 6. (a) Penggunaan alogaritma *houghlines*, (b) Hasil penggunaan alogaritma *cornerharris*.

Pada Gambar 6(a) dapat kita lihat terdapat garis tepi gawang yang saling menyinggung menghasilkan sudut, yang kemudian sudut tersebut dideteksi menggunakan alogaritma *cornerharris*. Sehingga hasil akhirnya robot dapat mengenali gawang seperti yang ditampilkan pada Gambar 6(b)

IV. HASIL DAN PEMBAHASAN

A. *Pengujian 1*

Pengujian ini menggunakan *hardware* sebuah Webcam logitech C922 dan sebuah mini PC Odroid Xu-4 sebagai pengolah citra digital yang di tangkap oleh Webcam. Pengujian ini dilakukan dalam beberapa tahap, yang pertama dengan meletakan objek berupa gawang diatas lapangan berwana hijau (Gambar 7(a)) dan diatas media yang bukan berwarna hijau (Gambar 7(b)).



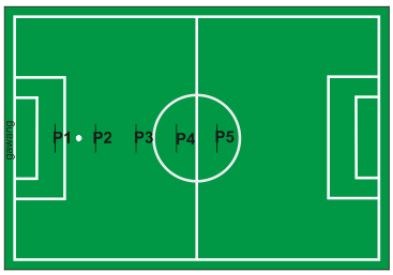
(a) (b)

Gambar 7. (a) Gawang di atas warna hijau, (b) Gawang diatas warna bukan hijau.

Dari gambar tersebut dapat kita lihat robot dapat mendeteksi gawang dengan baik berdasarkan titik sudut gawang pada objek gawang yang berada di atas warna hijau. Sementara pada Gambar 7(b) robot tidak dapat mendeteksi gawang karena objek tersebut tidak berada diatas warna hijau.

B. *Pengujian 2.*

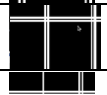
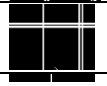
Pengujian kedua ini yang dilakukan yaitu mendeteksi objek berdasarkan jarak robot terhadap objek yang dideteksi. pengujian tersebut dilakukan seperti skema pada Gambar 8.



Gambar 8. Skema pengujian 2.

P1-P5 merupakan titik-titik dimana robot akan diletakan untuk pengujian. Antar titik tersebut berjarak satu meter, jarak gawang terhadap titik P1 berjarak satu meter, titik P1 juga berjarak satu meter terhadap titik P2, begitu seterusnya.
TABEL I. Menunjukan hasil dari sistem pendeteksian gawang berjasarkan jarak robot terhadap gawang.

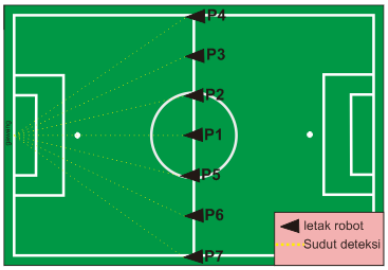
TABEL I. Pengujian objek berdasarkan jarak

Posisi	Filter canny	Metode Hough	Metode CornerHarris	Nilai (x,y) Sudut kanan	Nilai (x,y) Sudut kiri
P1				36, 54	316, 54
P2				52, 77	307, 74
P3				149, 92	307, 92
P4				144, 70	283, 70
P5				125, 67	254,70

Dari hasil uji yang dilakukan, pada TABEL I diperlihatkan pada pengujian P1-P5 penggunaan filter canny masih terdapat *noise* dari garis lapangan yang membentuk sudut robot, untuk itu penggunaan metode *houghlines* digunakan untuk menemukan garis tepi gawang dan membedakan gawang dengan garis lapangan, kemudian digunakan metode *cornerharris* untuk menemukan sudut persinggungan antar garis tepi gawang. Dengan penggunaan ketiga metode tersebut robot dapat mendeteksi objek dengan baik pada setiap letak robot yang diujikan.

C. *Pengujian 3.*

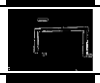
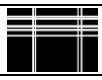
Pada pengujian yang kedua ini masih berdasarkan jarak robot terhadap gawang serta digabungkan dengan sudut letak robot terhadap gawang. Skema pengujian 3 yang dilakukan ditunjukan oleh Gambar 9.



Gambar 9. Skema pengujian 3.

Dari gambar tersebut dapat kita lihat P1-P7 merupakan titik dimana robot akan diletakan untuk pengujian. Masing-masing titik memiliki jarak dan sudut yang berbeda terhadap gawang atau objek yang akan di deteksi. TABEL II, menunjukan hasil dari pengujian 2 yang dilakukan dengan mengacu skema yang ditunjukan oleh Gambar 9.

TABEL II. Hasil pengujian berdasarkan sudut letak robot.

Posisi	Filter canny	Metode Hough	Metode CornerHarris	Nilai (x,y) Sudut kanan	Nilai (x,y) Sudut kiri
P1				69, 69	265, 69
P2				114, 81	300, 81
P3				-	241, 75
P4				-	255, 75
P5				47, 77	259, 77
P6				93, 77	271, 77
P7				122, 93	301, 93

Dari TABEL II, dapat kita lihat pada pengujian P3 dan P4 robot hanya dapat mendeteksi sudut gawang pada sebelah kiri.hal itu di sebabkan sudut pandang robot pada posisi yang terlalu jauh dengan sudut gawang sebelah kiri, sehingga sudut gawang tersebut tidak terdeteksi sebagai suatu sudut. Tetapi penggunaan logika OR pada strategi robot untuk mendeteksi gawang menyebabkan robot dapat mendeteksi gawang walaupun hanya satu sudut yang terdeteksi.

V. KESIMPULAN

1. Robot dapat mendeteksi gawang dengan baik, menggunakan metode Houghlines dan Cornerharris.

2. Berdasarkan pengujian robot dapat membedakan antara gawang dengan objek serupa yang berada di sekitar lapangan pertandingan.
3. Penelitian sistem pendeteksian gawang menggunakan metode houglines dan cornerharris tidak terpengaruh oleh jarak, sehingga cocok diterapkan pada robot sepak bola beroda.

REFERENSI

[1] Solichah uti, Muhammad Arifin, dan Muhammad Ardi Pradana, "Pendeteksian Gawang Menggunakan Algoritma *Ransac* Pada *Platform Darwin-OP* Berbasis Peraturan KRSBI 2016," The 4th Indonesian Symposium on Robot Soccer Competition 2016.

[2] Pambudi Wahyu Setyo, Irma Salamah, dan Alan Novi Tompunu, Deteksi dan Estimasi Jarak Obyek Menggunakan *Single Camera* Dengan Model Segmentasi HSV, Seminar Nasional Teknoin 2011 ISBN 978-979-96964-8-9.

[3] Payung, Musa, dan Bayu Saputera, "Robot Sepakbola dengan Pencarian dari Pengolahan Citra OpenCV dan Arduino," The 4th Indonesian Symposium on Robot Soccer Competition 2016.

[4] Mahesa, Jovi, dan Azizul Meliyana, "Rancang Bangun Sistem Tracking Pada Robot Sepak Bola Dengan Metode Thresholding Menggunakan Open-CV," The 4th Indonesian Symposium on Robot Soccer Competition 2016.

[5] Mandala, Hanjaya, Eko Rudiawan, dan Hendawan Soebhakti, "Sistem Deteksi Bola Berdasarkan Warna Bola Dan Background Warna Lapangan Pada Robot Barelang FC," The 4th Indonesian Symposium on Robot Soccer Competition 2016.

[6] Jovi Mahesa, dan Azizul, Meliyana. Tagawa, "Rancang Bangun Sistem Tracking Pada Robot Sepak Bola Dengan Metode Thresholding Menggunakan Open-CV," The 4th Indonesian Symposium on Robot Soccer Competition 2016.

Perancangan dan Realisasi Robot Underwater Autonomous Penjejak Garis dengan Metoda Thresholding dan Moments

Muliady*, Junaidi Sucipto
Program Studi Teknik Elektro
Universitas Kristen Maranatha
Bandung, Indonesia
mld_ang@yahoo.com

***Abstrak*—Eksplorasi di bawah laut menjadi hal penting dan akan lebih tepat jika dilakukan oleh robot yang dikenal sebagai robot *underwater*. Dalam pengembangan robot *underwater* masih diperlukan ide-ide baru sehingga banyak diselenggarakan kompetisi. Pada penelitian ini telah direalisasi robot untuk kategori robot *underwater autonomous* pada *Singapore Robotic Games* yang dapat menjejak garis berdasarkan pemrosesan citra pada Raspberry Pi 2 dengan memanfaatkan library OpenCV seperti Gaussian blur, thresholding, dan moment. Arduino Uno pada robot berfungsi sebagai *co-processor* dalam pengontrolan pergerakan robot. Pada robot ini terdapat sensor kamera untuk penglihatan, sensor orientasi IMU, dan sensor tekanan MPXV5004. Uji coba menunjukkan robot mampu menjejak garis, mendeteksi kargo, arena basket dan dapat mengatur kedalaman menyelam. Robot tidak dapat memindahkan barang dari kargo ke dalam arena basket karena medan magnet yang dihasilkan solenoid pengambil barang mengganggu kerja dari sensor IMU yang ada pada robot.**

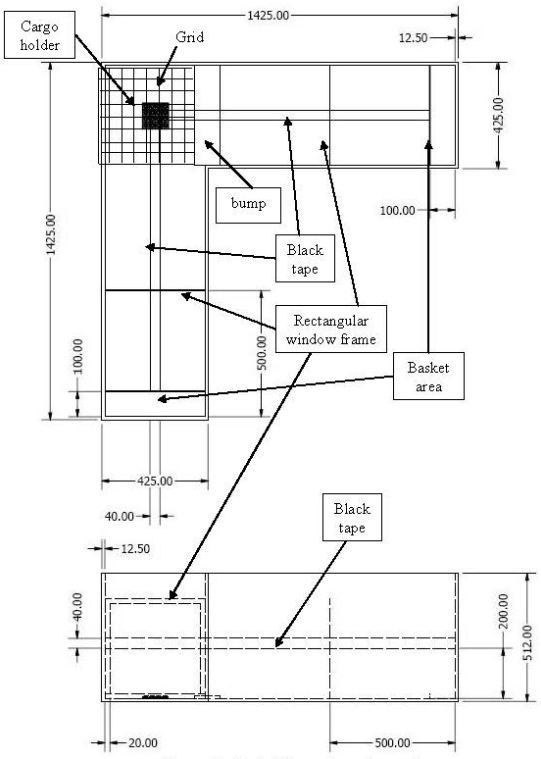
Kata Kunci—robot underwater autonmous; penjejak garis; thresholding; moment

I. PENDAHULUAN

Indonesia adalah negara maritim yang kekayaan alamnya sebagian besar terdapat di dalam laut, oleh karena itu teknologi eksplorasi di bawah air menjadi hal yang sangat penting. Tetapi akan beresiko jika manusia yang secara langsung melakukannya, sehingga robot menjadi opsi yang tepat dalam meminimalisir resiko.

Penelitian robot *underwater* menjadi topik yang menarik dengan berbagai misi, seperti menjaga keamanan pelabuhan, eksplorasi ilmiah bawah laut, atau pembersihan dalam laut. Untuk menstimulasi penelitian dan pembelajaran robot *underwater* telah banyak diselenggarakan kompetisi robot *underwater* sehingga dapat menciptakan ide-ide dalam perkembangan robot *underwater*. Salah satu kompetisi yang diselenggarakan adalah kategori RC (*remote control*) *underwater robot* dan *underwater autonomous robot* pada Singapore Robotic Games. Misi dari lomba ini adalah memindahkan barang-barang yang disebut kargo.

Gambar 1 merupakan arena perlombaan robot underwater pada kompetisi Singapore Robotic Games 2015 [1]. Arena berbentuk "L" dengan ukuran panjang 1425mm x 1425mm dan tinggi 512mm. Pada arena terdapat *black tape*, *basket area*, *cargo holder*, *bump*, dan *rectangular window frame*. *Black tape* merupakan garis berwarna hitam yang ditempel pada sisi kiri, sisi kanan, dan dasar arena. *Basket area* berada pada ujung arena yang berfungsi menampung kargo yang telah angkut. *Cargo holder* adalah tempat penyimpanan barang yang berupa mur besi. *Bump* merupakan sambungan dari dua belahan pada arena.



Gambar 1. Arena lomba robot *underwater* (satuan mm).

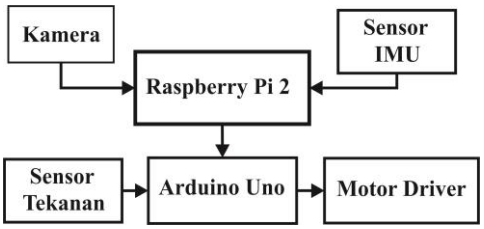
II. PERANCANGAN DAN REALISASI ROBOT

Robot *underwater* dirancang agar dapat bergerak secara otomatis di dalam air dan dapat menyelesaikan misi yaitu menjejak garis dengan kedalaman yang diatur, mengambil mur besi dari *cargo holder*, dan meletakannya dalam basket.

Perancangan dan reaslisasi ini dibagi menjadi tiga bagian yaitu sistem elektronika robot, kontrol robot, dan mekanika robot.

A. *Sistem Elektronika Robot*

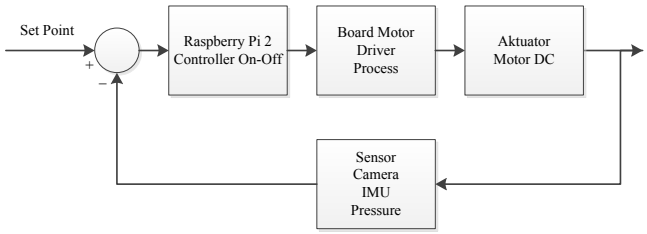
Robot *underwater* dirancang dengan bagian elektronika yang terdiri dari Raspberry Pi 2 [2] sebagai *processor* utama, modul kamera Raspberry [3] untuk *vision*, arduino uno [4] sebagai *co-processor*, sensor tekanan MPXV5004G6U yang dihubungkan dengan arduino uno sebagai umpan balik untuk kedalaman menyelam robot, sensor IMU Pololu MinIMU-9 v2 untuk data orientasi *heading*, dan motor driver untuk menggerakan motor pada robot. Diagram blok sistem elektronika robot dapat dilihat pada Gambar 2.



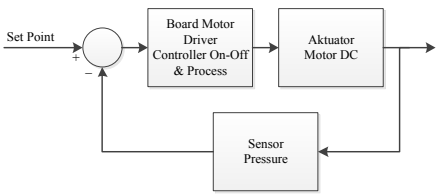
Gambar 2. Diagram blok sistem elektronika robot.

B. *Sistem Kontrol Robot*

Pada robot *underwater* diterapkan sistem kontrol *closed loop*. Set point pada *processor* utama merupakan titik tengah sumbu x pada kamera, besar koordinat ini adalah 320. Raspberry Pi akan melakukan pembacaan kamera secara terus menerus dan menggerakkan motor agar berada pada posisi set point. Mode pengontol yang akan digunakan adalah *on-off* dengan histerisis yang ditunjukkan pada Gambar 3 agar robot dapat menjejak garis. Set point untuk *co-processor* adalah kedalaman menyelam robot yang akan dijaga dengan menggunakan mode pengontrol *on-off* seperti pada Gambar 4.



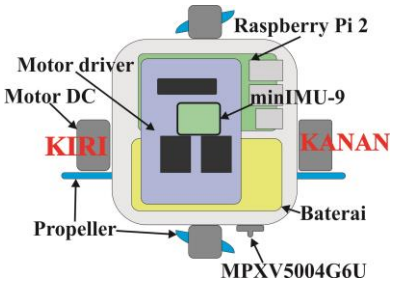
Gambar 3. Sistem kontrol *processor* utama pada robot.



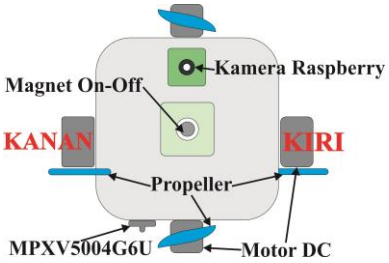
Gambar 4. Sistem kontrol *co-processor* pada robot.

C. *Sistem Mekanika Robot*

Robot *underwater* direalisasikan dengan *container* kedap air bentuk kotak berukuran11,5 x 11,5 x 6 cm (p x l x t). Robot menggunakan sistem gerak *differential drive*, Gambar 5 dan 6 menggambarkan rancangan robot yang terdiri dari 4 buah motor, Raspberry Pi 2, *driver motor*, minIMU-9, sensor tekanan MPXV5004G6U, kamera, dan sebuah magnet *on-off*.



Gambar 5. Tampak atas robot.

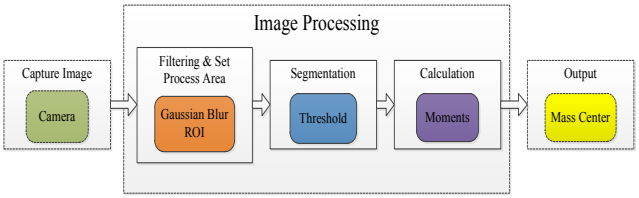


Gambar 6. Tampak bawah robot.

Motor yang berada di sisi kanan dan kiri robot menggerakkan robot agar dapat bergerak ke depan, ke kiri, ke kanan, dan ke belakang. Motor bagian depan dan belakang bertugas membuat robot menyelam pada kedalaman tertentu. Sedangkan untuk bagian pengambilan mur yang berada dalam *cargo holder* digunakan selenoid.

III. PENGOLAHAN CITRA DAN ALGORITMA ROBOT

Pengolahan citra pada robot bertujuan untuk mengetahui letak posisi garis. Citra yang diolah berupa frame gambar dari kamera secara *real time* dengan format warna *grayscale*. Pengolahan citra gambar menggunakan *library* OpenCV [5],[6], fungsi-fungsi yang akan digunakan adalah *Gaussian Blur* [7], *ROI (region of Interest), Thresholding* [8], dan *Moments*. Gambar 7 menunjukkan proses mendapatkan koordinat garis pada frame gambar yang ditangkap kamera.



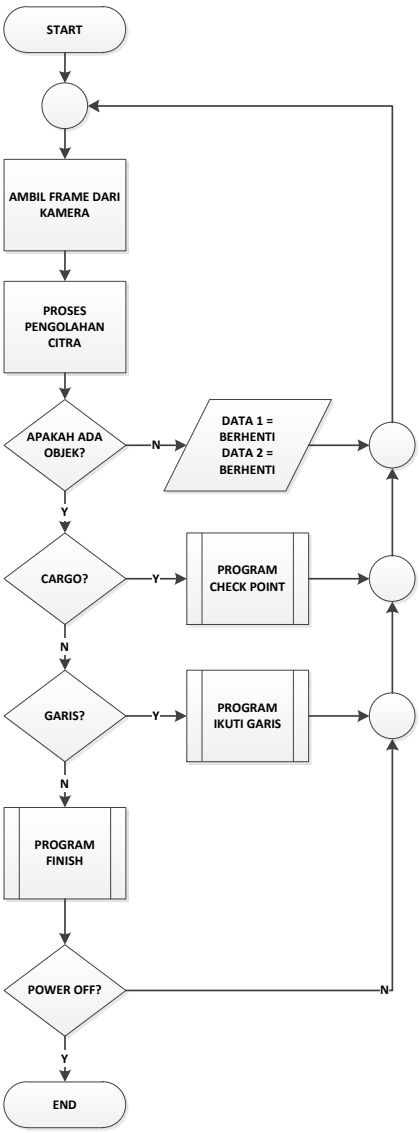
Gambar 7. Diagram blok pengolahan citra.

Algoritma pada robot underwater dirancang agar robot dapat bergerak secara autonomous di dalam air dalam

menyelesaikan misi yang telah ditentukan. Misi robot adalah mulai berjalan dari salah satu ujung arena menuju ke kargo dan kemudian memindahkan barang yang ada di kargo kesalah satu ujung arena secara satu persatu. Perancangan algoritma pada robot underwater memanfaatkan sensor IMU sebagai acuan orientasi heading robot dan sensor tekanan sebagai kedalaman robot menyelam.

Algoritma robot underwater dirancang terbagi menjadi dua bagian besar, yaitu algoritma *master* dan algoritma *slave*. Algoritma *master* untuk *processor* utama yaitu Raspberry Pi, algoritma *slave* untuk *co-processor* Arduino Uno. Kedua *processor* berkomunikasi dengan protokol komunikasi UART, dengan baud rate 9600 bps.

Pada algoritma *master* proses dimulai dengan pengambilan frame gambar oleh kamera, lalu dilakukan proses pengolahan citra seperti Gambar 7. Keluaran dari proses pengolahan citra akan menjadi penentu dari perintah yang akan diberikan *processor* utama kepada *co-processor*. Detil algoritma *master* ditunjukkan pada diagram alir yang Gambar 8.

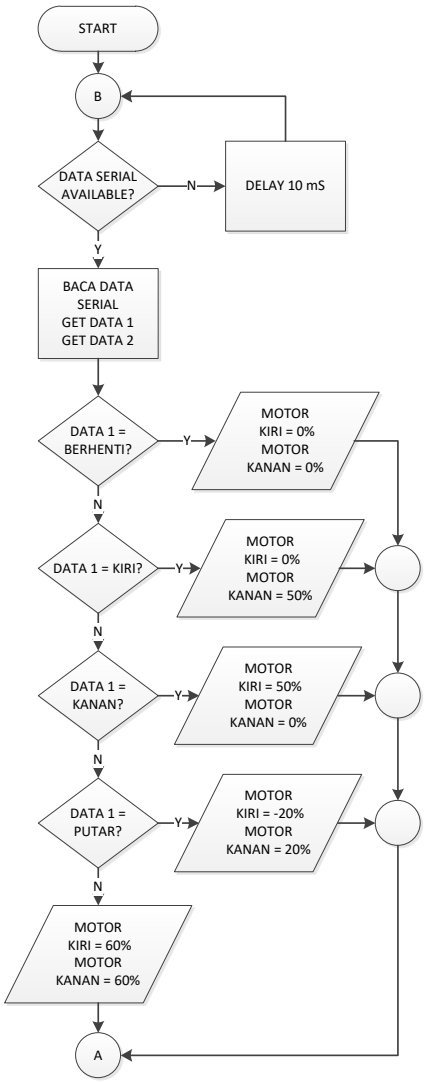


Gambar 8. Diagram alir algoritma *master*.

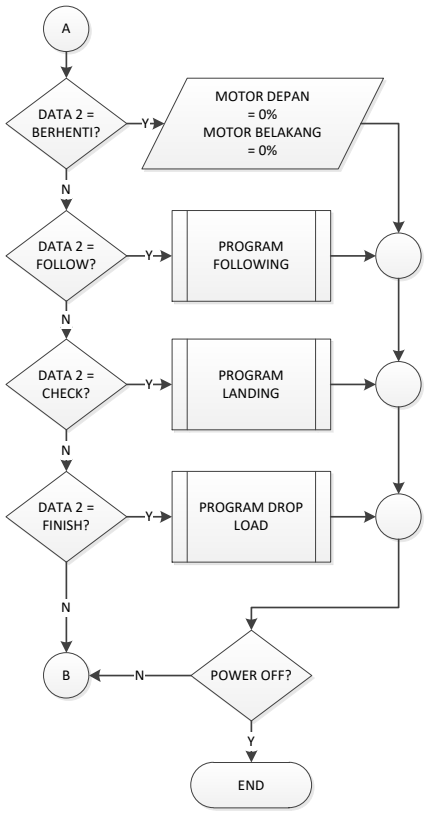
Perintah *processor* utama kepada slave akan dikirimkan melalui komunikasi serial dengan data yang dikirimkan yang diberi nama “data 1” dan “data 2”. Data 1 memiliki informasi perintah berhenti, belok kiri, belok kanan, maju, dan putar. Data ini untuk mengontrol aksi pergerakan motor yang berada pada sisi kiri dan sisi kanan robot. Data 2 memiliki informasi perintah berhenti, follow, check, dan finish. Perintah data 2 untuk mengontrol motor pada sisi depan dan belakang robot. Penjelasan perintah data 1 dan data 2 adalah sebagai berikut:

Data 1:
Berhenti: Motor kiri dan kanan berhenti berputar.
Kiri : Motor kiri berputar lebih lambat dari kanan.
Kanan : Motor kanan berputar lebih lambat dari motor kiri.
Maju : Motor kiri dan kanan berputar kecepatan sama.
Putar : Motor kanan akan berputar dan motor kiri diam.
Data 2:
Berhenti : Motor depan dan belakang berhenti berputar.
Follow : Motor depan, belakang menjaga ketinggian robot.
Check : Motor depan, belakang berhenti, aktifkan magnet.
Finish : Motor depan, belakang berhenti, matikan magnet

Proses yang akan ditangani oleh algoritma *slave* adalah mengontrol kecepatan motor pada robot menggunakan sinyal PWM, dan pembacaan nilai keluaran analog dari sensor tekanan. Gambar 9 menunjukkan diagram alir dari algoritma *slave* yang terdapat dalam mikrokontroler Arduino Uno.



Gambar 9. Diagram alir algoritma *slave.*



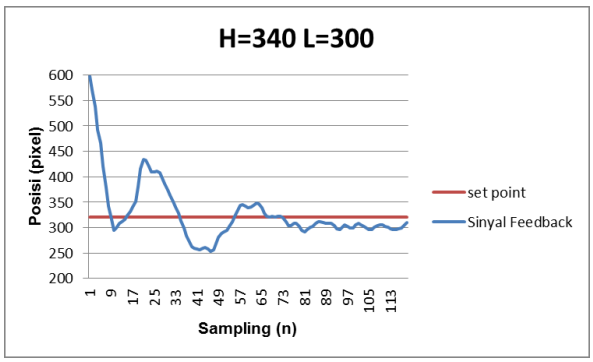
Gambar 9. (lanjutan) Diagram alir algoritma *slave.*

IV. PERFORMANSI ROBOT

Pengujian kontrol robot underwater ini dilakukan dengan tujuan mengamati performansi robot. Pengujian dilakukan pada pengaruh gap *on-off* histerisis, kecepatan robot, manuver robot, dan kontrol ketinggian robot menyelam.

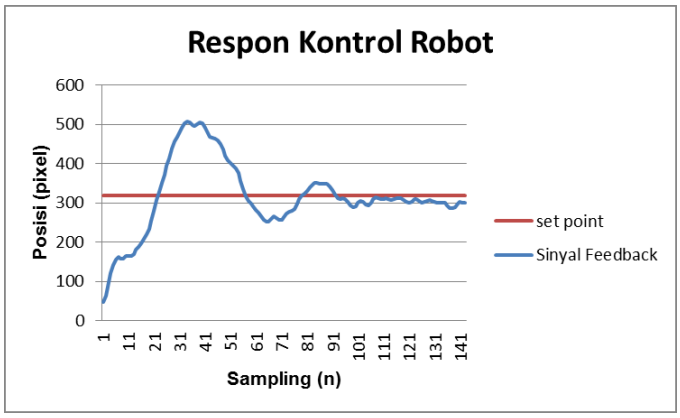
A. Pengujian Pengaruh Parameter Kontrol On-Off Histerisis

Pengujian pengaruh parameter kontrol *on-off* histerisis untuk mendapatkan besar nilai gap yang baik. Gap *on-off* batas atas dinyatakan dengan notasi H dan batas bawah dinyatakan dengan notasi L. Uji coba pada gap yang dinyatakan dalam pixel sebesar 20, 40, 60, 80, dan 100, menunjukkan gap 40 pixel memberikan performansi terbaik pada Gambar 10, dengan frekuensi osilasi relatif lebih kecil, dan amplitudo osilasi yang tidak terlalu besar.



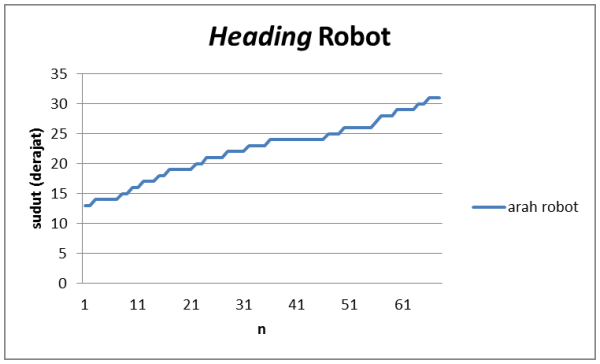
Gambar 10. Respon robot dengan gap 40 pixel.

Pengujian kecepatan robot dilakukan untuk mengetahui kecepatan maksimum yang dapat dicapai robot dan masih dalam keadaan stabil. Pengujian akan dilakukan dengan lebar gap yang telah diuji yaitu 300-340 pixel dan dengan *duty cycle* sebesar yaitu 30%, 40%, 50%, 60%, 70%, dan 80%. Gambar 11 menunjukkan pengujian respon kontrol robot dengan duty cycle sebesar 60% adalah yang maksimum, dengan tanpa banyak berosilasi.

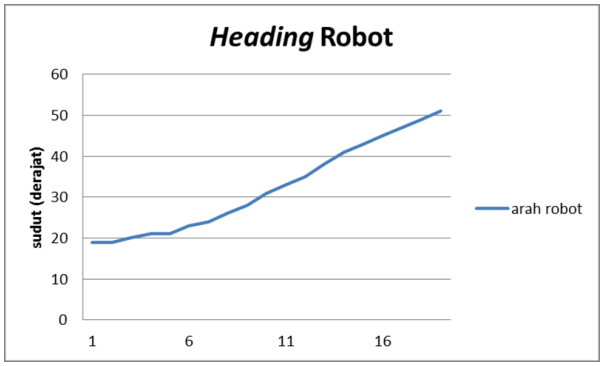


Gambar 11. Respon robot dengan duty cycle 60%.

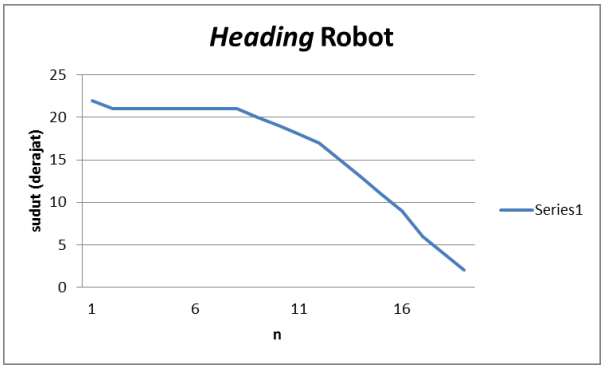
Pengujian manuver robot untuk melihat respon pergerakan robot ketika diberikan perintah maju, kiri, dan kanan. Data yang diambil adalah data dengan konfigurasi lebar gap kontrol *on-off* histerisis 40 pixel dan duty cycle 60%. Gambar 12 ditunjukkan grafik respon robot ketika diberikan perintah maju. Gambar 13 untuk grafik respon robot diberi perintah ke kanan. Gambar 14 untuk grafik respon robot diberi perintah ke kiri.



Gambar 12. Grafik *heading* robot dengan perintah maju.

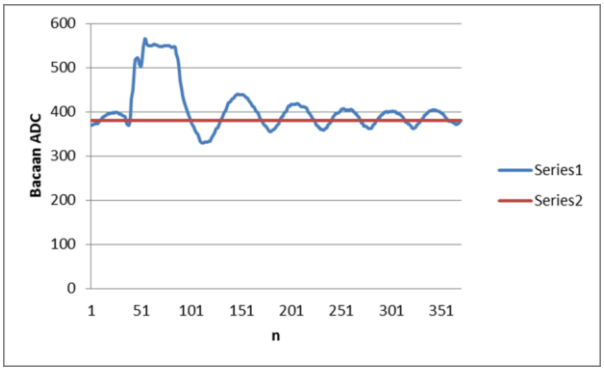


Gambar 13. Grafik *heading* robot dengan perintah ke kanan.



Gambar 14. Grafik *heading* robot dengan perintah ke kiri.

Pengujian untuk mengetahui respon sistem kontrol kedalaman menyelam robot. *Set point* kedalaman menyelam dari robot adalah sebesar 13 cm dengan pembacaan ADC sebesar 380. Gambar 15 menunjukkan respon robot terhadap kedalaman menyelam. Robot bergerak mengejar set point pada saat pertama robot dilepaskan dan terdapat osilasi pada sistem. Osilasi pada respon sistem ini masih terjadi walaupun sistem telah mencapai keadaan *steady*. Osilasi ini dapat diabaikan karena perubahan ketinggian menyelam pada robot tidak banyak.



Gambar 15. Grafik respon robot terhadap kedalaman menyelam.

Pengujian untuk mengetahui keberhasilan robot menyelesaikan misi keseluruhan. Misi yang akan dilakukan adalah pertama-tama berjalan mengikuti garis dari salah satu sisi ujung arena, kedua melakukan proses cek kargo, kemudian ketiga melakukan pendaratan di atas kargo, setelah itu kembali mengikuti garis hingga ke sisi ujung arena saat pertama kali

robot mulai mengikuti garis. Hasil ujicoba untuk misi secara lengkap dari lima percobaan belum dapat diperoleh misi berhasil secara utuh.

V. SIMPULAN DAN SARAN

A. Simpulan

Dengan memperhatikan data pengamatan dan analisis pada bab sebelumnya, dapat disimpulkan bahwa:

1. Pada robot underwater autonomous untuk menjejak garis dengan metoda thresholding dan moments telah berhasil direalisasikan.

2. Dari data pengujian data sensor IMU memperlihatkan bahwa robot tidak dapat memindahkan barang karena interferensi medan magnet yang ditimbulkan magnet pengambil mempengaruhi sensor ketika diaktifkan.

3. Dari pengujian deteksi objek menggunakan kamera dengan menggunakan pemrosesan citra yang telah dirancang memperlihatkan bahwa pendeteksian objek berhasil direalisasikan dengan benar.

4. Dari pengujian kontrol robot didapatkan bahwa kecepatan maksimum yang dapat dikontrol robot adalah sebesar 60% dengan lebar gap pengontrol *on-off* histerisis sebesar 40 pixel agar robot tetap bergerak dengan stabil

B. Saran

Saran-saran yang dapat diberikan untuk perbaikan dan pengembangan dari penelitian ini adalah sebagai berikut:

1. Sensor yang digunakan untuk mengukur ketinggian robot menyelam diganti dengan yang lebih baik karena sensor yang digunakan sekarang berpengaruh terhadap suhu yang ditimbulkan motor driver.

2. Ditambahkan aktuator atau sensor lainnya agar robot dapat mencari garis ketika garis tidak berada dalam jangkauan penglihatan sesnsor kamera robot.

3. Metoda pemindahan barang dari kargo ke sisi ujung arena dapat dikembangkan lagi agar tidak terjadi interferensi ke sensor IMU.

4. Motor penggerak yang digunakan lebih baik sehingga pergerakan robot dapat lebih stabil karena dapat diatur berputar dalam kecepatan rendah.

REFERENSI

[1] Singapore Robotic Games 2015. “Robot Underwater Rules”

[2] Monk, Simon. 2013. “Raspberry Pi Cookbook”. United States of America: O’Relly Media.

[3] Aplicaciones de la Visión Artificial. 2015. RaspiCam C++. http://www.uco.es/investiga/grupos/ava/node/40

[4] Arduino. 2015. Arduino Uno & Genuino Uno. https://www.arduino.cc/en/Main/arduinoBoardUno

[5] Bradski, Gary dan Kaebler , Adrian. 2008. “Learning OpenCV”. United States of America: O’Relly Media.

[6] Datasheet “The OpenCV Reference Manual”, Itseez.

[7] R. Fisher, S. Perkins, A. Walker and E. Wolfart. 2003. Gaussian Smoothing. http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm

[8] Opencv Dev Team. 2015 Basic Thresholding Operations. http://docs.opencv.org/doc/tutorials/imgproc/threshold/threshold.html

Application of Navigation Compass Android on Robot Soccer Based Arduino

Penerapan Navigasi Kompas Android Pada Robot Sepak Bola Beroda Berbasis Arduino

Muhammad Syahdan Maulana*, Muhammad Ainul Fikri, dan Aries Triwibowo
Universitas Negeri Semarang
Semarang, Indonesia
msyahdan608@gmail.com

***Abstrak*—Kemajuan teknologi di bidang robotika semakin pesat salah satunya pada acara yang diadakan setiap tahun oleh Direktorat Pendidikan Tinggi yaitu Kontes Robot Indonesia. Pada Tahun 2016, Kontes Robot Indonesia membuka divisi robot baru yaitu divisi Eksibisi Robot Sepak Bola Beroda (ERSB), dimana persyaratan untuk ikut serta dalam pertandingan robot tersebut adalah robot mampu mengejar objek bola dengan memanfaatkan kamera dan robot dapat menendang serta persyaratan dimensi mekanik yang telah di tentukan oleh pihak juri. Pada penelitian ini bertujuan untuk membuat robot sepak bola beroda yang memanfaatkan arduino sebagai mikrokontroler dan kamera android sebagai sistem diteksi serta penerapan kompas sebagai arah navigasi tendangan. Pengujian secara global terbagi menjadi 4 yaitu, pengujian sistem scanning, pengujian PWM dan pengujian akurasi tendangan terhadap sasaran navigasi kompas yang mengarah pada gawang lawan serta pengujian kelayakan oleh pakar ahli dan uji mekanik pada KRI 2016. Hasil pengujian kelayakan sistem oleh pakar ahli menunjukan nilai kelayakan sebesar 91.6 % pada desain mekanik dan elektronik, 83.3 % pada sistem deteksi objek dan navigasi dan 91.6 % pada tingkat kemudahan pengoprasian dan manfaat. Robot akan lebih akurat dalam proses navigasinya apabila menerapkan GPS yang dapat dengan kompas.**

Kata kunci—Android; GPS; Navigasi

I. PENDAHULUAN

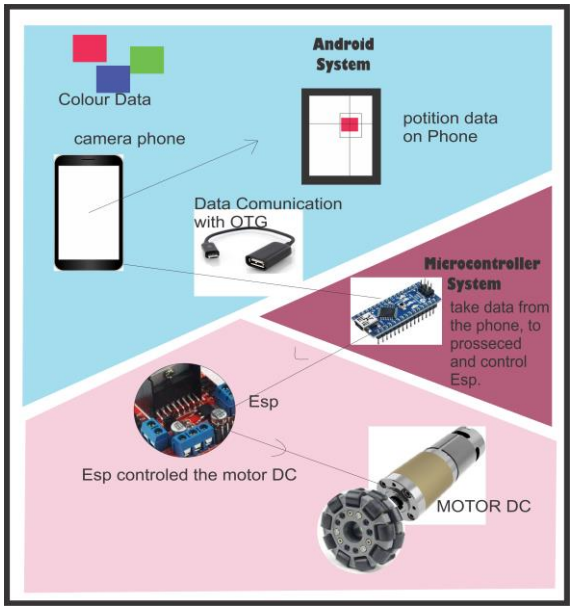
Kontes Robot Indonesia (KRI) adalah kegiatan kontes bidang robotika yang dapat diikuti tim mahasiswa dari institusi atau Perguruan Tinggi Negeri dan Swasta yang terdaftar di Kementerian Riset, Teknologi dan Pendidikan Tinggi yang dilakasanakan rutin setiap tahun. Pada Kontes Robot Indonesia (KRI) di tahun 2015 terdapat berbagai divisi robot, yaitu (1) Kontes Robot ABU Indonesia (KRAI), (2) Kontes Robot Pemadam Api Indonesia (KRPAI) tipe beroda, (3) Kontes Robot Pemadam Api Indonesia (KRPAI) tipe berkaki, (4) Kontes Robot Seni Tari Indonesia (KRSTI) (5) Kontes Robot Sepak Bola Indonesia (KRSBI). Khusus divisi KRSBI menggunakan tipe berkaki atau berbentuk manusia, salah satu permasalahan pada robot KRSBI yang terjadi saat pertandingan berlangsung adalah robot belum bisa membedakan arah gawang lawan sehingga menyebabkan tandangan robot tidak mengarah ke gawang lawan dengan tepat. Pada penelitian Robot Humanoid Pemain Bola [4] terdapat saran pengembangan pada penelitian tersebut yaitu, Dibutuhkan metode untuk mendeteksi posisi tengah dari gawang sehingga robot dapat lebih tepat menentukan arah gerak misalnya dengan bantuan sensor kompas. Pada KRI ditahun 2016 membuka divisi baru yaitu divisi ERSB (Eksibisi Robot Sepak Bola Beroda), dimana pada divisi ini menggunakan robot tipe beroda. Aturan umum robot yang harus dipenuhi yaitu masing-masing harus mempunyai kemampuan untuk mendeteksi, mengejar, dan menendang bola ke arah gawang lawan secara otomatis tanpa bantuan operator. Persyaratan robot yang harus dipenuhi yaitu robot harus memiliki kamera untuk sensor penglihatan objek bola, ukuran dimensi robot maksimum diameter 40 cm dan tinggi 30 cm. Spesifikasi lapangan sepakbola adalah berkarpet hijau dengan ukuran 6 x 4 meter, menggunakan 2 gawang dengan ukuran 40 x 80 cm dan menggunakan bola tenis berwarna jingga. Berdasarkan permasalahan yang terdapat pada divisi KRSBI dan telah dibukanya divisi baru ERSB, maka perlu dirancang robot sepak bola dengan penambahan sistem yaitu dengan menerapkan navigasi kompas berbasis Android dan robot sepak bola beroda sesuai aturan ERSB 2016 serta berbasis mikrokontroler Arduino.

II. PERANCANGAN SISTEM

A. Diagram Sistem

Diagram sistem secara umum, dapat dilihat bahwa perancangan sistem untuk pembuatan sistem ini melibatkan dua *Environtment* yakni *Environtment* Android dan *Environtment* Mikrokontroler. Kedua *Environtment* ini diharuskan bekerja secara sinkron agar sistem yang dibangun dapat berjalan dengan semestinya. *Environtment* Android disini difungsikan sebagai Sensor dan Sistem yang berfungsi mengkalkulasi Arah dan Kecepatan untuk mobile robot, sedangkan *Environtment* Mikrokontroler digunakan untuk mendapatkan data dan mengkonversikannya ke dalam sinyal yang dapat dimengerti oleh motor servo dan ESC (*Electronics Speed Control*). Untuk mengkomunikasikan dua *Environtment* ini, maka diperlukan sebuah protokol komunikasi yang dapat membuat sistem Android dapat mengirimkan data pada mikrokontroler. Pada tugas akhir ini digunakan protokol komunikasi *microbridge* yang memanfaatkan mode ADB pada sistem *Android*. Dikarenakan sistem yang dirancang mengharuskan untuk mengembangkan sistem untuk dua *Environtment*, maka Perancangan sistem secara detail akan

dibagi menjadi dua, yakni perancangan Sistem *Android* dan perancangan Sistem Mikrokontroler



Gambar 1. Diagram perancangan sistem.

B. *Perancangan Sistem Android*

Perancangan sistem *Android* secara Umum, Perancangan Sistem *Android* dapat dikategorikan menjadi 2 Sub bab yakni Perancangan Sistem Untuk Akses Sensor (GPS, Kompas) dan Perancangan Sistem untuk Komunikasi Perancangan Sistem Akses Sensor meliputi sensor apa saja yang akan digunakan, bagaimana pengambilan datanya, bagaimana pengolahan data mentahnya hingga data bisa dimanfaatkan.

1) *Perancangan Akses Sensor*

Perancangan akses sensor untuk navigasi Untuk dapat digunakan sebagai alat navigasi, maka sensor GPS dan kompas dari android harus diambil dan dikalkulasikan untuk mendapatkan nilai kemudi dan nilai kecepatan untuk *mobile robot.* Algoritma yang digunakan untuk menentukan arah *steering* dan *heading*, dapat dilihat bahwa *output* hasil kemudi dibatasi pada 3 keadaan yakni lurus, belok kanan, dan belok kiri. Sedangkan kecepatan juga dibatasi pada 3 state, yakni berhenti, kecepatan rendah dan kecepatan sedang. Pada tugas akhir ini arah lurus pada kemudi direpresentasikan dengan 90° pada motor servo sedangkan belok kanan dan kiri berturut-turut 70° dan 110° . Sedangkan kecepatan *mobile robot* yang digunakan adalah 5% dari kecepatan maksimal motor untuk kecepatan rendah dan 6% dari kecepatan maksimal motor untuk kecepatan tinggi. Data *steering* dan *speed* yang dikirimkan untuk mewakili arah dan kecepatan merupakan data yang berupa derajat servo. Hal ini digunakan untuk mempermudah mikrokontroler untuk mengkonversi menjadi sinyal kontrol untuk ESC dan motor servo. *Abbreviations and Acronyms.*

2) *Perancangan Komunikasi USB*

Perancangan komunikasi USB merupakan bagian yang berfungsi mengirimkan data setelah *device android* telah Units mendapatkan data *steering* dan *speed* hasil pengolahan data GPS dan kompas oleh *device android*. Karena data yang dikirimkan ada dua jenis, yakni data kemudi dan data kecepatan, maka data yang dikirimkan akan dibentuk sebuah *array* yang berurutan. Tujuan penggunaan array ini adalah agar data dapat diterima dengan baik oleh mikrokontroler sehingga tidak terjadi kesalahan penerimaan data atau data yang diterima menjadi terbalik.

C. *Perancangan Sistem Mikrokontroler*

Perancangan sistem Mikrokontroler sistem yang akan dirancang pada mikrokontroler adalah meliputi bagaimana perancangan *hardware* dan *software*. Perancangan *Hardware* yang dilakukan adalah meliputi bagaimana pemilihan *board* mikrokontroler yang digunakan dan bagaimana relevansinya dengan *final project* yang dibangun saat ini. Sedangkan pada bagian perancangan *software*, yang dilakukan adalah proses perancangan untuk pemrograman dari board yang bersangkutan. Secara umum, fungsi sistem mikrokontroler ini adalah untuk mengkonversikan data hasil kalkulasi dari *device Android* menjadi perintah yang dapat mengontrol *mobile robot,* dalam hal ini sinyal perintah motor servo dan ESC. Untuk dapat berkomunikasi dengan USB (Android), maka mikrokontroler harus menyesuaikan sistem kedalam sistem *logic* USB. Untuk itu, pada tugas akhir ini digunakan *board* mikrokontroler *Seeduino* ADK telah dilengkapi IC MAX3421E sebagai konverter dari USB menjadi *logic* yang dapat dimengerti oleh mikrokontroler. Komunikasi yang dilakukan antara *board minimum system* dengan IC MAX3421 memanfaatkan komunikasi SPI (*Serial Peripheral Interface*).

Untuk dapat memberikan sinyal kontrol pada motor servo dan ESC *mobile robot*, maka perlu dilakukan perancangan hardware yang sesuai. Karena ESC pada *mobile robot* telah memiliki *power supply* yang independen, maka sistem ESC pada *mobile robot* tidak perlu diberikan daya dari luar. Sedangkan pin VCC yang berada pada ESC mengeluarkan tegangan 6 V konstan yang dapat digunakan untuk mensupply motor servo kemudi. Dengan menggunakan desain tersebut, maka motor servo akan mengambil daya dari baterai utama *mobile robot* dan bukan dari mikrokontroler.

III. PENGUJIAN DAN ANALISIS

Pada bagian ini akan dilakukan pengujian navigasi menggunakan sistem yang dibangun. Pengujian yang dilakukan meliputi pengujian scanning bola, pengujian laju bola dan pengujian akurasi tendangan.

A. *Pengujian Scanning Bola*

Pengujian *scanning* bola di lakukan secara berulang dengan cara menyentuh layar android pada aplikasi android yang telah dibuat, dengan mengakses kamera pada android agar menampilkan objek tepat di depan robot dan kalibrasi

objek bola dilakukan dengan menyentuh layar dengan kamera yang telah di arahkan pada objek bola. Pengkalialibrasi objek bola dapat dilihat pada Tabel 1 berikut ini.

TABEL I. PENGUJIAN *SCANNING* BOLA

No	Kondisi	Gambar
1.	Saat sebelum menyentuh layar, deteksi warna belum mendeteksi warna objek apapun.	
2.	Saat menyentuh layar untuk deteksi warna objek bola.	
3.	Setelah menyentuh objek bola pada layar android, maka pada program deteksi akan memunculkan garis kotak deteksi pada objek bola.	

Dari hasil pengujian scanning warna pada objek bola pada tabel 1 menghasilkan deteksi pada warna jingga pada bola tersebut, sistem deteksi pada program scanning dapat mendeteksi warna orange karena pada sistem deketsi mengambil data warna dari kombinasi warna merah, hijau dan biru dan putih yang masing - masing memiliki nilai sebagai berikut Merah (253), Hijau (178), Biru (104), Putih (255), dimana data kombinasi warna yang di fungsikan sebagai sistem deteksi ini sudah di simpan pada program android setelah menyentuh layar android saat proses kalibrasi objek bola.

B. *Pengujian Laju Robot*

Pengujian laju robot dilakukan dengan menetukan nilai PWM dari setiap roda pada robot, untuk membuat laju robot maju, mundur, kiri, kanan dan rotasi, di perlukan kombinasi nilai PWM dan kombinasi arah putaran setiap roda pada robot yang menggunakan 3 roda dengan susunan dasar segitiga sama sisi. kombinasi PWM dan arah tiap roda yang di tentukan pada nilai kutub + atau – pada tiap motor. Pengujian laju robot sudah dapat berjalan 8 arah dan 8 arah tersebut di tentukan pada nilai PWM yang di tuliskan pada program Arduino. Berikut ini adalah salah satu penulisan program laju maju robot pada script program PWM. Setiap arah laju robot memiliki nilai PWM dan arah tertentu pada setiap 1 roda. Kombinasi nilai PWM dan pada setiap masing masing roda akan dapat menghasilkan laju yang berbeda-beda. Hasil kombinasi nilai PWM dari tiap roda terdapat pada tabel 2 berikut.

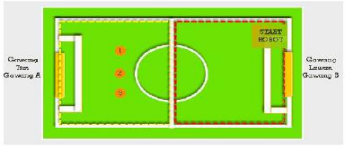
TABEL II. KOMBINASI PWM LAJU ROBOT

No.	Arah laju Robot	Nilai PWM
1.	Maju	rodaka + = LOW rodaka - = 160 rodate + = LOW rodate - = LOW rodaki + = 160 rodaki - = LOW
2.	Kanan	rodaka + = LOW rodaka - = 106 rodate + = LOW rodate - = 194 rodaki + = LOW rodaki - =106
3.	Kiri	rodaka + = 100 rodaka - = LOW rodate + = 170 rodate - = LOW rodaki + = 100 rodaki - = LOW
4.	Mundur	rodaka + = 160 rodaka - = LOW rodate + = LOW rodate - = LOW rodaki + = LOW rodaki - =160
5.	Rotasi Kiri	rodaka + = LOW rodaka - = 70 rodate + = 70 rodate - = LOW rodaki + = LOW rodaki - = 70
6.	Rotasi Kanan	rodaka + = 70 rodaka - = LOW rodate + = LOW rodate - = 70 rodaki + = 70 rodaki - = LOW
7.	Itari Kanan	rodaka + = LOW rodaka - = LOW rodate + = LOW rodate - = 130 rodaki + = LOW rodaki - = LOW
8.	Itari Kiri	rodaka + = LOW rodaka - = LOW rodate + = 130 rodate - = LOW rodaki + = LOW rodaki - = LOW

C. *Pengujian Akurasi Tendangan Berdasarkan Navigasi Kompas*

Pengujian akurasi tendangan berdasarkan navigasi kompas dilakukan dengan cara melakukan pengambilan data uji coba akurasi tendangan robot berdasarkan nilai *range* derajat kompas yang telah ditentukan pada program robot. Pada percobaan berikut akurasi *range* nilai kompas dengan sasaran gawang arah barat atau sasaran gawang A dengan melakukan uji tendangan pada peletakan bola sesuai letak bola 1, letak bola 2 dan letak bola 3. Setiap arah laju robot memiliki nilai PWM dan arah tertentu pada setiap 1 roda. Kombinasi nilai PWM dan pada setiap masing masing roda akan dapat menghasilkan laju yang berbeda - beda. Hasil kombinasi nilai PWM dari tiap roda terdapat pada tabel 3 berikut.

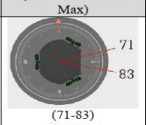
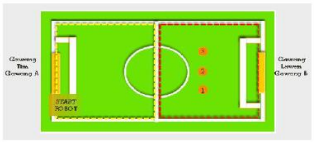
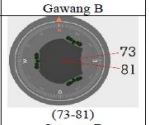
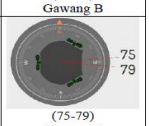
TABEL III. PENGUJIAN AKURASI TENDANGAN DENGAN SASARAN GAWANG A

No.	Nilai *Range* Kompas (Nilai Min – Nilai Max)	Lapangan Pengujian Tendangan Robot meliputi letak bola 1,2,3 serta *Start* ROBot
1.	251 263 (251-263) Gawang A	
2.	253 261 (253-261) Gawang A	
3.	255 259 (255-259) Gawang A	

Pada percobaan selanjutnya *range* nilai derajat kompas yang ditentukan adalah 71-83, 73-81, 75-79 dimana pada 3 nilai *range* tersebut posisi arah gawang lawan berada pada arah timur atau sasaran gawang B dengan melakukan uji

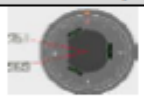
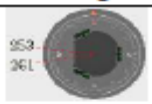
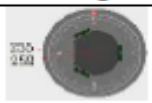
tendangan pada peletaan bola sesuai letak bola 1, letak bola 2 dan letak bola 3.

TABEL IV. PENGUJIAN AKURASI TENDANGAN DENGAN SASARAN GAWANG B

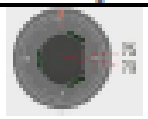
No.	Nilai *Range* Kompas (Nilai Min – Nilai Max)	Lapangan Pengujian Tendangan Robot meliputi letak bola 1,2,3 serta *Start* RObot
1.	 71 83 (71-83) Gawang B	
2.	 73 81 (73-81) Gawang B	
3.	 75 79 (75-79) Gawang B	

Pengujian akurasi tendangan robot berdasarkan nilai kompas di lakukan dengan melakukan 18 kali pengujian tendangan, hasil dalam pengujian tersebut dapat terlihat akurasi *range* yang tepat pada program robot sepak bola berdasarkan banyaknya hasil gol pada settingan nilai kompas yang di tentukan pada pengujian. Hasil gol dapat dilihat pada tabel 5 dan tabel 6 berikut ini.

TABEL V. HASIL PENGUJIAN AKURASI TENDANGAN DENGAN SASARAN GAWANG A

No.	Nilai *Range* Kompas (Nilai Min – Nilai Max)	Percobaan		
		1	2	3
1.	 (251-263) Gawang A	TIDAK GOL	GOL	TIDAK GOL
2.	 (253-261) Gawang A	GOL	GOL	TIDAK GOL
3.	 (255-259) Gawang A	GOL	GOL	GOL

TABEL VI. HASIL PENGUJIAN AKURASI TENDANGAN DENGAN SASARAN GAWANG B

	Nilai Max)			
1.	 (71-83) Gawang B	TIDAK GOL	GOL	TIDAK GOL
2.	 (73-81) Gawang B	TIDAK GOL	GOL	GOL
3.	 (75-79) Gawang B	GOL	GOL	GOL

Hasil analaisi pembahasan dari pengujian diatas adalah seperti berikut :

1. Akurasi Tendangan robot berdasarkan range kompas dengan sasaran gawang A Berdasarkan tabel 4.5 dapat di peroleh hasil akurasi gol dengan sasaran gawang A pada range kompas 251-263 dengan interval 12 di lakukan pengujian pada 3 titik lekta bola dan di peroleh gol sebanyak 1 kali sehingga presentase akurasi gol sebesar 33,3 %. Percobaan selanjutnya pada range kompas 253-261 dengan interval 8 di lakukan pengujian pada 3 titik lekta bola dan di peroleh gol sebanyak 2 sehingga presentase akurasi gol sebesar 66,6 %. Percobaan akhir pada range kompas 255-259 dengan interval 4 di lakukan pengujian pada 3 titik letak bola dan di peroleh gol sebanyak 3 sehingga presentase akurasi gol sebesar 100 %.
2. Akurasi Tendangan robot berdasarkan *range* kompas dengan sasaran gawang B Berdasarkan tabel 4.6 dapat di peroleh hasil akurasi gol dengan sasaran gawang A pada range kompas 71-83 dengan interval 12 di lakukan pengujian pada 3 titik lekta bola dan di peroleh gol sebanyak 1 sehingga presentase akurasi gol sebesar 33,3 %. Percobaan selanjutnya pada range kompas 73-81 dengan interval 8 di lakukan pengujian pada 3 titik lekta bola dan di peroleh gol sebanyak 2 sehingga presentase akurasi gol sebesar 66,6 %. Percobaan akhir pada range kompas 75-79 dengan interval 4 di lakukan pengujian pada 3 titik letak bola dan di peroleh gol sebanyak 3 sehingga presentase akurasi gol sebesar 100 %.

Dari uraian hasil jumlah gol berdasarkan nilai*range* kompas, dapat di simpulkan bahwa semakin kecil nilai *range* kompas maka tingkat akurasi gol akan semakin besar dan jika semakin besar penentuan nilai *range* kompas maka hasil akurasi gol akan semakin kecil.

IV. KESIMPULAN

Berdasarkan hasil penelitian penerapan navigasi kompas android pada robot sepak bola beroda berbasis arduino dan pengujian robot sepak bola beroda dapat diambil beberapa kesimpulan sebagai berikut :

1. Penerapan akses kompas android pada program android robot sepak bola dapat mendetaksi arah gawang lawan dan menendang sesuai arah gawang lawan dengan range nilai interval kompas android yang telah ditentukan.
2. Penerapan kamera android dan penggunaan *library* openCV pada program android, robot sepak bola beroda dapat mendeteksi warna objek bola dan menjadikan objek utama scanning robot, sehingga robot dapat mengejar dan menendang bola dengan memanfaatkan sistem deteksi warna pada objek bola tersebut.
3. Berdasarkan hasil perancangan desain robot sepak bola beroda dengan tinggi 30 cm dan diameter 19 cm serta susunan komponen dapat menghemat tempat yang menjadikan berat robot 1,4 Kg, maka robot sepak bola beroda ini layak untuk bertanding di divisi ERSB pada KRI 2016 karena rancang mekanik sudah sesuai aturan divisi ERSB pada KRI 2016.

Saran – saran yang dapat diberikan untuk pengembangan alat ini sebagai berikut:

1. Penerapan deteksi objek bola akan lebih peka dalam proses scanning bola.
2. Robot akan lebih akurat dalam proses navigasinya apabila menerapkan GPS yang dapat kaliborasi dengan kompas.
3. Robot akan lebih presisi dalam laju robot apabila menerapkan motor stepper sebagai aktuatornya.
4. Sensor GPS dan Kompas saja belum dapat melakukansistem navigasi darat secara optimal, hal ini dikarenakan akurasi dari GPS itu sendiri maksimal hanya 2 meter, padahal lebar jalan dan medan yang harus dilalui kebanyakan kurang dari 2 meter. Untuk itu perlu ditambahkan *secondary sensor* untuk dapat membaca kondisi di sekitar *robot*, semisal sensor jarak *ultrasonic,* atau *infrared* untuk mendeteksi penghalang.
5. Algoritma untuk pengontrolan arah kemudi perlu diperbaiki dengan algoritma yang lebih baik , semisal PID, Fuzzy, ataupun *neural network* agar hasil pengontrolan lebih optimal.
6. *Smartphone Android* memiliki fitur lain yang bisa ditambahkan dan dieksplorasi kedalam sistem navigasi, semisal kamera, *Bluetooth,* WiFi, *Mobile Internet*, *Accelerometer*, *Gyroscope*, dll

REFERENSI

[1] Askolani, Imam. 2012. “Rancang Bangun Humman Interaction Berbasis Android Pada Robot Penerima Tamu”. *Skripsi*. Fakultas Teknik. Universitas Indonesia.

[2] Budiharto, W. 2015. *Robotika Modern*. Edisi Revisi. Cetakan Kedua. Andi Offset. Yogyakarta.K.

[3] Kadir, A. 2015. *Buku Pintar Pemrograman Arduino*. Edisi Pertama. Cetakan Pertama. MediaKom. Yogyakarta

[4] Muliady dkk, 2012. Robot Humanoid Pemain Bola. *Laporan Penelitian*. Universitas Kristen Maranatha. Bandung

[5] Rajaie, H., O. Zweigle, K. Haussermann, U.P. Kappeler, A. Tamke dan P. Levi. 2011. Hardware design and distributed embedded control architecture of a mobile soccer robot. *Jurnal Mechatronic Elsevier* (21): 455-468.

[6] Smyth, Neil. 2015. *Android Studio Development Essentials*. Edisi Kedua. eBookFrenzy.

[7] Sugiyono. 2009. *Metode Penelitian Kuantitatif Kualitatif dan R&D*. Bandung : ALFABETA.

[8] Triwibowo, Aries. 2017. Penerapan Navigasi Kompas Android Pada Robot Sepak Bola beroda Berbasis Arduino. Skripsi. Fakultas Teknik. Universitas Negeri Semarang.

Sistem Deteksi Bola Menggunakan Pustaka OPENCV dengan Kombinasi Dua Kamera (Logitech 720p & Kinnect XBOX 360) di Robot Sepak Bola Beroda

Deny Prasetyo*, Zainal Abidin
Studi Otomasi dan Kendali Trunojoyo (SOKET)
Universitas Trunojoyo Madura
Bangkalan, Indonesia
denyprasetyo12345@gmail.com

***Abstrak*—Deteksi objek pada robot sepak bola merupakan suatu proses yang cukup kompleks untuk dilakukan oleh karena itu diperlukan penelitian tentang pengolahan citra digital. Pengolahan citra digital (*digital image processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Fungsi pengolah Citra adalah gambar yang berasal dari sensor vision (webcam) sebagai sistem pendeteksian dan pelacakan objek (bola). Pustaka yang digunakan untuk mengolah citra adalah *OPENCV* karena bersifat *open-source*. Dalam penelitian ini robot menggunakan dua kamera *logitech 720p* di kamera *omnivision* dan kamera Kinnect XBOX 360 di kamera depan. Hasil penelitian menunjukkan bahwa robot dapat koordinasi dua kamera dan mendeteksi bola.**

Kata kunci—robot soccer; pengolah citra; deteksi objek; kamera

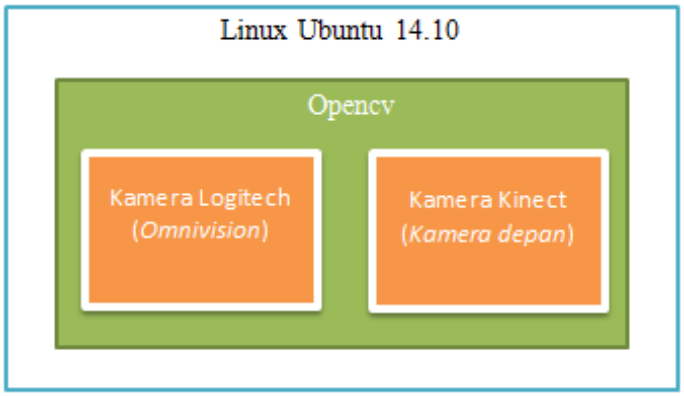
I. PENDAHULUAN

Perkembangan sistem *computer vision* saat ini telah banyak membantu manusia, contohnya sebagai pendeteksi sebuah objek untuk dapat diolah lebih lanjut seperti deteksi bola untuk robot sepak bola. *Computer vision* atau pengolahan citra digital adalah ilmu pemrograman computer untuk memproses dan akhirnya memahami gambar dan video, atau membuat computer dapat melihat [1]. Deteksi objek dalam pengolahan citra digital adalah suatu proses yang digunakan untuk menentukan keberadaan objek tertentu di dalam suatu citra digital [2]. Pengolahan citra digital menggunakan teknologi computer vision saat ini banyak digunakan sebagai obyek penelitian. Bagian dari pengolahan citra adalah dengan menggunakan pengolahan berdasarkan warna. Analisis warna dalam pengenalan citra digital ini ada beberapa model diantaranya, model RGB, CMY, HSI, HSV dan normalized RGB [3-4]. Dalam penelitian ini robot dipasangi dengan dua kamera *logitech 720p* di kamera *omnivision* dan kamera Kinnect XBOX 360 di kamera depan. Kamera *logitech* adalah kamera standart untuk webcamp, dan kamera *kinect* adalah *controller-free gaming* dan pengalaman hiburan oleh Microsoft dan Xbox 360 video game platform, dan mungkin digunakan untuk Windows 8. Perangkat ini memiliki kamera RGB, sensor kedalaman dan mikrofon yang berjalan di perangkat software khusus, yang menyediakan kemampuan untuk menangkap gerak secara 3D, mengenali wajah dan mengenali suara [5]. Harapannya bahwa dengan penelitian ini akan mampu membuat robot mendeteksi bola dengan lebih akurat.

II. METODE PENELITIAN

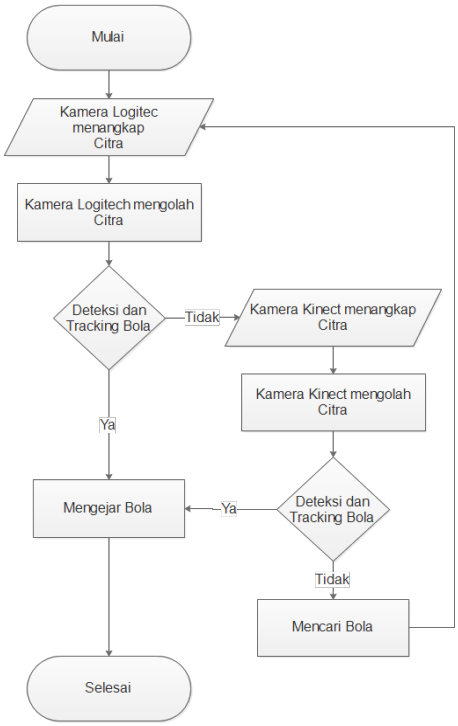
A. Analisis Sistem

Sistem yang akan dirancang untuk penelitian ini merupakan suatu sistem pendeteksian dan tracking bola berbasis pengolahan citra dengan pustaka *OpenCV*. Rancangan sistem ditunjukkan pada diagram blok rancangan sistem pada Gambar 1.



Gambar 1. Diagram blok rancangan system.

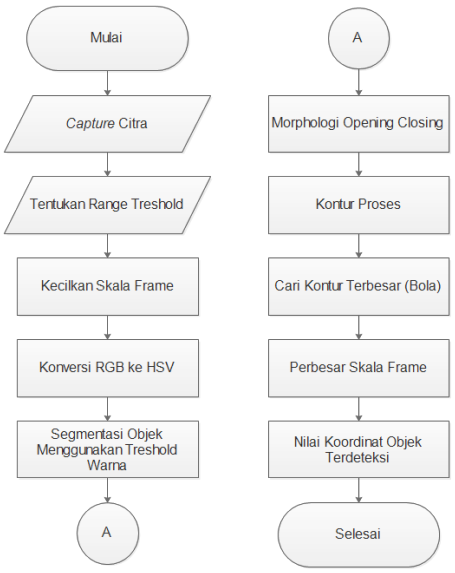
Ketika robot dihidupkan maka yang pertamakali di jalankan adalah program kamera logitech sebagai kamera *omnivision* jika dari kamera ini mendeteksi adanya bola maka robot akan langsung menjalankan proses kejar bola, jika tidak mendeteksi adanya bola maka jalankan proses *kinect*. Proses *kinect* digunakan saat bola berada benar-benar jauh dan tidak ada dalam jangkauan kamera logitech. Jika bola terdeteksi oleh kamera *kinect* maka jalankan proses kejar bola *kinect*, jika bola berhasil didekati oleh proses kamera kinect dan kemudian terdeteksi kamera logitech maka jalankan proses kejar bola kamera logitech , jika tidak maka kembali ke proses awal program dijalankan.



Gambar 2. Flowchart alur system.

B. *Rancangan Pengolahan Citra Digital (Deteksi Bola)*

Perancangan pengolahan citra digital meliputi algoritma mendeteksi dan tracking objek bola menggunakan software code editor pada laptop dengan sistem operasi Linux Ubuntu 14.10 serta pustaka *OpenCV*. *OpenCV* adalah pustaka *open-source*, yang secara luas digunakan dalam aplikasi pengolahan citra untuk mendapatkan pengakuan instan untuk aplikasi tertanam [6].

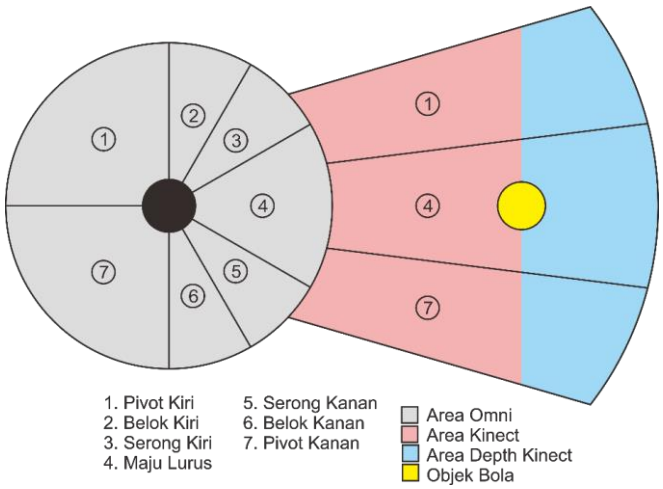


Gambar 3. *Flowchart* deteksi bola.

Proses *image processing* dimulai dari *capture* citra secara *real-time* dari kamera logitech dan kamera kinect, selanjutnya dilakukan penentuan range segmentasi warna HSV menggunakan metode optimasi ketetanggaan warna pada titik objek yang diklik. Untuk memperingan proses komputasi, skala frame diperkecil terlebih dahulu kemudian dilakukan konversi RGB ke HSV sebelum proses segmentasi citra bola. Untuk menghilangkan derau hasil segmentasi menggunakan metode morphologi opening, closing dan seleksi area hasil *contour*. Terakhir ukuran citra hasil segmentasi diperbesar sesuai skala awal, kemudian tentukan titik tengah dari objek sebagai acuan posisi dalam satuan (x, y) pixel.

C. *Rancangan Penyesuaian Posisi Robot terhadap Bola*

Rancangan yang akan dibuat untuk menyesuaikan posisi robot terhadap bola seperti gambar di bawah ini.



Gambar 4. Gambar area cangkupan 2 kamera.

Pada bagian frame kamera *logitech* (*omnivision*) dibagi menjadi 7 bagian untuk menentukan proses pergerakan robot terhadap bola pada area dekat. Sedangkan kamera kinect digunakan untuk mendeteksi bola pada area yang jauh (diluar jangkauan *omnivision*), pada bagian *frame kinect* pergerakan robot dibagi menjadi 3 bagian. Ketika bola tidak terdeteksi oleh kamera *logitech* dan *kinect* maka robot akan melakukan pergerakan pivot untuk mencari bola di sekeliling robot.

III. HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai hasil dari pengujian sistem serta pembahasannya. Secara garis besar dibagi menjadi dua bagian yaitu pengujian kemampuan sistem dalam melakukan pendeteksian bola serta pengujian keberhasilan sistem dalam melakukan penyesuaian posisi robot terhadap posisi bola.

A. Pengujian deteksi bola

1) Pengujian jarak bola

Pengujian deteksi bola bertujuan untuk menangkap gambar bola untuk diproses lebih lanjut agar dapat di *tracking* dengan menggunakan nilai penentuan range segmentasi warna HSV menggunakan metode optimasi ketetanggaan warna pada titik objek yang diklik. Pengujian ini dilakukan di kedua kamera secara bersamaan dan dengan menonaktifkan fungsi Depth di kamera *kinect*. Menurut Pamungkas dkk (2017:7), variasi untuk menguji kontur bola dengan jarak robot yang digunakan adalah 100 cm, 200 cm, 300 cm, 400 cm, 500 cm, 600 cm, 700 cm, 800 cm dan 900 cm [7].



Gambar 5. Uji coba pendeteksian jarak bola menggunakan kamera *kinect*.

TABEL I. DATA KINERJA PENDETEKSIAN JARAK BOLA DENGAN KONTUR

Jarak Robot ke Bola	Tingkat keberhasilan kontur *logitech* dalam pixel (px)	Tingkat keberhasilan kontur *logitech* dalam persen (%)	Tingkat keberhasilan kontur *kinect* dalam pixel (px)	Tingkat keberhasilan kontur *kinect* dalam persen (%)
100 cm	64 px	100 %	7500 px	100 %
200 cm	16 px	25 %	2500 px	33,3 %
300 cm	0 px	0 %	1300 px	17,3 %
400 cm	0 px	0 %	920 px	12,2 %
500 cm	0 px	0 %	710 px	9,4 %
600 cm	0 px	0 %	550 px	7,3 %
700 cm	0 px	0 %	480 px	6,4 %
800 cm	0 px	0 %	300 px	4 %
900 cm	0 px	0 %	220 px	2,9 %
Rata - rata	8,89 px	13,89 %	1608,89 px	21,45 %

Berdasarkan tabel diatas, dapat disimpulkan dari uji coba deteksi bola berdasarkan kontur menggunakan kamera *logitech* rata-ratanya adalah 13,89 %. Sedangkan dari uji coba deteksi bola berdasarkan kontur menggunakan kamera *kinect* rata-ratanya adalah 21,45 %. Terdapat perbedaan yang signifikan diantara keduanya dan jika dilihat dari persentase kamera *kinect* lebih unggul dalam ukuran pengambilan jarak bola.

2) Pengujian sudut bola

Dengan tujuan yang sama yaitu untuk menangkap gambar bola untuk diproses lebih lanjut agar dapat di *tracking* dan dengan menggunakan nilai penentuan range segmentasi warna HSV menggunakan metode optimasi ketetanggaan warna pada titik objek yang diklik, akan tetapi dalam uji coba kali ini yang diukur adalah jangkauan sudut yang dapat di ambil oleh dua kamera. Variasi untuk menguji sudut bola dengan kontur adalah jarak 100 cm, dan sudut uji 360 dibagi menjadi 4 bagian.



Gambar 6. Uji coba pendeteksian sudut bola menggunakan kamera *logitech*.

TABEL II. DATA KINERJA PENDETEKSIAN SUDUT BOLA DENGAN KONTUR

Sudut	Jangkauan kamera *logitech* terhadap bola	Jangkauan kamera *kinect* terhadap bola
0/360	ada bola	ada bola
90	ada bola	tidak ada bola
180	ada bola	tidak ada bola
270	ada bola	tidak ada bola

Berdasarkan tabel diatas, dapat disimpulkan dari uji coba deteksi sudut bola berdasarkan kontur menggunakan kamera *logitech* adalah 4 dari 4 sudut bola terdeteksi. Sedangkan dari uji coba deteksi sudut bola berdasarkan kontur menggunakan kamera *kinect* adalah hanya 1 dari 4 sudut terdeteksi . Terdapat perbedaan yang signifikan diantara keduanya dan jika dilihat dari hasil uji coba kamera *logitech* lebih unggul dalam ukuran pengambilan sudut bola.

B. Pengujian tracking bola dan posisi robot terhadap bola

Dengan berdasarkan data dari ujicoba sub-bab 1) dan 2), dengan hasil adalah kamera *kinect* unggul dalam ukuran jarak dan kamera *logitec* unggul dalam ukuran sudut, maka peneliti mencoba menggabungkan kedua kamera tersebut dala m satu program pengolah citra digital yaitu mendeteksi dan mentracking bola. Aplikasi IDE yang digunakan peneliti adalah Codeblocks dengan bahasa pemrograman c++. Algoritma yang digunakan adalah algoritma pemilihan

sederhana yaitu jika kamera *logitech* tidak menangkap bola maka robot akan berputar melakukan scanning menggunakan kamera *kinect*. Output yang diinginkan adalah area dari garis yang mengelilingi bola tidak sering hilang. Hasil dari uji coba dapat di lihat pada gambar-gambar di bawah ini:



Gambar 7. Keberhasilan pendeteksian menggunakan kombinasi *dual camera* (bola dalam area *logitech*) robot bergerak lurus pelan.

Pada Gambar 7 tampak terlihat bola masih dalam jangkauan kamera *logitech* tapi tidak dalam jangkauan kamera *kinect* dan aksi yang dikirimkan ke *microcontroller* adalah karakter "v" dan aksinya adalah robot akan lurus pelan. Pada kondisi seperti ini maka kamera *logitech* yang akan menjadi acuan dalam *tracking* bola.



Gambar 8. Keberhasilan pendeteksian menggunakan kombinasi *dual camera* (bola dalam area *kinect*) robot bergerak lurus cepat.

Pada Gambar 8 tampak terlihat bola masih dalam jangkauan kamera *logitech* tapi tidak dalam jangkauan kamera *kinect* dan aksi yang dikirimkan ke *microcontroller* adalah karakter "1" dan aksinya adalah robot akan lurus cepat. Pada kondisi seperti ini maka kamera *kinect* yang akan menjadi acuan dalam *tracking* bola. Akan tetapi jika masuk area kamera *logitech* maka akan menggunakan kondisi seperti gambar 7. Dari rangkaian percobaan diatas terbukti bahwa dapat mengurangi titik buta robot sepak bola beroda.

IV. HASIL DAN PEMBAHASAN

Berdasarkan penelitian yang dilakukan telah berhasil dibuat sistem pendeteksian dan *tracking* bola dengan metode *switching* dual kamera *kinect* dan *logitech*. Berdasarkan data yang didapat dan pembahasan yang dilakukan dapat diambil kesimpulan sebagai berikut:

1. Jarak maksimal yang dapat ditangkap oleh kamera *logitech* dalam menangkap bola adalah 200cm.
2. Jarak maksimal yang dapat ditangkap oleh kamera *kinect* dalam menangkap bola lebih dari 900cm.
3. Uji coba jarak mendeteksi bola berdasarkan kontur menggunakan kamera *logitech* rata-ratanya adalah 13,89 %. Sedangkan dari uji coba jarak mendeteksi bola berdasarkan kontur menggunakan kamera *kinect* rata-ratanya adalah 21,45 %. Terdapat perbedaan yang signifikan diantara keduanya dan jika dilihat dari persentase kamera *kinect* lebih unggul dalam ukuran pengambilan jarak bola.
4. Uji coba sudut deteksi bola berdasarkan kontur menggunakan kamera *logitech* adalah 4 dari 4 sudut bola terdeteksi. Uji coba sudut deteksi bola berdasarkan kontur menggunakan kamera *kinect* adalah hanya 1 dari 4 sudut terdeteksi. Terdapat perbedaan yang signifikan diantara keduanya dan jika dilihat dari hasil uji coba kamera *logitech* lebih unggul dalam ukuran pengambilan sudut bola.
5. Dari rangkaian percobaan diatas terbukti bahwa dapat mengurangi titik buta robot sepak bola beroda.

V. SARAN

Berikut adalah saran-saran dari peneliti untuk penelitian selanjutnya yang sejenis :

1. Metode pengolahan citra digital dapat dikembangkan dengan menggunakan metode pengenalan pola, bentuk atau dengan metode ekstraksi fitur agar objek yang dideteksi tidak hanya menggunakan warna.
2. Sensor Depth dalam kamera *kinnect* dapat di aktifkan untuk menghilangkan noise-noise yang tidak diperlukan dalam jarak yang ditentukan.
3. Jika perlu gunakan pengiriman data serial yang menggunakan data koordinat (x,y) agar pergerakan robot lebih tepat ke arah objek bola.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam mengerjakan penelitian ini.

REFERENSI

[1] Culjak, I., Abram, D., Pribanic, T., Dzapo, H. dan Cifrek, M., 2012, A brief introduction to OpenCV, MIPRO, 2012 Proceedings of the 35th International Convention, Croatia, May 21-25.

[2] Nagataries, D., Hardiristanto, S., Purnomo, M.H., 2012, Deteksi Objek pada Citra Digital Menggunakan Algoritma Genetika untuk Studi Kasus Sel Sabit.

[3] T. Sutoyo, E. Mulyanto, V. Suhartono, O.D. Nurhayati, Wijanarto, Teori Pengolahan Citra Digital, Penerbit Andi, Yogyakarta, 2009, p.256.
[4] F. Guo, Q. Cao, Proceedings of the 5th World Congress on Intelligent Control and Automation, Hangzhou, P.R. China, 2004.
[5]"Kinect", [Halaman Online] https://id.wikipedia.org/wiki/Kinect , Diakses 1 Juni 2017.
[6] Deepthi, R. dan Sankaraiah, S., 2011, Implementation of mobile platform using Qt and OpenCV for image processing applications, Open Systems (ICOS), 2011 IEEE [Online] 284–289. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6079235
[7] Pamungkas dkk. 2017, Sistem Pendeteksi dan Pelacakan Bola dengan Metode *Hough Circle Transform*, *Blob Detection*, dan *Camshift Received October 20th* Menggunakan AR.Drone, **IJEIS**, Vol.7, No.1, April 2017, pp. 1~12.

Motion Planning in Dynamic Environment for Middle Size League using Theta* and Polynomial Trajectory Generator

Fahri Ali Rahman*, Rudy Tri Hartanto, Revaldy Imani Chairistian, Yahya Abdurrahman, Okta Fajar Suryani, Maria Veronica Astrid, Resha Dwika Hefni Al-Fahsi, dan Igi Ardiyanto
Department of Electrical Engineering and Information Technology Faculty of Engineering
Universitas Gadjah Mada
Yogyakarta, Indonesia
ali.rahman.fahri@gmail.com

***Abstract*—In recent years, robot soccer competition is gaining popularity as a benchmarking platform for robotic research since it addresses many robotics problem such as localization, motion planning and multi-agent coordination. In RoboCup MSL, the gameplay is very dynamic and therefore the task of motion planning should be done in real-time. In this paper, we describe our approach to achieve a real-time motion planning for Middle-Size robot soccer in dynamic environment using grid-based path planning and polynomial trajectory generator. We evaluate our approach using 3D dynamic simulator gazebo. The simulation results shows that our approach could perform real-time motion planning in robot soccer scenario.**

Keywords—path planning; trajectory generation; motion planning

I. INTRODUCTION

RoboCup is an international robot soccer competitions that aims to promote research and developed of an autonomous soccer robot. In RoboCup Middle-Size League (MSL) teams of robots are competing each other in a soccer game. The gameplay of RoboCup MSL is very dynamic and challenging. Thus, the ability to move from one point to another without collide with other robot is necessary.

In this paper, we address the problem of path planning, trajectory generation and control of omnidirectional mobile robot in robot soccer domain. We assume that the state of the robot and the position of the obstacles are known and the robot is given a task to move to a certain point in the field. The challenge is to find shortest path without colliding obstacle and move the robot to follow the path.

This paper is organized as follows: section II reviews the basic Theta* path planning approach. In section III, we describe our approach to realize the generated path using polynomial trajectory generator. Afterwards, we discuss the implementation details in section IV. Section V discuss the simulation results. Finally, we conclude in section VI.

II. THETA* ALGORITHM

A. Theta* Path Planning

Basic Theta* is a path planning algorithm in a grid world. This algorithm is a variant of A* algorithm that find shorter and more realistic paths [1]. Unlike A*, the parent of a vertex in Theta* is not limited to the neighbor vertex but could be any vertex. Like A*, Theta* focuses it search using heuristics. The heuristic approximate the goal distance from the current vertex. In A*, each vertex has two values that is: the length of current vertex to the start vertex and parent vertex to extract path after the goal is found. Besides values for each vertex, A* algorithm also maintain two global data structure that is: open list and closed list. The open list is a priority list that store vertex to perform expansion. The closed list contains vertex that has been expanded. The algorithm of Theta* is similar to A* except that Theta* assigns a parent of an unexpanded vertex considering if the parent of current vertex has line-of-sight with the unexpanded vertex. This line-of-sight check allows Theta* to have any angle paths while in A* the angle of paths is artificially constrained by the connection of the vertices. The algorithm of Theta* is shown on figure 1. In the figure, function g(s) returns the cost to move from start vertex s. The function c(s,s') calculate the cost to move from vertex s to vertex s' while the function h(s) calculate the distance from vertex s to goal vertex. In this research, we use euclidean distance as heuristic. The euclidean distance is calculated as follows:

$$h(s) = \sqrt{(g_x - s_x)^2 + (g_y - s_y)^2} \quad (1)$$

B. Bresenham Line Algorithm

As explained above, Theta* needs line-of-sight checking that returns true if one point is visible from the other. This check could be performed by drawing a line from one point to another one and check if it collides with obstacle. The algorithm above use line Of Sight routine in line 20 to evaluate if the parent of current vertex s could be assigned as parent to neighbor vertex s'.

```
1. Main()
2.   g(s_start) := 0
3.   parent(s_start) := s_start
4.   clear open_list
5.   insert s_start to open_list with cost g(s_start)+h(s_start)
6.   clear closed_list
7.   while open_list is not empty
8.      s := first vertex of open_list
9.      if s = s_goal then
10.             return success
11.     add s to closed_list
12.     for each s', neighbor of s
13.             if s' is not in closed_list
14.         if s' is not in open_list
15.           g(s') := infinite
16.           parent(s') := null
17.               updateVertex(s,s')
18.  return failed


19. updateVertex(s,s')
20.   if lineOfSight(parent(s), s')
21.     if g(parent(s)) + c(parent(s) to s') < g(s')
22.       g(s') := g(parent of s) + c(parent(s) to s')
23.         parent(s') := parent(s)
24.         if s' is in open_list
25.           remove s' from open_list
26.         insert s' to open_list with cost(g(s') + h(s'))
27.   else
28.     if g(s) + c(s to s') < g(s')
29.        g(s') := g(s) + c(s to s')
30.        parent(s') := s
31.        if s' is in open_list
32.           remove s' from open_list
33.        insert s' to open_list with cost(g(s') + h(s'))
```

Fgr. 1. Basic theta* algorithm.

This procedure is one major difference between A* algorithm and Theta* algorithm is that the latter has more realistic path since the angle of the paths is not restricted to neighbor vertex. To implement line Of Sight routine in a grid world, one could utilize bresenham's line algorithm. Bresenham's line algorithm determines a set of points that approximate straight line between two points. If one of those points is identified as an obstacle, then the two point has no line of sight and therefore the update Vertex routine would select a basic A* vertex evaluation. The line Of Sight routine is shown in fgr. 2.

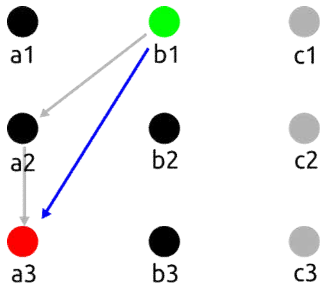
The illustration of the algorithm is shown in figure 3. The figure shows parent assignment of a vertex. In the figure, the circle illustrate the vertices, while arrows show the parent of vertices. In A* algorithm, the parent of vertex b1 expanded from vertex a2 should be vertex a2. While in Theta* algorithm, consider if vertex b1 is visible from the parent of current vertex a2, that is vertex a3, if it is visible then the parent of vertex b1 should be vertex a3. This parent assignment allows Theta* algorithm to have any-angle path. Hence the resulting path of Theta* algorithm is more realistic than A* algorithm.

III. Trajectory Generation

Omnidirectional robots have superior maneuver capability compared to car-like robots. They have ability to move in any direction and simulaneously reach target orientation. To make the robot move from one point to another, one should compute the velocity of the robot that makes the robot eventually reach the goal. In this paper, to accomplish such task, we use kinematic trajectory generation to make the robot reaches the goal.

```
34. lineOfSight(s,s')
35.       check_list := 0
36.       dx := s'x - sx
37.       dy := s'y - sy
38.       step_x := sign(dx)*1
39.       step_y := sign(dy)*1
40.       if abs(dy) > abs(dx)
41.                           slope := dy/dx
42.                           pitch := sy - slope*sx
43.                           x := sx
44.                           while x is not s'x
45.                                   y :=
round(slope*x+pitch)
46.                                   if isObstacle(x, y)
47.                                           return false
48.                                   x += step_x
49.       else
50.                           slope := dx/dy
51.                           pitch := sx - slope*sy
52.                           y := sy
53.                           while y is not s'y
54.                                   x :=
round(slope*y+pitch)
55.                                   if isObstacle(x, y)
56.                                           return false
57.                                   y += step_y
58.       return true
```

Fgr. 2. Line of sight checking algorithm.



Fgr. 3. Illustration of the algorithm.

A. Polynomial Trajectory Generator

The reference trajectory is generated using third order polynomial. Starting with a known position and velocity, the reference trajectory of the robot could be computed using the following equation [2] :

$$z_{ref}(t)=\left[\frac{-2(z_f-z_0)}{t_f}+v_f+v_0\right]\frac{t^3}{t_f^2}+\left[\frac{3(z_f-z_0)}{t_f}-v_f-2v_0\right]\frac{t^2}{t_f}+v_0 t+z_0 \quad (2)$$

where,

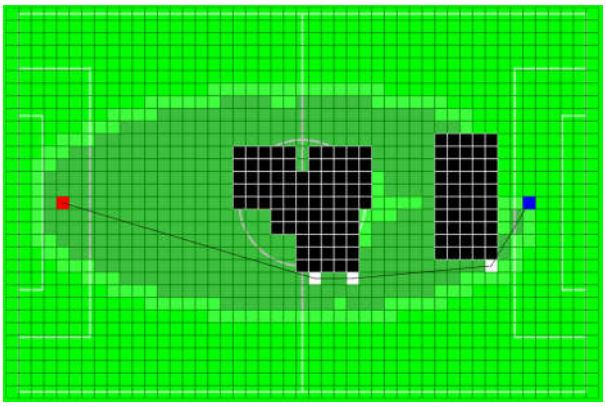
$$z_{ref}(0)=z_0 \quad (3)$$

$$\dot{z}_{ref}(0)=v_0 \quad (4)$$

$$z_{ref}(t_f)=z_f \quad (5)$$

$$\dot{z}_{ref}(t_f) = v_f \tag{6}$$

The equations above could generate a continuous trajectory from t=0 to t=tf while satisfying the given initial position z_0, initial velocity v_0, final position z_f and final velocity v_f. Here, the position and velocities are expressed in global coordinate. Ideally, the robot could achieve the final state by following the reference velocities. But in the presence of disturbance, one need a closed loop control to follow the reference trajectory.



Fgr. 4. Path generated by theta*.

B. Trajectory Tracking

After the reference trajectory is generated, the problem is then to make the robot follow the desired trajectory. We define the tracking error as:

$$e = z - z_{ref} \tag{7}$$

We then use PD control to track the desired trajectory. Thus, the control signal that is given to robot is:

$$\dot{z}_{control} = \dot{z}_{ref} + k_p e + k_d \dot{e} \tag{8}$$

Where k_p and k_d are positive constants of PD controller.

IV. Implementation Details

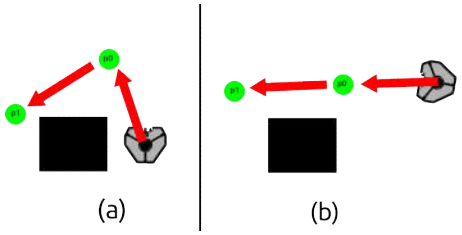
A. Path Planning

To find obstacle-free path, the robot is defined as a single vertex in a grid world. The map of the world is generated using fixed decomposition[3]. Theta* algorithm is then used to find obstacle-free path. The path is extracted by tracing the parent of the vertex of the grid starting from the goal vertex. Figure 4 shows the result of Theta* algorithm. In the figure, black cells represent the obstacles, red cell indicates the start vertex, blue cell indicates the goal vertex and white cells represent the obstacle-free path generated by the algorithm.

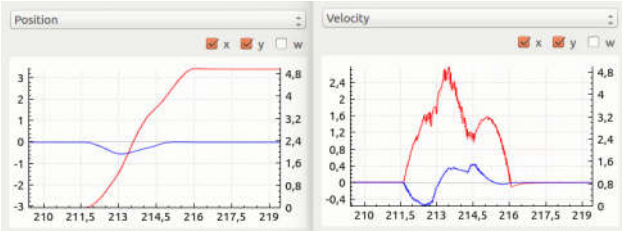
B. Motion Control

The generated path is then treated as sub-goal for the trajectory generator. The final velocity of the sub-goal is computed considering the position of the next sub-goal to achieve smooth trajectory. We consider the turning angle between sub-goals when computing the final velocity of sub-goals. Figure 4 shows the illustration of this idea.

In fgr. 5(a), the one should configure the robot so that the trajectory generator does not generate excessive final velocity when reaching sub-goal p0. In other words, the robot should slow down when reaching p0 in figure 5(a). While in figure 5(b), the trajectory generator should generate a sufficient final velocity or speed up when reaching sub-goal p0.



Fgr. 5. In case (a), robot should slow down when reaching p0. While in case (b) robot could speed up when reaching p0.



Fgr. 6. Trajectory of the robot.

Generated trajectory is defined in global coordinate. Thus, the control input should be transformed into robot's local coordinate. The relationship between the velocity in local coordinate and velocity in global coordinate are:

$$z = [x_w \; y_w \; \theta]^T \tag{9}$$

$$u = (\dot{x}_r \; \dot{y}_r \; \omega)^T \tag{10}$$

$$u = T\,\dot{z} \tag{11}$$

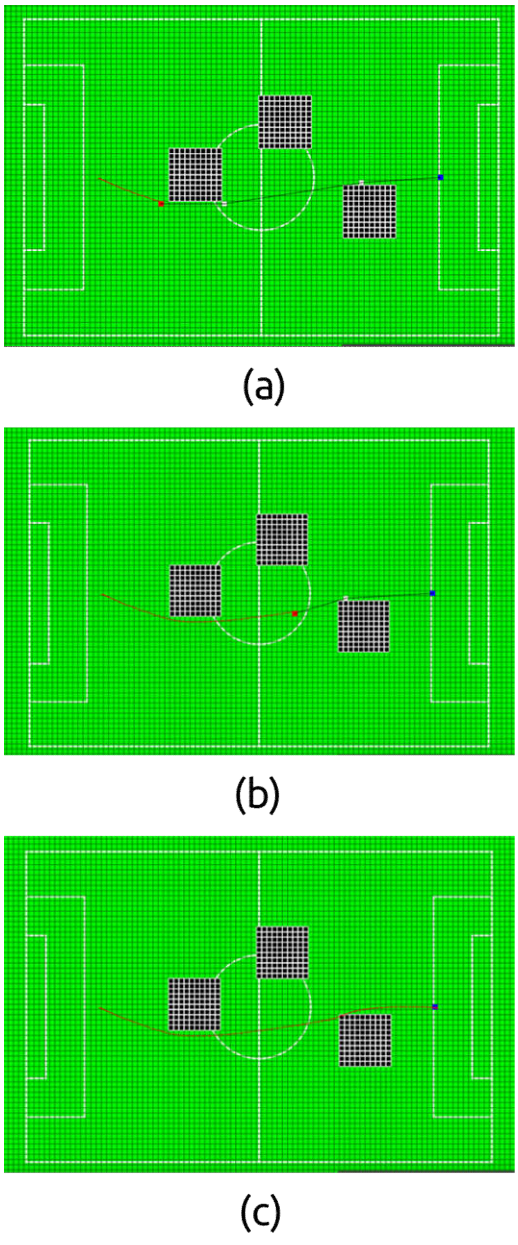
$$T = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{12}$$

V. Simulation Result

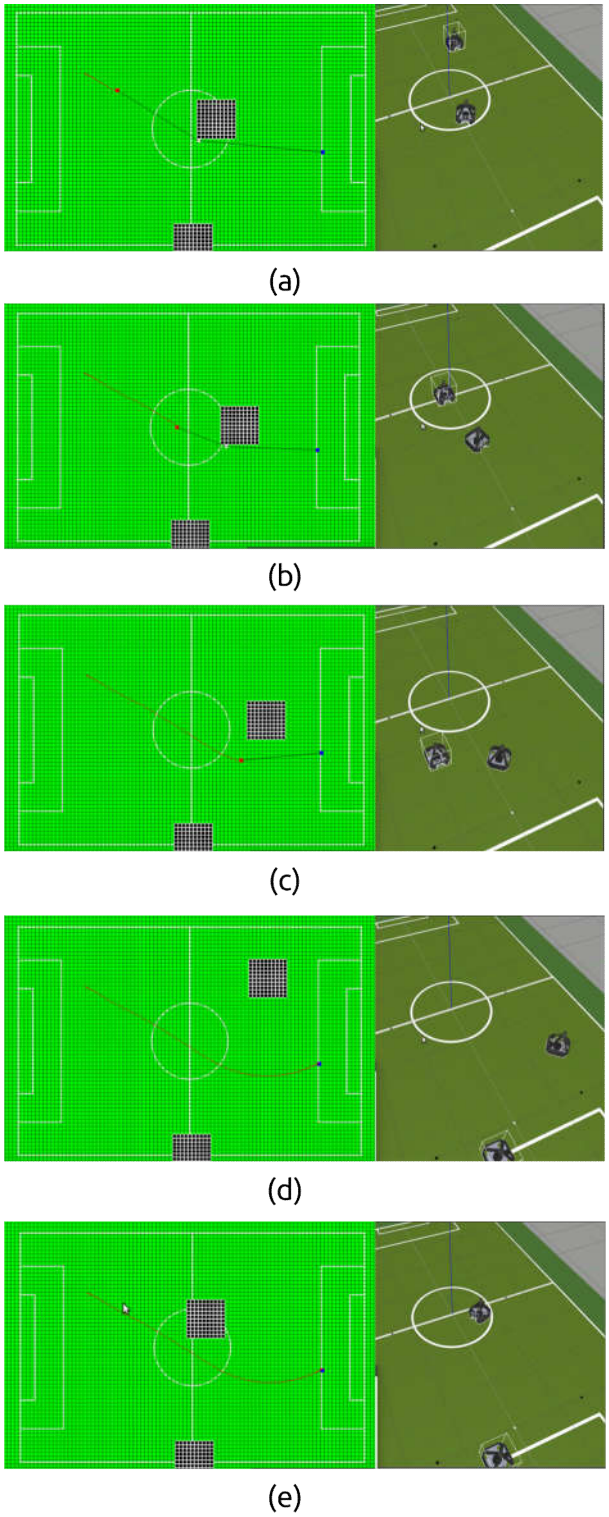
We test our algorithm using dynamic simulator gazebo[]. All of implementations are done in Laptop with intel core i3 processor, 4 GB RAM, Ubuntu 16.04. We implement our algorithm as a *node* in ROS environment for better reusability.

We perform our simulation in a Middle-Size League football field with size of 9x6 m. We consider both case with

static and dynamic obstacle. Simulation result with static obstacle is shown in fgr. 7. In the figure, the current position of the robot shown with red cell, the goal is shown with blue cell. The red line shows the executed trajectory of the robot from the trajectory controller. Black lines show planned path from Theta* algorithm. Fgr. 6. shows the plot of the robot trajectory, with red lines show position and velocity of x-axis in global coordinate and blue lines show position and velocity of y-axis in global coordinate system.



Fgr. 7. Simulation result with static obstacle



Fgr. 8. Simulation result with moving obstacle.

Fgr. 8. shows the simulation result when the obstacle is dynamic. As shown in the figure, the robot replanned the path continuously to deal with moving obstacle.

VI. CONCLUSION

We presented our approach to to realize motion planning in dynamic environment in robot soccer scenario. We used Theta* algorithm combined with polynomial trajectory generator to generate the motion of the robot. We show our effectiveness of our algorithm in 3D dynamic simulation using gazebo simulator. The result shows that our algorithm could deal with dynamic obstacle.

REFERENCES

[1] Nash, Alex, et al. "Theta*: Any-Angle Path Planning on Grids." *AAAI*. 2007.

[2] Kalmár-Nagy, Tamás. "Real-time trajectory generation for omni-directional vehicles by constrained dynamic inversion." *Mechatronics* 35 (2016): 44-53.

[3] Siegwart, Roland, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to autonomous mobile robots*. MIT press, 2011.

Development of Middle-Sized Mobile Robot FUKURO for Robot Soccer Competition

Igi Ardiyanto, Rudy Tri Hartanto*, Fahri Ali Rahman, Revaldy Imani Chairistian, Maria Veronica Astrid, Okta Fajar Suryani, Yahya Abdurrahman, Resha Dwika Hefni Al-Fahsi, Muhammad Hamam Afandi, Denanda Hendra, Ma'ruf Shidiq, Linggom Enrico C, Olga Priandana Yudi, Doni Aria Wisesa, Irziq Misbahul Arif, Dimas Arya Dewangga, dan Ezra Febrinanda

Team FUKURO, Gadjah Mada Robotic Team
University of Gadjah Mada
Yogyakarta, Indonesia
rudy.tri.h@mail.ugm.ac.id

***Abstract*—Nowadays, robotic soccer is a popular research platform for developing an autonomous intelligent robot. This paper describes the omnidirectional mobile robot platform developed by robotic soccer team FUKURO of University of Gadjah Mada, Indonesia. *Kontes Robot Sepak Bola Beroda Indonesia (KRSBI-Beroda),* a national robotic soccer competition in Indonesia, was identified as perfect testbed and testing for our autonomous intelligent robot. First, we describe our hardware platform and electronic system. Afterwards, we describe our approach to achieve modular robotic soccer software.**

Keywords— holonomic mobile robot; autonomus robot; robot soccer; middle-sized league

I. INTRODUCTION

RoboCup is an international competition to promote research on autonomous, intelligent, multi robot system. The ultimate of RoboCup is that the winner of RoboCup could win against human team that win the FIFA WorldCup by the year of 2050. *Kontes Robot Sepak Bola Beroda Indonesia (KRSBI-Beroda)* is a national robotic soccer competition in Indonesia that adopts RoboCup Middle-Size League. The year of 2017 is the first time *KRSBI-Beroda* competition [1].

FUKURO is the *KRSBI-Beroda* soccer team of the University of Gadjah Mada, Indonesia. The poject consists of undergraduate student from several fields contributing for the development of all components of the robots, from the hardware to the software.

This paper describes the current development of the FUKURO Middle-Sized robot soccer. This paper is organized as follow: Section II describe the mechanical design of the robot, Section III describe the electronic system of the robot, and Section IV describe the modular Software Architecture. Finally, Section V concludes this paper.

II. MECHANICAL DESIGN

FUKURO is a wheeled robot soccer consisting of five base tiers. At the first base, this base is filled by the system of the robot drive, where in the robot uses three omni wheels these are connected by the planetary gear motor with torque of 20 kgf.cm [2]. On the base two there is a dribbler and kicker system, where the dribbler system functions to dribbling the ball so that the robot can keep the ball to stay attached to the robot in a state of motion and the kicker system itself is equipped by a motor that serves to pull the spring in order to kick the ball [3]. This robot uses a laptop as a hardware controller that is placed in the base three. For the vision of the team FUKURO using a convex mirror placed on the camera so that the robot has 360˚ vision [4]. The mechanical design of FUKURO shown in Fgr. 1. Overall, each robot of FUKURO has specification shown in TABLE I.



Fgr. 1. FUKURO mechanical design.

TABLE I. ROBOT SPESIFICATION

Dimension (cm)	43 × 50 × 78
Weight (kg)	19.4

A. *Drive Mechanism*

The first base such as Fgr. 2. consists of Planetary Gear 45 (PG-45) motors, omni wheel, lithium polymer (Li-Po) batteries and motor drivers. One of the challenges faced by many teams including FUKURO is when the space installation of mechanical and electronic parts is limited at the first base. The team solves this problem by transmitting power from motor to omni wheel using chain and sprocket. The advantage of transmitting the power from motor to the omni wheel is the space at the center of the base one will be empty when compared to the omni wheel directly connected to the motor. In addition to the empty center space, transmitting the motor power to the omni wheel can also vary the output because of the ratio of the motor sprocket to the adjustable diameter sprocket of the required torque and rotational speed.

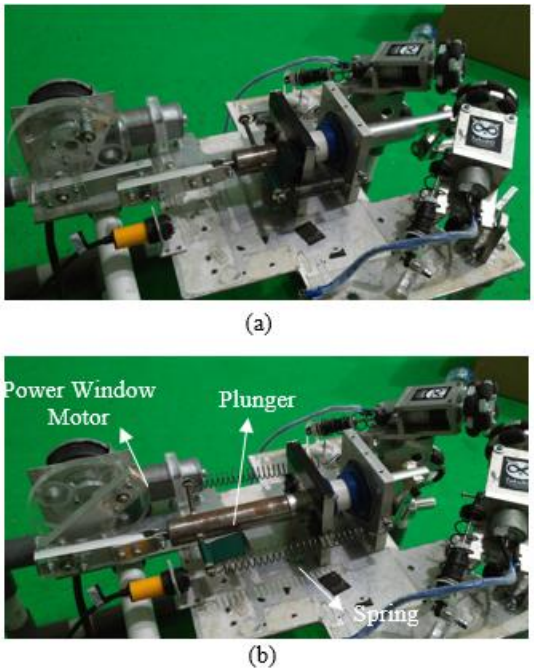


Fgr. 2. The lower base of FUKURO.

B. *Kicking Mechanism*

Main purpose of a soccer game is to score as much as possible goals and try to minimize opponents goal. Just like human soccer game, wheeled soccer robot also has the same missions to score goals as much as possible, and this is the challenge of every team in the world because the robot is required to be able to kick the ball.

FUKURO uses a mechanical kick system. The system use a spring to transmit kicking power to the plunger. This mechanical kick system has the same concept with the way the piston works on the motor that using combustion system, by changing the rotation movement into translation. This motor type is a dc power window motor. This motor is selected because this motor includes one motor with a very large average torque of 3 Nm, so it will have a very small rotation even up to 80 rpm. This motor is connected to structure that will connect to a plunger (kicker). This structure use energy from spring potential energy. Plunger in addition to being connected to the motor by the structure, will also be moved by a spring where the spring is attached to a fixed spot such as Fgr. 3.



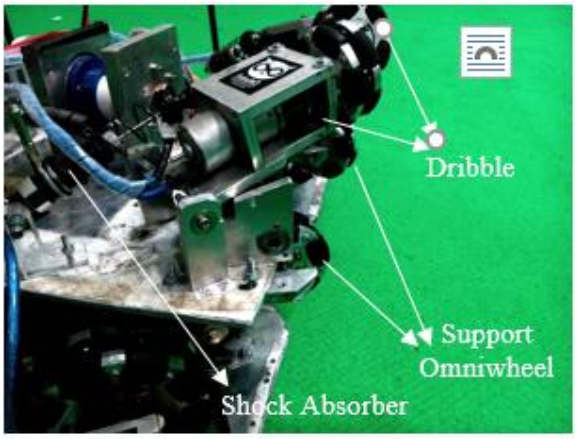
Fgr. 3. (a) Springs Before Being Stretched, (b) Springs are Being Stretched.

C. *Dribbling Mechanism*

The dribbling mechanism such as Fgr. 4. on the wheeled soccer robot serves to attach the ball to stay in the robot's grip. One type of dribbling mechanism on wheeled soccer robots is to use two wheels placed in front of the robot [5]. The mechanism have the advantage that can hold the ball in a state of robot maneuver.

Besides of those advantages, the disadvantages is that it becomes difficult to position the right dribble about the ball position. This difficulty can be overcame by adding a shock absorber to the dribble that works besides holding the incoming balls, but also to make the dribbler flexible so that dribble can grasp the ball well.

In order to be able to dribble well, FUKURO uses Planetary Gear motors connected to a 5cm diameter omni wheel as a dribble system. We choose omni-wheel so that the dribbler can accept the coming ball from the front and side of the robot. At the bottom of the dribble, we provide additional wheels that serves as freewheel to support the dribble so that the ball can be carried by the robot. The rotation direction of the dribble motor should lead to the robot in order to keep the ball in the robot's range even when moving forward, sideways or backward.



Fgr. 4. FUKURO dribbling mechanism.

III. Electrical Design

The electrical design of FUKURO consists of main controller board, arduino board, power circuit, motor controller. Since our robot space is limited, we design our system in a compact way to fit the available space. The overview of the system is depicted in Fgr. 5.

A. Main Controller Board

The STM32F446-Nucleo is used as main microcontroller in our robot. This microcontroller controls the wheel motors, dribbler motors, and kicking system. It also processes the input data from encoder and infrared sensor. The main board communicates with laptop as main controller using serial communication.

B. Arduino Board

In addition to the main board, we also use additional board to provide user input from the push button. This additional board use Arduino nano as the controller. The board also process data from compass to provide heading reference. This board also use serial communication with laptop.

C. Power Circuit

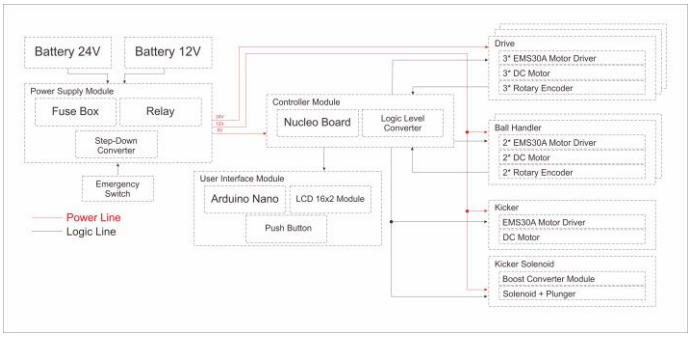
There are 2 batteries in our electronic system, 6-cell LiPo 3000 mAh and 3-cell lipo 2000 mAh. The 3-cell lipo battery operates at 12 V and supplying kicking system. The 6-cell lipo battery operates at 24 V and it is used for supplying motor also used to power the whole robot. XL4015 has been used for regulating and supplying 12V to the motor PG-28. Ubec is used to regulating and supplying 5V to the rest of the circuit. To supply 3.3 volts to the main board, LM1117 voltage regulator has been used.

D. Motor Controller

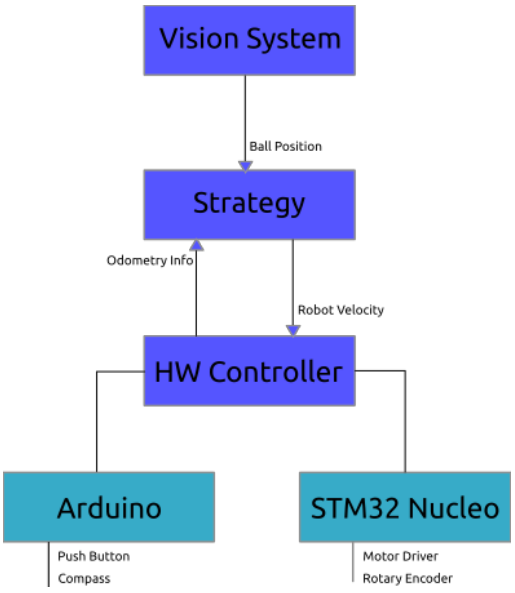
EMS 30 A H-Bridge motor controller module is used to control the DC motors in our robots. The module is designed to operate in 2 driving direction with 30 A continues current at voltage range of 5.5 V to 36 V. The same motor controller has been used to control both the wheel and dribbler motors.

IV. Software Architecture

The software system such as Fgr. 6. is distributed among several computational units. Higher computational load is processed on the laptop while lower computational load is processed on microcontrollers. We use two microcontroller as sub-controller. Arduino microcontroller provides user input via push button and process compass data. STM32-Nucleo controls the motor driver and reads encoder inputs. On the laptop side, we divide the overall process to three sub-process. We use ROS as middleware to synchronize the sub-process.



Fgr. 5. FUKURO electrical design diagram.



Fgr. 6. FUKURO software architecture design.

A. Vision System

Our Vision System is based on a custom-made omni-directional camera. The omni-directional camera consists of Logitech C920 web camera facing upward with hyperbolic

mirror to reflect object around the robot [6]. With this configuration, we could detect ball in every direction. The camera delivers image with VGA resolution at 30 fps. Ball detection is obtained by performing color segmentation using color lookup table. The lookup table is defined in HSI color space. Ball detection's result is then published as ROS Topic to be used by the other *node*. Our vision system is is shown in Fgr. 7. In the figure, ball detection is shown with yellow borderline. The red circle shows Region Of Interest (ROI) of the image. As shown in the figure, the camera is located in the center of the image, thus the robot could see the ball from all directions.



Fgr. 7. Ball detection result.

B. Hardware Controller

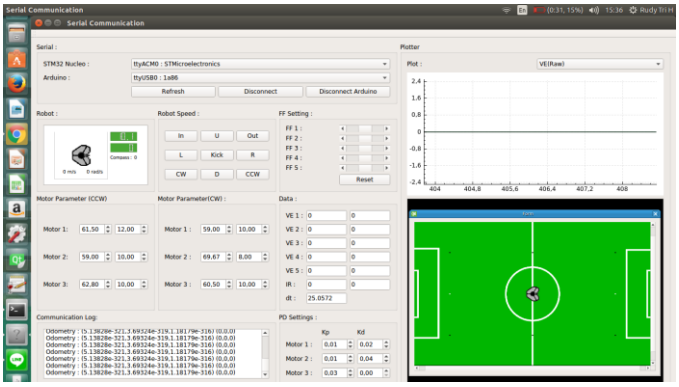
This *node* process all computations and services that need to be execute by the Nucleo and Arduino Board. The *node* also serve as tuning interface for motor control. To make easy such functionality, we create a Graphical User Interface (GUI) for this *node*. The GUI is shown in Fgr. 8. In the GUI, we plot encoder reading and visualize odometry information. Additionally, there are buttons for tuning motion control of the robot.

C. Odometry

The odometry computation is processed on the HW Controller *node.* The HW Controller *node* receives rotary encoder readings from three wheels via UART Serial Communication with STM32-Nucleo. Odometry provides robot pose estimation in 2D space by integrating robot velocity information from rotary encoder data [7]. The robot velocity is calculated using forward kinematics. Every time the HW Controller receives new encoder data, the odometry is updated. The Odometry result is then published as ROS topic to be used by another *node*.

D. Motion Control

Our robot is based on holonomic platform, thus, could transverse in any direction. Current implementation of our robot motion control is performed by open loop control. The motion control is achieved by setting feed-forward PWM value for each motor. The motor speed computation is performed on the HW Controller *node* using Inverse Kinematics. It receive robot velocity vector (v_x, v_y, ω) and translate it into individual wheel speed. The velocity vector is obtained by subscribing a ROS Topic published by another *node,* that is strategy *node*.



Fgr. 8. HW controller graphical user interface.

E. Strategy

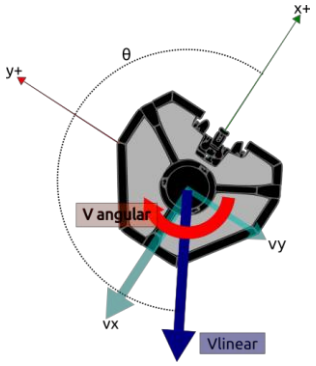
Our Strategy sub-process are responsible to decide the behavior of the robot. The module receives ball position information from the vision sub-process and publish robot velocity vector. This module also receives input from push button and odometry information from HW Controller sub-process. The behaviors of the robot are defined using Finite State Machine approach. Current implementation of robot's basic behavior are GotoHome, SearchBall, ApproachBall, DribbleBall, and Shoot.

The robot velocity vector is calculated using PD position control. Setpoint for the PD controller are defined by the robot's current behavior. For example, *ApproachBall* behavior gives ball's current position as setpoint to the controller. The velocity computation is performed by assuming that linear velocity and angular velocity of the robot are independent. Ball position is defined in polar coordinates, the distance is used for linear velocity calculation while the angle is used for angular velocity calculation. Computed linear velocity is then decomposed into local velocity on local x-axis (v_x) and local velocity on local y-axis (v_y) of the robot current pose. Fgr. 9. illustrate the computation of v_x and v_y. The velocities are computed as follows:

$$v_x = v_{linear} \cos \theta \quad (1)$$

$$v_y = v_{linear} \sin \theta \quad (2)$$

Where v_x is the robot velocity in x axis, v_y is the robot velocity in y axis, v_{linear} is linear velocity of robot and θ is angle of robot.



Fgr. 9. Velocity vector and decomposition.

V. Conclusion And Future Works

This paper described the development of our robot soccer team, FUKURO. The team has developed omni-directional robot platform, electronic system and modular software architecture. Future Works should ideally center around the implementation of self-localization for position estimation of the robot. Additionally, the work should also be done on improving robot control and motion control to achieve better locomotion of the robot.

References

[1] Robocup. *Middle Size Robot League Rules and Regulation. Robocup, 2016.*

[2] Dias, R., et al. "CAMBADA 2013 Team Description Paper." *Proceedings Robocup* 2013 (2013).

[3] Hibbeler, Russell C., Kai Beng Yap, and S. C. Fan. *Mechanics for Engineers: Dynamics*. Macmillan, 1985, pp. 344, 459-461.

[4] Yu, Wentao, et al. "NuBot team description paper 2010." *Proc. RoboCup 2010 Singapore, CD-ROM* (2010).

[5] LU, Sheng-cai, et al. "Design of the active ball handling mechanism of RoboCup MSL robot [J]." *Mechanical & Electrical Engineering Magazine* 11 (2009): 003.

[6] Kasaei, S. Hamidreza Mohades, et al. "Modeling and implementation of omnidirectional soccer robot with wide vision scope applied in RoboCup-MSL." *BRAIN. Broad Research in Artificial Intelligence and Neuroscience* 1.3 (2010): 65-74.

[7] Inthiam, Jiraphan, and Chirdpong Deelertpaiboon. "Self-localization and navigation of holonomic mobile robot using omni-directional wheel odometry." *TENCON 2014-2014 IEEE Region 10 Conference*. IEEE, 2014.

Sistem Penendang Bola Mekanis pada Robot Sepakbola Beroda

Dimas Arya Dewangga, Doni Aria Wisesa, Ezra Febrinanda, Irziq Misbahul Arif*, dan Olga Priandana Yudi
Departemen Teknik Mesin dan Industri, Fakultas Teknik
Universitas Gadjah Mada
Yogyakarta, Indonesia
irziq.misbahul.a@mail.ugm.ac.id

***Abstract*— The most important part of the wheeled-robosoccer is kicker. In this paper, kicking mechanism is using Power Window motor and spring as the main part. Using piston mechanism, this kicking mechanism also changes the rotational motion into translational motion as the main principle. By a simple and strong structure, high coefficient of spring, and high torque of motor, this kicking mechanism will produce high force to kick the ball. Plunger using in this mechanism is formed by aluminium and steel to produce a high momentum. Then the ball can move at a speed of 7 m/s.**

Keywords— kicker; piston mechanism; wheeled robosoccer

I. PENDAHULUAN

Kontes Robot Indonesia (KRI) adalah kegiatan kontes bidang robotika yang dapat diikuti tim mahasiswa dari institusi atau Perguruan Tinggi Negeri dan Swasta yang terdaftar di Kementerian Riset, Teknologi dan Pendidikan Tinggi [1].

Salah satu faktor penting untuk memenangkan pertandingan dalam Kontes Robot Sepakbola Indonesia divisi beroda adalah dengan sistem penendang yang baik. Sistem penendang yang baik yaitu sistem penendang yang mampu menendang bola sesuai tujuan. Menendang dengan arah yang sesuai dengan keinginan dan tepat sasaran.

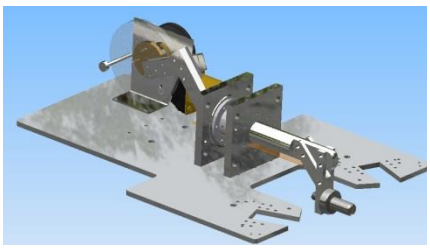
Ada berbagai macam mekanisme penendang yang dapat digunakan pada robot sepakbola beroda. Salah satu yang paling sering digunakan adalah dengan penendang pneumatik. Dengan memanfaatkan tekanan udara yang besar, dan instalasi serta konstruksi yang sederhana menjadikan sistem ini menjadi pilihan bagi banyak tim robot sepakbola beroda dalam negeri maupun luar negeri.

Dalam semua kelebihannya, ada beberapa kekurangan yang pada sistem ini, antara lain ruang yang dibutuhkan untuk menyimpan udara sangat memakan tempat, rawan terhadap kebocoran, semakin sering dipakai maka daya yang dikeluarkan akan semakin menurun. Hal inilah yang menyebabkan dibutuhkannya inovasi penendang yang lebih efisien.

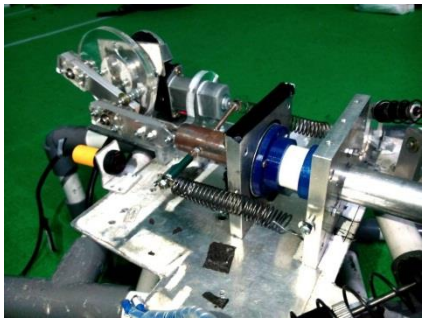
Salah satu inovasi dari tim robot sepakbola beroda dari Universitas Gadjah Mada yaitu Fukuro menciptakan sistem penendang yang diharapkan lebih baik, yaitu dengan menggunakan sistem yang akan di jelaskan pada paper ini.

II. DESAIN SISTEM PENENDANG MEKANIS

Sistem penendang yang dibuat oleh tim robot sepakbola beroda dari Universitas Gadjah Mada menggunakan sistem mekanis dengan mengadopsi prinsip kerja piston yaitu mengubah gerakan rotasi menjadi gerak translasi. Pada perancangannya dibutuhkan beberapa bagian seperti Motor *Power Window*, pegas tarik, *plunger*, dan stuktur pendukung. Bagian-bagian tersebut di susun sedemikian rupa dalam satu lantai robot. Kemudian bagian-bagian yang memerlukan penyokong seperti motor menggunakan plat aluminium dengan tebal 2 mm yang dibentuk menyesuaikan bentuk motor. Begitu juga untuk bagian yang lain seperti *plunger* digunakan plat aluminium dengan tebal 10 mm untuk menjaga agar *plunger* dapat bergerak translasi dengan kuat. Gambar 1 dan 2 berikut adalah penampakan dari desain penendang mekanis.



Gambar 1. Desain kicker mekanis.



Gambar 2. Kicker mekanis.

A. Kekekalan Energi

Prinsip kerja dari penendang ini menggunakan hukum kekekalan energi dimana energi yang masuk sama dengan energi yang keluar pada sistem.

$$E_{in} = E_{out}$$

Dengan hukum tersebut, sistem ini memanfaatkannya untuk membuat susunan struktur dengan tenaga masukan menggunakan motor *Power Window*. *Power window* yang digunakan menggunakan input tegangan 12 V. Dengan tegangan sebesar itu dapat menarik pegas yang digunakan sebagai penyimpan energi potensial. Energi tersebut nantinya akan menjadi gaya yang digunakan untuk menendang bola. Gaya yang dihasilkan oleh pegas dapat dihitung dengan mengalikan konstanta pegas dengan pertambahan panjang pegas. Inilah yang disebut Hukum Hooke pada pegas.

Pada tahun 1676, Robert Hooke mengusulkan satu hukum fisika yang menyangkut pertambahan panjang sebuah benda elastis yang dikenai oleh suatu gaya. Menurut Hooke, pertambahan panjang berbanding lurus dengan yang diberikan pada benda. Secara matematis, hukum Hooke ini dapat dituliskan sebagai [2]

$$F=k.\Delta x$$

Dengan F=gaya, k=konstanta pegas, *Δx=pertambahan panjang*.

Energi potensial yang disimpan oleh pegas di dapat dari persamaan

$$Ep=½\ k\ \Delta x^2$$

Dengan energi potensial tersebut dapat dilakukan perhitungan agar didapat energi potensial terbesar dan mampu menendang bola dengan kuat. Serta dapat pula digunakan sebagai acuan dalam memilih pegas dan mendesain struktur.

Perhitungan untuk mendapatkan tenaga terbesar dapat dilakukan dengan menghitung persamaan kekekalan energi dan kondisi lapangan. Lapangan yang digunakan pada Kontes Robot Sepakbola Indonesia divisi beroda berukuran 9 x 6 meter persegi. Untuk itu kita perlu kecepatan gerak bola minimal 9 meter/detik untuk dapat memaksimalkan tendangan ke gawang *(shooting)*. Hal ini juga diasumsikan dengan kecepatan kiper dalam mendeteksi bola. Kiper akan mendeteksi bola dalam waktu 0.5 detik. Maka dari itu untuk dapat menendang bola ke gawang dan mencetak gol, bola harus dapat melebihi kecepatan gerak kiper dan deteksinya.

Permasalahan itu menjadi salah satu dasar dalam menentukan kekuatan tendangan. Dengan memanfaatkan hukum kekekalan energi pula, dapat diperoleh persalamaan untuk menentukan satuan yang dapat menjadi acuan untuk memperoleh kekuatan tendangan yang baik. Salah satu nilai yang dicari dalam hal ini adalah kecepatan *plunger* dalam kontak dengan bolanya. Dari hukum kekekalan energi di peroleh nilai sebagai berikut,

$$Ein = Eout \quad (1)$$

$$Ep_{pegas} = Ek + E_{gesek} \quad (2)$$

$$\frac{1}{2}k\Delta x^2 = \frac{1}{2}mv^2 + \mu Ns \quad (3)$$

Dengan mengabaikan nilai E_{gesek}, maka dari persamaan (3) kita dapat meperolah persamaan sebagai berikut,

$$v = \sqrt{\frac{k\Delta x^2}{m}} \quad (4)$$

Dari persamaan (4) diatas, dapat diambil kesimpulan bahwa yang mempengaruhi kecepatan *plunger* dalam bergerak adalah konstanta pegas *k , Δx, dan m.* Untuk mendapatkan momentum yang besar ketika *plunger* mengenai bola, maka massa *plunger* haruslah bernilai tinggi dan tetap [3]. Oleh karena itu, nilai yang dapat berubah adalah nilai *k* dan *Δx*. Nilai dari *Δx* terbatas oleh desain dan ruang yang termuat dalam robot. Jadi nilai *k* menjadi nilai yang mempengaruhi kekuatan tendangan.

Pemilihan pegas perlu memperhatikan sifat elastisitas materialnya. Elastisitas adalah sifat pada suatu material yang memungkinkan benda untuk kembali ke bentuk semula setelah mengalami deformasi. Pegas adalah elemen mesin yang mendesak sebuah gaya ketika mengalami deformasi [4].

Untuk mencapai kecepatan 9 meter/detik, maka dengan mengasumsikan nilai *m* dan *Δx* masing-masing sebesar 0.8 kg dan 10 cm, maka nilai *k* yang dicari dapat diperoleh sebagai berikut,

$$9 = \sqrt{\frac{k(0.1)^2}{0.8}}$$

$$k = 6480$$

Dengan demikian, nilai konstanta *k* yang dibutuhkan adalah 6480. Nilai *k* ini menjadi nilai minimum yang harus dimiliki oleh pegas yang di pakai. Namun nilai konstanta pegas yang di pakai juga dipengaruhi hal-hal lain seperti kekuatan struktur dan gaya gesek yang ada. Dan dengan melewati percobaan dan perhitungan. Kecepatan yang diperoleh dari robot tim Fukuro ini tidak mencapai target 9 meter/detik.

Untuk mengatasi hal tersebut, digunakanlah strategi untuk menyerang berupa *kick and rush*. Strategi ini merupakan strategi dengan langkah membawa bola hingga dekat dengan gawang lawan, kemudian melakukan tendangan. Dengan seperti itu, tendangan yang memiliki kecepatan kurang tersebut masih memiliki kesempatan untuk dapat masuk gawang lawan karena jarak yang dekat.

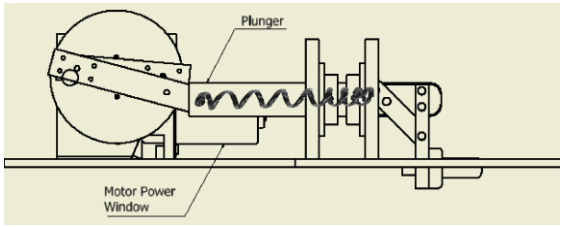
B. Penempatan Posisi

Pemasangan bagian-bagian dari sistem penendang mekanis ini harus tepat. Dengan memperhatikan kekuatan dari masing-masing material penyusun bagian-bagiannya seperti plat aluminium yang digunakan sebagai dudukan motor, plat aluminium sebagai dudukan *plunger* dan struktur dinamis yang kekuatannya juga sangat diperhatikan, maka kita perlu menyusun penempatan dan *assembly* dari sistem penendang ini.

C. Langkah Kerja

Langkah kerja dari sistem penendang mekanik ini adalah sebagai berikut. Motor PW akan memutar piringan yang terhubung dengan struktur yang mentransmisikan energi ke *plunger* yang terhubung dengan pegas tarik. Bentuk piringan

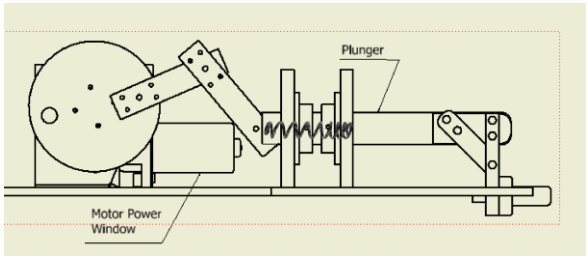
adalah lingkaran dan memiliki pengait untuk mengaitkan profil struktur.



Gambar 3. Posisi saat penendang siap untuk menendang.

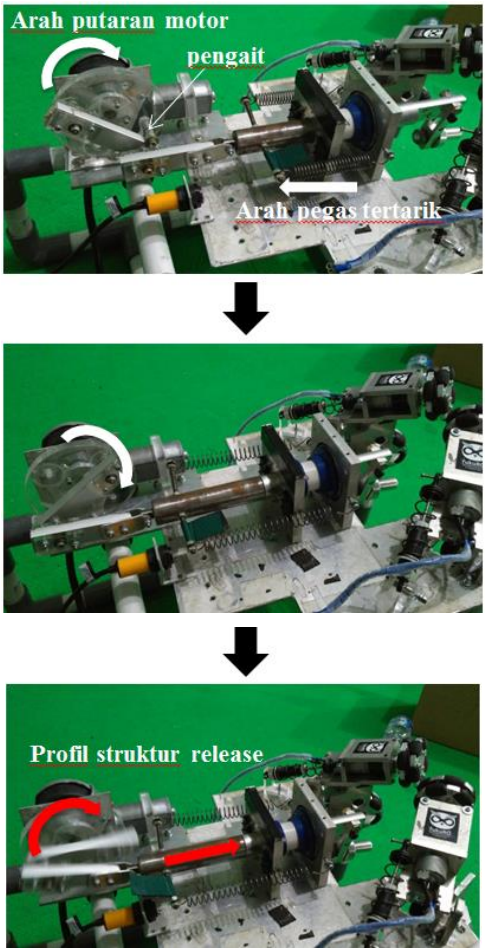
Posisi saat penendang siap untuk menendang yaitu ketika pertambahan panjang pegas telah mencapai titik maksimum. Yaitu ketika pengait pada piringan telah mengait profil struktur dan membawanya memutar hingga ke titik terjauh dari ujung pegas yang tetap. Posisi itu dapat dilihat pada Gambar 3.

Kemudian setelah mencapai titik maksimum pertambahan panjang pegas, maka profil akan lepas dari pengait dan gaya pegas akan mendorong *plunger*. Dengan torsi motor yang tinggi dapat menarik pegas dengan koefisien pegas yang cukup tinggi untuk menghasilkan gaya reaksi pegas yang tinggi pula. Kemudian energi potensial yang dimiliki pegas di transimisikan ke *plunger* yang kemudian menghasilkan gerak translasi, kemudian diarahkan pada bola agar menghasilkan momentum yang akhirnya akan membuat bola bergerak sesuai arah tendangan.



Gambar 4. Posisi saat penendang telah melakukan tendangan.

Gambar 4 menunjukkan posisi saat penendang telah melakukan tendangan dimana pegas kembali pada panjang mula-mula dan *plunger* telah berubah posisinya akibat terdorong oleh pegas. Pada saat inilah energi potensial pegas di transmisikan secara cepat menjadi energi untuk menendang bola.



Gambar 5. Langkah kerja sistem penendang mekanis.

Gambar 5 di atas merupakan langkah kerja sistem penendang mekanis yang telah dijelaskan pada paragraf sebelumnya. Langkah kerja seperti ini membutuhkan kekuatan dari masing masing bagian agar dapat melakukan tugasnya tanpa mengalami kegagalan. Maka dari itu dipilihlah material-material yang kuat namun tetap ulet untuk memberikan toleransi gerak dari gaya yang ditimbulkan oleh pegas.

III. BAGIAN ELEKTRONIS

Dalam penggunaannya, sistem penendang mekanis ini menggunakan input tegangan pada motor sebesar 12 V. Untuk menghentikan putaran saat mencapai pertambahan panjang pegas maksimum digunakan *limit switch* dan dari data yang di terima akan dikirim ke *microcontroller*. Kemudian setelah robot siap menendang, motor diputar lagi yang mengakibatkan pegas kembali ke posisi semula atau tidak ada pertambahan panjang. Itu terjadi secara cepat karena itu adalah proses lepasnya struktur profil dari pengait.

Dengan pemasangan bagian elektronis yang tidak terlalu rumit membuat sistem penendang ini menjadi lebih ringkas dan mudah dalam instalasinya.

IV. HASIL PERCOBAAN DAN ANALISIS

Percobaan sangat perlu dilakukan dalam menentukan kekuatan dari sistem penendang mekanis ini. Teknik yang digunakan untuk melakukan percobaan yaitu dengan memasang semua bagian, kemudian dicoba dengan menggunakan variasi pegas dengan kekakuan yang berbeda-beda. Kemudian diimbangi dengan perbaikan sambungan struktur.Hasil percobaan menunjukkan bahwa kekakuan pegas akan mempengaruhi kekuatan struktur yang digunakan.

V. KESIMPULAN

Paper ini menjelaskan tentang inovasi penendang pada robot sepakbola beroda yang dikonteskan pada Kontes Robot Indonesia. Sistem penendang ini mengadopsi prinsip kerja piston yang mengubah gerak rotasi menjadi gerak translasi. Untuk menyimpan energi potensial digunakan pegas dengan besar konstanta pegas yang disesuaikan dengan kekuatan yang diinginkan serta kekuatan struktur dinamis yang dipakai. Dengan input tegangan sebesar 12 V untuk memutar motor penggerak maka akan menghasilkan energi yang digunakan untuk menarik pegas dengan konstanta tertentu, serta struktur dinamis yang dibuat sedemikian dapat menghasilkan gaya untuk menendang bola. Pada pengembangan selanjutnya diharapkan dapat menendang dengan arah melambung dan memiliki kecepatan yang diharapkan.

REFERENSI

[1] Kontes Robot Indonesia 2017,"Kontes Robot Indonesia (KRI)" [online]. Available : https://www.kontesrobotindonesia.org/kri-2017/

[2] Foster Bob, Fisika Terpadu 2004:122-123.

[3] Hibbeler R.C, Engineering Mechanic Dynamic 12nd ed, 2010.

[4] Budynas Richard G, Nisbett J Keith, Mechanical Engineering Desain, 10th ed, 2014.

Desain Solenoida Sebagai Mekanisme Penendang Robot Sepak Bola Beroda

Muhammad Hamam Afandi*, Ma'ruf Shidiq, Linggom Enrico, dan Denanda Hendra
Universitas Gadjah Mada
Yogyakarta, Indonesia
Email: hamam.afandi@gmail.com

***Abstract*—In this study, a solenoid system is designed as football kicking actuator for mobile robotic application. The actuator is used to shoot a regular sized soccer ball by applying different force levels directly to the ball. A capacitor bank is used to buffer energy from the battery trough DC-DC converter. By using insulated gate bipolar transistor, the energy is transferred from the capacitor to the actuator. Consequently, by applying pulse width modulation, the force applied by the actuator can be adjusted to enable a variable shooting power.This design is capable to shoot a ball with a maximum starting speed of 5,68 m/s**

Keywords— solenoid mechanism; transient analysis; variable shooting power

I. PENDAHULUAN

Kontes Robot Indonesia (KRI) adalah kegiatan kontes bidang robotika yang dapat diikuti tim mahasiswa dari institusi atau Perguruan Tinggi Negeri dan Swasta yang terdaftar di Kementerian Riset, Teknologi dan Pendidikan Tinggi [1].

KRSBI Beroda merupakan divisi terbaru pada KRI yang merupakan pengembangan dari KRPAI Beroda. Aturan kompetisi mengacu pada RoboCup Middle Size League (MSL). Ukuran panjang dan lebar robot antara 30 hingga 52 cm, tinggi antara 40 hingga 80 cm. Robot bermain di lapangan pertandingan berukuran 6 x 9 m. Bola menggunakan bola futsal yang diberi warna orange (Ukuran No.4, ukuran lingkar 62-64 cm, berat 390-430 gram). Sensor yang digunakan harus berada pada robot. Tidak ada sensor yang diletakkan di luar robot (tidak ada kamera di atas lapangan) [2].

Paper ini fokus pada pembahasan mekanisme penendang. Untuk mendefinisikan kebutuhan sistem tendangan, pemodelan lanjut seperti massa bola, diameter bola perlu dilakukan. Selain itu mekanisme tendangan juga harus cukup kuat untuk menendang bola futsal dengan kecepatan yang dibutuhkan oleh tim.



Gambar 1. Robot sepak bola beroda tim FUKURO UGM.

Berdasarkan kebutuhan robot yang akan digunakan untuk menendang bola, sistem pneumatis terlihat bisa menjadi solusi. Namun agar cukup kuat untuk menendang bola, maka dibutuhkan ruang yang banyak untuk menampung botol yang berisi udara sebagai bagian dari mekanisme pneumatis tersebut. Selain sistem pneumatis, penggunaan motor BLDC (*Brushless Direct Current Motor*) juga bisa menjadi pilihan. Motor BLDC dapat digunakan untuk menarik pegas kemudian pegas dilepaskan untuk menendang bola. Namun kelemahan sistem ini adalah tendangan tidak bisa dikontrol dan kekuatan tendangan akan menurun seiring berjalannya waktu karena daya tarik pegas lama kelamaan akan berkurang. Motor BLDC juga dapat digunakan untuk mengayunkan batang yang ujungnya digunakan untuk menendang bola. Tetapi sistem ini akan memakan banyak tempat pada robot dan tendangan susah dikendalikan. Supaya robot mampu menendang bola futsal dalam beberapa variasi level tendangan dan sistem tendangan tidak memakan banyak tempat maka tim robot sepak bola beroda UGM "FUKURO" memilih mekanisme penendang dengan menggunakan solenoida. Gambar 1 menunjukan robot KRSBI Beroda UGM "FUKURO".

II. DESAIN SISTEM TENDANGAN SOLENOIDA

A. Kebutuhan Energi

Ukuran lapangan yang digunakan pada pertandingan adalah 6 x 9 m. Anggap robot penjaga gawang lawan membutuhkan waktu 0.5 detik untuk merespon pergerakan bola. Sehingga untuk mencetak gol dari tengah lapangan bola harus bergerak dengan kecepatan lebih dari 9 m/s, anggap bola solenoida harus mampu menggerakkan bola dengan kecepatan (v_{ball}) 10 m/s. Untuk menghitung energi yang dibutuhkan solenoida untuk menggerakkan bola dengan kecepatan awal 10 m/s dilakukan perhitungan sebagai berikut [3]:

$$E_{ball} = \frac{1}{2} m_{ball} {v_{ball}}^2 + \frac{1}{2} J_{ball} {\omega_{ball}}^2 \quad (1)$$

$$v_{ball} = 10[\tfrac{m}{s}]$$

$$m_{ball} = 0{,}43[kg]$$

$$r_{ball} = 0{,}11[m]$$

$$J_{ball} = \frac{2}{3} m_{ball} {r_{ball}}^2 = 3{,}4687.10^{-3}[kg.m^2] \quad (2)$$

$$\omega_{ball} = \frac{v_{ball}}{r_{ball}} \approx 91[\tfrac{rad}{s}] \quad (3)$$

Sehingga energi yang yang dibutuhkan

$$E_{ball} \approx 35{,}9[J]$$

Sesuai hukum konservasi energi maka

$$E_{solenoid} = E_{ball} = 35{,}9[J] \quad (4)$$

B. Rangkaian Elektronis

Robot memiliki 2 baterai sebagai sumber tegangan yaitu baterai Lippo 6 sel 22,5 V dan baterai Lippo 3 sel 12 V. Baterai 22,5 V digunakan untuk menyuplai motor penggerak robot sementara baterai 12 V digunakan untuk keperluan solenoid sebagai penendang. Gambar 2 menunjukan bentuk *hardware* dari konverter DC-DC, rangkaian kapasitor bank dan solenoida.

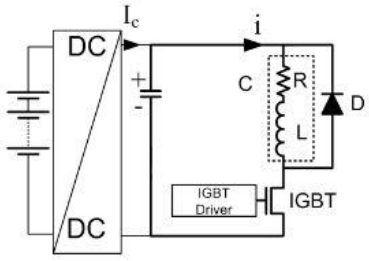


Gambar 2. Konverter dc-dc, kapasitor bank, solenoid.

Karena robot harus mampu menendang bola futsal maka dibutuhkan energi puncak yang cukup besar. Kapasitor dengan nilai kapasitansi (C) 5 mF dengan rating 450 V digunakan untuk penyimpanan energi sementara. Kapasitor diisi (*charging*) menggunakan baterai yang tegangannya (V) telah dinaikkan menjadi 311 V dengan menggunakan DC-DC converter. Energi yang tersimpan dalam kapasitor (E_C) ditunjukan pada persamaan (5) :

$$E_C = \frac{1}{2}CV^2 = 241{,}8025\,[J] \quad (5)$$

Energi yang tersimpan dalam kapasitor tersebut akan dilepas (*discharge*) melalui solenoid. IGBT (*Insulation Gate Bipolar Transistor*) digunakan untuk mengontrol arus yang melewati kumparan solenoid dengan nilai induktansi (L) 20 mH dan resistansi (R) 5 Ω. Dengan menggunakan sinyal lebar pulsa / *pulse width modulation* (PWM) untuk mengontrol IGBT, energi yang ada pada solenoida juga dapat dikontrol. Untuk menghindari tegangan *spike* ketika IGBT *switch off*, digunakan *flyback diode* (D) yang dipasang parallel dengan solenoida seperti yang terlihat pada Gambar 3.



Gambar 3. Rangkaian elektronis sistem solenoida [4].

Ketika IGBT menutup rangkaian elektronis dapat dianggap sebagai osilator teredam dengan persamaan diferensial orde dua sebagai berikut [4] :

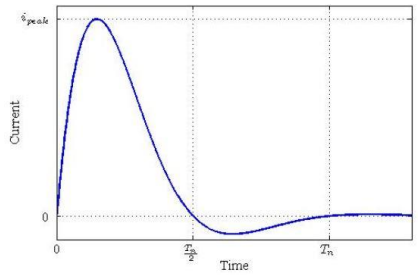
$$L\frac{d^2i}{dt^2} + R\frac{di}{dt} + \frac{1}{C}i = 0 \quad (6)$$

Bentuk tipikal respon dari rangkaian tersebut digambarkan pada Gambar 4 dimana T_n merupakan *invers* dari frekuensi natural yang teredam [5]. Frekuensi natural teredam (ω_d) didefinisikan sebagai :

$$\omega_d = \sqrt{1-\zeta^2}\sqrt{\frac{1}{LC}} \quad (7)$$

Dimana ζ adalah faktor redaman

$$\zeta = \frac{R}{2\sqrt{\frac{L}{C}}} \quad (8)$$



Gambar 4. Arus yang melewati solenoid ketika IGBT menutup [4].

III. SIMULASI TRANSIEN

Sebuah tim robot tentu memerlukan level tendangan yang berbeda-beda sehingga tendangan dapat disesuaikan ketika robot akan melakukan operan ke sesama anggota tim (*passing*) dan menendang bola kearah gawang lawan (*shooting*) sesuai kebutuhan. Untuk mendapatkan tendangan dengan level yang berbeda-beda digunakan PWM untuk mengontrol IGBT. Pengontrolan dilakukan dengan mengatur interval waktu ketika IGBT menutup sehingga arus puncak pada solenoid akan berbeda-beda.

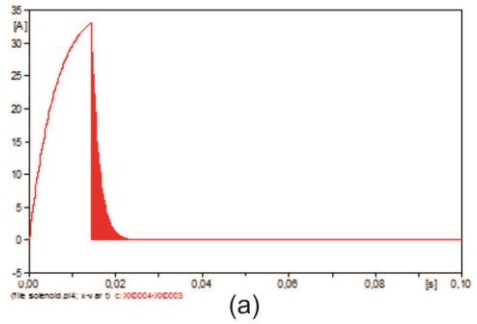
Tendangan maksimum didapat dengan memberikan PWM yang memiliki periode 16 ms dengan duty-cycle 90%. Level tendangan dibawahnya duty-cycle diubah menjadi 80% begitu seterusnya sampai tendangan level terendah IGBT di control dengan PWM berduty-cycle 10%.

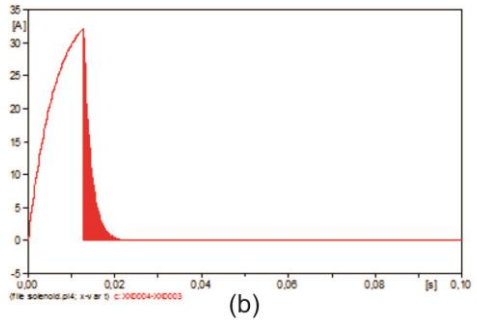
IV. HASIL DAN PEMBAHASAN

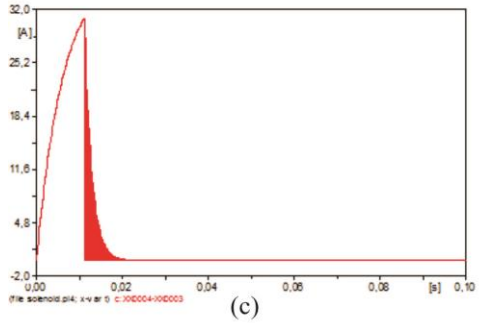
Setelah dilakukan simulasi transien dengan menggunakan Atpdraw didapatkan data sebagai berikut :

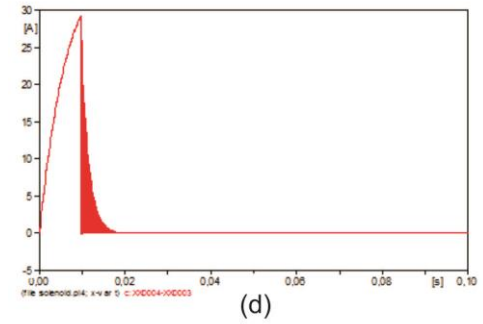
TABEL I. PERFORMA SOLENOIDA

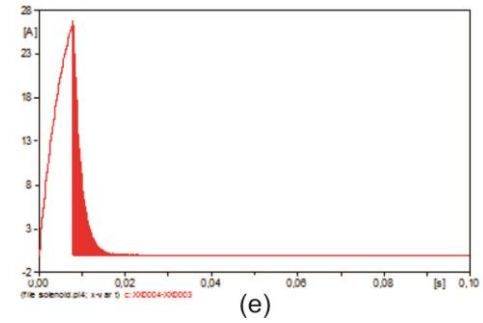
No	Hasil simulasi transien dengan Atpdraw		
	Duty-cycle PWM	Arus puncak	Energi puncak
1	90%	34 A	11.56 J
2	80%	32.5 A	10.5625 J
3	70%	30.86 A	9.5234 J
4	60%	29 A	8.41 J
5	50%	27 A	7.29 J
6	40%	24 A	5.76 J
7	30%	20 A	4 J
8	20%	14.5 A	2.1025 J
9	10%	8.3 A	0.6889 J



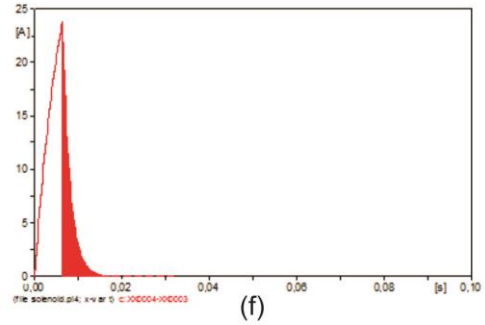


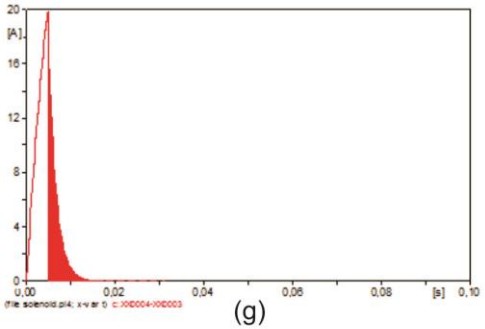


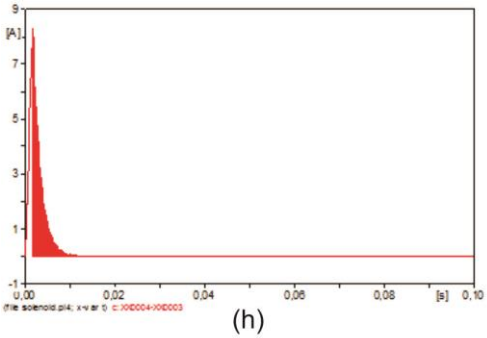


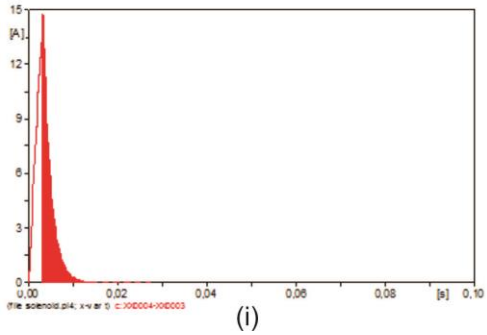


Gambar 5. Arus melewati solenoida dengan simulasi Atpdraw saat *duty-cycle* PWM : (a) 90% , (b) 80%, (c) 70%, (d) 60%, (e) 50%.









Gbr. 6. Arus melewati solenoida dengan simulasi Atpdraw dengan *duty-cycle* PWM : (f) 40% , (g) 30%, (h) 20%, (i) 10%.

Hasil simulasi Atpdraw ditunjukan oleh Gambar 5 dan Gambar 6. Dari data yang didapat diketahui jika kecepatan bola maksimum saat ini adalah 5,68 m/s. Hal ini belum mencapai target awal yakni bola mampu melesat dengan kecepatan maksimum 10 m/s. Sehingga untuk mencetak gol bola harus ditendang dalam jarak yang kurang dari 2,84 m kea rah gawang lawan dengan asumsi robot penjaga gawang lawan mampu merespon pergerakan bola dalam waktu 0,5 detik.

V. KESIMPULAN

Tujuan dari studi ini adalah untuk mendapatkan mekanisme tendangan dengan sistem solenoida yang memiliki ukuran minimum dan mampu melakukan variasi tendangan dalam beberapa level.

Dari hasil simulasi variasi tendangan dalam beberapa level sudah didapatkan. Namun untuk melakukan tendangan agar bola melesat dengan kecepatan maksimum 10 m/s masih belum tercapai. Untuk saat ini kecepatan bola maksimum yang dapat dicapai adalah 5,68 m/s. Sehingga untuk mencetak gol bola harus ditendang dalam jarak yang kurang dari 2,84 m dengan asumsi robot penjaga gawang lawan membutuhkan waktu 0,5 detik untuk merespon pergerakan bola.

Tentu saja mekanisme tendangan solenoida ini masih bisa dikembangkan lagi. Energi solenoida dapat ditingkatkan dengan membuat solenoida yang memiliki nilai induktansi lebih besar atau dengan membuat arus puncak yang lebih tinggi dengan cara menaikkan level tegangan pada kapasitor bank dan atau menurunkan nilai resistansi dan impedans dari sistem.

REFERENSI

[1] Kontes Robot Indonesia 2017, "Kontes Robot Indonesia (KRI)" [online]. Available : https://www.kontesrobotindonesia.org/kri-2017/

[2] Kontes Robot Indonesia 2017, "KRSBI Kontes Robot Sepak Bola Indonesia Beroda"[online].Available : https://www.kontesrobotindonesia.org/kri-2017/

[3] J.L. Meriam, L.G. Kraige, Engineering Mechanics Dynamics,4th edition.

[4] Meessen, K.J., Paulides, J.J.H., Lomonova, E. "A Football Kicking High Speed Actuator for a Mobile Robotic Application," In: Proceedings of the 36th Annual Conference of the IEEE Industrial Electronics Society, pp. 1659-1664, 2010.

[5] W. J. Palm, "Modeling analysis and control of dynamic system," John Wiley and Sons, 1994.

Rancang Bangun Lengan Robot Pendeteksi Api Menggunakan Metode Pengolahan Citra

Try Ahmad Agus*, Rama Okta Wiyagi, Anna Nur Nazilah Chamim, dan Danardono
Jurusan Teknik Elektro, Fakultas Teknik
Universitas Muhammadiyah Yogyakarta
Yogyakarta, Indonesia
3ahmadagus@gmail.com

***Abstrak*—Robot pendeteksi api merupakan robot yang digunakan untuk mendeteksi api. Perancangan sistem pada robot pendeteksi api tergolong kompleks yang terdiri dari sistem mekanik, sistem elektronik dan sistem kontrol. Dalam mendeteksi titik api, robot pendeteksi api memerlukan beberapa sensor seperti Thermophile Array, *photodiode*, UV sensistive sensor dan lain sebagainya. Namun beberapa sensor tersebut memiliki kelemahan dalam pembacaan nilai sensor dan terpengaruh oleh rentang jarak. Untuk menggantikan sensor-sensor tersebut digunakan sebuah kamera. Penggunaan kamera sebagai sensor pemindai gambar, tidak terlepas dari kemampuan kamera yang dapat mengidentifikasi objek berupa bentuk atau warna dengan metode pengolahan citra. Oleh karena itu dirancang sebuah lengan robot pendeteksi api menggunakan metode pengolahan citra dengan pengendali utama berupa *Single Board Computer* (SBC) Raspberry Pi 3 model B dengan pustaka OpenCV. Dengan menggunakan modul kamera Raspberry Pi Cam v2 sebagai sensor pemindai gambar yang diletakan pada lengan robot, maka sistem dapat melakukan pendeteksian dan mengikuti objek (titik api) dengan baik.**

Kata Kunci—lengan robot; deteksi api; pengolahan citra; kamera, OpenCV; single board computer (SBC)

I. PENDAHULUAN

Robot pendeteksi api merupakan robot yang digunakan untuk mendeteksi api. Perancangan sistem pada robot pendeteksi api sangat kompleks yang terdiri dari sistem mekanik, sistem elektronik dan sistem kontrol. Sistem mekanik robot berupa fungsi gerak atau derajat kebebasan. Sistem elektronik robot berupa rangkaian elektronika. Sistem kontrol berupa program untuk mengontrol pergerakan robot.Dengan sistem yang kompleks pada robot pendeteksi api, diperlukan banyak sensor agar robot pendeteksi api bisa bekerja secara maksimal. Dalam mendeteksi titik api, robot pendeteksi api memerlukan beberapa sensor seperti *Thermophile Array Sensor, photodiode, UV-Tron* dan lain sebagainya.

Sensor-sensor tersebut memiliki kelemahan dalam pembacaan nilai, terpengaruh oleh cakupan sudut dan jarak. Untuk menggantikan sensor-sensor tersebut digunakan sebuah kamera. Penggunaan kamera sebagai sensor pemindai gambar, tidak terlepas dari kemampuan kamera yang dapat mengidentifikasi objek berupa bentuk atau warna dengan metode pengolahan citra sekaligus.

Oleh karena itu dirancang sebuah lengan robot pendeteksi api menggunakan metode pengolahan citra dengan pengendali utama berupa *single board computer* (SBC) Raspberry Pi 3 model B dengan pustaka OpenCV. Dengan menggunakan modul kamera Raspberry Pi v2 sebagai sensor pemindai gambar yang diletakan pada lengan robot.

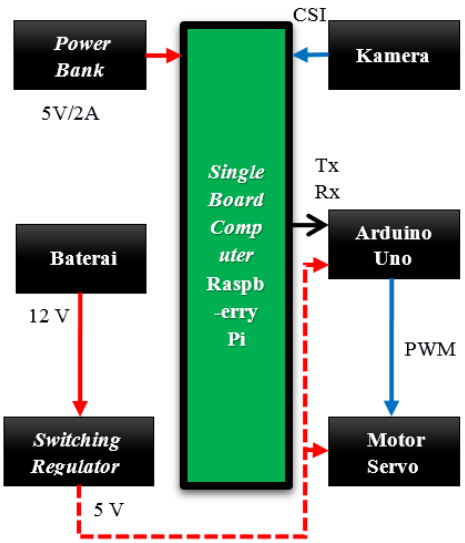
II. TUJUAN

Merancang dan membuat lengan robot pendeteksi api menggunakan metode pengolahan citra dengan kamera sebagai sensor.

III. PERANCANGAN SISTEM

A. Perancangan Perangkat Keras

Sistem pendeteksian titik api lilin pada lengan robot dibangun menggunakan beberapa perangkat keras dan perangkat lunak. Adapun perangkat yang digunakan dapat dilihat pada Gambar 1.

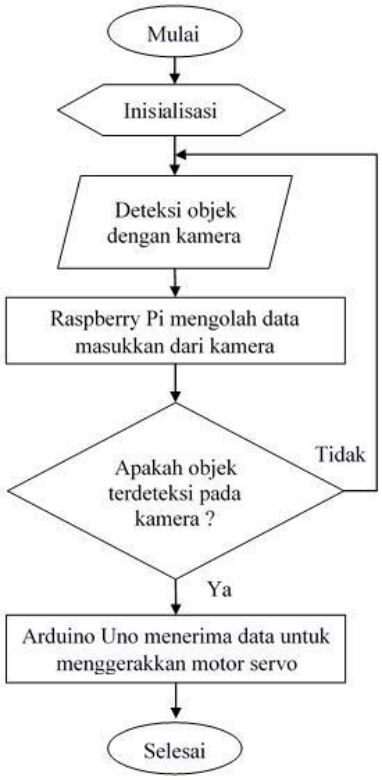


Gambar 1. Blok diagram rangkaian.

Desain prototipe perancangan dalam gambar perspektif dan penempatan perangkat keras ditunjukan pada Gambar 1.

B. *Perancangan Perangkat Lunak*

Secara garis besar, tahap awal sistem bekerja dengan mendapatkan informasi citra dari kamera. Citra yang didapat diolah pada Raspberry Pi. Pengolahan citra dilakukan untuk mendeteksi objek berupa titik api lilin dan mendapatkan posisi absolut objek pada frame citra. Data posisi objek akan digunakan untuk menentukan posisi sudut servo [1]. Saat titik api lilin terdeteksi maka Raspberry Pi akan mengirim data ke arduino menggunakan komunikasi serial UART. Arduino digunakan sebagai pengendali servo. Arduino akan mengirimkan data untuk menggerakkan servo sesuai posisi pada objek (titik api lilin). Seperti terlihat pada Gambar 2, diagram alir perangkat lunak berikut.



Gambar 2. Diagram alir perangkat lunak.

Pengolahan citra pada SBC Raspberry Pi diawali dengan melakukan konversi citra yang didapat oleh kamera kedalam ruang warna HSV (*Hue, Saturation, Value*). Ruang warna HSV dipilih agar sistem mudah melakukan deteksi berdasarkan presepsi. Setelah melakukan konversi ke dalam ruang warna HSV dilakukan *thresholding* pada citra dengan menggunakan nilai rentang HSV yang sesuai dengan objek deteksi [2]. Nilai rentang HSV ini didapatkan dengan menguji objek deteksi pada perangkat lunak seleksi HSV. Hasil tahap *thresholding* ini adalahseleksi objek deteksi. Objek deteksi yang berupa titik api lilin akan diberi nilai 1 yang ditandai dengan dengan warna putih, sedangkan selain titik api lilin akan diberi nilai 0 ditandai dengan warna hitam. Untuk mengurangi *noise* yang dihasilkan pada tahap *thresholding,* diterapkan teknik erosi dan dilatasi pada citra [3][4]. Setelah didapatkan seleksi citra objek deteksi yang baik, tahap selanjutnya adalah melakukan deteksi dan pengenalan objek dengan menganalisis bentuk objek [5]. Tahap ini diterapkan dengan fungsi *contours*. Untuk mendapatkan koordinat objek titik api lilin pada frame citra dilakukan dengan mengevaluasi momen citra setelah tahap *contours*.

Pada tahap ini, nilai koordinat objek yang terdeteksi akan diambil sebagai data untuk menggerakan lengan robot sesuai dengan Persamaan 1.

$$servo\ angle = \frac{(x - in\ Min_{Pixel}) * (out\ Max_{Servo} - out\ Min_{Servo})}{(in\ Max_{Pixel} - in\ Min_{Pixel}) + out\ Min_{Servo}} \quad (1)$$

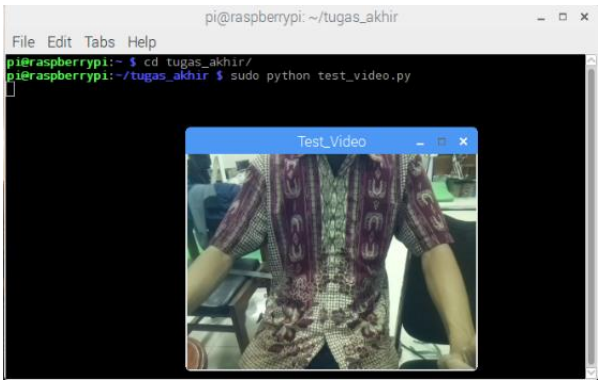
Resolusi citra yang digunakan pada sistem deteksi ini adalah 320 x 240 piksel. Persamaan diatas akan melinierisasi nilai koordinat posisi objek dengan sudut servo. Servo yang digunakan pada lengan robot sebanyak dua buah. Masing-masing servo merepresentasikan sumbu horizontal dan vertikal. Dengan persamaan diatas nilai koordinat objek titik api lilin digunakan untuk mengerakan servo sehingga lengan robot dapat selalu memposisikan lengan mengikuti titik api lilin.

IV. HASIL UJI COBA DAN PEMBAHASAN

Hasil yang diperoleh dalam penelitian ini adalah lengan robot pendeteksi api yang dapat mendeteksi api dengan menggunakan metode pengolahan citra. Lengan robot yang memiliki dua derajat kebebasan ini juga mampu mengikuti objek berupa titik api lilin dengan baik.

A. *Hasil Pengujian Modul Kamera Raspberry Pi v2*

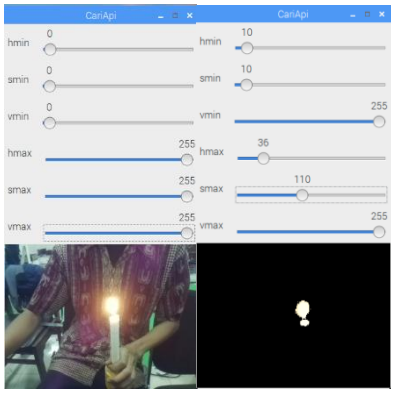
Pengujian digunakan untuk mengambil video secara *realtime*. Kamera Raspberry Pi v2 dipasang paad *single board computer* Raspberry Pi. Pengujian kamera dilakukan dengan menjalankan perintah pada terminal yang berjalan pada sistem operasi raspbian pada Raspberry Pi. Gambar 3 menunjukan tampilan keluaran ketika pengujian dilakukan.



Gambar 3. Pengujian modul kamera Raspberry Pi.

B. *Pengujian Nilai HSV*

Pengujian ini dilakukan untuk mengetahui nilai HSV oada titik api lilin untuk mengetahui objek dapat terdeteksi dengan baik. Perangkat lunak yang digunakan untuk mendapatkan nilai rentang HSV ditunjukan pada Gambar 4. Sedangkan TABEL I menunjukan perbandingan nilai HSV Titik Api Lilin sesudah dan sebelum pengujian.



Gambar 4. Pengujian nilai HSV titik api lilin.

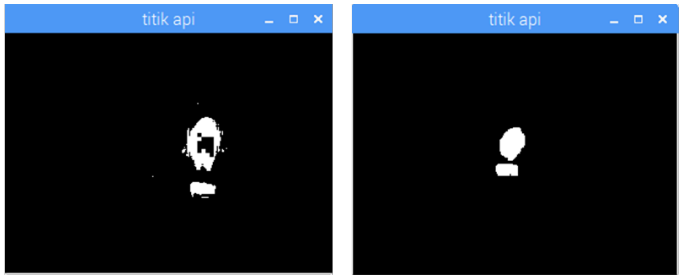
TABEL I. HASIL UJI COBA MENCARI NILAI HSV

Parameter	Nilai Sebelum	Nilai Sesudah
Nilai Hmin	0	10
Nilai Smin	0	10
Nilai Vmin	0	255
Nilai Hmax	255	36
Nilai Smax	255	110
Nilai Vmax	255	255

Nilai sebelum merupakan hasil citra yang ditangkap oleh modul kamera. Dengan kata lain citra normal sebelum dilakukan pengolahan citra. Sedangkan nilai sesudah merupakan nilai setelah melewati proses pengolahan citra dengan melakukan pengaturan pada setiap komponen nilai HSV hingga diperoleh nilai HSV objek titik api lilin.

C. *Pengujian Objek Terdeteksi*

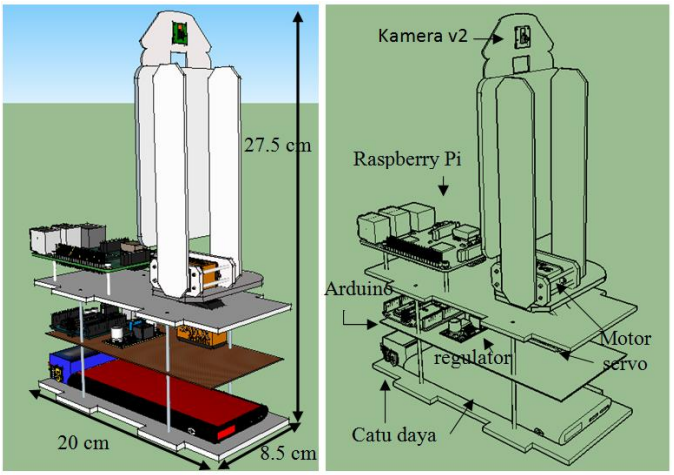
Citra yang ditangkap oleh kamera kemudian diproses dengan melakukan filter untuk mengurangi *noise*. Filter diterapkan dengan mengaplikasikan metode erosi dan dilatasi. Gambar 5 menunjukan hasil citra sebelum dan sesudah dilakukan filter. Setelah dilakukan pemfilteran, dicari konturterluar dari objek terdeteksi.



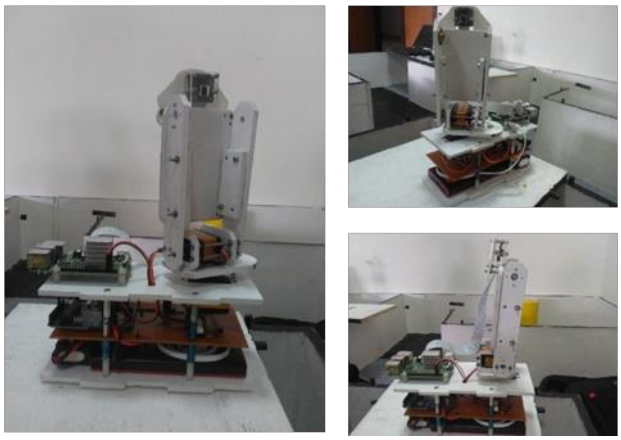
Gambar 5. Hasil pengolahan citra titik api lilin.

D. *Pengujian Keseluruhan*

Kemampuan pendeteksian objek juga dipengaruhi kondisi cahaya sekitar saat pengujian. Pengujian dilakukan dengan mengukur terangnya cahaya pada ruangan menggunakan lux meter kemudian dibandingkan dengan jarak yang dapat terdeteksi. Pada TABEL II menunjukan pengaruh cahaya ruang terhadap kemampuan deteksi sistem terhadap objek. Dengan nilai lux yang semakin besar (terang) maka kemampuan kamera menangkap objek terdeteksi semakin baik.



Gambar 6. Sketsa prototipe sistem.



Gambar 7. Realisasi pembuatan lengan robot.

TABEL II. PERBANDINGAN TERANG CAHAYA DENGAN JARAK OBJEK TERDETEKSI

Cahaya pada ruang	Jarak terdeteksi (cm)
232 lux (Terang)	70
198 lux (Sedang)	100
16,61 lux (Gelap)	130

V. KESIMPULAN

Berdasarkan analisis dan hasil pengujian dapat diambil kesimpulan sebagai berikut :

- Sistem dapat melakukan pendeteksian objek berupa titik api lilin dengan menggunakan kamera sebagai input sistem.
- Dengan menggunakan 2 derajat kebebasan pada lengan robot, lengan robot pendeteksi api bisa bergerak menuju titik api lilin pada sumbu horizontal dan vertikal.
- Lengan robot pendeteksi titik api lilin mampu mendeteksi objek sampai dengan jarak 130 cm pada ruang dengan cahaya sekitar 16,6 Lux.

REFERENSI

[1] J. Iovine, PIC Robotics. McGraw-Hill, New York, 2004.
[2] R. O. Wiyagi, I. Soesanti, and A. Susanto, “Identifikasi Titik Api Lilin Berbasis Nilai HSV , Threshold dan Momen Citra untuk Aplikasi Robot Pemadam Api,” Jurnal Semesta Teknika, vol. 17, no. 1, pp. 38–44, Mei 2014.
[3] H. Zhou, J. Wu, and J. Zhang, "Digital Image Processing: Part II", 1st ed. 2010.
[4] R.F.Gusa, “Pengolahan Citra Digital untuk Menghitung Luas Daerah Bekas Penambangan Timah Rika Favoria Gusa,” Jurnal Nasional Teknik Elektro, vol. 2, no. 2, pp. 27–34, 2013.
[5] R. O. Wiyagi and M. Y. Mustar, “Deteksi Jarak Objek Bercahaya Secara Real Time Menggunakan Kamera Tunggal,” The 3rd Indonesian Symposium on Robot Soccer Competition 2015, vol. 3, pp.138-141, June 2015.

PID Controller-based Object Tracking and Speed Control System for Wheeled Soccer Robot

Arbayu Maulana*, Wisesa Wibisono, Rizki Adhi Pamungkas, Indra Riyanto, Akhmad Musafa, dan Sujono
Dept. of Electrical Engineering, Fac. of Engineering Center for Robotics Studies Image and Spatial Analysis Research Center for Environmental Studies
Universitas Budi Luhur
Jakarta, Indonesia
1352500043@student.budiluhur.ac.id

***Abstract*—This paper discussed about the detection system of ball objects and speed control system of wheeled robot football using PID controller. The object detection system uses behavior-based control methods based on the orientation of the robot position against the object position. The position of the robot is controlled to keep it perpendicular to the ball. When the position of the robot is perpendicular to the ball, the robot will move with a certain speed that is controlled by the PID controller based on the distance between the robot and the ball. Test results in static conditions (static balls) show the robot can detect the object of the ball, either when the position of the object perpendicular, to the right, or left of the robot. In testing the dynamic state (moving ball) with random directions, the robot can also move dynamically as the ball position changes.**

Keywords—object tracking; position control; speed control; PID controller; wheeled soccer robot

I. INTRODUCTION

One of the abilities that must be owned by a wheeled soccer robot is to find the ball position, move towards the ball position, then hold the ball to further dribble and kick the ball into the opposing goal. To complete this task, the robot must be equipped with a camera sensor and ball position detection algorithm as well as setting the speed of movement to the ball position. The position of the ball in a wheeled soccer wheeled game is dynamic, meaning the ball moves and the position of the ball can move. For object detection, camera position is controlled to follow moving objects automatically by using PID controller. The camera must be controlled with the correct movement speed in order to produce the best detection results. If camera movement is too slow, the camera will lose the object to be captured, whereas if it is too fast, the captured image will be damaged [1].

The environment around the wheeled soccer robot is dynamic, where the position of the opposing robot and the ball position constantly changing. Therefore it is necessary to apply a control system on robots that have the ability to read these dynamic environmental conditions. One of them is using behavior-based control and fuzzy logic controller on football robot navigation system. The robot is designed to have the ability to find the target (ball), avoid obstacles and kick the ball into the opposing goal [2]. In order for the wheeled soccer robot to move as freely as possible in anticipating dynamic environmental conditions, the movement of the robot must be designed in such a way, including the use of omni-directional drive system. With this drive system the position of the robot can be controlled with algorithms such as fuzzy logic, so that the movement of the robot can be more free and flexible. This greatly helps the robot in tracking down and avoiding all obstacles [3].

Based on the conditions described earlier, this paper discusses the detection system of ball objects and speed control systems on wheeled soccer robots. Object detection systems use behavior-based control methods, while speed control systems use PID controllers.

II. SYSTEM DESIGN

A. Mechanical Design

This wheeled soccer robot is designed with length of 51 cm, width of 43 cm and height of 40 cm. The robot body is made with the main material of a square aluminum pipe with a size of 10 mm × 20 mm and 3 mm and 5 mm thick acrylic sheets. Robotic mechanics are made into 2 levels: first level for battery and wheel drive motor, second level for sensors, electronic circuit, and gripper. The mechanical design of the robot is shown in Fig.1. The wheel of a soccer robot uses three omnidirectional wheels for the robot to move in any direction, the principle of robotic movement shown in Fig.2.

B. The block diagram of the system

The system consists of a camera (CMUCam5) used as a ball detection sensor, an ultrasonic sensor to measure the distance of the robot to the ball, the Arduino Mega2560 as the system controller, and the DC motor and its driver to drive the robot wheel. The control system block diagram of the wheeled soccer robot is shown in Fig.3.

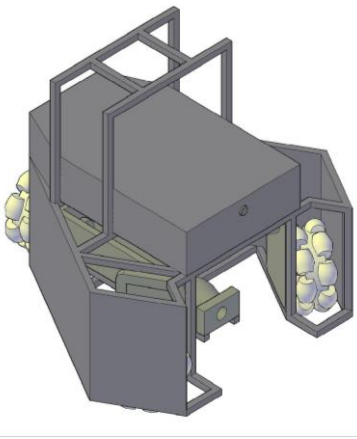


Fig. 1. Mechanical design of robot

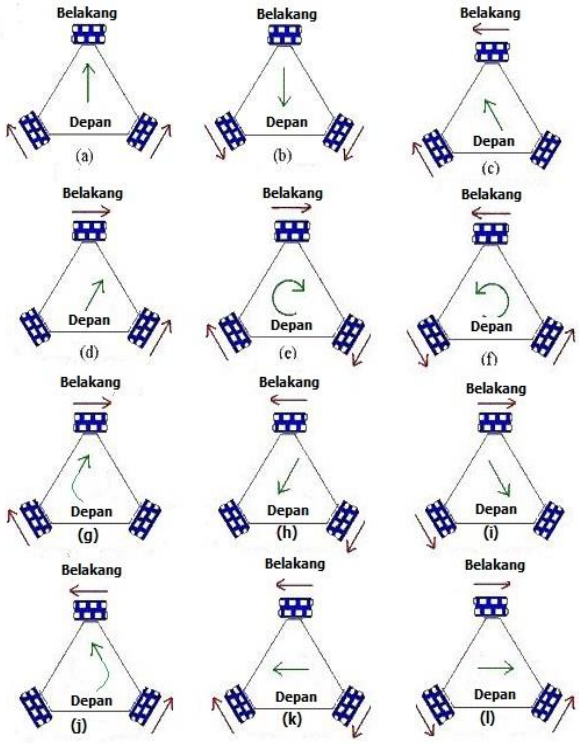


Fig. 2. The concept of robot movement with omnidirectional drive system

C. *The principle of ball detection*

When the robot is activated, the CMUcam5 camera sensor will be active and look for a ball that is placed arbitrarily. CMUcam5 has a reading angle range of 75° (θ_1). If the robot does not find the ball, the robot will rotate to the right to find the ball. If the robot detects the ball, the robot will move to the right or left by θ_2 to make the robot camera's point of view perpendicular to the ball. The concept of ball detection by the robot is shown in Fig.4. Once the ball is detected and centered, the ultrasonic sensor will measure the distance between the robot and the ball (path c). If the distance of the robot with the ball is smaller or equal to 5 cm, the motor gripper will immediately move to hold the ball, while if the ball is more than 5 cm, the robot will move towards the ball. Robot movement speed towards the ball will be maximum (PWM wheel motor value = 255) when the distance of the robot with the ball more than 50 cm. The speed of the robot will be reduced when the distance of the robot to the ball is equal to 50 cm, until finally the robot will stop when the distance of the robot with the ball is equal to 5 cm. The speed of wheel motors based on the distance between the robot and the sphere is set by using the PID controller.

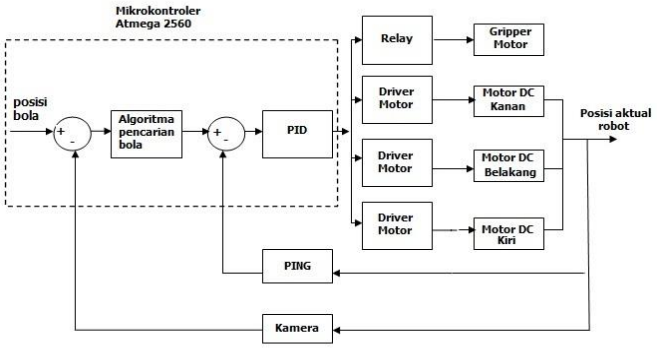


Fig. 3. The block diagram of a wheeled soccer robot controls

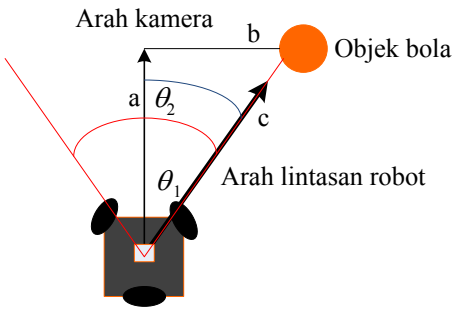


Fig. 4. The concept of object detection

D. *Speed Control System use PID Controller*

Then the ultrasonic sensor will measure the distance between the robot with the ball (path c in Fig.4). If the distance of the robot with the ball is smaller or equal to 5 cm, the motor gripper will immediately move to hold the ball. If not, if the robot's distance with the ball is more than 5 cm, the robot will move towards the ball. The speed of movement of the robot to the ball will be maximum (PWM value for motor driver is 255) when the distance of the robot with the ball more than 50 cm. The speed of robot movement will decrease when the distance of the robot to the ball is equal to 50 cm, and continue to decrease in proportion to the reduced distance of the robot with the ball. The robot will stop when the distance of the robot with the ball is 5 cm. This speed control is done based on the distance of the robot with the ball, using the PID controller.

The structure of pid controller used is a parallel structure, show in Fig.5. r(t) is the set point of distance between the

robot and the ball, c(t) is the actual distance between the robot and the ball, that measured use ultrasonic sensor, and u(t) is the PWM signal of the PID controller, to control the motor speed of the robot wheel drive. Kp, Ki and Kd are the parameter of PID controller.

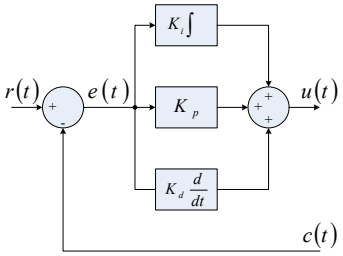


Fig. 5. The block diagram of Parallel PID Controller

The control signal of the PID controller is expressed by (1).

$$u(t) = K_p.e(t) + K_i\int e(t)dt + K_d\frac{de(t)}{dt} \quad (1)$$

where $u(t)$ = output of PID Controller (PWM signal)
$e(t)$ = error signal
K_p = Proportional gain of PID Controller
K_i = Integral gain of PID Controller
K_d = Differential gain of PID Controller

In its field implementation for wheeled soccer robots, the concept of ball detection is shown in Fig.6.

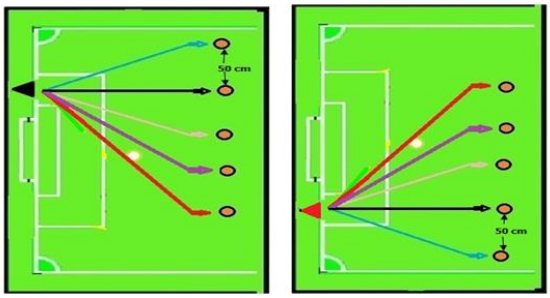


Fig. 6. The concept of detecting a ball object

The concept of search and the detection of ball objects in implementation on the match field for wheeled soccer robots is shown in Fig.5. The wheeled football robot is placed on the right or left side of the goal in a straight position against the ball and then the robot moves to the specified ball position, for example the spacing between position-1 and position-2 is 50 cm.

E. Electronics Systems

The electronic control system on the robot uses Arduino Mega2560 board with clock of 16 MHz. Detection of objects using CMUCam5 camera. CMUCam5 detects the centroid of the object, in this case is an orange ball. The output of the camera is used in the Arduino program to determine if the robot has found the ball or not. The microcontroller system circuit also receives input from the ultrasonic sensor to detect the distance between the robot and the ball. The ultrasonic sensor detection results and data from the camera sensors are used as a reference for the robot position. The output of the microcontroller system is connected to the motor driver circuit. The design of electronics control systems in robots is shown in Fig. 7, 8, and 9.

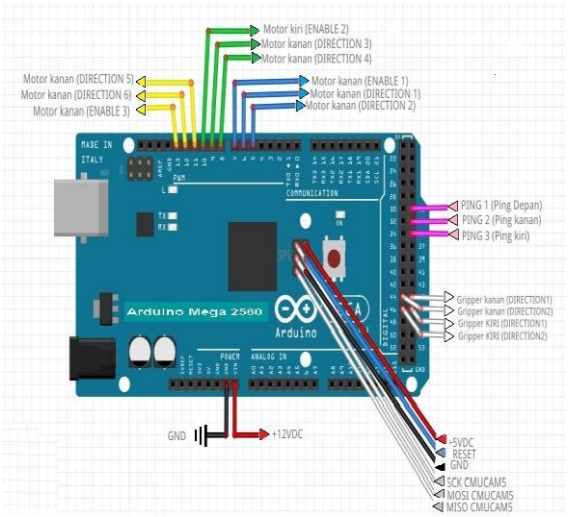


Fig. 7. Design of electronics control system with Arduino Mega2560

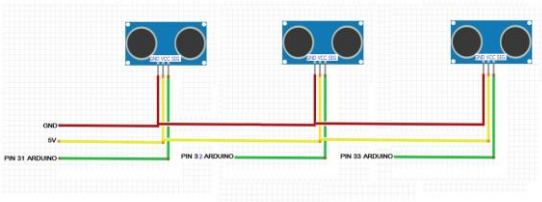


Fig. 8. Wiring diagram of ultrasonic sensor

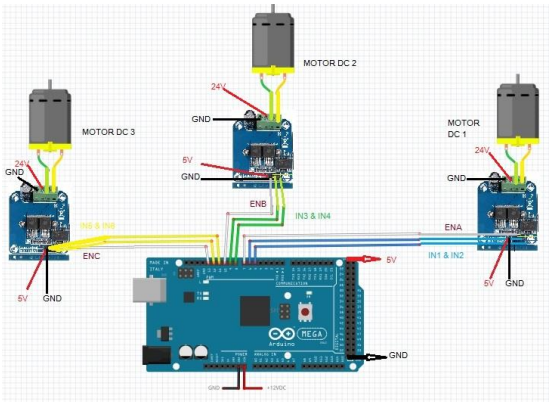


Fig. 9. Wiring diagram of driver motor

III. TESTING SET-UP

Tests conducted to determine the robot's performance in detecting ball objects in both static and moving conditions (dynamic). In testing on static conditions, testing is done to determine whether the ball object successfully detected by the

robot so that the coordinates of the object position is known. The test is done by arranging the object straight in front of the robot camera, the object on the left side of the robot, and the object on the right side of the robot, the detection distance is 40 cm, 80 cm, 120 cm, 160 cm and 200 cm. Data retrieval is done by displaying x and y coordinate data, and height and weight using serial monitoring from Arduino IDE software and PixyMon software. Illustrations of the target marker positioning sensor testing can be seen in Fig. 10.

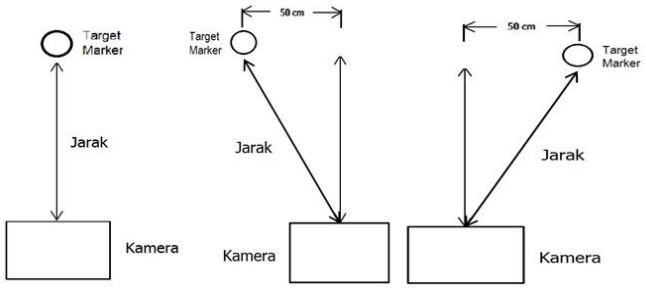


Fig. 10. Sensor testing with target marker: directly in front, offset to the left, and offset to the right

IV. Results and Discussion

A. Testing for the position of the object straight in front of the camera (centered)

Testing is done by placing the object (target marker) at various distances: 40 cm, 80 cm, 120 cm, 160 cm and 200 cm. Target marker is placed at the center position of the camera sensor. The result of the camera test when the target marker is in the center of the sensor is shown in Fig.11 and Table 1.

Table 1 shows the results of CMUcam5 camera sensor testing related to position, width, and target marker height. For the value of x it is seen that the value ranges from 171-178, this is due to non-precise left-right position at the time of target shift. The value of y changes from 162 to 115 because the greater the distance of the sensor with the target marker the target marker's position will be higher, so the y value is smaller. Results of sensor readings with a value of x ranging from 170 ± 10 can be interpreted that the target is in the center.

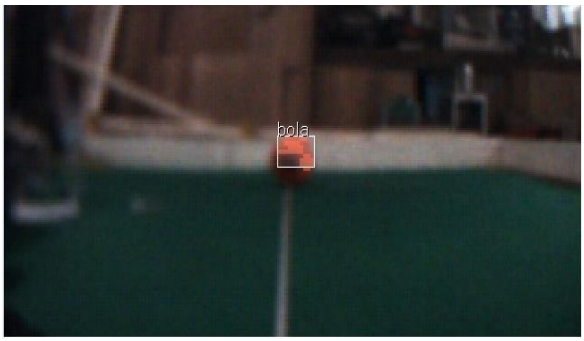


Fig. 11. The result of object detection with the position centered against the robot at a distance of 200 cm

TABLE I. Target Detection Test Result On Center Position

Target marker distance (cm)	**Sensor readings *(integer)***			
	Coordinates		***Width (pixels)***	***Height (pixels)***
	x	***y***		
40	178	162	105	74
80	177	131	17	39
120	175	125	41	23
160	173	117	20	17
200	171	115	22	16

B. Testing with target marker offset to the left

Testing is done by placing the object (target marker) to the left direction of the robot camera at a distance of 40 cm, 80 cm, 120 cm, 160 cm and 200 cm. The test results are shown in Fig.12 and the data in Table 2.



Fig. 12. The results of object position detection test to the left of the robot with a distance of 200 cm

TABLE II. Target Detection Test Result On Left-offset Position

***Target marker* distance (cm)**	**Sensor reading *(integer)***			
	Coordinates		***Width (pixels)***	***Height (pixels)***
	x	y		
40	3	149	6	9
80	22	126	40	35
120	55	120	31	26
160	73	115	22	17
200	98	112	17	12

Based on Table 2 it can be seen for x coordinates that the values are between 3 and 98, if the value of x less than 170 can mean that the target marker is to the left of the center. The value for y coordinates between 112-149 due to the greater the distance of the sensor with the target marker, then the target marker will look higher on the camera so that the y value also increases.

C. Testing with target marker offset to the right

Testing is done by placing the object (target marker) to the right direction of the robot camera at a distance of 40 cm, 80

cm, 120 cm, 160 cm and 200 cm. The test results are shown in Fig.13 and the data in Table 3. Based on the data in Table 3, the value of x is in the range from 209 to 300. For the sensor readout value, if the value of x greater than 180 can be interpreted that the target marker is to the right of the midpoint. The value for y coordinates is in the range of 114 to 151, the greater the distance of the sensor with the target marker, the target marker will appear higher so that the y value also increases.



Fig. 13. The results of object position detection test to the right of the robot with a distance of 200 cm

TABLE III. TARGET DETECTION TEST RESULT ON RIGHT-OFFSET POSITION

***Target marker* distance (cm)**	**Sensor reading *(integer)***			
	Coordinates		***Width (pixels)***	***Height (pixels)***
	x	y		
40	300	151	42	45
80	270	131	34	35
120	254	120	32	30
160	230	118	26	22
200	209	114	21	19

D. Detection Test for Moving Ball (Dynamic) Conditions

The detection test for the ball in moving (dynamic) condition is done by means of a robot directed to the ball, then when the robot manages to detect the ball and moves towards the ball, the ball will be driven by a human (dribbled) with random directions. Test results show the robot can dynamically follow the direction of movement of the ball. Control system testing for robot movement speed is done by varying the distance of the ball to the robot and observing the response speed. Variations of distance tested were 5 cm, 25 cm, 50 cm, 75 cm, 100 cm, 150 cm, and 200 cm. The test results show the speed of the robot can be controlled based on the distance between the robot with the ball. The shorter the robot's distance to the ball, the robot's speed will decrease.

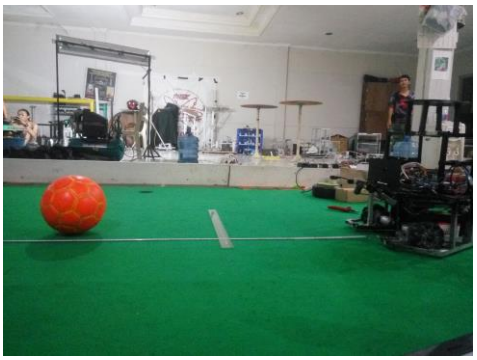


Fig. 14. Robot speed control test for dynamic ball position

V. CONCLUSION

Based on the results from the research that has been done, when the ball is static, the object detection system can work well. For dynamic ball conditions, although the robot is able to follow the movement of the ball position, the response is still slow but the speed of movement of the robot based on the distance conditions between the ball with the robot, speed control system can work as expected.

REFERENCES

[1] D. Tristianto, "Aplikasi Perancangan Kontroler Kamera Personal Komputer untuk Mengikuti Objek Bergerak," Jurnal Monitor, Vol. 2, No. 2, Juli 2013.

[2] A. Yulianto and H.P. Handoyo,"Penerapan Behavior-Based Control dan Fuzzy Logic Controller Pada Sistem Navigasi Robot Soccer." CENTRE, Civil and Electrical Engineering Journal, Vol.7 No.1,Juni 2012.pp.16-23

[3] M. Rahman and H. Aprilianto, "Penerapan Metode Fuzzy Pada Robot Beroda Menggunakan Omni-Directional Wheels," JUTISI, Vol.5, No.2 Agustus 2016, pp.1075-1082

Segmentasi dan Estimasi Jarak Bola dengan Robot Menggunakan Stereo Vision

Irfan Marzuqi*, Gilang Prilian Arinata, Z.M.A Putra, dan Agus Khumaidi
Teknik Kelistrikan Kapal
Politeknik Perkapalan Negeri Surabaya
Surabaya, Indonesia
marzuqiirfan@gmail.com

***Abstrak*—Kemampuan robot pemain sepak bola dalam menemukan letak bola adalah hal utama yang harus dimiliki oleh robot. Salah satu metode yang digunakan untuk melakukan segmentasi adalah dengan menggunakan metode Color Threshold. Segmentasi citra dengan menggunakan ruang warna HSV (Heu, Saturation,Value) di nilai lebih efektif dibandingkan dengan ruang warna RGB dikarenakan intensitas warna dan cahaya dapat dibedakan sehingga dalam keadaan tertentu HSV sangat dibutuhkan. Pada robot ini dipasang dua kamera sejajar dengan sumbu putar 360º. Satu objek hasil segmentasi dikenali dari dua buah citra yang diambil sesuai dengan prinsip stereo vision. Titik tengah dari setiap citra dijadikan referensi dalam perhitungan jarak antara kamera dan ojek dengan menggunakan titik perpotongan dua garis. Dengan terlebih dahulu melalui proses Stereo Calibration dan Stereo Rectification. Proses kalibrasi ini digunakan untuk mendapatkan faktor-faktor *intrinsic* dan *extrinsic*. Dari hasil percobaan robot dapat mengetahui jarak robot (koordinat dunia) dan bola dengan tingkat ketelitian 95%.**

Kata kunci*—RGB; HSV; Stereo vision; intrinsic; extrinsic.

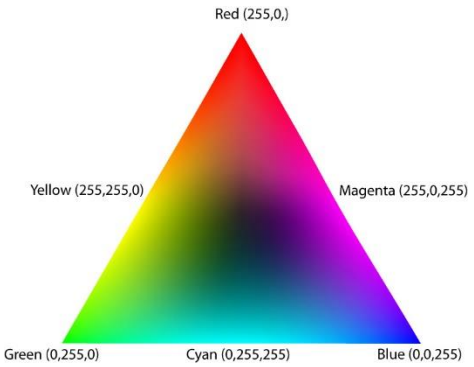
I. PENDAHULUAN

Cara pasif yang sering digunakan dalam menentukan jarak objek adalah dengan dengan menggunakan citra stereo atau stereo vision pada kamera. Hal ini dikarenakan adanya ilusi jarak yang dapat dimunculkan dari citra stereo tersebut. Dua buah kamera dipasang sejajar yang dipisahkan oleh jarak, kedua kamera mengambil gambar pada saat yang bersamaan. Semakin dekat sebuah objek dengan kedua kamera, maka semakin besar perbedaaan posisi bola yang ditangkap oleh kamera. Perbedaan ini dinamakan *disparity*. Peletakan dua kamera yang tidak sejajar akan menyebabkan posisi horizontal dari dua citra yang diambil oleh dua kamera tersebut tidak sama. Hal ini akan menyebabkan proses pengukuran jarak tidak akan sesuai dengan nilai yang sebenarnya. Namun, pada praktiknya, peletakan dua kamera agar benar-benar sejajar memang sangat sulit untuk dilakukan. Untuk mengatasi hal tersebut, dibutuhkan beberapa proses komputasi yang dapat membuat posisi horizontal dari dua citra menjadi sama, yaitu Stereo *Calibration* dan Stereo *Rectification*.

II. DASAR TEORI

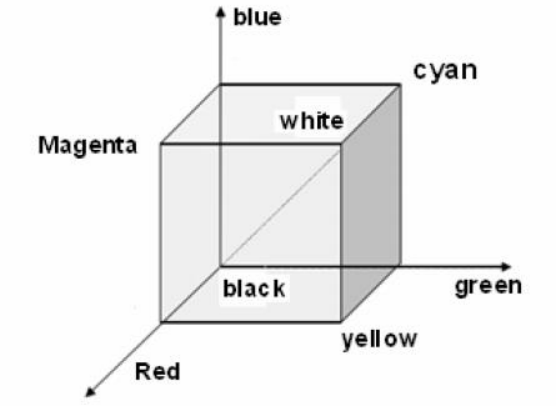
2.1. Ruang Warna RGB

Model ruang warna RGB terdiri atas tiga channel warna yaitu merah, hijau, dan biru, yang disebut dengan tiga warna utama *(the three primary colors)*. Banyaknya variasi warna dapat dilakukan dengan modulasi kombinasi dari ketiga warna tersebut. Setiap *channel* warna dibatasi antara nilai 0 sampai 255. Dengan demikian masing-masing komponen terdapat 256 tingkat. Apabila dikombinasikan maka ada 256 x 256 x 256 atau 16.777.216 kombinasi warna RGB yang dapat dibentuk [1]. Seperti pada Gambar 1.



Gambar 1. Ruang warna RGB

Dari *Gambar 2* dapat diketahui koordinat dari ruang RGB, merah terletak di koordinat (1,0,0), hijau terletak pada koordinat (0,1,0), biru terletak pada koordinat(0,0,1), hitam terletak pada koordinat (0,0,0), dan putih terletak pada koordinat (1,1,1). Sedangkan tingkat keabuan adalah distribusi di daerah diagonal pada kubus yaitu (0.5, 0.5, 0.5).

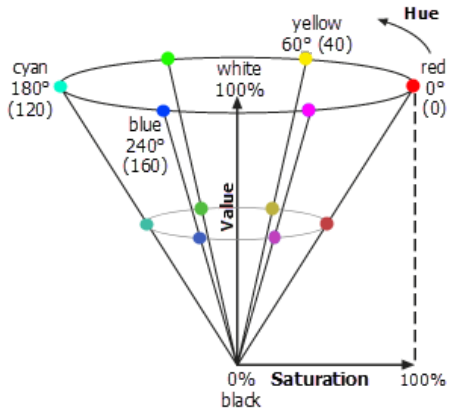


Gambar 2. Kubus warna RGB

Ruang warna RGB adalah representasi warna sederhana. Dan jika ingin mengubah warna dari citra, ketiga komponen harus berubah agar dapat benar-benar dapat dilihat oleh visual mata manuasia [1].

2.2. Ruang Warna HSV

HSV mendefinisikan warna dalam terminologi Hue, *Saturation* dan *Value*. Model warna HSV dapat digambarkan melalui heksagonal piramid terbalik seperti pada Gambar 3. permukaan paling atas terbentuk dari segi enam dengan arah positif terhadap sumbu S, ini digunakan untuk representasi nilai Heu (H) yaitu kemurnian warna [3], perubahan nilai berlawanan dengan arah jam (CCW). Keenam sudut tersebut mewakili posisi warna merah, kuning, hijau, cyan, biru, dan magenta. Sudut interval masing-masing warna adalah 60^0 . Saturasi (S) menunjukkan seberapa putih warna tesebut [3], jarak dari V-axis ke *hexagonal border* dengan rentan 0-1 atau 0-100%. Nilai S pada permukaan paling luar mempunyai nilai maksimum yaitu 1 dan nilai S pada V-axis mempunyai nilai minimum yaitu 0.



Gambar 3. Heksagonal piramid

Sumbu piramid menunjukkan tingkat kecerahan (V) dengan nilai antara 0-1 atau 0-100%. Dengan V=0 menunjukkan hitam, dan 1 menunjukkan warna putih [1].

2.3. Konversi RGB ke HSV

Untuk dapat merubah nilai RGB ke HSV dapat dilakukukan dengan menggunakan persamaan berikut dengan R adalah intensitas warna merah, G intensitas warna hijau, dan B intensitas warna biru dalam satu pixel [1]:

1. Mendapatkan nilai H

$$R' = \frac{R}{255} \quad (1)$$

$$G' = \frac{G}{255} \quad (2)$$

$$B' = \frac{B}{255} \quad (3)$$

$$C_{max} = Max(R', B', C') \quad (4)$$

$$C_{min} = Min(R', B', C') \quad (5)$$

$$\Delta = Cmax - Cmin \quad (6)$$

$$H = \begin{cases} 0 & ,0 \\ 60x\left(\frac{G'-B'}{\Delta} \bmod 6\right) & ,Cmax = R' \\ 60x\left(\frac{B'-R'}{\Delta} + 2\right) & ,Cmax = G' \\ 60x\left(\frac{R'-G'}{\Delta} + 4\right) & ,Cmax = B' \end{cases} \quad (7)$$

2. Mendapatkan nilai S

$$S = \begin{cases} 0 & ,C_max \\ \frac{\Delta}{Cmax} & ,C_{max} \end{cases} \quad (8)$$

3. Mendapatkan nilai V

$$V = C_{max} \quad (9)$$

2.4. Stereo Vision

Stereo vision menggunakan dua citra yang diambil dalam waktu yang hampir bersamaan pada suatu area dan arah yang sama. Dengan menggunakan stereo vision, jarak objek-objek yang ada di kedua citra bisa diukur dengan menggunakan beberapa metode pengukuran jarak berdasarkan stereo vision. Untuk melakukan pengukuran jarak objek menggunakan ada syarat yang harus dipenuhi, yaitu:

1. Kamera harus benar-benar sejajar dan mengarah pada arah yang sama.
2. Pengambilan dua citra harus bersamaan.

Mekanisme penglihatan robot didesain dengan menggunakan dua kamera yang diberi lensa *fish eye* polaroid, sebuah motor

stepper dan mikrokontroller sebagai pengendali gerakan kamera. Bentuk mekanik penggerakan kamera seperti yang ditampilkan di Gambar 4.



Gambar 4. Peletakan kamera stereo

Stereo *Calibration* merupakan proses penghitungan hubungan geometris dari dua kamera di suatu ruang [2]. Sedangkan stereo *rectification* merupakan penghitungan matriks intrisik dan ekstrinsik kamera untuk me-remap citra stereo agar kedua citra sejajar dalam arah horizontal. Hasil dari stereo *calibration* adalah parameter-parameter intrinsik dan ektrinsik dari dua kamera tersebut. Parameter intrinsik merupakan parameter yang mengoreksi kesalahan-kesalahan yang dapat disebabkan oleh kondisi fisik internal dari kamera, seperti distorsi akibat pemasangan lensa di dalam kamera atau distorsi akibat kecacatan lensa kamera itu sendiri. Sedangkan parameter ekstrinsik mengoreksi ketidak sejajaran dari dua kamera yang berupa kesalahan translasi maupun kesalahan rotasi.

2.5. *Metode Pengukuran*

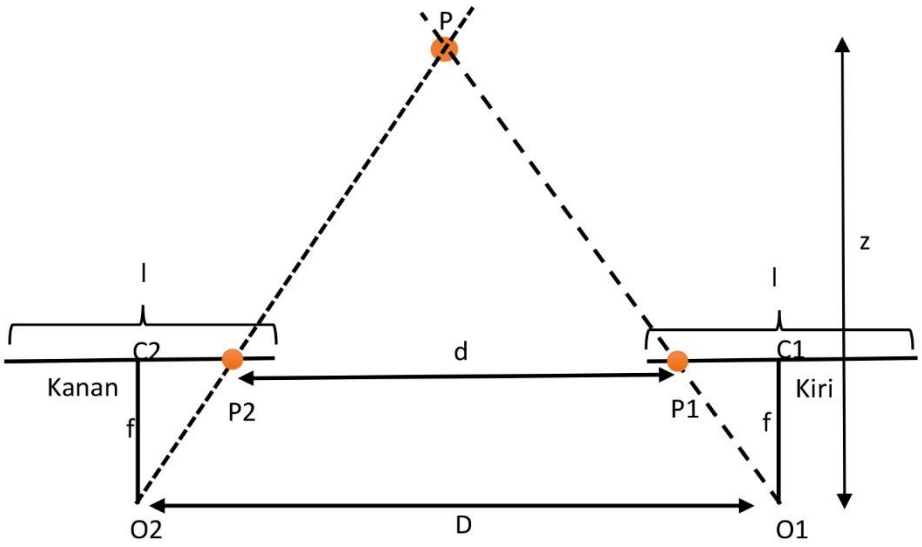
Metode pengukuran jarak yang memanfaatkan perbedaan posisi objek pada dua citra yang diambil pada waktu yang hampir bersamaan dengan stereo *vision* [4]. Dengan mengasumsikan bahwa kedua citra diambil dari dua kamera yang benar-benar sejajar. Dari Gambar 5, O2 dan O1 adalah pusat proyeksi, P1 dan P2 adalah hasil proyeksi objek pada bidang proyeksi, C1 dan C2 adalah *principal point* dari bidang proyeksi dengan nilai didapatkan melalui kalibrasi yang dijelaskan pada bab selanjutnya, d *(disparity)* adalah selisih nilai dari koordinat benda pada titik x,y pada bidang satu dengan bidang proyeksi yang lainnya d=P1(x) - P2(x), sedangkan D adalah jarak antar pusat proyeksi. Dengan menggunakan persamaan dua garis maka didapatkan persamaan untuk mencari nilai z:

$$\frac{x - x1}{x2 - x1} = \frac{z - z1}{z2 - z1} \tag{10}$$

$$(x - x1)(z2 - z1) = (z - z1)(x2 - x1) \tag{11}$$

$$\underbrace{(x - x1)(z2 - z1)}_{a} - \underbrace{(x2 - x1)(z - z1)}_{b} = 0 \tag{12}$$

$$a(x - x1) + b(z - z1) = 0 \tag{13}$$



Gambar 5. Konsep pengukuran jarak

$$ax + bz = \underbrace{ax1 + bz1}_{c}$$

Sehingga didapatkan persamaan garis seperti pada persamaan 14:

$$ax + bz = c \quad (14)$$

Kemudian untuk mendapatkan perpotongan kedua garis (P) dengan menggunakan eliminasi didapatkan nilai seperti pada persamaan 15 untuk nilai z dan persamaan 16 untuk mendapatkan nilai x.

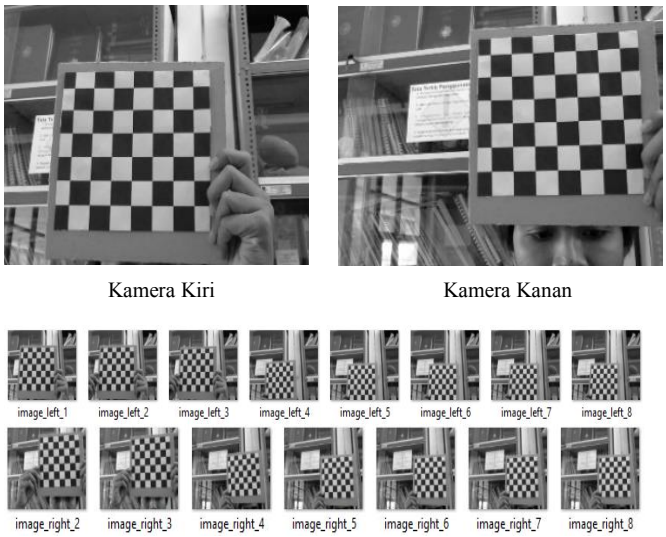
$$z = \frac{a_2c_1 - a_1c_2}{a_2b_1 - a_1b_2} \quad (15)$$

$$x = \frac{c_2b_1 - c_2b_1}{a_1b_2 - a_2b_1} \quad (16)$$

III. HASIL PENGUJIAN

3.1. *Hasil Kalibrasi*

Pada percobaan kali ini kalibrasi menggunakan kamera bertipe sama dengan objek berupa pola papan catur *(chessboard)* yang memiliki grid 7 x 6 dengan lebar 2.6 Cm, objek diletakkan pada area yang dapat dilihat oleh kedua kamera dengan pengambilan pola sebanyak 40 kali, dengan pengambilan gambar secara bersamaan. Hasil pengambilan gambar seperti pada Gambar 6.



Gambar 6. Hasil Pengambilan gambar stereo kamera saat kalibrasi

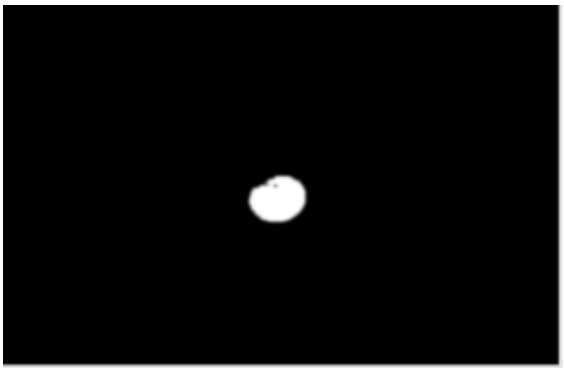
Dari hasil kalibrasi didapatkan data seperti pada Tabel 1 untuk menentukan *focal length* x (fx) dan y (fy), *principal point* x (cx) dan y (cy) pada titik tengah citra, serta nilai proyeksi pada sumbu x, y, dan z.

TABLE I HASIL KALIBRASI FAKTOR INTRINSIK DAN EKSTRINSIK.

Kamera	Parameter	Nilai
Kiri	fx	1039
	fy	1039
	cx	310
	cy	44
Kanan	fx	1039
	fy	1039
	cx	308
	cy	51
Translasi	X	8,2
	Y	-3,4
	Z	-5,02

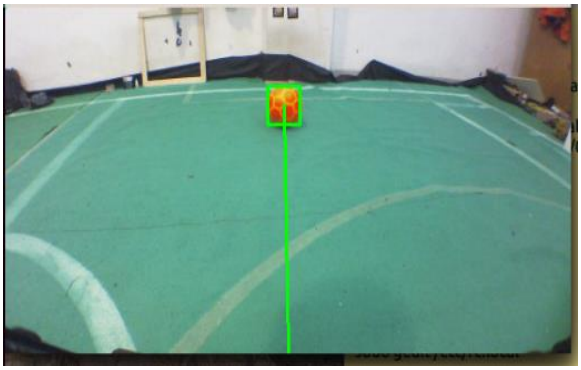
3.2. *Filter Warna HSV*

Pada percobaan kali ini objek yang dideteksi adalah sebuah bola berwarna orange dengan kontur bulat, adapun proses segmentasi menggunakan *threshold* HSV jika dari penjelaskan subbab 2.2 diketahui bahwa warna *orange* terletak pada sudut Heu antara 0º sampai 60º pada skala 360º dan karena OpenCV nilai Heu mempunyai range 0 – 179º. Nilai Saturasi dan *Value* pada umumnya seperti yang dijelaskan subbab 2.2 bernilai dari rentang nilai 0-1 tetapi pada OpenCV bernilai 0-255. Gambar 7 menunjukkan filter untuk nilai H dengan rentang nilai antara 0 sampai 21, menunjukkan filter S dengan nilai antara 227 sampai 255, menunjukkan filter V dengan nilai antara 125 sampai 217. Dan hasil dari ketiga filter tersebut dapat dilihat bahwa objek yang dideteksi berwarna putih dan yang lainnya berwarna hitam.



Gambar 7. Color threshold untuk deteksi bola.

Tahap selanjutnya setelah filter warna selesai dilanjutkan dengan deteksi kontur untuk mendapatkan kontur lingkaran dan mencari titik berat dari lingkaran *(centroid)* [5]. *Centroid* ditunjukkan seperta pada Gambar 8.



Gambar 8. Hasil penggabungan ketiga filter.

3.3. *Hasil Pengukuran Jarak*

Dengan menggunakan persamaan garis yang telah dijelaskan pada subbab 2.5 maka didapatkan hasil jarak antara objek dan kamera. Pengujian bertujuan untuk mendapatkan selisih nilai output pada *image processing* dengan jarak yang sebenarnya, sehingga didapatkan error seperti pada Tabel II.

TABLE II HASIL PENGUKURAN JARAK

No	Jarak Sebenarnya (Cm)	Image Processing (Cm)	Error (%)
1	30	30	0
2	25	24	4
3	23	23	0
4	18	17	5
5	16	16	0
6	14	14	0

IV. KESIMPULAN

Dari percobaan yang telah dilakukan dapat disimpulkan bahwa:

1. Ruang warna HSV mempunyai komposisi nilai sebagai berikut Heu mempunyai enam sudut dengan masing-masing sudut merefleksikan warna-warna tertentu dengan perbedaan sudut masing-masing 60º. Saturasi dan *Value* mempunyai nilai antara 0 sampai 1, semakin mendekati nilai 0 maka warna akan semakin berwarna putih untuk saturassi dan semakin gelap untuk nilai Value.
2. Dari Tabel 2 tingkat akurasi data error antara jarak yang ditunjukkan oleh hasil pengolahan citra dan pengukuran maksimal 5%.

REFERENSI

[1] Y.-P. Z. Fu-Qiang, "Color Feature Extraction Of Hainan Li Brocade Image Based On RGB And HSV," Haikou, 2015.

[2] L. d. L. Y. Zou, " Method of Stereo Vision Matching Based on OpenCV," dalam *nternational Conference on Audio, Language, and Image Processing (ICALIP)*, Shanghai, 2010.

[3] J. W. Shipman, "Introduction to color theory," 16 10 2012. [Online] Available: http://www.nmt.edu/ tcc/help/pubs/colortheory/web/hsv.html [Diakses 2 1 2017].

[4] A. Maulidi, "Rancang Bangun Sistem Pengukuran Posisi Target dengan Kamera Stereo untuk Pengarah Senjata Otomatis," *Jurnal Teknik ITS,* vol 5, no. 2, 2016.

[5] Wikipedia, "Image Moment," wikipedia, 27 11 2016. [Online]. Available https://en.wikipedia.org/ wiki/Image_moment. [Diakses 16 1 2017].

[6] Y. T. Akira Ohashi, "Fisheye stereo camera using equirectangular images," dalam *17th International Conference on Research and Education in Mechatronics (REM)*, Osaka, Japan, 2016.

Desain Robot Sepak Bola Beroda Tim Abimanyu 2017

Julian Sahertian*, Ardi Sanjaya, M. Denny Yusuf H, Agus Sutrisno, Maulana Malik V, dan Ahmad Makinun A.
Prodi Teknik Informatika, Fakultas Teknik
Universitas Nusantara PGRI Kediri
Kediri, Indonesia
juliansahertian@unpkediri.ac.id

***Abstrak*—Kontes Robot Indonesia (KRI) khususnya pada divisi beroda setiap tahun terus berubah mengikuti perkembangan lomba robot internasional. Tahun 2017 divisi beroda menjadi Kontes Robot Sepak Bola Indonesia (KRSBI) beroda dimana dalam perlombaan robot harus mampu melakukan berbagai perilaku pemain sepak bola seperti mengejar bola, menggiring, menendang, dan lain sebagainya. Desain robot yang mampu melakukan perilaku pemain sepak bola tersebut sampai sekarang masih menjadi riset dari berbagai insitusi yang mengikuti KRSBI beroda. Makalah ini membahas tentang perkembangan riset desain robot sepak bola beroda tim Abimanyu dari Universitas Nusantara PGRI Kediri. Pembahasan mencakup mekanik, *hardware* elektronik dan *software*. Perkembangan yang paling signifikan adalah desain mekanik drible, mekanik penendang, dan *vision* robot untuk mendeteksi bola maupun gawang.**

Kata kunci—robot sepak bola beroda; abimanyu; mekanik; computer vision

I. PENDAHULUAN

Abimanyu adalah tim robot *soccer* beroda dalam Kontes Robot Sepak Bola Indonesia (KRSBI) beroda dari Universitas Nusantara PGRI Kediri. Terbentuk pada tahun 2014 tim ini terdiri dari dosen, mahasiswa dan staff laboratorium. Pada awalnya tim ini dikhususkan pada pengembangan robot beroda yang mengikuti Kontes Robot Indonesia (KRI) pada divisi Kontes Robot Pemadam Api Indonesia (KRPAI) beroda pada tahun 2015. Setiap tahun aturan pertandingan pada divisi robot beroda semakin berkembang, pada tahun 2017 divisi beroda pemadam api diganti menjadi divisi sepak bola beroda sehingga perlu adanya riset yang lebih mendalam agar robot yang dikembangkan mampu mengikuti perkembangan pertandingan tersebut.

Perkembangan terbaru dari robot *soccer* atau robot sepak bola khususnya divisi beroda dibandingkan dengan divisi beroda sebelumnya adalah robot sepak bola harus memanfaatkan teknologi *Computer Vision* dalam mendeteksi objek-objek dalam pertandingan seperti bola, gawang, dan robot kawan maupun lawan. Selain itu mekanik robot juga harus menyesuaikan aturan sepak bola seperti yang dilakukan oleh manusia yaitu dapat men*drible*, menendang bola, mengumpan, dan lain sebagainya.

Terdapat beberapa penelitian yang mengkaji tentang pengembangan robot sepak bola beroda seperti penelitian [1] dan [2] yang membahas tentang sistem *vision* dari robot, penelitian [3] yang membahas tentang desain mekanik dribel dan penelitian [4] membahas tentang desain mekanik penendang, dan terdapat pula penelitian yang membahas algoritma *high-level* seperti koordinasi antar robot dan strategi permainan [5].

Berdasarkan studi literatur dari penelitian-penelitian terdahulu pada makalah ini akan dibahas hasil riset robot sepak bola yang telah dilakukan oleh tim Abimanyu dalam mengikuti KRSBI beroda 2017. Pembahasan dimulai dari arsitektur robot yang dikembangkan, desain mekanik, hardware elektronik, dan *software* termasuk *vision* dari robot.

II. DESKRIPSI ROBOT

Pada bagian ini membahas tentang bagian-bagian dari robot sepak bola yang dikembangkan. Gambar 1. menunjukkan bentuk fisik dari robot *soccer* tim Abimanyu.



Gambar. 1.Robot sepak bola tim Abimanyu

A. Mekanik

Pembahasan mekanik dibagi menjadi tiga bagian yaitu mekanik roda, mekanik *drible*, dan mekanik penendang.

1) Base Mekanik

Base mekanik dari robot sepak bola tim Abimanyu dibagi menjadi dua yaitu *base* mekanik penyerang dan *base* mekanik penjaga gawang. *Base* mekanik penyerang menggunakan desain roda omni 3wd yang mengacu pada [6] sedangkan penjaga gawang menggunakan desain roda omni 4wd [4].



(a) (b)

Gambar. 2. Base mekanik robot sepak bola tim Abimanyu (a) Penyerang (b) Penjaga gawang

2) Mekanik Drible

Terdapat dua pendekatan yang telah dilakukan peneliti terdahulu dalam mendesain mekanik *drible*, yaitu pendekatan *drible* pasif [7] dan *drible* aktif [3].

Berdasarkan dari riset yang dilakukan Robot tim Abimanyu menggunakan pendekatan *drible* aktif karena setelah diimplementasikan pendekatan *drible* aktif lebih efektif dalam menahan dan menggiring bola daripada pendekatan *drible* pasif. Mekanik dribel robot tim Abimanyu dapat dilihat pada Gambar 3.



Gambar. 3.Mekanik dribel aktif

3) Mekanik Penendang

Mekanik penendang robot tim Abimanyu menggunakan *pneumatic* seperti yang pernah diimplementasikan oleh [8]. Akan tetapi disini kita melakukan pengembangan pada bagian kaki penendang yaitu menggunakan desain jungkat jungkit dimana silinder *pneumatic* tidak dipasang paten akan tetapi dapat bergerak mengikuti posisi tendangan. Lebih jelasnya dapat dilihat pada Gambar 4.



(a) (b)

Gambar. 4. Mekanik penendang (a) Posisi awal (b) Posisi menendang

Berdasarkan riset yang telah dilakukan desain mekanik seperti ini efektif digunakan selain hanya membutuhkan satu silinder *pneumatic* berbeda dengan [8] yang menggunakan dua silinder *pneumatic*. Selain itu kekuatan menendang dari desain ini lebih baik yaitu mampu melakukan tendangan lob daripada desain silinder *pneumatic* yang dipasang paten.

B. Hardware

Robot sepak bola tim Abimanyu menggunakan mini-PC *Raspberry pi 3* dan *Arduino Mega* sebagai perangkat kerasnya. *Raspberry pi 3* digunakan untuk melakukan pemrosesan *vision* yang didapat dari gambar video *Webcam*. Sedangkan Arduino mega digunakan untuk mengontrol aktuator robot berupa motor dc *gearbox base* dan *drible* serta *pneumatic valve* penendang. Selain itu *Arduino Mega* juga digunakan untuk mengontrol masukan dari sensor seperti orientasi, dan ultrasonik. Mini-PC *Rasberry pi 3* dan *Arduino Mega* saling melakukan komunikasi melalui jalur serial USB. *Vision* dari robot menggunakan perangkat kamera depan *Logitech c270* dengan pendekatan kamera *pan-tilt*.

C. Software

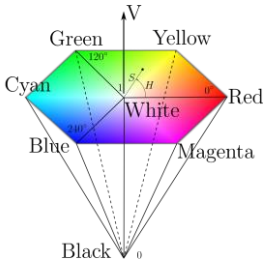
Komputasi yang dilakukan oleh robot dibagi menjadi dua bagian yaitu *low-level* dan *high-level*. Pada layer *low-level* dilakukan pengontrolan terhadap aktuator dari robot yaitu motor pada *base* mekanik yang digunakan untuk pergerakan robot, motor pada *drible*, dan *pneumatic valve* untuk penendang. Sedangkan pada layer *high-level* dilakukan komputasi untuk *vision* dari robot dan kontrol *behaviour* robot seperti mengikuti bola, mendeteksi gawang, dan pengambilan keputusan menendang bola.

III. VISION ROBOT

Vision dari robot tim Abimanyu berfungsi sebagai deteksi objek-objek dilapangan. Saat ini robot sudah dapat mengenali objek bola dan objek gawang.

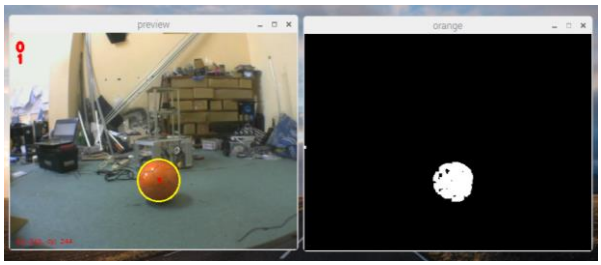
A. Deteksi Bola

Bola yang digunakan dalam aturan KRSBI beroda pada tahun 2017 berwarna jinggga (*orange*) dengan ukuran standar FIFA. Pendekatan yang dilakukan untuk mendeteksi bola menggunakan metode yang sederhana yaitu dengan melakukan filter warna. Warna jingga diambil dari ruang warna HSV. Model dari HSV berbentuk kerucut seperti pada Gambar 5. Warna dasar didapat dari perubahan *hue* (sudut), tingkat pewarnaan diatur oleh *saturation* (radius kerucut, 0, 1), sedangkan tingkat terangnya warna diatur oleh *intensity* atau *value* (bernilai antara 1, putih, dan 0, hitam) [9].



Gambar. 5.Representasi ruang warna HSV

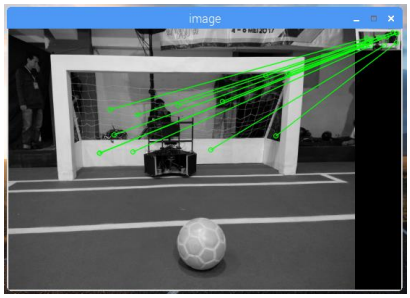
Berdasarkan dari persepsi dari ruang warna HSV tersebut apabila kita melakukan filtering warna kita cukup menggunakan channel warna *hue* dalam menentukan warna yang ingin difilter berbeda dengan ruang warna RGB yang ketiga channelnya berkontribusi pada warna. Gambar 6. menunjukkan hasil deteksi bola dari robot tim Abimanyu.



Gambar. 6.Hasil deteksi bola

B. *Deteksi Gawang*

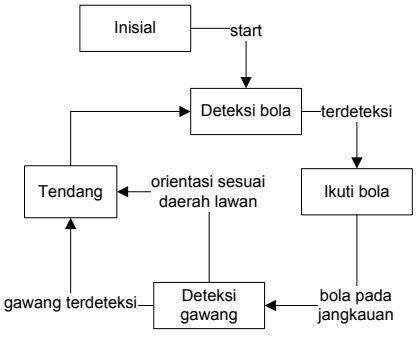
Gawang yang digunakan pada KRSBI beroda tahun 2017 menggunakan warna putih. Apabila menggunakan filter warna, putih akan sulit dideteksi karena akan sangat dekat dengan *noise* yang ada dilapangan seperti iluminasi dari cahaya lampu. Sehingga untuk mendeteksi gawang digunakan pendekatan deteksi objek menggunakan fitur SIFT [10]. Sebenarnya metode ini tidak secepat apabila menggunakan filter warna akan tetapi metode ini tetap digunakan dalam mendeteksi gawang karena robot tidak selalu melakukan deteksi gawang tetapi menunggu bola ditangkap dulu baru melakukan deteksi gawang. Gambar 7. Menunjukkan hasil deteksi gawang robot tim Abimanyu.



Gambar. 7.Hasil deteksi gawang

IV. Behaviour Robot

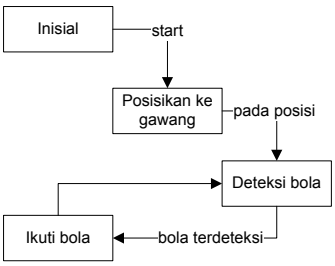
Perilaku (*behaviour*) robot penyerang dijelaskan oleh diagram pada Gambar 8.



Gambar. 8.Diagram perilaku robot penyerang

Pertama robot dalam keadaan *standby* atau inisial. Ketika permainan dimulai secara manual robot diberi sinyal *start*. Setelah itu robot mulai masuk ke lapangan dan mulai melakukan deteksi bola. Ketika bola sudah dideteksi robot akan bergerak menuju bola sampai bola berada pada jangkauan dan ditahan oleh mekanik *drible*. Setelah bola berhasil ditangkap, robot kemudian melakukan *scanning* gawang lawan proses ini selain menggunakan *vision* dari robot juga menggunakan informasi dari sensor orientasi. Ketika gawang lawan sudah ditemukan maka robot akan meluruskan posisi dengan gawang dan bola ditendang, akan tetapi apabila gawang tidak ditemukan bola juga akan tetap ditendang sesuai dengan orientasi gawang lawan. Proses akan kembali pada deteksi bola setiap robot kehilangan bola.

Sedangkan untuk perilaku dari robot penjaga gawang sedikit berbeda dengan robot penyerang ditunjukkan oleh Gambar 9.



Gambar. 9.Diagram perilaku robot penjaga gawang

Untuk robot penjaga gawang pertama pada kondisi inisial. Pada saat permainan dimulai secara manual diberi sinyal *start*. Robot akan memposisikan diri kegawang dengan cara masuk ke lapangan terlebih dahulu setelah itu berjalan ke gawang sendiri sampai jarak gawang terdeteksi oleh sensor jarak. Setelah berhasil memposisikan diri di gawang robot mulai melakukan deteksi bola. Pada bola berada di kanan atau kiri robot, robot akan melakukan pergerakan ke kanan maupun ke kiri yang pergerakannya dibatasi oleh sisi tiang gawang. Akan tetapi apabila bola berada lurus dengan robot akan berada pada kondisi *standby*.

V. Kesimpulan dan Penelitian Lebih Lanjut

Pada makalah ini dibahas tentang deskripsi robot sepak bola tim Abimanyu dari Universitas Nusantara PGRI kediri dalam mengikuti Kontes Robot Sepak Bola Beroda (KRSBI) 2017. Robot yang dikembangkan sudah mampu melakukan beberapa perilaku pemain sepak bola yang sebenarnya seperti mengetahui posisi bola, menggiring bola dan menendang bola ke gawang.

Dalam melakukan deteksi bola robot menggunakan metode filter warna jingga pada ruang warna HSV sedangkan untuk melakukan deteksi gawang robot menggunakan metode SIFT dibantu dengan sensor orientasi. Pergerakan robot menggunakan desain *omnidirectional* sehingga pergerakan dapat secara *holonomic*. Mekanik *drible* yang menggunakan desain *drible* aktif mampu dengan baik menahan bola untuk digiring. Mekanik penendang yang menggunakan *pneumatic* sudah mampu melakukan tendangan lob.

Penelitian lebih lanjut untuk robot sepak bola tim Abimanyu berfokus pada pengembangan fitur-fitur yang

belum dapat diimplementasikan seperti komunikasi dengan *reffbox*, *base station*, maupun antar robot. Pada bagian *vision* robot perlu dikembangkan, secara *hardware* yaitu menggunakan *omnivision* agar robot tidak perlu berputar terlebih dahulu untuk melakukan *scanning* objek-objek di lapangan. Penentuan lokasi robot juga perlu dikembangkan agar nantinya dapat ditentukan strategi permainan menggunakan *path planning*. Hal ini dijadikan pertimbangan berdasarkan dari studi yang telah dilakukan, kontes robot sepak bola yang diadakan ini mengacu pada aturan RoboCup dimana robot yang bertanding sebanyak 5 robot sehingga koordinasi antar robot yang didukung oleh penentuan posisi antar robot sangat dibutuhkan.

REFERENSI

[1] Joao Silva, Antonio J. R. Neves, Nuno Lau, "Ball identification in the RoboCup Middle Size League", Proc. of the 16th Portuguese Conf. on Pattern Recognition, RECPAD 2010, Vila Real, Portugal, October 2010.

[2] Trifan, Alina, António JR Neves, and Bernardo Cunha. "An overview on the application of machine vision in soccer robots.", Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2013.

[3] De Best, Jeroen, Rene Van de Molengraft, and Maarten Steinbuch. "A novel ball handling mechanism for the RoboCup middle size league." Mechatronics 21.2 (2011): 469-478.

[4] Schoenmakers, Ferry, et al. "Tech united eindhoven team description 2013." Proceedings of the 17th RoboCup International Symposium (May 2013). 2013.

[5] Neves, António JR, et al. "CAMBADA soccer team: from robot architecture to multiagent coordination.", Robot Soccer. InTech, 2010.

[6] Aangenent, W. H. T. M., et al. "Tech united eindhoven team description 2009.", CD proceedings of RoboCup 2009 symposium. 2009.

[7] Ichinose, Takaaki, et al. "HIBIKINO-MUSA SHI." RoboCup 2006 Bremen, CD-ROM Proc. of RoboCup 2006 (2006).

[8] Liu, Yanfei, Fort Wayne, and Jiaxin Zhao. "A Kicking Mechanism for an Autonomous Mobile Robot."

[9] Acharya, Tinku, and Ajoy K. Ray. Image processing: principles and applications. John Wiley & Sons, 2005.

[10] Se, Stephen, David Lowe, and Jim Little. "Vision-based mobile robot localization and mapping using scale-invariant features." Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on. Vol. 2. IEEE, 2001.

Rancang Bangun Robot Sepak Bola Beroda

Rizki Wira Pratama*, Gemma Retal Ananda, Yudha Dwi Putra, Pratama Mahadika, dan Yunus Maulana
Program Studi Teknik Elektro, Program Studi Teknik Komputer, Program Studi Teknik Komputer, Program Studi Teknik Elektro, Program Studi Teknik Komputer
Universitas Indonesia
Depok, Indonesia
rizkiwirapratama@gmail.com

***Abstrak*—Permainan sepak bola robot biasanya diimplementasikan dengan menggunakan robot humanoid yang memiliki rangka gerak menyerupai manusia. Kekurangan dari robot humanoid adalah belum tercapainya pergerakan yang gesit dari robot tersebut. Oleh karena itu dibuatlah suatu alternatif untuk mempermudah dalam pengendalian robot yaitu robot sepak bola beroda. Harapan Robot sepak bola beroda ini diikutsertakan dalam perlombaan Kontes Robot Sepakbola Indonesia (KRBSI) Beroda.**

Kata kunci—Robot Beroda; Sepak Bola; KRSBI.

I. PENDAHULUAN

Telah dilakukan banyak penelitian tentang robot sepak bola terutama pada robot sepak bola humanoid. Robot humanoid adalah robot sepak bola yang menyerupai manusia dalam segala hal permainan bola. Robot humanoid dibuat sedemikian sehingga berjalan, menendang dan menggiring bola menyerupai manusia. Karena tingkat kesulitan dalam pembuatannya robot humanoid masih memiliki beberapa kekurangan antara lain kebebasaan gerak, keseimbangan, dan biaya pembuatan. Kebebasan gerak robot humanoid sangat terbatas karena menggunakan dua buah kaki. Robot juga harus memiliki keseimbangan yang baik untuk tetap tegak saat berjalan dan melakukan perannya dalam bermain bola. Selain itu, biaya yang digunakan untuk membuat robot dengan segala perangkatnya membutuhh biaya yang cuku besar. Dengan biaya yang terbilang banyak mengakibatkan tidak semua kalangan bisa membuat robot humanoid. Oleh karena itu, dibuatkan robot yang lebih sederhana yaitu robot sepak bola beroda.

Robot sepak bola beroda memiliki tingkat kesukaran yang lebih rendah karena jumlah penggerak robot jauh lebih sedikit dan memiliki biaya yang lebih terjangkau dibandingkan robot humanoid. Meskipun begitu robot beroda belum banyak dibuat di Indonesia padahal sudah sejak lama kompetisi robot sepak bola beroda diselenggarakan di kancah internasional. Oleh karena itu mulai tahun 2017 telah diadakan cabang lomba KRSBI beroda pada perlombaan KRI.

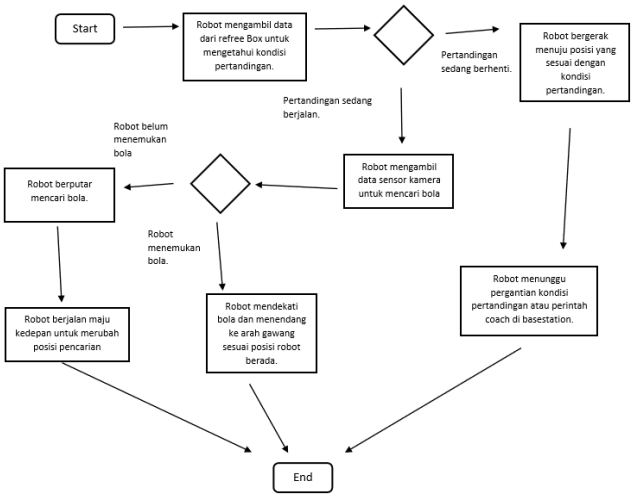
Kontes Robot Indonesia (KRI) adalah perlombaan robot bagi mahasiswa yang diadakan setiap tahunnya dan memiliki beberapa cabang perlombaan yang diadakan oleh Direktorat Jenderal Pendidikan Tinggi (Dikti). Pada tahun 2017 Kontes Robot Indonesia (KRI) mengubah cabang lomba Kontes Robot Pemadam Api (KRPAI) Beroda menjadi Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda.

II. SPESIFIKASI PIRANTI KERAS

A. Desain Sistem

Robot yang dirancang secara umum memiliki sistem masukan (*input*) dan keluaran (*output*). Masukan (*input*) robot berasal dari kamera dan perangkat sensor lainnya. Sementara keluaran (*output*) robot berupa gerakan robot serta komunikasi robot.

Sistem robot secara lebih detail memiliki tiga bagian yaitu masukan (*input*), proses, dan keluaran (*output*). Pada proses terdapat tiga bagian utama yaitu *planning* unit, *executing* unit, dan *knowledge* unit. *Knowledge* unit merupakan bagian tempat pengumpulan data-data masukan sari sensor dan sistem penglihatan robot. Data-data tersebut dianalisis dan dikonversi menjadi data-data sebagai masukan dari pengambilan keputusan. Dari *knowledge* unit data-data tersebut diteruskan ke *planning* unit. *Planning* unit berfungsi sebagai pengambilan keputusan cerdas (*high level intelligant decisions*) untuk menentukan apakah robot akan menendang bola, menggiring bola, dan lain sebagainya. Unit ini juga berfungsi sebagai penentu strategi permainan, seperti apakah robot akan berperan sebagai pemain bertahan atau menyerang. Setelah pengambilan keputusan, perintah untuk melakukan keputusan tersebut dilakukan oleh *executing* unit. Pada *executing* unit, perintah dari *planning* unit diproses dan dianalisis berdasarkan pergerakan dasar robot seperti maju, berhenti, berputar, atau menendang. Kemudian, perintah tersebut diteruskan ke bagian *navigation controller* untuk dieksekusi ke perangkat keras pada robot.

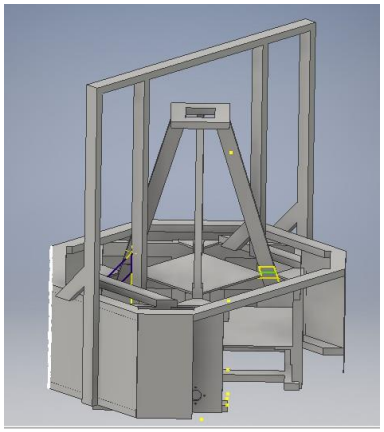


Gambar 1. Alur Diagram Robot.

B. *Desain Platform Robot*

Desain robot berdasarkan peraturan perlombaan KRSBI Beroda yaitu tinggi maksimal robot 60 cm dan memiliki luas 52 cm x 52 cm. Robot yang dibuat memiliki tinggi maksimal dan luas 50 cm x 51 cm. Berat maksimal robot yang diperbolehkan adalah 45 kg tetapi robot yang dirancang hanya memiliki berat 10 kg, sehingga material penyusun rangka yang digunakan adalah alumunium karena memiliki berat yang ringan dan kuat. Selain itu robot didesain agar memiliki pergerakan yang efisien dan mampu mengendalikan bola dengan baik. Bola yang digunakan dalam pertandingan adalah bola futsal berukuran standar yaitu memiliki diameter 68 – 70 cm dan berat 0,4 – 0,44 kg. Ukuran bola juga termasuk dalam pertimbangan bentuk depan robot agar dapat melakukan pengendalikan bola.

Desain mekanika robot dibuat sedemikian rupa agar robot tahan terhadap benturan, ringan dan tidak mudah terbalik ketika tertabrak tetapi robot tetap memiliki gerakan yang lincah dan efisien seperti yang diinginkan karena dalam permainan sepak bola agar terjadi benturan antar robot saat memperebutkan bola di lapangan. Mekanika robot didesain seperti pada Gambar 2.



Gambar 2. Desain Mekanika Robot

Penggerak robot yang diinginkan adalah penggerak yang mampu membuat robot bergerak ke segala arah dengan kecepatan yang konstan untuk setiap arahnya. Hal ini dilaukan dengan sistem penggerak *omni-directional*. Setelah itu dibutuhkan suatu modul untuk memanipulasi gerak bola agar dapat menggiring, menerima, mengendalikan bola saat robot bergerak dan menendang bola. Karena dirancang robot sepak bola sederhana sehingga dibuatlah penggiring bola pasif.

Sistem penggiring bola pasif dibuat untuk mengendalikan bola agara dapat bergerak secara selaras dengan gerakan robot namun tidak dapat mengubah arah rotasi bola. Hal ini membatasi gerakan menggiring bola pada robot misalnya robot tidak dapat menggiring bola mundur atau bergerak terlalu agresif karena dapat membuat bola terlepas dari pengendalian robot.

Penggerak robot terdiri dari motor driver *h-bridge*, mikrokontroler, dan *Direct Current* (DC) 12 V. Keluaran (*output*) dari modul ini dikendalikan oleh pengendali utama yaitu Stm32f407vg dengan memperhitungkan *feedback* dari motor. *Feedback* dari motor tersebut diterima oleh mikrokontroler dengan komunikasi serial. Informasi yang diterima diolah dengan algoritma kontrol Proporsional, Integratif, dan Derivatif (PID) sehingga perbedaan kecepatan *real* dan kecepatan yang diinginkan dapat di kompensasi.

III. SPESIFIKASI PIRANTI LUNAK

Perangkat lunak yang digunakan pada robot terdiri dari masukan (*input*) dan keluaran (*output*). Pada sistem masukan dilakukan pembacaan pada modul penglihatan robot, kompas dan *inertial measurment* unit (IMU). Dan pada sistem keluaran (*output*) robot merupakan komunikasi dan pergerakan robot. Dari sistem masukan robot dilakukan integrasi sehingga robot dapat mendeteksi posisi dirinya di lapangan, menentukan posisi bola, dan menentukan posisi gawang lawan. Dari pembacaan tersebut robot dapat melakukan tindakan sesuai dengan kondisinya.

A. *Sistem Citra Robot*

Modul pengolahan citra berfungsi sebagai pengambil data *visual* dari lingkungan untuk diproses menjadi masukan pada algoritma pemecahan masalah pada robot. Modul ini terdiri dari kamera CMUcam 5 Pixy. Dilakukan proses penyaringan gambar di lingkungan sehingga robot dapat membedakan bola dari lingkungan disekitarnya. Sensor kamera ini dihubungkan ke *main controller*. Kamera yang digunakan memiliki resolusi maksimal sebesar 2 megapixel. Namun, pada saat pemprosesan citra, hasil dari sensor kamera akan dikompres sehingga menjadi lebih kecil agar pemprosesan data yang dilakukan lebih cepat. Jenis kamera yang digunakan adalah ...

Pada robot sepak bola dilakukan pengolahan citra untuk membedakan komponen komponen di lingkungan robot seperti bola, gawang dan garis lapangan. Pertama, digunakan metode segmentasi warna pada citra objek dan kordinat objek untuk menentukan jenis apa objek tersebut. Namun proses segmentasi warna ini sangat bergantung pada pencahayaan lingkungan sehingga memiliki realibilitas yang kurang baik.

B. *Navigasi Robot*

Navigasi robot berfungsi sebagai petunjuk arah untuk pergerkan robot. Navigasi yang digunakan pada robot sepak bola beroda adalah kompas yang dapat mengetahui posisi sudut.. sensor yang digunakan adalah sensor magnetik kompas *HMC5833L.* Sensor kompas ini memberikan petunjuk bagi robot dimana posisi gawang lawan dan gawang tim.

IV. HASIL PENGUJIAN

A. *Platform Robot*

Berdasarkan hasil pengurukuran, robot memili luas 50 x 51 cm dan berat 10 kg sehingga dimensi yang dimiliki robot telah sesuai dengan peraturan perlombaan. Pengendali gerakan robot telah diintegrasikan dengan penggerak robot secara keseluruhan untuk melihat kemampuan dasar robot. Hasilnya robot dapat bergerak sesuai dengan harapan yaitu bergerak maju, berputar dan menggiring bola.

B. *Pengolah Citra*

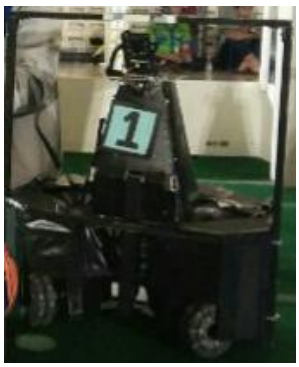
Modul pengolah citra diimplementasikan menggunakan sensor kamera yang terhubung langsung pada modul pengontrol utama. Dilakukan segmentasi warna untuk mengetahui posisi dan jenis objek yang ada di lapangan.

C. *Navigasi*

Modul navigasi berfungsi untuk mengetahui posisi robot dilapangan berdasarkan medan magnet bumi. Modul ini diimplementasikan dengan sensor kompas magnetik HMC8533L

D. *Integrasi*

Setelah dilakukan pengujian dari setiap kompnen robot, dilakukan integrasi pada sistem. Pengujian dilakukan secara keseluruhan dengan memberikan perintah kepada robot untuk menemukan bola, menggiring bola, dan menendang bola. Hasilnya robot dapat melakukan semua tugas trsebut dengan baik.



Gambar 3. Integrasi Robot

V. KESIMPULAN

Desain robot sepak bola beroda telah diimplementasikan dan diuji. Pengujian dilakukan pada masing masing komponen kemudian secara keselurahn sehingga menjadi robot utuh. Hasilnya robot dapat melakukan tugasnya dengan baik seperti mengenali bola, menggiring bola dan menendang bola ke arah gawang lawan.

REFERENSI

[1] *Prociding_ISRC_2016*
[2] Panduan KRSBI Beroda 2017 - Kontes Robot Indonesia, http://www.kontesrobotindonesia.org/datakontes/2017/PanduanKRSBIBeroda2017.pdf Diakses pada Juni 2017
[3] CMUCAM5 Data Sheet, http://www.cmucam.org/projects/cmucam5/ Diakses pada Juni 2017
[4] HMC5883L Data Sheet, https://media.digikey.com/PDF/HMC5883L.pdf Diakses pada Juni 2017
[5] STM32f407vg Data Sheet, http://www.st.com/en/microcontrollers/stm32f407vg.html Diakses pada Juni 2017
[6] J.J.T.H. de Best, M. J.G. van de Molengraft, and M. Steinbuch. A novel ball handling mechanism for the RoboCup Middle Size League. Mechatronics, 21(2), 2011.
[7] K.J. Meessen, J.J.H. Paulides, and E. Lomonova. A football kicking high speed actuator for a mobile robotic application. Proceedings of the 36th Annual Conference of the IEEE Industrial Electronics Society, pages 1659–1664, 2010.

Image Normalization for Reducing Light Effect Based on HOG Object Detection

Abdurrohman Ahmad Rofiq, Agil Athallah, Mufti Irawan Fikri*, Ilham Nur Ahmad, M. Ridho Fuadin, Hanif Baskoro, Tribagus Novandi Winantyo, Ikrima Sabri, Arifandhi Nur Muhamad, Farchan Hakum Raswa, Muhammad Hadyan Akbar, Teguh Nur Iman, Muhammad Haritsah Mukhlis, Gabrielle Satya Mahardika, Bakhtiar Alldino Ardi Sumbodo, dan Dini Nur Annisa

Gadjah Mada Robotics Team
Universitas Gadjah Mada
Bulaksumur, Yogyakarta, Indonesia, 55284
mufti.irawan.f@mail.ugm.ac.id

***Abstract*—This paper present robust algorithm to recognize white ball. The main focus is to recognize with low information color of ball using image processing. This project consist of four main stages such as image segmentation, contour analyze, feature extraction, and object recognition. Image segmentation based on procces to make threshold region object. Contour analyze to analyze shape, contour to compute region that we want. Feature extraction can work after we has best result of contour analyze. The last is, object recognition was recognized by training data object based SVM method and we use HOG to detect feature of training data image, so we can detect the object ball and can make difference which object ball or not object ball. The effectiveness of purpose method has been revealed in this paper through experimental result. True positive detection is ball detection and recognition and will be implemented robot to tracking ball.**

Keywords—HOG; SVM; Image Segmentation; Contour Analyze; Feature Extraction; Object Recognition.

I. INTRODUCTION

Each year, Robocup soccer leagues update their rules, to force participating teams for develop more advanced feature for their robot and make competitions more similar to real soccer leagues. Since Robocup 2015, rules of humanoid kidsize league has been changed to use 50% white ball which previously use orange ball and the goal post are now white instead of yellow. It will make affect more effort to recognize the ball because the goal and line are white color, and we try to reduction of color coding ball on the field.

For win the competition, this will need method for a fast, effective, reliable, robust and good sensitivity to recognize ball. Thus, it should have a low complexity computation and short time to recognize ball. The more challenge are to make minimum miss detection other object on the field such as white line, boundary white line, white robot, and white goal post.

Machine learning has become one of many trends of information technology with ever increasing of data becoming available there is good reason to believe that smart data analysis will become even more pervasive as necessary ingredient for help much of humans for technology progress. Taste of machine learning has many effect to solve much of problems and provide good guarantee for the solutions[1].

Machine learning include two aspects: the character descriptor of operator and a learning algorithm. Character descriptor operator has the features of haar wavelet[2], the HOG (Histogram Oriented Gradient) [3], LBP (Local Binary Pattern) [4], Shapelet features[5], edge character (edgelet) [6] and so on. Learning algorithm has support vector machine (SVM),neural network, etc.

Ball recognition has been many research until now. Mu *et al* , for ball detection has been making by combination of two different techniques such as one is a fast circle detection algorithm, contour feature method (CF), based on contour information. While another one, modified gradient Circle Hough Transform (GCHT) is introduced as supplement for the CF[6]. Farazi *et al,*as member of Nimbro team, to recognize ball has been using 2 stages consist of first stage of processing, ball candidates are generated based on color segmentation, color histograms, shape, and size. In the second stage of processing, a dense of histogram of oriented gradient (HOG) descriptor[3] is applied in the form of a cascade classifier, with use of AdaBoost technique[7]. In this year, CIT-Brains team has achieve to uses deep learning and the mehod is CNN (Convolutional Neural Network) to recognize ball, goal posts, and opponents. They are currently trying to shorten the collecting time for training data using their own method[8]. In their recent work, NUbots team has been develop a fully-autonomous method of look-up table adaptation for changing lighting condition to recognize ball[9].

The main topic of this paper is to recognize ball using segmentation ball, contour analyze, shape, and features of ball candidates that had extracted by Histogram of Oriented Gradient (HOG), will be checked by a file that consist of database of training data examples that has been developed by Support Vector Machine (SVM). The training data examples are consist of positive data sets and negative data sets. Positive data set consist of image of ball and negative data set consist of all environment images that not include the ball.

II. METHOD

In order to carry out the robust algorithm in recognize ball, Figure 1 depicted the flowchart of system's method.

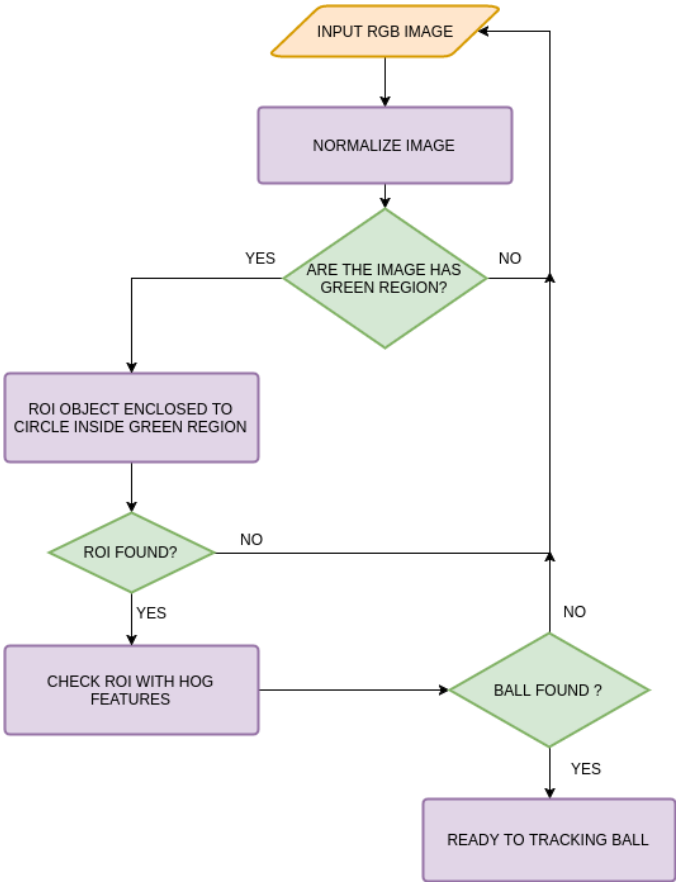


Fig. 1. Flowchart of ball recognition method

RBG (Red Green Blue) image The RGB color space defines specific color into combinations of Red, Green, and Blue. RGB is the most color space that in used for input image. Most of the image capture devices such as camera, use RGB color space. RGB has some advantages, but also has many disadvantages in several conditions.

Normalize Image Normalizing Image means that we divide each pixel with total intensities of the image. Normalizing RGB image, as example, means that we divide each pixel with intensities of R, G and B in the respective channels. Sometimes, we want to get rid of the image from some noises that occurs because of the light effect or shadows. Normalizing of an image can be effective way for achieving this.

The normalized value will be:

R/S, G/S, and B/S where S=R+G+B.

The new RGB value after normalization can represent the color image that more clear than before. The figure below show how the image normalization works.



Fig. 2. Image before normalization



Fig. 3. Image after normalization

Now, we want to get only the green component of the image. We can use the G channel after normalization, then find the object based on G channel. There are some probabilities to find out the green region. As example, If we find more than one green region (there are many contours at G channel), we can try to take only the large contour that represent the green field. After that, focus on the object inside green region.

ROI (Region of Interest) The goals of ROI is to select a small pieces of region inside the image which we want to focus on it. The idea of ROI is that an image can be crop or divide into small parts in order to restricting of area that will be analyzed. Because we want to find a ball, we may use Hough transform or other method to get the object that enclosing to circle. We can make the ROI object based on the circle found in the green region.

HOG Features Checking HOG, stands for Histogram of Oriented Gradient, are the object detection method which is based on gradient matching. The basic idea is that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions, even without precise knowledge of the corresponding gradient or edge positions[10]. In order to implementing the HOG, SVM (Support Vector Machine) usually used for training and making the features. HOG features are file which has an extension .yaml . The circle ROI found in the previous step then will be checked by HOG features. This is the final step to deciding whether the object in the ROI are a ball or not. If there are more than one ROI, means that there are more than one object

enclosing to circle, they all must be checked by HOG features one by one until the ball found.

III. Experimental Result

We succesfully build algorithm for white ball recognize in our robot, then region frame of ball at the center position of the frame camera and robot can follow the ball. In this experiment, we give value for expossure, brightness and contrast camera setting to make white ball look good and then we have "threshold" variable use for callibrate green field. The experiment result is shown at the table below.

TABLE I FALSE DETECTION OF HOG FEATURE

		False Detection			
Brightness	Contrast	HOG without Normalize	%false	HOG with Normalize	%false
50	50	38 of 386	9,84%	10 of 386	2,59%
50	70	45 of 386	11,66%	9 of 385	2,34%
50	90	49 of 384	11,72%	9 of 384	2,34%
50	100	50 of 385	12,73%	9 of 385	2,34%
70	50	37 of 386	9,61%	7 of 386	1,81%
70	70	48 of 385	12,47%	11 of 385	2,86%
70	80	52 of 385	13,51%	11 of 385	2,86%
70	100	65 of 385	16,88%	13 of 385	3,38%
90	50	43 of 385	11,17%	8 of 385	2,08%
90	70	46 of 385	11,95%	9 of 385	2,34%
90	90	58 of 385	15,06%	12 of 385	3,12%
90	100	65 of 385	16,88%	18 of 385	4,68%

Table 1 shows that false detection using image with normalization is less than image without normalization. But even if using normalization, there still has some false detection in the implementation. It is because of many factors, such as quality of HOG features, over brightness or over contrast of the input image, fail at normalization stage, or fail at finding the circular object. In other hand, this normalization are proved to reducing the disturbance caused by light.

Based on the table 1, we can see ability of HOG feature of SVM to deciding frame with detection. If false detection of all frame experiment more fewer than before, the experiment has succesfully, because if false detection of all frame experiment increases, robot will confuse because it is can distract to tracking object of ball.

The best of ball recognition while brightness get value 70 of 100 and contrast get value 50 of 100, so HOG give false detection without normalise is 9.61 % and with normalise is 1,81 %. The precentage of HOG normalise is good with 1,81 % is and it is enough for implemented to robot.

For example, true and false detection during experiment can be shown in figure 4 and figure 5 below.

There are the problem of this algorithm. this algortihm are to diffilcut for calibrate detecion white ball because this algorithm is not adapt for light and luminance change. Light and luminance change give effect for white region of ball, so that algorithm can not detail recognize ball for far distance of ball from robot.

When robot try to recognize ball at the far distance, robot will detect many of object or false positive detection and robot will confuse to detect and recognize object of ball.



Fig. 4. true detection



Fig. 5. false detection

IV. Conclussion and Future Works

In this paper, we present our approach to detect and recognize ball at the far distance from robot. In this algorithm that we have implemented to our robot, we succesfullly build robot can detect and recognize region of ball as far as 1.5 meters from robot.

In the future works, we would focus to develop our algorithm to robust ball recognize as far as we can do to implemented to our robot and at the proccesing image of ball to minimaze disturbance from the light and luminance.

Acknowledgement

We convey our gratitude to Universitas Gadjah Mada, all members of Alfarobi team, and all Gadjah Mada Robotics Team supervisors and member to support this project.

Project's funding was provided by Gadjah Mada Robotics Team (GMRT) under research grant of Universitas Gadjah Mada for research period 2016 – 2017.

REFERENCES

[1] A. Smola and S.V.N. Vishwanathan. Introduction to machine learning.

[2] Viola P, JonesM, and Snow D. Detecting pedestrians using pattern of motion appereance. Proceedings of the IEEE International Conference on Computer Vision, Beijing,2003: 734-741.

[3] N. Dalal and B. Triggs, "Histogram of Oriented Gradients for Human Detection". IEEE Conference on Computer Vision and Pattern Recognition , San Diego, CA, USA: IEEE, 2005, 1:886-893.

[4] Ojala T,, Pietik imen M, Harwood D. "A comparativr study of texture measures with classification based on featrure distribution", Pattern recognition,1996,29:51-59

[5] Sabzmeydani P, Mori G., "Detecting pedestarians by Learning Shapelet Features. IEEE Conference Computer Vision and Pattern Recognition", Minneapolis, Minnesota, USA: IEEE, 2007

[6] J. Mu and Y. Li, "A new Efficient Real-Time Arbitary Colored Ball Recognition Method for a Humanoid Soccer Robot", 2016 12th World Congress on Intelegent Control and Automation(WCICA), Guilin China, June 2016, pp. 494-499

[7] H. Farazi, P. Allgeuer, and S. Behnke, "A Monocular Vision for Playing Soccer in Low Color Information Environments", in *Proceedings 15th IEEE-RAS International Conference on Humanoid Robots*, Seoul Korea, November 2015.

[8] Y. Seki, and Friends, "CIT Brains (Kid Size League), in Proceedings Team Description Paper Robocup Humanoid League 2017, Nagoya Japan, 2017.

[9] M. Amos, and Friends, "The NUbots Team Description Paper 2017", in Proceedings Team Description Paper Robocup Humanoid League 2017, Nagoya Japan, 2017.

[10] B. Kishor B, R. U. Shekokar, "Human Body Detection in Static Images Using HOG & Piecewise Linear SVM", in Internaional Journal of Innovative Research & Development. June 2014, pp. 180-184

Algoritma Multi Planning Pada Robot Soccer

Indah Permata Suryaningsih*, Ahmad Risal, dan Hendra Jaya
Dept. Of Electronics Engineering, Faculty of Engineering
Universitas Negeri Makassar
Makassar, Indonesia
permataindah.ips@gmail.com

***Abstract*— Control System plays a very important role in the world of robotics because it allows the robot to move automatically only with the commands that have been made (Program). So that robots can act like humans and can someday be used in a wider scope. One of the sports that entered the world of robotics is football. Robot soccer (football robot) is a intelligent control system with multiprogramsystem. Control of robot using automatic control of soccer robot with path planning method, Mobile robot soccer work automatically to recognize the movement of ball and direction of opponent's goal. From Planning Path method then developed algorithm with Multi Planning method in the form of menu processing. In the Robot Soccer System contained The initial design of soccer robot strategy is the individual movement of each robot to avoid the existing obstacles without collision. The robot uses a vision sensor captured from the camera and the robot can move itself as desired.**

Keywords— Multiplanning; Soccer Robot.

I. PENDAHULUAN

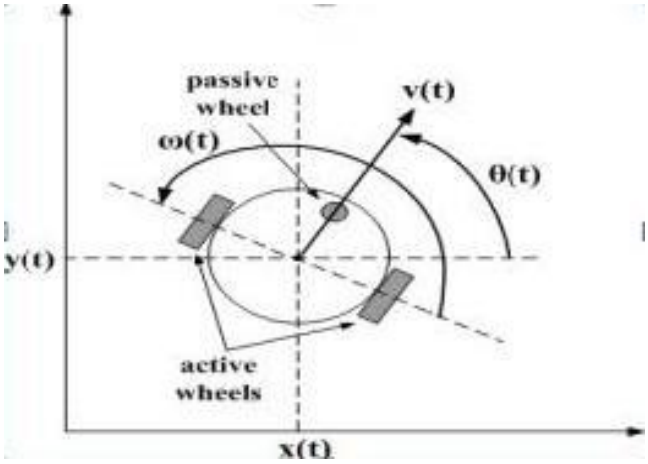
Penerapan robot selama ini telah dapat membantu manusia di berbagai bidang terutama bidang Industri. Sistem Kontrol memegang peran yang sangat penting dalam dunia robotika karena memudahkan robot dapat bergerak secara otomatis hanya dengan perintah yang telah dibuat (Program), sedangkan sensor merupakan indra yang sangat penting dalam memberikan informasi ke sistem kontrol. Sensor merupakan devais atau komponen yang berfungsi mendeteksi (hasil) gerakan atau fenomena lingkungan yang diperlukan oleh sistem kontroler yaitu logika Ada atau tiada (0/1). Dapat dibuat dari sistem yang paling sederhana seperti sensor ON/OFF menggunakan *limit switch*, sistem analog, sistem bus parallel, sistem bus serial, hingga sistem mata kamera [1]. Pengembangan dunia robotika lebih jauh, para peneliti mencoba membuat robot untuk melakukan berbagai macam jenis olahraga dan permainan yang dibuat manusia. Harapannya, agar robot dapat bertindak seperti manusia dan suatu saat dapat digunakan dalam lingkup yang lebih luas. Salah satu cabang olahraga yang dimasuki dunia robotika adalah sepak bola. Olahraga yang dimainkan di seluruh dunia ini memiliki unsur kerja sama tim yang tinggi, strategi yang kompleks, dan permainan yang beragam dalam penyelesaian misi.

Robot *soccer* (sepakbola robot) merupakan *system* pengendalian cerdas (inteligent control) dengan multiprogram system [2]. Pengendalian robot menggunakan kendali otomatis pada robot *soccer* dengan metode *path planning*, *path planning* adalah perencanaan jalan yang akan dilalui oleh robot. Jalan yang akan dilalui robot dipilih berdasarkan bahwa jalan tersebut adalah jalan yang tercepat dalam proses penyelesaian misi. *Mobile* robot *soccer* bekerja secara otomatis untuk mengenali pergerakan bola dan arah gawang lawan. Dari metode *path Planning* kemudian dikembangkan algoritma dengan metode *Multi Planning* dalam bentuk pengolahan menu. Dengan metode Algoritma tersebut dibuatlah aturan-aturan tertentu (*if-then rules*) yang dirancang sedemikian rupa sehingga robot mobil tersebut dapat berpikir seperti seorang manusia dalam proses pengambilan keputusan (*human behavior based*) dalam hal ini adalah operator yang merancang algoritma tersebut. Pengambilan keputusan yang dimaksud adalah perilaku robot dalam melakukan kemampuan-kemampuannya seperti yang telah disebutkan sebelumnya..

II. TINJAUAN PUSTAKA

A. Robot Soccer

Pada robot sering digunakan istilah-istilah untuk menunjukkan geometri dan spesifikasi robot seperti "poros gerakan" dan "derajat kebebasan". Setiap benda bergerak berarti menempuh jarak dari satu titik ke titik lain yang berada pada satu bidang. Pergerakan ini mempunyai garis sumbu atau titik yang tidak berubah yang disebut poros. Derajat kebebasan adalah jumlah arab bebas (*independent*) dari suatu robot (*end effector*) yang menentukan arah dari pergerakannya [3]. Parameter-parameter yang bertindak sebagai variabel masukan dari robot *soccer* adalah koordinat posisi (x, y) yaitu titik pusat robot dan sudut arah (Φ) yaitu sudut antara sumbu x dengan poros robot [4]. Sebagai variabel keluarannya adalah sinyal sudut-pengarah (θ) yaitu sudut antara poros dengan arah roda. Pada Gambar 1 ditunjukan diagram simulasi dari posisi dan sudut arah robot.



Gambar 1. Simulasi dari posisi dan sudut arah robot

Pada Sistem Robot *Soccer* terdapat Rancangan awal pembuatan strategi robot sepak bola adalah gerakan individu dari masing-masing robot untuk menghindari rintangan yang ada tanpa tabrakan. Robot menggunakan sensor visi yang ditangkap dari kamera dan robot dapat bergerak sendiri seperti yang diinginkan. Robot menghasilkan parameter kecepatan untuk memindahkan dirinya dari titik awal ke titik sasaran tanpa menabrak objek-objek lain yang ada disekitarnya. Pada setiap langkah diperlukan data input yang diperoleh secara waktu nyata dari sistem visi, kemudian robot menghasilkan referensi linear sudut arah pergerakan dan kecepatan untuk menuju ke titik sasaran. Untuk menghasilkan referensi sudut pergerakan dan kecepatan robot mengambil *input* data yang meliputi jarak serta sudut antara hambatan, tujuan pergerakan dan tempat-tempat kosong disekitar robot tersebut. Robot mengambil nilai-nilai sebagai berikut : 1) jarak menuju ke tujuan 2) jarak menuju halangan terdekat 3) sudut antara orientasi robot saat ini dan arah menuju ke sasaran 4) sudut antara orientasi robot saat ini dan arah ke rintangan terdekat [4].

B. Algoritma

Algoritma dapat diartikan urutan penyelesaian masalah yang disusun secara sistematis menggunakan bahasa yang logis untuk memecahkan suatu permasalahan. Dalam menentukan algoritma untuk menyelesaikan suatu permasalahan, mungkin dihadapkan oleh beberapa pilihan algoritma. Oleh karena itu harus memiliki rambu-rambu dalam menentukan pilihan algoritma. Pertimbangan dalam pemilihan algoritma adalah, pertama, algoritma haruslah benar. Artinya algoritma akan memberikan keluaran susuai seperti yang diharapkan dari sejumlah masukan yang diberikan. Tidak peduli sebagus apapun algoritma, jika memberikan keluaran yang salah, maka sudah pasti algoritma tersebut bukanlah algoritma yang baik. Pertimbangan kedua yang harus diperhatikan adalah harus mengetahui seberapa baik hasil yang dicapai oleh algoritma tersebut. Hal ini pentingterutama pada algoritma yang memerlukan aproksimasi hasil yaitu algoritma yang hasilnya hanya berupa pendekatan. Algoritma yangbaik harus mampu memberikan hasil yang sedekat mungkin dengan nilaiyang sebenarnya.Ketiga adalah efisiensi algoritma. Efisiensi algoritma dapat ditinjau dari dua hal yaitu efisiensi waktu dan memori. Meskipun algoritma memberikan keluaran yang benar atau paling mendekati, tetapi jika harus menunggu lama untuk mendapatkan hasil semisal berjam-jam untuk mendapatkan keluarannya maka biasanya algoritma tersebut biasanya tidak akan menjadi pilihan utama, setiap orang menginginkan keluaran yang relatif cepat. Begitujuga dengan memori, semakin besar memori yang terpakai maka semakin jelek algoritma tersebut. Dalam kenyataannya, setiap orang bisa membuat algoritma yang berbeda untuk menyelesaikan suatu permasalahan, walaupun terjadi perbedaan dalam menyusun algoritma, tentunya mengharapkan keluaran yang mirip atau sama. Jika dihadapkan pada permasalahan seperti ini maka sebaiknya pilih algoritma yang paling efisien dan cepat. Tujuan dari belajar logika dan algoritma adalah agar dapat membiasakan diri melakukan suatu perencanaan apabila menyelesaikan suatu masalah. Karena suatu permasalahan yang diselesaikan dengan suatu perencanaan yang matang maka akan mendapatkan solusi yang lebih optimal dibandingkan menyelesaikan masalah tanpa menggunakan suatu perencanaan [5].

III. SISTEM ALGORITMA

A. Path Planning

Path planning, merupakan hal penting dalam dunia robotika khususnya bagi robot *mobile* otonom yang memungkinkan robot menemukan jalur terpendek atau optimal antara dua titik. *Path planning* membutuhkan kondisi sekitar dan robot untuk mengetahui lokasinya sehubungan dengan sekitarnya [6]. Dengan demikian robot mampu menyesuaikan dirinya sendiri agar mampu melewati rintangan yang berada disekitarnya. Ada 2 jenis sistem navigasi yang biasa digunakan pada robot mobil, yaitu:

1). *Global Path Planning*

Pada metode ini robot telah memiliki informasi tentang keseluruhan lingkungan termasuk objek-objek diam yang berada pada lingkungan tersebut. Pada pendekatan ini robot telah memiliki informasi lengkap jalur yang akan dilewati bahkan sebelum dia bergerak.

2) *Local Path Planning*

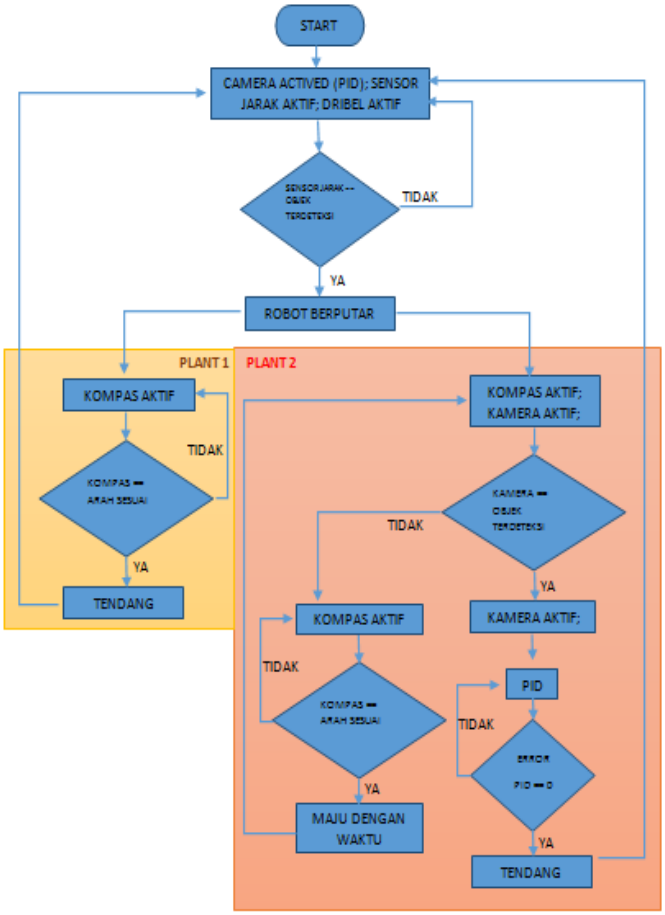
Sedangkan pada metode robot tidak mengetahui informasi mengenai lingkungan dimana dia ditempatkan namun dengan pendekatan ini robot dapat membuat sendiri jalur yang baru untuk bisa ke tempat tujuan. Pada perancangan ini digunakan metode yang kedua karena pada pendekatan yang ke dua robot akan dapat beradaptasi dengan cepat dan mudah dengan perubahan lingkungan.

B. Multiplanning

Pada perancangan selain *path planning* , juga terdapat algoritma sebagai komponen pendukung dan dasar pergerakan dari robot mobil [3]. Algoritma yang digunakan adalah *if-then rules*. Algoritma *if-then rules* dibuat berdasarkan pola pikir manusia (*human behaviour based*) dalam hal ini adalah operator yang merancang algoritma. Fungsi utama dari algoritma adalah untuk mencari jarak terdekat untuk pergerakan robot dari posisi awal menuju posisi tujuan.

C. *Algoritma Path Planning*

T Masalah untuk menemukan jalur terpendek dari suatu titik ke titik lainnya melalui grafik sangat diminati di beberapa bidang ilmu, seperti penentuan paket optimal untuk paket data dalam jaringan, dan jarak fisik dalam bidang robotik. Salah satu algoritma paling dasar dan paling sederhana adalah algoritma Dijkstra. Algoritma Dijkstra merupakan algortima yang dapat memecahkan masalah pencarian jalur terpendek dari suatu graf pada setiap simpul yang bernilai tidak negatif. Dikjstra merupakan algoritma greedy, yaitu algoritma yang sering digunakan untuk memecahkan masalah yang berhubungan dengan suatu optimasi.



Gambar 2 Flowchart Program Multiplanning Robot Soccer

Dalam pencarian jalur terpendeknya algoritma dijkstra bekerja dengan mencari bobot yang paling minimal dari suatu graf berbobot, jarak terpendek akan diperoleh dari dua atau lebih titik dari suatu graf dan nilai total yang didapat adalah yang bernilai paling kecil [7]. Flowchart Program *Multiplanning* Robot *Soccer*.

Misalkan G adalah drag berarah berlebel dengan titik-titik V(G) = {v1,v2,....,vn} dan path terpenek yang dicari adalah dari v1 ke vn. Algoritma dijkstra dimulai dari titik v1. Dalam iterasinya, algoritma akan mencari satu titik yang jumlah bobotnya dari titik 1 terkecil. Titik-titik yang terpilih dipisahkan, dan titik-titik tersebut tidak diperhatikan lagi dalam iterasi berikutnya.

IV. KESIMPULAN DAN SARAN

A. *Kesimpulan*

Berdasarkan hasil penelitian ini, dapat disimpulkan beberapa hal Algoritma *Multi Planning* Pada Robot *Soccer*:

1. Robot *soccer* (sepakbola robot) merupakan system pengendalian cerdas (*inteligen control*) dengan multi program *system*.
2. *Path planning* adalah perencanaan jalan yang akan dilalui oleh robot. a) *Global Path Planning*. Pada metode ini robot telah memiliki informasi tentang keseluruhan lingkungan termasuk objek-objek diam yang berada pada lingkungan tersebut. Pada pendekatan ini robot telah memiliki informasi lengkap jalur yang akan dilewati bahkan sebelum dia bergerak.; b) *Local Path Planning*, Sedangkan pada metode robot tidak mengetahui informasi mengenai lingkungan dimana dia ditempatkan namun dengan pendekatan ini robot dapat membuat sendiri jalur yang baru untuk bisa ke tempat tujuan.
3. Pada robot sering digunakan istilah-istilah untuk menunjukkan geometri dan spesifikasi robot seperti "poros gerakan" dan "derajat kebebasan".

B. *Saran*

Berdasarkan kesimpulan penelitian, maka penulis merekomendasikan berupa saran-saran sebagai berikut :

1. Untuk meningkatkan efisiensi penggunaan kontroler dalam Robot *Soccer* perlu rasionalisasi program dengan perangkat keras yang digunakan kembali struktur perangkat/ komponen yang ada sekarang yang disesuaikan berdasarkan tuntutan kebutuhan robot.
2. Pengelolaan data *planning* Robot *soccer* yang hanya menekankan pada pendekatan *multiplanning* harus disempurnakan melalui perubahan kontroler yang sesuai.
3. Penelitian ini belum maksimal, karena hanya melihat penggunaan *multiplanning* pada robot *soccer*, maka untuk kebutuhan penelitian berikutnya bagi tim Krsbi selanjutnya untuk menerapkan *system* kontroler yang lebih maksimal.

ACKNOWLEDGMENT

Terimakasih diucapkan pada Dosen Pembimbing (Dr. Hendra Jaya, M.T), ketua Jurusan Pendidikan Teknik Elektronika FT UNM (Dr. Darlan Sidik, M.Pd), dan Kepala Laboratorium PTA FT-UNM (Ridwansyah, S.T., M.T), serta Ucapan terimakasih yang setinggi-tingginya kepada Dirjen Belmawa Kemristek Dikti yang telah memberikan kesempatan untuk mengikuti Kontes Robot Indonesia.

REFERENCES

[1] Hendra Jaya. Desain Dan Implementasi Sistem Robotika Berbasis Mikrokontroller. Penerbit: Edukasi Mitra Grafika, 2016.

[2] Jong-Hwan Kim, dkk, 1998. Path Planning and Role Selection Mechanism for Soccer Robots, E-journal, Dept. of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Proceedings of the 1998 IEEE International Conference on Robotics& Automation Leuven, Belgium.

[3] Selvia Lorena Br Ginting, Dkk, Penerapan Algoritma Best-Path Planning Untuk Aplikasi Pencarian Rute Transportasi Publik Berbasis Android, E-Journal Teknologi, Universitas Komputer Indonesia, Universitas Riau/PhD Student at Curtin University, Universitas Komputer Indonesia

[4] Kim, C.J., M.S. Park, A.V. Topalov, D. Chwa and S.K. Hong., 2007. Unifying strategies of obstacle avoidance and shooting for soccer robot systems. Proceedings of International Conference on Control, Automation and Systems, Oct. 17-20, Seoul, Korea, pp. 207-211. DOI: 10.1109/ICCAS.2007.4406909

[5] Ali Ridho Barakbah. LOGIKA DAN ALGORITMA. Program Studi Teknik Informatika Departemen Teknik Informatika dan Komputer Politeknik Elektronika Negeri Surabaya, 2013.

[6] Tenday Agus Setya Wibawa, dkk, Rancang Bangun Robot Soccer Wireless Berbasis Mikrokontroller, E-Journal, Politeknik Elektronika Negeri Surabaya Institut Teknologi Sepuluh Nopember.

[7] Prabancoro Adhi Catur Widagdo, Dkk, Sistem Koordinasi Dan Kecerdasan Buatan Untuk Strategi Bertanding Pada Robot Soccer, E-Journal, Program Studi D4 Teknik Mekatronika, Departemen Teknik Mekanika dan Energ Politeknik Elektronika Negeri Surabaya.

Pembuatan Dataset Pelatihan dengan Optimalisasi Pseudocoloring Berbasis Depth untuk Faster Region-Proposed Convolutional Network Network

Yudi Pratama*, Widyawardana Adiprawita, dan Kusprasapta Mutijarsa
Program Studi Teknik Elektro
STEI – Institut Teknologi Bandung
Bandung, Indonesia
yudi.prtm@yahoo.com

***Abstrak*---** **Pada proses evakuasi bencana alam, tim SAR *(Search and Rescue)* biasanya mengalami kendala berupa intensitas cahaya yang rendah bahkan kondisi gelap total. *Semantic vision* mampu menginterpretasi citra berupa korban bencana pada kondisi cahaya dengan intesitas rendah. *Semantic vision* menggunakan kamera RGBD berupa kamera Kinect Xbox 360 untuk melakukan interpretasi citra lingkungan berupa *raw depth data*. Pada bagian pendeteksian objek, *semantic vision* menggunakan *deep neural network* (DNN) dengan arsitektur Faster RCNN yang dikembangkan dengan Caffe. Untuk melatih DNN, maka dilakukan pendefinisian kelas objek berupa manusia berdiri, manusia duduk atas, manusia duduk bawah, dan manusia tidur, jumlah *dataset* sejumlah 6255 gambar, dan *image preprocessing* berupa *pseudocoloring* untuk *raw depth dataset*. Hasil yang diperoleh dari *semantic vision* antara lain memiliki mAP 56,62 % untuk 1039 citra pengujian, dapat melakukan pendeteksian manusia pada kondisi intensitas cahaya rendah yakni kecil dari 22 lux, dan mampu melakukan pendeteksian manusia dengan jarak optimal 2-3 meter.**

***Kata kunci*---kamera RGBD; pseudocoloring; raw depth; Caffe.**

I. PENDAHULUAN

Indonesia merupakan negara kepulauan yang memiliki konfigurasi geografis yang beragam. Kondisi iklim tropis dan aktivitas geologi yang intens sering kali menyebabkan timbulnya bencana alam. Data menyebutkan bahwa pada tahun 2011-2015 terdapat 1881 korban meninggal dunia dan 231 korban hilang akibat bencana alam[[1]]. Tanah longsor merupakan penyumbang angka paling banyak dalam memakan korban jiwa. Selain bencana alam tentunya bencana dapat terjadi akibat kecelakaan dan tindakan terorisme. Bencana tersebut dapat berupa kebakaran, gedung runtuh, maupun pemboman.

Untuk membantu proses SAR, perlu dilakukan penilaian bencana. Adanya penilaian bencana bertujuan memberikan gambaran-gambaran atau informasi untuk pengambilan keputusan yang bertujuan mengurangi risiko saat bencana terjadi. Informasi yang didapatkan akan sangat berguna untuk *search and rescue*. Tujuan dilakukannya penilaian bencana pada tahap awal sebuah bencana adalah memberikan informasi untuk tindakan darurat (SAR), lokasi tepat, status lokasi-lokasi yang membutuhkan perhatian (terdapat korban selamat), dan akses menuju ke lokasi tersebut [1]

Dalam makalah ini, akan dijelaskan solusi bagian dari sebuah robot berupa *semantic vision* yang mampu membantu memudahkan manusia dalam penanganan bencana, khususnya menjawab beberapa tantangan penilaian bencana (*disaster assessment*). *Semantic vision* dapat mengenali dan membedakan pose korban di sekitarnya pada kondisi dengan cahaya cukup maupun gelap total. Dengan ini diharapkan angka korban meninggal dunia dan korban hilang dapat ditekan karena robot akan berupaya membedakan korban masih hidup dan tidak. Robot diharapkan dapat mengumpulkan informasi yang bersifat obyektif, sehingga proses *Total Disater Risk Management (TDRM)* dapat dilakukan dengan baik serta pengambilan kebijakan dapat terbantu untuk mentukan langkah yang akan diambil.

II. TEORI PENDUKUNG

A. Depth Encoding

Citra yang dihasilkan hanya dari *raw depth data* hanya memiliki satu *channel*. Pada *deep learning framework* yang umumnya digunakan untuk pengolahan citra atau *computer vision* membutuhkan *input* tiga *channel*. Oleh karena itu, diperlukan sebuah *image enhancement process* untuk memanipulasi masalah tersebut. Terdapat beberapa metode yang dapat digunakan [2], antara lain:

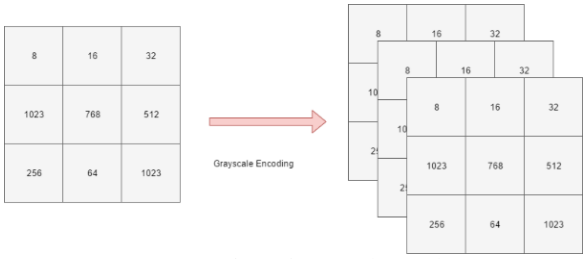
1. *Grayscale encoding*
2. *Surface normal*
3. HHA *encoding*
4. *Pseudocolor encoding*

Berikut adalah penjelasan mengenai metoda *image enhancement* pada *depth data:*

Grayscale Encoding

Metode ini merupakan metode paling sederhana. *Depth data* 1 *channel* yang diperoleh sebelumnya disalin ke 2 *channel* lain. Ilustrasi dan gambar di bawah ini menunjukkan proses *grayscale encoding* dan hasil *encoding*.

[1] Bencana banjir, tanah longsor, gempa bumi, tsunami, kebakaran, puting beliung, dan letusan gunung api.



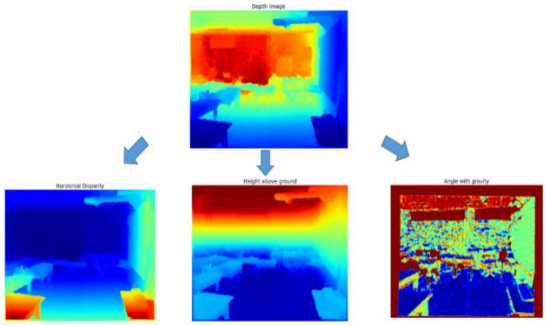
Gambar 1. Ilustrasi *grayscale encoding*



Gambar 2. *Depth encoding* dengan menggunakan *grayscale encoding*

HHA Encoding

Encoding dilakukan pada setiap *pixel* menjadi tiga *channel*, yaitu *horizontal disparity*, *height above ground*, dan sudut normal permukaan lokal *pixel* dibuat dengan arah *inferred gravity* [3]. Semua nilai *pixel* diskalakan secara linier ke rentang 0-255. HHA melakukan *encode* pose geosentris yang menekankan pada diskontinuitas komplementer pada sebuah citra, seperti *depth*, *height*, dan normal permukaan) [3]. Struktur pada HHA memiliki kemiripan dengan RGB, sehingga *neural network* yang dirancang untuk citra RGB dapat digunakan untuk citra HHA. Tepi-tepi pada *disparity* dan sudut dengan arah gravitasi berhubungan dengan batas-batas sebuah objek dan mirip dengan apa yang terjadi pada RGB [3]. HHA *encoding* memerlukan *cost* besar dalam melakukan perhitungan untuk menghasilkan citra yang diinginkan. Penjelasan mengenai proses *encoding* yang mengubah *depth data* menjadi HHA tidak dijelaskan pada makalah [3].



Gambar 3. HHA Encoding [3]

Jika masukan berupa *pointcloud* (memerlukan data RGBD dengan RGB untuk x, y dan D untuk z sebuah titik), proses *encoding* dijelaskan oleh Nicolai Harich di dalam *master thesis* miliknya [4] sebagai berikut:

Height over Ground (H)

Perhiungan dilakukan per elemen *pixel*. Setiap titik dihitung jaraknya terhadap *floor*. Nicolai Harich mengasumsikan *static groundplane*, sehingga bidang tersebut dapat dirumuskan dengan:

$$ax + by + cz + e = 0$$

Keterangan:
a, b, c, dan e adalah parameter model
x, y, z menunjukkan koordinat titik dalam ruang

Menghitung jarak antara antara bidang dengan titik (x_p, y_p, z_p) dilakukan dengan persamaan berikut ini.

$$d = \frac{ax_p + by_p + cz_p}{\sqrt{a^2 + b^2 + c^2}}$$

Horizontal Disparity (H)

Diperlukan dua nilai, yakni *baseline* B dan *focal length* F untuk memetakan *depth* ke *disparity*. Perhitungan dapat dilakukan secara langsung dengan menggunakan nilai z yang merupakan jarak titik tertentu terhadap kamera.

$$d = \frac{B.f}{z}$$

Angle (A)

Nilai ini menunjukkan sudut tertutup α antara normal permukaan $\vec{n}$ dan vektor gravitasi $\vec{g}$. Vektor gravitasi diasumsikan tetap. Titik-titik normal permukaan *groundplane* memiliki arah yang mirip dengan vektor gravitasi.

$$\alpha = cos^{-1}\left(\frac{\vec{n}.\vec{g}}{|\vec{n}||\vec{g}|}\right)$$

Pseudocolor Encoding

Menurtu Eitel dkk [2], metode *pseudocolor encoding* memiliki performa yang jauh lebih baik dibandingkan dengna HHA *encoding*. Metode ini pertama kali melakukan proses normalisasi terhadap *depth image* yang masuk ke nilai 0-255 (*depth data* dapat memiliki nilai di atas 1023). Selanjutnya, proses *jet colormap* dilakukan pada *input image* ya mengubah *input* dari 1 channel ke 3 channel dengan pewarnaan *depth*. Pada setiap *pixel* (*i,j*) untuk *depth image* dengan ukuran WxH, dilakukan pemetaan *depth* ke warna antara merah dan biru, yang secara dasar mendistribusikan *depth data* ke RGB *channel* [2]. Tepi-tepi pada hasil transformasi ini berhubungan dengan batas-batas objek yang menjadi perhatian [2].

Tentunya, hasil ini memiliki kesamaan struktur dengan RGB *image* sehingga dapat dijadika *input* bagi *deep* neural *network*. Gambar di bawah merupakan hasil pemrosesan dengan *pseudocolor encoding*.



Gambar 4. *Depth encoding* dengan menggunakan *pseudocolor encoding*

B. Caffe Deep Learning Framework

Caffe merupakan salah satu deep learning framework yang dikembangkan dengan C++ *library* bersama Python dan MATLAB untuk melatih dan menjalankan *Convolutional Neural Network* (CNN) dan model yang lainnya pada arsitektur komoditasnya. Caffe didukung dengan komputasi CUDA GPU, dapat melakukan proses untuk lebih 40.000.000 gambar dalam sehari pada satu K40 atau Titan GPU (dengan rata-rata waktu komputas 2.5 ms per gambar). Caffe dikembangkan oleh Barkeley Vision and Learning Center (BVLC) dengan bantuan komunitas di GitHub. Caffe mendukung riset projek, aplikasi industri, protipe untuk *vision*, *speech*, dan multimedia.

Caffe menyediakan toolkit lengkap untuk pelatihan, percobaan, penyetelan parameter, dan menjalankan model dengan sumber dokumentasi dan contoh yang baik.

Penyimpanan Data

Caffe menyimpan dan mengomunikasikan data dalam 4-D *array* yang disebut dengan blobs. Blobs menyediakan satu kesatuan antar muka memori, mengambil data gambar (atau tipe data lain), parameter, atau mengupdate parameter [7]. Blobs menyatukan komptasi dan menggabungkan operasi CPU/GPU dengan mensinkronisasi dari host CPU ke perangkat GPU. Secara praktiknya, data diambil dari disk kemudian diinisialisasi ke blob pada CPU, memanggil CUDA kernel untuk melakukan komputasi GPU, melanjutkan pada layer selanjutnya, mengabaikan rincian *low-level* ketika melakukan proses *high-level.* Memori pada *host* CPU dan perangkat GPU dialokasikan berdasarkan permintaan untuk mengefisiensi penggunaan memori. Model disimpan kedalam *disk* dengan beberapa fitur penting antara lain ukuran *binary string* yang minimal, serialisasi yang efisien, format teks yang dapat dibaca manusia dengan versi *binary*, memuliki antar muka pengimplementasian yang efisien untuk beberapa bahasa, pada umumya C++ dan Python.

Konvesional dimesi dari blob untuk sejumlah data gambar adalah jumlah N x channe K x tinggi H x lebar W. Memori blob adalah bari utama pada susunan, sehingga dimensi terakhir berubah paling cepat. Pada kondisi 4D blob, nilai dari indeks (n,k,h,w) ditempatkan pada indeks ((n*K+k)*H+h)*W+w [7].

- Jumlah atau N merupakan ukuran tumpukan data. Prose penumpukan data dicapai dengan lebih baik melalui komunikasi dan pengolahaan *device*.
- Channel atau K merupakan dimensi fitur dalam hal ini untuk gambar RGB K=3.

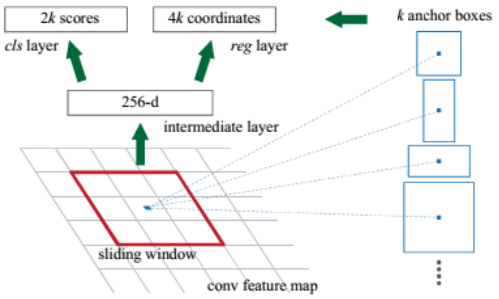
Pada kondisi citra *grayscale* dilakukan pengoalahan data terlebih dahulu. Karena pada Caffe terdapat 3 channel masukkan untuk RGB maka *grayscale* yang hanya 1 channel dilakukan duplikasi dari citra *grayscale* untuk setiap channel.

C. Layer

Layer Caffe merupakan representasi dari layer *neural network* (NN). Ini bisa menggunakan satu atau leboh blobs pada masukan dan menghasilkan satu atau lebih blobs pada keluaran. Laye memiliki dua fungsi utama dalam pengoperasian dari network secara keseluruhan yaitu forward pass untuk mengambil data masukan dan menghasilkan data keluaran, dan backwad pass untuk menghasilkan *gradient* yang berpengaruh pada keluaran dan memperhitungkan *gradient* yang berpengaruh pada parameter dan masukan. Caffe menyediakan tipe-tipe layer yang dapat digunakan untuk mendefinisikan *neural network* (NN) seperti convolution, pooling, inner products, nonlinearities seperti rectified linear dan logistic, local respone normalization, element wise operations, losses seperti softmax dan hinge.

D. Region Proposal Network (RPN)

RPN merupakan fully convolutional network (FCN) yang mengambil masukan gambar untuk sembarang ukuran dan keluaran berupa kotak dari *object proposal* yang memiliki nilai skor objektivitas [5].



Gambar 5. Region Proposal Network

Untuk menghasilkan region proposal, dilakukan *sliding network* terhadap *convolutional feaure map* hasil keluaran dari *convolutional layer* dari citra yang digunakan secara bersama untuk layer RPN dan Fast RCNN. [5] *Network* ini mengambil *input* sebuah *window* dengan ukuran $n \times n$ dari *convolutional feature map*. Setiap *window* dipetakan ke *feature* dengan ukuran lebih kecil. *Feature* tersebut dimasukkan ke dua buah

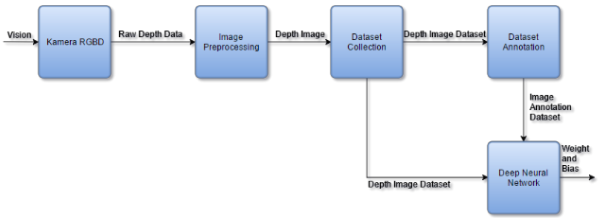
fully connected layer, yaitu *box regression layer* dan *box classification layer*. Pada kasus ini digunakan n= 3, dikarenakan *mini-network* ini beroperasi dengan mode *sliding windows,* sehingga *fully-connected layer* bekerja secara spasial pada setiap lokasi pada citra.

Pada setiap lokasi dari *sliding windows,* dilakukan pendeteksian beberapa *region proposal* yang mana jumlah maksimal kemungkinan proposal untuk setiap lokasi adalah k. Sehingga, *reg layer* memiliki 4k keluaran yang berarti koordinat dari k kotak. Dan *cls layer* memiliki keluaran 2k yang mengestimasi kemungkinan ada atau tidaknya objek pada masing-masing proposal. k proposal merupakan nilai relatif terhadap k kotak atau disebut sebagai *anchors*.

III. PERANCANGAN SISTEM

A. *Perancangan Sistem Secara Umum*

Aliran data dan proses dari sistem *semantic vision* ini dapat digambarkan oleh dua diagram di bawah ini. Terdapat dua proses utama dalam membentuk sistem ini. Proses pertama adalah melakukan *deep neural network training* untuk mendapatkan *weight* dan *bias* dari setiap *layer*. Kemudian, nilai *weight* dan *bias* ini digunakan untuk melakukan *inferencing* saat sistem digunakan.



Gambar 6. Data Flow dan Skema Proses Tahap Training

Penjelasan secara singkat mengenai diagram di atas adalah sebagai berikut:

1. Citra dari lingkungan diambil oleh kamera RGBD dan menghasilkan *raw depth data.*
2. Algoritma *image preprocessing* digunakan untuk mengoleh *raw depth data* menjadi *depth image* agar dapat diolah oleh *deep neural network*.
3. Algoritma *dataset collection* digunakan saat melakukan pengumpulan *depth image* yang berisi manusia dengan empat pose untuk menghasilkan *depth image dataset*.
4. *Depth image dataset* yang tersedia akan melalui tahapan anotasi yang memberikan *bounding box* dan *label* setiap manusia dan posenya. *Bounding box* dan *label* disimpan dalam bentuk teks.
5. *Deep neural network* menerima *depth image*, *bounding box*, dan *label* untuk melakukan *training*. *Weight* dan *bias* setiap *layer* akan diperoleh dari tahapan ini yang akan digunakan pada tahapan i*nferencing*.

B. *Perancangan dan Implementasi Pengumpulan Dataset*



Gambar 7. Diagram pengumpulan *dataset*

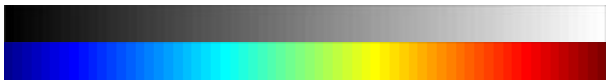
Pada bagian pengumpulan *dataset,* ada empat proses yang dilakukan oleh program pengambilan dataset yakni dimulai dengan *stream kinect data* dalam kasus ini *raw data* yang diakases adalah *depth data*. *Raw depth data* diakses dengan menggunakan *library* libfreenect yang dibangun dengan C *language*. Untuk program pengumpulan dataset, program dibangun dengan menggunakan Python *language* dan *library* OpenCV. Pada libfreenect terdapat *python warapper* yang merupakan program hasil pemanggilan akses data yang dilakukan oleh C *language* pada bagaian dasar *steaming data* dari libfreenect. Python wrapper ini ditulis dalam Cython. Cython merupakan *optimising static compiler* untuk *Python language* dan *extended cython* (berdasarkan Pyrex). Ini membuat penulisan ekstensi C untuk Python semudah penulisan Python.

C. *Perancangan dan Implementasi Image Preprocessing*



Gambar 8. Diagram proses *image preprocessing*

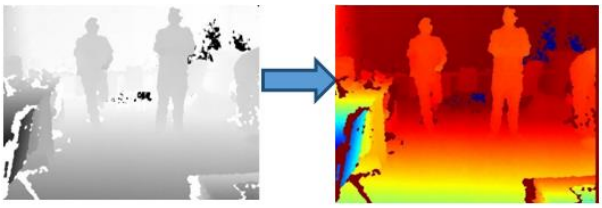
Dalam pengolahan citra, normalisasi adalah suatu proses untuk mengubah rentang dari intensitas pixel dari suatu representasi citra. Penggunaan normalisasi ini dalam pegolahan citra adalah untuk memperbaiki ketidakmerataan kontras pada citra akibat intensitas cahaya yang tinggi pada satu area di citra. Normalisasi biasa disebut sebagai *contrast streching* atau *histogram stretching*. Pada umumnya dalam pengolahan data, normalisasi diartikan juga sebagai *dynamic range expansion.*



Gambar 9. Skala warna untuk *gray-colormap* (atas) dan *jet-colormap* (bawah)

Pada dasarnya, penglihatan manusia dalam menginterpretasikan *grayscale image* sangat terbatas karena jangkauan warna representasi dari *grayscale image* sangat kecil yakni dari putih ke hitam. Begitu juga dalam membuat training data untuk sistem *semantic vision, training data* yang dibuat harus dapat memberikan perbedaan yang signifikan untuk setiap objek yang akan didefinisikan. *Pseudo colouring*

merupakan salah satu metoda yang tepat dalam meningkatkan kualitas dari *grayscale image*.



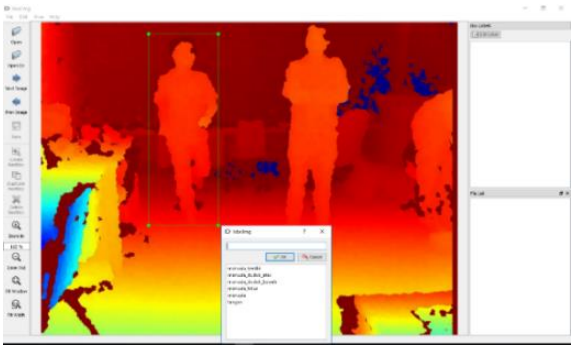
Gambar 10. Hasil *Presudocoloring*

E. Perancangan dan Implementasi Anotasi Dataset



Gambar 11. Diagram proses anotasi dari *dataset*

Pada bagian ini, proses anotasi pada dataset mulai dilakukan. *Colour image dataset* meupakan hasil *grayscale image* yang telah dioptimasi. Colour image dataset ini mempermudah pemberian anotasi pada gambar, karena hasil optimasi memperjelah bentuk dan komposisi warna dari gambar yang akan dianotasi. Dengan mengguanakn *annotation tool* yakni *LabelImg*, pemberian anotasi pada setiap gambar menjadi lebih mudah. *Annotation tool* ini memiliki graphical user interface yang mudah digunakan dikembangkan dengan menggunakan *pyhon languange* dan Qt platform. Pemberian label dari setiap gambar yang terdapat pada *dataset* menggunakan format PASCAL VOC [10]. Berikut hasil pendefinisian *bounding box* dengan menggunakan LabelImg,



Gambar 12. Mendefinisikan daerah objek dengan representasi *boounding box*

IV. HASIL PENGUJIAN

A. Pengujian *mAP*

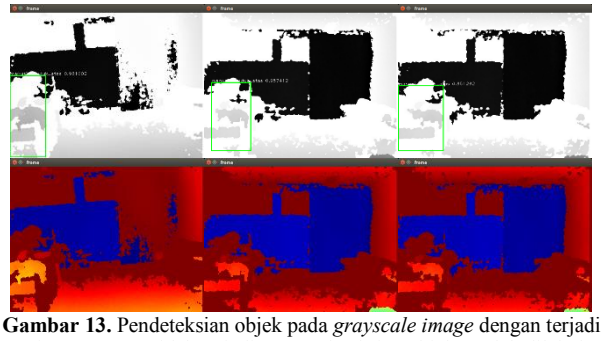
Berikut adalah hasil pengujia dari mAP yang diperoleh dengan dua metoda *depth encoding*,

TABLE I HASIL PENGUJIAN mAP

Depth Encoding	mAP	Overfitting
Grayscale	65.41 %	Iya
Pseudocoloring	56.62 %	Tidak

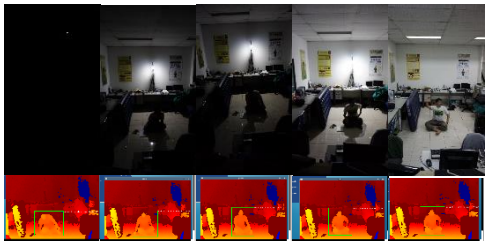
Dari hasil kedua mAP, diperoleh nilai lebih tinggi dibandingkan spesifikasi yang dibutuhkan, yakni mAP > 0,5. *Precision* tinggi menunjukkan FP rendah dan *recall* tinggi menunjukkan FN rendah. Jika semakin tinggi AP, maka semakin rendah FP dan FN.

Berdasarkan hasil pengujian, nilai mAP dari kemampuan pendeteksian objek pada *deep neural network* (DNN) yang telah dilakukan optimalisasi berupa *pseudocoloring* terjadi penurunan yakni dari 65.41% menjadi 56.62%, namun hasil optimalisasi dataset dengan *prseudocoloring* dapat menurunakan kondisi overfitting dari pendeteksian objek. Kondisi *overfitting* merupakan kondisi dimana model terlalu sesuai dengan *training dataset* yang diberikan. Hal ini membuat model susah untuk mengenali contoh objek yang baru diluar dari dataset.



Gambar 13. Pendeteksian objek pada *grayscale image* dengan terjadi *overfitting* (atas). Tidak terjadinya pendeteksian objek setelah dilakukan optimalisasi berupa *pseudocoloring* (bawah)

B. Pengujian Pendeteksian Manusia pada Berbagai Kondisi Pencahayaan

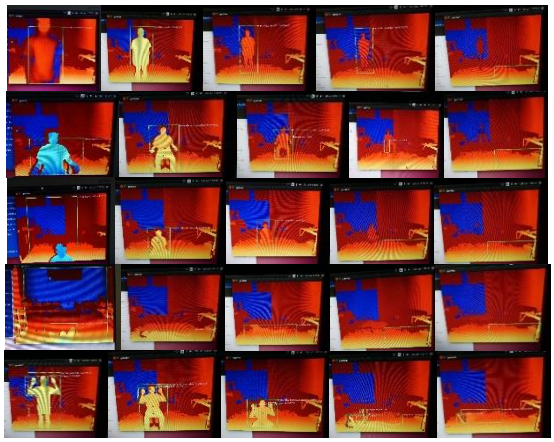


Gambar 14. Hasil pengujian pendeteksian pada pencahayaan 0 lux, 2.2 lux, 3.8 lux, 8.5 lux, dan 22 lux

Pada pengujian diatas, dapa dilihat bahwa sistem ini berhasil melakukan pendeteksian pose manusia duduk bawah dalam kodisi gelap total atau dalam penglihatan manusia biasa tidak ada objek yang terlihat. Kemampuan kamera RGBD yakni kamera Kinect Xbox 360 [9] untuk merepresentasikan data lingkungan dalam bentuk *raw* RGB dan *raw depth data* dapat dimanfaatkan untuk menangkap citra objek dari lingkungan pada kondisi lingkungan yang susah terdefinisiakan dalam hal

ini pencahayaan yang berbeda-beda dari kondisi gelap total sampai terang total yang dalam pengujian ini direpresentasikan dalam kondisi 0.1 lux, 2.2 lux, 3.8 lux, 8.5 lux, dan 22 lux.

C. *Pengujian Pendeteksian Manusia pada Berbagai Jarak*



Gambar 15. Hasil pengujian pendeteksian pada pada kondisi jarak 1, 2, 3, 4, dan 5 meter dari berbagai pose manusia dan kelas tangan

Pada pengujian ini, dilakukan untuk mengukur kemampuan sistem *semantic vision* dalam mendeteksi berbagai pose manusia seperti mansuia berdiri, manusia duduk atas, manusia duduk bawah, dan manusia tidur untuk kondisi jarak 1 meter, 2 meter, 3 meter, 4 meter, dan 5 meter. Pada setiap posisi jarak akan dilakukan perubahan pose untuk tanpa mengangkat tangan dan mengangakat tangan.

Pada kondisi jarak 1 meter, beberapa hasil dari pendeksian terdapat kesalahan seperti seharusnya manusia duduk atas, sistem *semantic vision* mendeteksi manusia berdiri. Hasil pendeteksian ini berbeda dan tidak sesuai dengan yang seharusnya karena pada jarang terdekat bentuk shape dari manusia berdiri, manusia duduk atas, manusia duduk bawah hampis sama dan menutupi sebagian dari pembacaan kamera. Sehingga, hasil pendeteksian akan digeneralisir menjadi hasil *confidence level* tertinggi yakni manusia berdiri yaitu 0.9742. Pada saat melakukan pendeteksian manusia tidur, sistem semantic vision mendeteksi adanya mansuia berdiri pada bagian kondisi kosong tanpa objek. Hal ini karena keterbatasan hasil pembacaan kamera kinect Xbox 360 memiliki keterbatasan jarak deteksi yakni 1.2 meter sampai 3.5 [9]. Sehingga untuk kondisi jarak diluar jangkauan akan didefinisiakan sebgai warna biru pada metoda *image preprocessing* yaitu *pseudocolor*.

Dari batas kemampuan kamera Kinect Xbox 360 untuk menangkap hasil citra dari lingkuangan dengan kondisi jarak 1.2 meter sampai 3.5 meter [9], hasil pengujian diatas memberikan hasil bahwa kondisi jarak 2 meter dan 3 meter dari sistem *sematic vision* hasil pendeteksian sangat optimal dan akurat.

V. KESIMPULAN

Secara umum pembuatan dataset dengan optimalisasi *pseudocoloring* dapat meningkatkan performa dari pendeteksian dengan meminimalisir terjadinya *overfitting* pada pendeteksian. Hasil yang diperoleh antara lain semantic vision memiliki mAP 56,62 % untuk 1039 citra pengujian. *Semantic vision* mampu melakukan pendeteksian manusia untuk kondisi pencahayan rendah yaaitu kecil dari 22 lux. Sistem mampu melakukan pendeteksian manusia dengan jarak optimal 2-3 meter.

REFERENSI

[1] Skavdal, Terje. *Introduction to Disaster Assessment and Assessment Methodologies.* Kobe (2003).ITU-R, "*Technical Characteristics for an Automatic Identification System Using Time-Division Multiple Access in the VHF Maritime Mobile Band*," ITU-R, Geneva, 2010.

[2] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal Deep Learning for Robust RGB-D Object Recognition". July 2015.

[3] Saurabh Gupta, Ross Girshick, Pablo Arbelaez, Jitendra Malik. "Learning Rich Features from RGB-D Images for Object Detection and Segmentation". July 2014.

[4] Harich, N., "Fully Convolutional Network for Semantic Segmentation from RGB-D Images". Maret 2016.

[5] Ren, S., He, K., Girshick, R., and Sun, J., "Faster R-CNN: Towards Real-Time Object Detection with Regional Proposal Networks". Cornell University. 2016.

[6] G, Ross., "Fast R-CNN". Microsoft Research. 2015

[7] Jia, Y., "Caffe: An open source convolutional architecture for fast feature embedding". http://caffe.berkeleyvision.org/ (2013)

[8] Evaluation of Results using Mean Average Precision. Tersedia: https://deshanadesai.github.io/notes/Evaluation-of-Results-using-Mean-AvgPrecision diakses pada 27 April 2017 23.00 WIB.

[9] C, Andujar., "Kinect", Conputer Science Department, Universitat Politecnica De Catalunya, 2017

[10] Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J. and Zisserman, A. "The PASCAL Visual Object Classes Challenge: A Retrospective". International Journal of Computer Vision, 111(1), 98-136, 2015

Strategi Pergerakan Penjaga Gawang pada Tim Rimau FC

Benedict Owen Napitupulu* dan Bhakti Yudho Suprapto.
Departemen of Electrical Engineering
Sriwijaya University
Palembang, Indonesia
Benedictowen09@gmail.com

***Abstrak*—Kontes Robot Sepak Bola Indonesia Beroda (KRSBI Beroda) adalah kompetisi robot beroda yang setiap robot dapat bermain bola layaknya permainan sepakbola pada umumnya. Setiap tim harus memiliki pemain berposisi sebagai penjaga gawang. Penjaga gawang bergerak sesuai nilai *setpoint* bola yang didapat dari kamera *android.* Nilai *setpoin* yang didapat adalah 620. Sensor PING digunakan untuk membatasi gerak robot agar tidak menabrak tiang gawang. Hasil yang didapat adalah nilai *setpoint* sensor PING adalah 5 cm. Dengan kecepatan 80.**

Kata kunci— Strategi; penjaga gawang; PING.

I. PENDAHULUAN

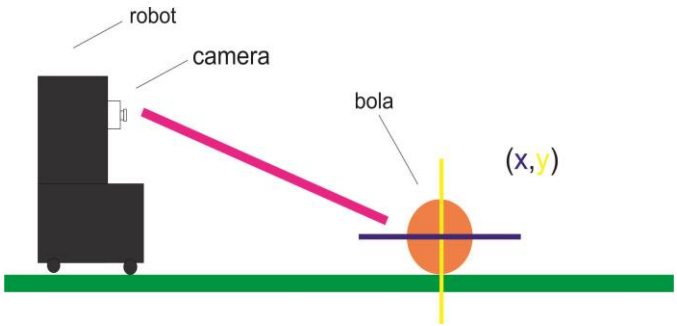
Kontes robot sepakbola Indonesia beroda (KRSBI beroda) adalah kompetisi robot beroda yang setiap robot dituntut bergerak otomatis bermain sepakbola layaknya pertandingan sepakbola pada umumnya. Kamera digunakan untuk mendeteksi posisi bola. Bola yang digunakan adalah bola futsal standar FIFA berwarna *orange*. Lapangan menggunakan karpet berwarna hijau dan cat putih sebagai garis lapangan.[2]

Seperti permainan sepakbola pada umumnya, setiap tim harus memiliki pemain pada posisi penjaga gawang. Dengan pengolahan citra pada kamera penjaga gawang membaca keberadaan bola yang ditendang robot lawan. Robot penjaga gawang akan bergerak ke arah kanan atau kiri gawang sesuai arah bola. Akantetapi robot penjaga gawang dapat menabrak tiang gawang ketika robot melakukan *manuver* pergerakan menghadang bola yang melebar atau tidak ke arah gawang.

Berdasarkan alasan diatas, Tim Rimau FC menggembangkan strategi pergerakan penjaga gawang dengan memanfaatkan sensor PING.

II. SISTEM PENGLIHATAN PENJAGA GAWANG

Penjaga gawang Rimau FC mempunyai sistem penglihatan yang terpasang dibagian atas robot yaitu kamera. Kamera yang digunakan adalah kamera *android*. Kamera berfungsi untuk mendeteksi keberadaan bola dilapangan.

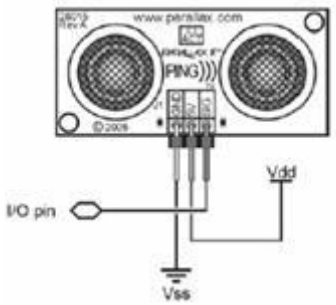


Gambar 1. Sistem deteksi bola penjaga gawang

Robot tersebut menerima informasi yang dikirimkan oleh *android* menggunakan media *bluetooth*. Proses akan berulang setiap saat, sejak *android* menerima informasi dari kamera sampai dengan robot-robot tersebut mengambil tindakan atas perintah yang diberikan. Dari gambar 1, dapat dijelaskan robot mendeteksi bola dan mendapatkan nilai (*x, y*) bola, kemudian robot akan mengolah data tersebut dan melakukan pergerakan sesuai dengan algoritma pergerakan robot.

III. PEMBATAS PERGERAKAN PENJAGA GAWANG

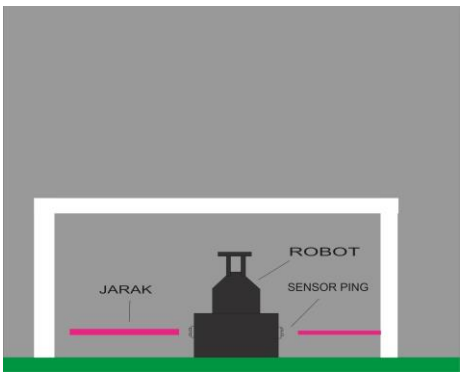
Robot penjaga gawang Rimau FC menggunakan sensor PING sebagai pembatas pergerakan robot agar tidak menabrak tiang gawang ketika melakukan *manuver* pergerakan menghadang bola ke kanan dan kiri.



Gambar 2. Sensor PING

Sensor PING merupakan sensor ultrasonik yang dapat mendeteksi jarak obyek dengan cara memancarkan

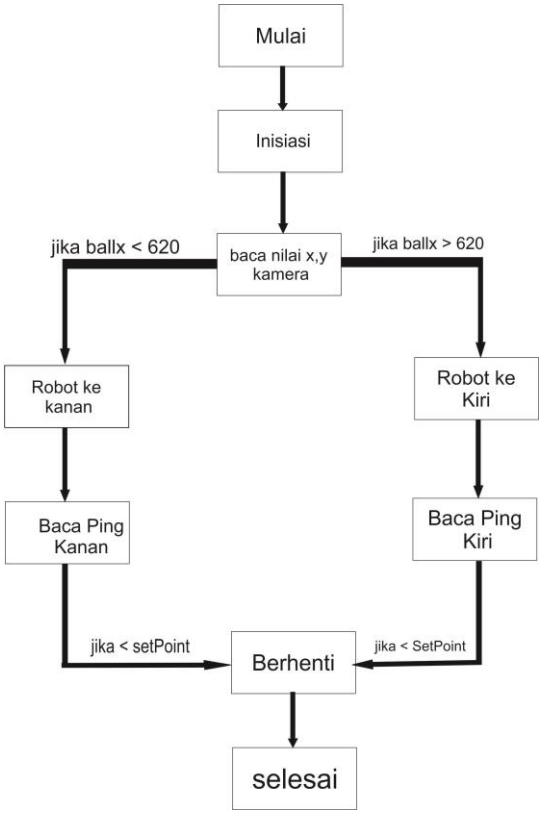
gelombang ultrasonik dengan frekuensi 40 KHz dan kemudian mendeteksi pantulannya. Sensor ini dapat mengukur jarak antara 3 cm sampai 300 cm. keluaran dari sensor ini berupa pulsa yang lebarnya merepresentasikan jarak. Lebar pulsanya bervariasi dari 115 uS sampai 18,5 mS. Pada dasanya, PING terdiri dari sebuah chip pembangkit sinyal 40KHz, sebuah speaker ultrasonik dan sebuah mikropon ultrasonik. Speaker ultrasonik mengubah sinyal 40 KHz menjadi suara sementara mikrofon ultrasonik berfungsi untuk mendeteksi pantulan suaranya.[1]



Gambar 3.Sensor PING pada robot.

IV. *FLOW CHART*

Berikut ini adalah *flowchart* penjaga gawang tim Rimau FC:



Gambar 4. *Flowchart* penjaga gawang

Berdasarkan *flow chart* pada gambar 4. Robot penjaga gawang dari tim Rimau FC membaca nilai (*x, y*) yang diterima dari kamera *android.* Jika robot menerima nilai kurang dari 620 robot bergerak ke kiri. Jika robot menerima nilai nilai lebih dari 620 robot bergerak ke kanan. Sensor PING sebagai pengukur jarak ke tiang gawang. Saat robot melakukan manuver pergerakan ke kanan sensor PING kanan membaca jarak robot dan tiang kanan gawang. Jika nilai sensor PING kanan kurang dari *setpoint,* maka robot akan berhenti. Jika robot menerima nilai kurang dari 620 robot bergerak ke kiri. Saat robot melakukan manuver pergerakan ke kiri sensor PING kiri membaca jarak robot dan tiang kiri gawang. Jika nilai sensor PING kiri kurang dari *setpoint,* maka robot akan berhenti.

V. HASIL DAN PEMBAHASAN

Strategi pergerakan robot penjaga gawang diatas dilakukan percobaan secara *real* pada robot penjaga gawang tim Rimau FC. Robot diberikan nilai (*x,y*) 630 dan 610. Kemudian motor diberi kecepatan 50, 80, 100 dan 150. Nilai *setpoint=5cm dan 3cm.*

Dari Tabel 1 dan Tabel 2 dapat dilihat saat jarak 5cm. Speed maksimal 80. Jika nilai speed lebih dari 80, maka robot akan menabrak tiang gawang.

TABLE I PERCOBAAN I

No	Speed	Jarak	x	Arah	Hasil
1	50	5 cm	630	Kiri	Baik
2	80	5 cm	630	Kiri	Baik
3	100	5 cm	630	Kiri	Kurang Baik
4	130	5 cm	630	Kiri	Kurang Baik

TABLE II PERCOBAAN II

No	Speed	Jarak	x	Arah	Hasil
1	50	5 cm	600	Kanan	Baik
2	80	5 cm	600	Kanan	Baik
3	100	5 cm	600	Kanan	Kurang Baik
4	130	5 cm	600	Kanan	Kurang Baik

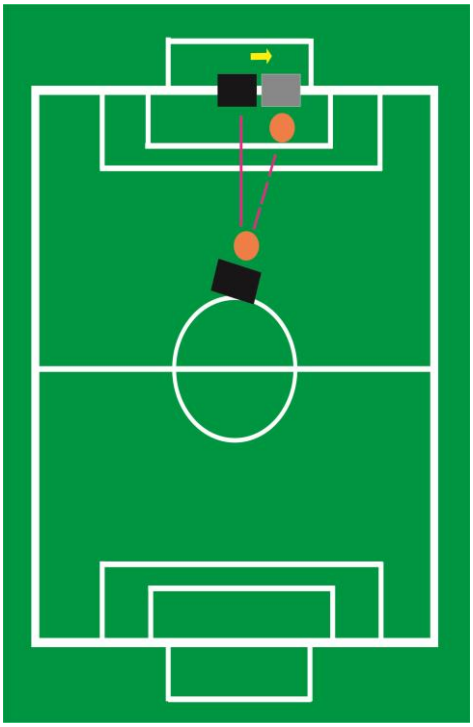
Dari Tabel 3 dan Tabel 4 dapat dilihat saat jarak 3cm. Speed maksimal 100. Jika nilai speed lebih dari 100, maka robot akan menabrak tiang gawang.

TABLE III PERCOBAAN III

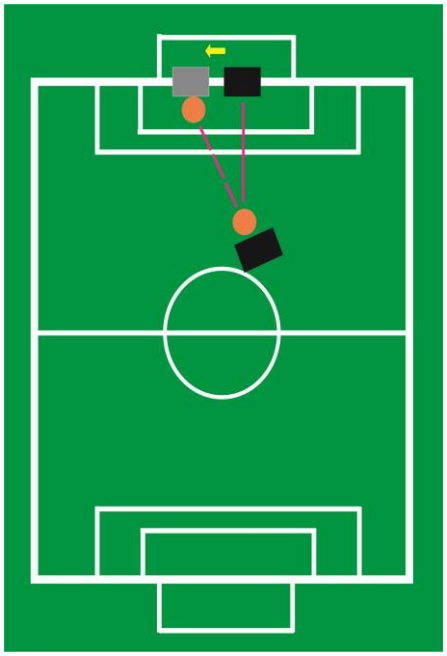
No	Speed	Jarak	x	Arah	Hasil
1	50	3 cm	630	Kiri	Baik
2	80	3 cm	630	Kiri	Baik
3	100	3 cm	630	Kiri	Baik
4	130	3 cm	630	Kiri	Kurang Baik

TABEL IV PERCOBAAN IV

No	Speed	Jarak	x	Arah	Hasil
1	50	3 cm	600	Kanan	Baik
2	80	3 cm	600	Kanan	Baik
3	100	3 cm	600	Kanan	Baik
4	130	3 cm	600	Kanan	Kurang Baik



Gambar 5. Kondisi ini ketika bola bergerak kearah kanan gawang.



Gambar 6. Kondisi ini ketika bola bergerak kearah kanan gawang.

Robot bergerak kekiri ketika robot mencapai batas minimum jarak sensor PING maka robot akan berhenti.

VI. KESIMPULAN

Berdasarkan hasil pengujian , strategi penjaga gawang pada tim rimau FC mampu menghadang bola yang datang dari posisi kiri maupun kanan sehingg bola tidak memasuki gawang *Setpoint* sensor PING terbaik adalah 5 cm dan kecepatan motor terbaik 80.

REFERENSI

[1] PING Sensor Product Guide v.2.0.2013.Parallax Inc.
[2] Panduan KRSBI Berodav.221216. 2017. Ristekdikti

Artificial Neural Network Untuk Pengukuran Posisi Bola Menggunakan Kamera Omnidireksional Pada Robot Sepak Bola Beroda

Yohan Prakoso, Adnan Rachmawan, Muhammad Abizhar Multazam, Alam Ar Raad Stone, Aulia Aditya Rachman, Rudy Dikairono*, Muhammad Azhar Ismail, Helga Dendy Ardianika, dan Kamal Arief
Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111
rudydikairono@ee.its.ac.id

***Abstrak*—Pada penelitian kali ini kamera 360 derajat /Omnidireksional digunakan sebagai sensor *imaging* pada robot sepak bola beroda. Citra kamera Omnidireksional diciptakan dengan menggunakan webcam dengan tambahan lensa *fisheye* yang dipasang dibagian atas robot. Distorsi yang dihasilkan oleh lensa kamera menyebabkan pengukuran jarak terukur dengan satuan pixel dari citra tidak linear terhadap jarak sesungguhnya yang terukur terhadap robot. Metode Artificial neural network digunakan untuk mendapatkan karakteristik dari jarak yang diperoleh dengan satuan pixel pada citra terhadap jarak sesungguhnya di lapangan. Setalah didapatkan karakteristik dari kamera, hal itu digunakan sebagai dasar pengukuran posisi bola terhadap robot. Dari hasil pengujian didapatkan bahwa sistem kamera 360 derajat mampu mendeteksi bola pada jarak 40-400 cm dengan error rata-rata sebesar 2.52 %.**

Kata kunci*—Omnidireksional; artificial neural network; pengukuran posisi.

I. PENDAHULUAN

Robot adalah alat yang dapat membantu banyak kebutuhan manusia secara fisik baik dengan kotrol oleh manusia ataupun secara otomatis dengan program yang telah ditanamkan terlebih dahulu pada robot. Untuk memacu pengembangan pengetahuan tentang robot, ada banyak sekali kontes yang diadakan setiap tahunnya dengan tema-tema yang beraneka ragam. Salah satu kontes yang tersebut adalah Kontes Robot Sepak Bola Indonesia kategori beroda.

Salah satu tantangan dari robot ini adalah mentukan posisi bola dan objek lain terhadap robot. Kamera adalah alat yang sering digunakan untuk mendapatkan citra dari lingkuangan dan mengidentifikasi posisi dari bola. Penggunaan kamera tunggal membuat sudut pandang dari robot terhadap lingkungan sekitar terbatas sehingga diperlukan bantuan gerakan mekanik untuk membantu robot mendapatkan pandangan terhadap lingkungan yang lebih luas dan mendapatkan objek yang diinginkan baik berupa bola, gawang, kawan dan lawan untuk lebih lanjut diproses menjadi gerakan robot yang harus dilakukan.

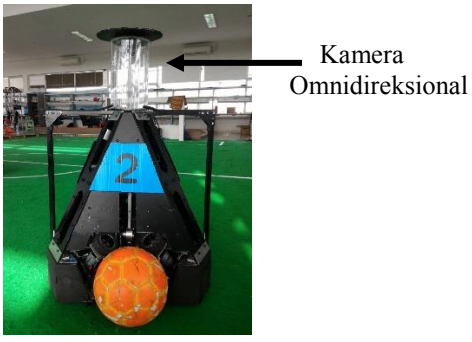
Dengan menggunakan kamera dengan sudut pandang 360 derajat tentunya akan membuat sudut pandang robot terhadap lingkungan semakin luas sehingga tidak diperlukan lagi terlalu banyak gerakan mekanik untuk mencari bola sehingga robot dapat lebih cepat untuk menentukan keputusan selanjutnya.

Kamera akan menghasilkan citra lingkungan yang jika diproses lebih lanjut akan mendapatkan posisi bola pada citra. Posisi bola pada citra dapat di representasikan dengan x, y dan jari-jari dari bola dengan satuan *pixel*. Namun posisi x, y, dan jari-jari bola pada citra tidak linear terhadap posisi bola sesungguhnya terhadap robot sehingga diperlukan proses pengolahan lebih lanjut untuk mendapatkan posisi bola terhadap robot. Beberapa metode yang dapat digunakan adalah *Artificial Neural Network* (ANN)[1]–[3] atau regresi polinomial[4].

ANN adalah sebuah model matematik yang berupa kumpulan unit yang terhubung secara paralel yang bentuknya menyerupai jaringan saraf pada otak manusia. Dengan menggunakan metode ini kita dapat menggunakan beberapa *input* data dan *output* yang seharusnya untuk dijadikan sebagai data pembelajaran. Dari data-data yang telah diberikan, sistem ini dapat menentukan secara mandiri *output* untuk *input-input* lain. Dengan demikian, menggunakan metode ini dapat digunakan untuk mengatasi permasalahan ketidak linearan antara posisi objek pada citra dengan posisi objek pada lapangan sebenarnya terhadap robot.

II. TEORI PENUNJANG

A. Kamera Omnidireksional



Gambar 1. Kamera Omnidireksional

Dalam fotografi omnidireksional kamera adalah sebuah kamera yang memiliki sudaut pandang 360 derajat pada area horizontal atau sudut pandang yang melingkupi hampir seluruh bola[3]. Kamera omnidireksional digunakan pada fungsi yang memerlukan sudut pandang luas seperti pada fotografi panorama dan beberapa keperluan robotika.

Dalam robotika, kamera omnidireksional sering digunakan untuk odometri secara *visual* dan untuk menyelesaikan *simultaneous localization and mapping* (SLAM). Karena kemampuan kamera omnidireksional untuk mengambil gambar dengan sudut pandang 360 derajat, hasil yang lebih baik akan didapatkan untuk *optical flow* dan *feature selection and matching.*

B. OpenCV

OpenCV adalah sebuah *library open source* untuk vision komputer yang disediakan pada link *http://SourceForge.net/projects/opencvlibrary*. Library ini ditulis dalam bahasa C dan C++ dan dapat dijalankan pada sistem operasi Linux,Windows dan Mac OS X. Selain itu terdapat pengembangan pada Python, Ruby, Matlab, dan bahasa pemrograman lain [1]. OpenCV adalah singkatan dari *Open Source Computer Vision*. OpenCV merupakan sebuah *software library* bebas (*open source*) yang digunakan untuk operasi *computer vision* dan *machine learning*. OpenCV telah dibangun untuk menyediakan sebuah infrastruktur umum untuk beberapa aplikasi *computer vision* dan untuk mempercepat penggunaan dari mesin persepsi dalam produk komersial. OpenCV mempermudah bisnis-bisnis untuk memanfaatkan dan memodifikasi kode. *Library* OpenCV mempunyai lebih dari 2500 algoritma yang telah dioptimalkan dimana meliputi sebuah himpunan menyeluruh dari keduanya yaitu klasik dan seni beberapa algoritma *computer vision* dan *machine learning.* Algoritma-algoritma tersebut dapat digunakan untuk mendeteksi dan mengenali wajah, mengidentifikasi objek, mengklasifikasi tindakan manusia dalam video, mengikuti jejak perpindahan objek, mengekstrak model-model 3D objek, menghasilkan titik awan 3D dari kamera stereo, dan lain sebagainya.

OpenCV dapat diterapkan pada pemrograman C++, C. Python, Java dan MATLAB. OpenCV mendukung untuk sistem operasi Windows, Linux, Android dan Mac OS.

C. Artificial neural network

Artificial neural network adalah sebuah bentuk proses yang menyerupai kerja dari otak manusia yang memiliki neuron dan interkoneksi. Sistem ini berperan layaknya otak manusia yaitu memiliki persepsi, mememiliki kemampuan belajar dan pengenalan terhadap suatu bentuk dan pola.

Elemen dasar dari sebuah sistem artificial neural network adalah neuron layaknya elemn dasar pada saraf manusia. Neuron merupakan model dari sebuah neuron natural. Setiap neuron memiliki x_1, x_2, …, x_m yang masuk pada neuron. Pada neoron manusia masukan-masukan ini merupakan tingkatan stimulus. Pada sistem artificial neural network setiap masukan x_i, memiliki sebuah bobot pengali yaitu w_i. Hasil perkalian dari $x_i w_i$ ini kemudian masuk kedalam neuron [2].

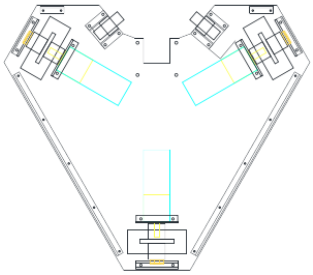
D. Back propagation

Back propagation adalah metode pembelajaran untuk sistem *artificial neural network* yang dirumuskan oleh Werbos dan dipopulerkan oleh McClelland dan Rumelhart. Algoritma ini mempuyai sitem kerja untuk mengkoreksi error yang dihasilkan oleh sebuah sistem *artificial neural network*. Selanjutnya error yang diperoleh tersebut digunakan sebagai pertimbangan bagi algoritma tersebut untuk mengkoreksi bobot dengan tujuan memperkecil error yang dihasilkan.

III. PERANCANGAN SISTEM

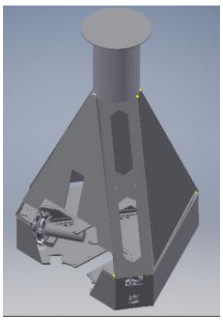
A. Perancangan Mekanik

Mekanik robot didesain menggunakan bahan dasar alumunium padat untuk bodinya dan akrilik untuk beberapa bagian lain semisal penyangga sensor dan alat elektronik lainnya. Desain dari *base body* robot ditunjukkan pada Gambar 2. Robot di desain dengan menggunakan penggerak 3WD. Desain ini ditujukan agar bobot robot menjadi relatif lebih ringan. Mengingat satu sistem penggerak yang terdiri dari Motor DC berencoder dan roda omni memiliki berat yang cukup besar, pemilihan penggunaan sistem 3WD dirasa lebih baik dari pada 4WD. Selain itu penggunaan sistem 3WD dapat memberi ruang yang lebih besar untuk penempatan komponen-komponen lain.



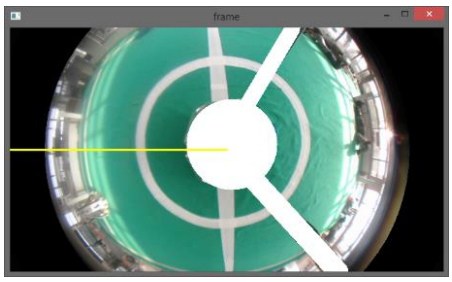
Gambar 2. Desain 2D *body base robot*

Robot sepak bola ini di desain mengerucut yaitu bagian bawah besar dan semakin ke atas semakin kecil dan disatukan oleh lempeng segitiga diatas robot. Dibagian atas diberikan silinder tabung akrilik sebagai pelindung kamera dan bagian paling atas adalah tempat kamera omnidireksional. Tujuan robot disesain mengerucut adalah memberikan kamera omnidireksional sudut pandang yang yang lebih luas serta meminimalisir titik buta terutama di daerah yang dekat dengan robot. Desain 3D dari robot dapat dilihat pada Gambar 3.



Gambar 3. Desain 3D robot

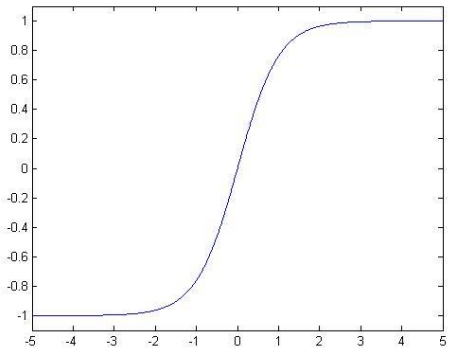
Kamera yang digunakna pada robot adalah kamera webcam dengan tambahan lensa *fisheye*[5] yang diletakkan pada depan kamera. Lensa *fisheye* digunakan untuk menciptakan citra kamera omnidireksional dengan memperluas sudut pandang dari kamera sebesar 235 derajat. Pemasangan sistem kamera omnidireksional ini dapat dilihat pada Gambar 4.



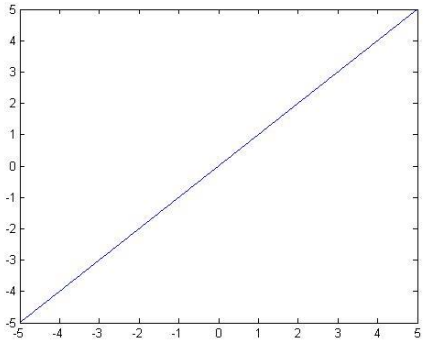
Gambar 4. Citra yang dihasilkan oleh kamera omni direksional

B. Perancangan Software Learning (Preproccess)

Tahap ini adalah tahap *preproccess* yaitu tahap pembelajaran untuk mendapatkan nilai bobot w dan b pada sistem ANN. Proses pembelajaran dilakukan dengan menggunakan metode *backpropagation*. Sistem ANN terdiri dari 2 layer memiliki 1 *input* dan 1 *output*. *Hidden layer* memiliki fungsi transfer log sigmoid dengan 10 *output*. *Output layer* terdiri dari 10 *input* dengan fungsi transfer linear dan 1 *output*. Grafik fungsi transfer log sigmoid dan linear ditunjukkan pada Gambar 5 dan Gambar 6.



Gambar 5. Fungsi Transfer Log Sigmoid



Gambar 6. Fungsi Transfer Linear

Tahap pembelajaran dimulai dengan memberikan nilai inisialisasi pada pembobot w dan b. Inisialisasi dilakukan dengan memberikan nilai acak yang terdistribusi merata antara -1 sampai 1.

Untuk melakukan proses pembelajaran digunakan beberapa sampel dari data percobaan hubungan antara jarak terukur dalam *pixel* dan jarak sesungguhnya bola pada lapangan. Data ini disebut dengan *training set.* Data ini digunakan sebagai contoh bagi sistem ANN untuk dipelajari karakteristiknya.

Sebelum dilakukan proses pembelajaran terlebih dahulu dilakukan normalisasi pada data training set agar fungsi transfer dapat bekerja secara optimal. Setelah proses tersebut dilakukan proses selanjutnya adalah dengan melakukan propagasi maju untuk mendapatkan nilai *output* sistem ANN dengan nilai bobot yang telah ditentukan. Setelah didapatkan *output* dari sistem, dicari nilai kesalahan dari nilai tersebut terhadap nilai pada *training set.* Nilain kesalahan tersebut digunakan untuk sebagai bahan bahan pertimbangan untuk melakukan pembaruan nilai bobot w dan b pada sistem pada proses propagasi balik. Proses tersebut diulangi sampai didapatkan nilai kesalahan relatif kurang dari 0.01%.

C. Perancangan Software Pengukuran (Inproccess)

Pada saat permainan (*inproccess*) terdapat beberapa proses yang dilakukan untuk mendapatkan posisi bola terhadap robot. Proses pertama adalah pengambilan citra warna dari kamera dalam citra RGB. Langkah selanjutnya adalah dengan mengkonversi citra RGB menjadi citra HSV. Hal ini dilakukan karena citra HSV memisahkan antara informasi warna dan informasi kecerahan. Dengan demikian diharapkan proses seleksi warna pada bola dapat dilakukan dengan meminimalisir efek dari pencahayaan di lapangan. Proses berikutnya adalah memfilter warna dari bola dan lapangan berdasarkan rentang ketiga kanal HSV. proses ini menghasilkan citra biner yang terdiri dari citra berwarna putih sebagai objek dan citra berwarna hitam sebagai *background*. Proses pendeteksian bola dilakukan dengan mencari objek lapangan yang berada diatas objek lapangan. Jika objek bola tidak berada diatas objek lapangan sistem menganggap objek tersebut bukanlah bola. Setelah objek bola ditemukan proses selanjutnya adalah mendapatkan informasi dari posisi bola tersebut.

Posisi bola terhadap robot pada citra dapat dinyatakan dalam koordinat polar yang terdiri dari jarak dan sudut. Titik origin

koordinat tersebut terletak pada titik tengah citra yang pada citra merupakan titik tengah robot. Dengan menggunakan sistem kamera omnidireksional koordindinat sudut pada citra diasumsikan sama dengan koordinat sudut pada kondisi sebenarnya. Koordinat objek bola pada citra dapat diperoleh dengan persamaan berikut:

$$r_p = \sqrt{(x_o - x_c)^2 + (y_o - y_c)^2} \quad (1)$$

$$\theta = \tan^{-1} \frac{(y_o - y_c)}{-(x_o - x_c)} \quad (2)$$

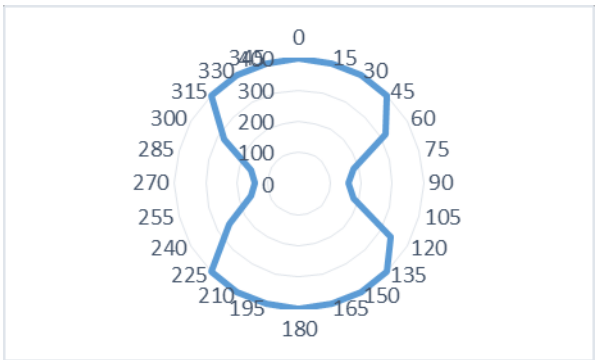
x_c, y_c merupakan titik tengah dari citra yang juga merupakan titik tengah dari robot pada citra. Sedangkan x_0, y_0 adalah posisi bola pada citra terhadap titik origin citra. Setelah diperoleh posisi bola pada citra, informasi tersebut dproses menggunakan ANN dengan menggunakan struktur dan bobot yang di dapat pada saat pembelajaran pada proses sebelumnya untuk mendapatkan jarak sesungguhnya di lapangan.

IV. HASIL PENGUJIAN

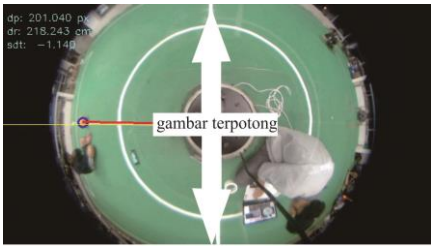
A. Uji Pendeteksian Bola

Pengujian dilakukan dengan meletakkan bola dibeberapa posisi dan mengambil nilai pengukuran jarak dengan satuan *pixel* yang terukur. Posisi bola dinyatakan dalam sudut terhadap arah depan robot dan seberapa jauh bola tersebut dari robot. Hasil pengujian tersebut di interpretasikan pada Gambar 7.

Dari percobaaan yang telah dilakukan dapat dilihat bahwa sistem mampu mendeteksi bola hingga jarak 400 cm. Namun pada sudut tertentu sistem hanya mampu mendeteksi bola hingga jarak 160 cm. Hal ini dikarenakan resolusi dari kamera yang digunakan bertipe lebar dengan perbandingan panjang : lebar adalah 16:9. Hal ini menyebabkan keterbatasan sudut pandang pada bagian atas dan bawah citra. Dapat dilihat pada Gambar 8. Citra terpotong pada bagian atas dan bawah bahwa citra terpotong pada bagian atas dan bawah sehingga menyebabkan keterbatasan pendeteksian bola untuk jarak yang relatif jauh.



Gambar 7 Uji pendeteksian bola



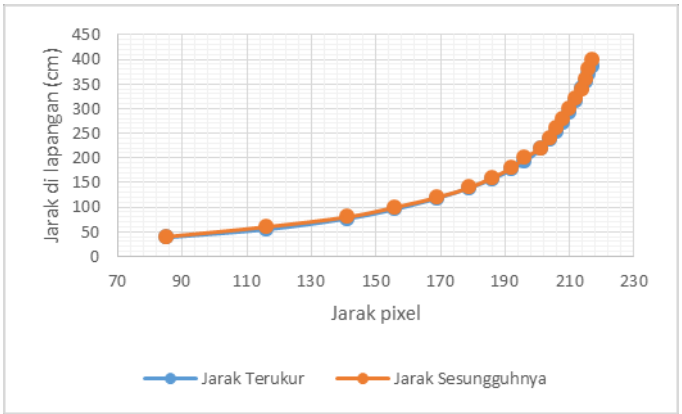
Gambar 8. Citra terpotong pada bagian atas dan bawah

B. Pengujian Hasil Dari Sistem ANN Terhadap Karakteristik Jarak Terukur Dalam Pixel Dan Jarak Sesungguhnya

Pada bagian ini dilakukan pengujian untuk mendapatkan relasi dan karakteristik dari data jarak bola terdeteksi terhadap titik tengah robot dengan satuan *pixel* dibandingkan dengan jarak sesungguhnya. Relasi tersebut dibandingkan dengan hasil pengukuran oleh sistem ANN. Berikut hasil dari pengujian tersebut disajikan pada Tabel dibawah ini:

TABLE I PENGUJIAN HASIL PENGUKURAN TERHADAP KARAKTERISTIK DARI KAMERA OMNIDIREKSIONAL

No	dp (cm)	dm (cm)	dr (cm)	error(%)
1	85.024	38.593	40	3.51
2	116.039	54.943	60	8.43
3	141.032	76.832	80	3.96
4	156.051	96.418	100	3.58
5	169.074	118.517	120	1.24
6	179.07	139.487	140	0.37
7	186.043	157.301	160	1.69
8	192.042	176.401	180	2.00
9	196.064	192.496	200	3.75
10	201.04	218.243	220	0.79
11	204.022	238.053	240	0.81
12	206.039	253.847	260	2.37
13	208.038	271.734	280	2.95
14	210.038	292.09	300	2.64
15	212.038	315.119	320	1.53
16	214.021	340.724	340	0.21
17	215.021	354.683	360	1.48
18	216.037	369.577	380	2.74
19	217.021	384.651	400	3.83
		Error rata-rata		***2.52***



Gambar 9. Grafik pengujian hasil pengukuran terhadap karakteristik dari kamera omnidireksional

Dari tabel dan grafik yang telah disajikan di atas nampak bahwa bahwa karakteristik hubungan antara jarak sesungguhnya dan jarak yang terukur dengan satuan *pixel* kamera menunjukkan sifat eksponensial. Pada nilai *pixel* rendah perubahan jarak dalam satuan *pixe*l yang sedikit mengindikasikan perubahan jarak yang sedikit pula pada jarak sesungguhnya. Namun pada nilai jarak satuan *pixel* yang besar, perubahan kecil mengindikasika perubahan jarak yang signifikan pada jarak yang sesungguhnya. Hal ini menunjukkan bahwa sistem memiliki ketelitian pengukuran yang semakin kecil seiring bertambahnya jarak bola terhadap robot.

Dari pengujian yang telah dilakukan didapatkan bahwa sistem ANN yang dibuat telah mampu mengikuti karakteristik hubungan antara jarak bola terukur dalam *pixel* dan jarak bola sesungguhnya. Dari pengujian tersebut terlihat bahwa pengukuran tidak benar-benar sempurna. Pengukuran menunjukkan nilai error relatif rata-rata sebesar 2.52 %

V. KESIMPULAN DAN SARAN

Sistem kamera 360 derajat mampu mendeteksi bola pada jarak 40 – 400 cm dari robot. Namun di sudut tersentu terhadap depan robot, sistem hanya mampu mendeteksi bola pada jarak maksimal 160 cm dikarenakan citra gambar yang terpotong. Hal ini disebabkan spesifikasi kamera yang digunakan adalah *wide resolution* sehingga bagian atas dan bawah citra sudut pandangnya terbatas.

Karakteristik hubungan antara jarak sesungguhnya dan jarak yang terukur dengan satuan *pixel* kamera menunjukkan sifat eksponensial. Pada nilai *pixel* rendah perubahan jarak dalam satuan *pixel* yang sedikit mengindikasikan perubahan jarak yang sedikit pula pada jarak sesungguhnya. Namun pada nilai jarak satuan *pixel* yang besar, perubahan kecil mengindikasika perubahan jarak yang signifikan pada jarak yang sesungguhnya. Hal ini menunjukkan bahwa sistem memiliki ketelitian pengukuran yang semakin kecil seiring bertambahnya jarak bola terhadap robot.

Sistem ANN mampu mengikuti karakteristik hubungan antara jarak bola yang terukur dengan satuan *pixel* dan jarak sesungguhnya dengan kesalahan relatif pengukuran rata-rata sebesar 2.52 %.

Beberapa saran yang dapat diberikan penulis untuk pengembangan penelitian ini adalah sebagai berikut:

Penggunaan lensa *fish eye* penelitian ini dapat digantikan dengan penggunaan cermin atau lensa *fish eye* lain yang lebih luas sehingga sistem mampu mendeteksi bola dalam lingkup yang lebih luas. Selain itu penggunaan metode *learning back propagation* cenderung membutuhkan waktu relatif lama. Oleh karena itu dapat digunakan metode lain selain *back propagation* yang dapat mempelajari karakteristik hubungan jarak terukur dengan satuan *pixel* kamera terhadap jarak sesungguhnya dengan lebih cepat dan efektif.

REFERENSI

[1] S. Wang, Z. Song, H. Ding, and H. Shi, “An Improved Reinforcement Q-Learning Method with BP Neural Networks in Robot Soccer,” 2011, pp. 177–180.

[2] B. M. Faria, L. P. Reis, N. Lau, and G. Castillo, “Machine Learning algorithms applied to the classification of robotic soccer formations and opponent teams,” 2010, pp. 344–349.

[3] Y.-R. Hou, “Moving target detection in the robot system based on Omni-Vision and Laser Rangefinder,” 2015, pp. 1162–1166.

[4] F. R. Gideon Manalu, “Double target potential field: Planning the movement on passing and kicking the ball in soccer robot,” 2014, pp. 1–5.

[5] B. Tian, Chuen-Leong Ng, and C.-M. Chew, “Self-localization of humanoid robots with fish-eye lens in a soccer field,” 2010, pp. 522–527.

Implementasi Model Inverted Pendulum pada Proses Akselerasi dan Deselerasi Gerakan Berjalan Robot Sepak Bola Humanoid

Anas Mufid Nurrochman*, Muhammad Arifin, Djoko Purwanto, dan Muhtadin.
Departemen Teknik Elektro, Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember (ITS)
Surabaya, Indonesia
anasmfd@gmail.com

***Abstrak*—Penelitian ini dilakukan dalam rangka pengembangan Robot Sepak Bola humanoid. Penelitian ini dilakukan pembuatan robot humanoid dengan derajat kebebasan 20 DoF (*Degree of Freedom*) dan tinggi 88 cm. Pada Pertandingan Robot Sepak bola, kemampuan robot untuk bekerja dengan cepat dan tepat menjadi penentu utama dalam sebuah pertandingan sepak bola. Salah satu hal yang paling penting dalam pertandingan robot sepak bola adalah kecepatan robot untuk mengejar bola. Namun, seperti halnya manusia robot tidak bisa bergerak dari kondisi diam secara langsung dengan kecepatan tinggi ataupun berhenti secara langsung dari kecepatan tinggi sehingga diperlukan akselerasi dan deselerasi. Pada saat melakukan akselerasi atau deselerasi, bagian badan robot akan cenderung mempertahankan posisi awal sehingga keseimbangan robot akan terpengaruh. Untuk itu, dalam penelitian ini dilakukan percobaan model inverted pendulum untuk mengontrol keseimbangan robot saat melakukan pergerakan akselerasi ataupun deselerasi.**

***Kata Kunci— robot humanoid; inverted pendulum; akselerasi; deselerasi*.**

I. PENDAHULUAN

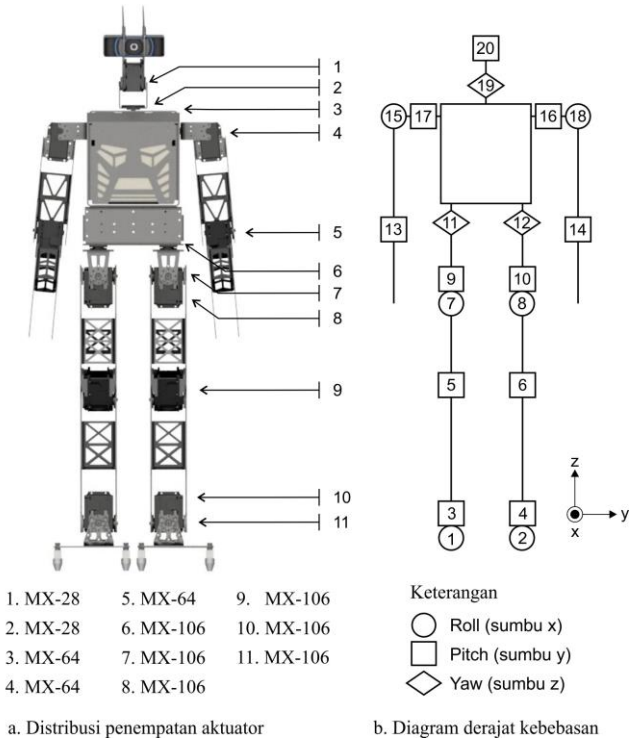
Kemajuan teknologi robotika berkembang dengan pesat. Hal tersebut dipengaruhi oleh kebutuhan bidang-bidang lain akan robotika yang semakin meningkat, seperti bidang medis, industri, sipil, pemerintahan, manajemen bencana, bahkan olahraga. Penelitian-penelitian di bidang robotika yang disesuaikan dengan bidang-bidang tersebut juga semakin banyak. Demikian juga penelitian tentang robot humanoid. Robot humanoid digunakan untuk kepentingan riset pada berbagai bidang ilmu pengetahuan dengan tujuan bermacam-macam, mulai dari robot untuk kesenian, olah raga hingga robot untuk penyelamatan bencana. Peran robot humanoid yang anatominya dibuat mirip dengan anatomi tubuh manusia di bagian tertentu sangatlah penting untuk menyesuaikan tugasnya.

Dalam perkembangannya, sebagai salah satu tolok ukur perkembangan robot humanoid di dunia, setiap tahun dilakukan pertandingan robot humanoid untuk bermain sepak bola. Pada tahun 1997, dilakukan kompetisi robot sepak bola pertama di dunia dengan nama Robocup. *Event* tersebut dilakukan untuk menggenjot perkembangan robot humanoid melalui kompetisi. Dengan adanya kompetisi tersebut diharapkan peneliti-peneliti dari berbagai belahan dunia bersaing untuk mengembangkan robot humanoid untuk kepentingan ilmu pengetahuan dan juga diharapkan bisa diaplikasikan ke bidang ilmu lain.

Dalam rangka mengembangkan Robot Sepak Bola humanoid, dalam penelitian ini dilakukan pembuatan robot humanoid dengan 20 DoF dan tinggi 88 cm. Robot didesain meniru anatomi manusia. Anggota gerak pada robot seperti kaki atau anggota gerak bagian bawah lainnya juga didesain menyerupai anggota tubuh manusia. Namun, karena kompleksnya anggota gerak manusia, dalam pengembangannya Robot Sepak Bola humanoid hanya meniru beberapa bagian anggota gerak yang penting saja untuk digunakan dalam pertandingan Sepak Bola.

Pada Pertandingan Robot Sepak bola, robot juga diharuskan melakukan gerakan–gerakan yang menirukan gerakan manusia seperti berjalan, berlari, menendang, dan berkoordinasi. Kemampuan robot untuk bekerja dengan cepat dan tepat menjadi penentu utama dalam sebuah pertandingan sepak bola. Salah satu hal yang paling penting dalam pertandingan robot sepak bola adalah kecepatan robot untuk mengejar bola. Dengan memprediksi jarak bola, robot diharapkan bergerak secepat mungkin untuk mengejar bola tersebut. Namun seperti halnya manusia, robot tidak bisa bergerak dari kondisi diam secara langsung dengan kecepatan tinggi ataupun berhenti secara langsung dari kecepatan tinggi. Sehingga diperlukan Akselerasi dan Deselerasi.

Pada saat melakukan akselerasi atau deselerasi kecepatan berjalan, seperti yang dikatakan dalam Hukum I Newton bagian badan robot akan cenderung mempertahankan posisi awal sehingga keseimbangan robot akan terpengaruh. Untuk itu, dalam penelitian ini dilakukan percobaan model *inverted* pendulum untuk mengontrol keseimbangan robot saat melakukan pergerakan akselerasi ataupun deselerasi..



a. Distribusi penempatan aktuator b. Diagram derajat kebebasan

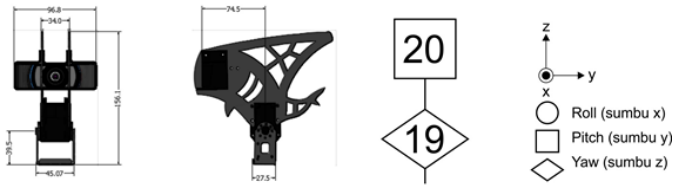
Gambar. 1. Distribusi aktuator dan diagram derajat kebebasan robot Humanoid Ithaaro

II. KONSTRUKSI ROBOT HUMANOID ITHAARO

A. Desain Mekanik

Robot Ithaaro memiliki 20 DoF dapat dilihat pada gambar 3.2, 12 DoF untuk kaki, 6 DoF untuk tangan, dan 2 DoF untuk kepala. Tinggi total robot adalah 88 cm. Rangka robot dibuat menggunakan aluminium lembaran yang dipotong menggunakan mesin CNC laser. Pada bagian kaki dan pinggul menggunakan aluminium dengan tebal 3mm, sedangkan pada bagian badan, tangan, dan kepala menggunakan aluminium dengan tebal 2mm. Semua aluminium yang digunakan menggunakan aluminium tipe 5053. Aktuator yang digunakan pada robot ini adalah motor servo Dynamixel MX-106, MX-64, dan MX-28.

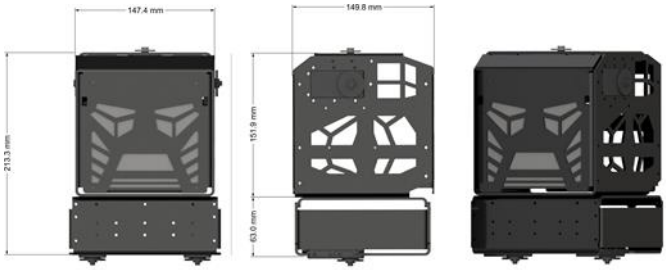
Motor servo MX-106 digunakan pada 12 DoF di kaki, motor servo MX-64 digunakan pada 6 DoF di tangan, sedangkan motor servo MX-28 digunakan pada 2 DoF di kepala.Pemilihan motor servo tersebut digunakan sesuai kebutuhan torsi. Pada kaki digunakan motor servo MX-106 yang mempunyai torsi 106 kg/cm karena merupakan bagian yang menopang bagian tubuh lain. Visualisasi distribusi aktuator pada robot Ithaaro dapat dilihat pada gambar 1.



Gambar. 2. Ukuran dan derajat kebebasan kepala robot Humanoid Ithaaro

Kepala robot mempunyai dua Derajat kebebasan untuk gerakan *pitch* dan yaw. Gerakan ini mirip gerakan manusia saat menggeleng dan mengangguk. Pada kepala terdapat dua aktuator MX 28 dan satu buah sensor berupa kamera. Kepala robot dibuat seringan mungkin agar tidak terlalu membe- bani kerja aktuator. Desain, ukuran, dan derajat kebebasan kepala robot dapat dilihat pada gambar 2.

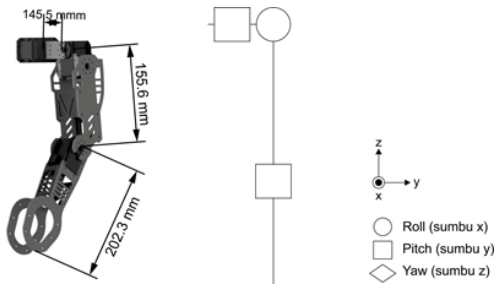
Badan berfungsi sebagai wadah dan pelindung perangkat elektronik dari gangguan luar dan saat robot jatuh. Komponen elektronik yang terdapat pada badan diantaranya; CPU, kontroler, kompas, *power supply* distributor, dan baterai. Selain itu, pada badan terdapat dua buah aktuator *pitch* untuk bahu, satu aktuator yaw untuk kepala, dan dua aktuator yaw untuk kaki. Desain dan ukuran badan dapat dilihat pada gambar 3.



a. tampak depan b. tampak samping

Gambar. 3. Desain dan ukuran badan robot

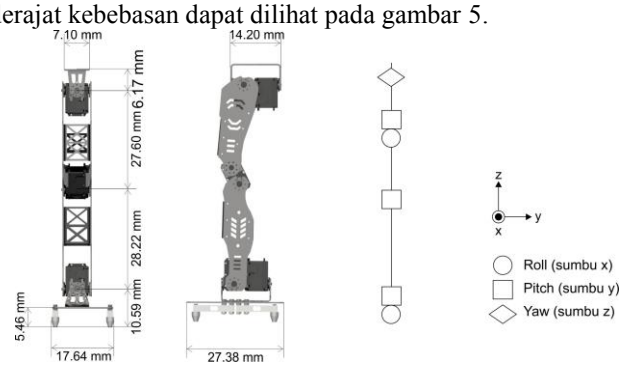
Lengan robot mempunyai tujuan utama untuk gerakan bangun dan membantu keseimbangan berjalan. Oleh karena beban berat mengangkat badan saat gerakan bangun, aktuator yang digunakan pada lengan robot adalah MX 64 yang mempunyai torsi 74kg.cm pada input tegangan maksimal 14.8V. Tangan robot memiliki 3 Dof, dua dof pada bahu untuk gerakan *pitch* dan *roll*, serta satu dof pada siku untuk gerakan pitch. Desain lengan robot dan diagram derajat kebebasan dapat dilihat pada gambar 4.



Gambar. 4. Desain mekanik dan diagram derajat kebebasan lengan robot

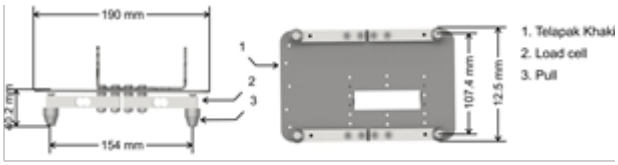
Desain kaki robot bertujuan untuk menopang badan robot pada saat berdiri, meniru gerakan berjalan manusia, dan menendang bola. Aktuator yang digunakan pada kaki robot adalah motor servo MX 106 yang mempunyai torsi 102kg.cm pada tegangan input maksimal 14.8V.

Setiap kaki memiliki enam derajat kebebasan, tiga derajat kebebasan pada pangkal paha untuk gerak- an yaw, *pitch*, dan *roll*, satu derajat kebebasan pada lutut untuk gerakan *pitch*, dan dua derajat kebebasan pada pergelangan kaki untuk gerakan *pitch* dan *roll*. Desain mekanik kaki robot dan diagram derajat kebebasan dapat dilihat pada gambar 5.



Gambar. 5. Desain mekanik dan diagram derajat kebebasan kaki robot

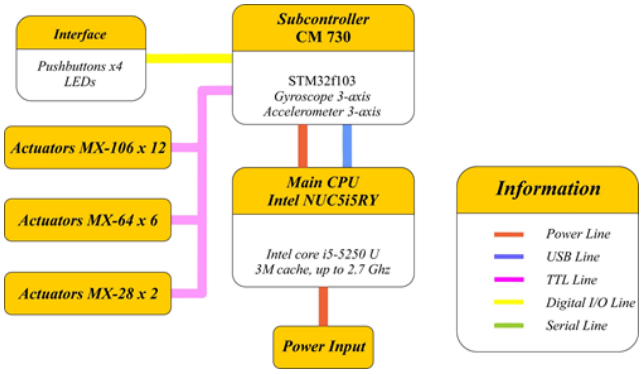
Desain Telapak kaki pada robot Ithaaro dibuat khusus untuk dapat berjal- an di rumput. Telapak kaki robot Ithaaro mempunyai empat *loadcell* yang disusun di bagian kanan dan kiri setiap telapak kaki. Diujung setiap *loadcell* terdapat *pull* sepeti pada sepatu pemain sepak bola. Tujuan ditambahkannya *loadcel*l adalah untuk keseimbangan tekanan robot saat berdiri, sedangkan penambahan pull adalah untuk menabahka *grip* pada saat robot berjalan maupun menendang bola. Desain mekanik telapak kaki robot dapat dilihat pada gambar 6.



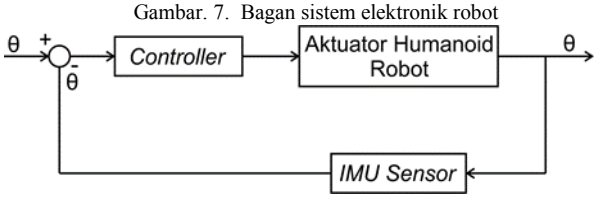
Gambar. 6. Desain mekanik dan telapak kaki robot

B. *Desain Elektronik Robot*

Sensor untuk sistem keseimbangan robot menggunakan accelerometer dan gyrometer yang tertanam dalam kontroler CM730. Data dari accelerometer dan gyrometer dapat diakuisisi langsung oleh cpu melalui komunikasi serial dengan kontroler cm730. Sensor IMU pada robot digunakan un- tuk keseimbangan berjalan, dan juga digunkan sebagai parameter kondisi robot apakah sedang jatuh atau tidak. Bagan sistem elektronik dapat di- lihat pada gambar 7. Diagram blok sistem keseimbangan robot secara keseluruhan dapat dilihat pada gambar 8.



Gambar. 7. Bagan sistem elektronik robot



Gambar. 8. Diagram blok sistem keseimbangan robot secara umum

III. SISTEM KESEIMBANGAN PADA AKELERASI DAN DESELERASI BERJALAN ROBOT

A. *Akelerasi dan Deselerasi Robot*

Akselerasi dan deselerasi pada robot berfungsi ketika robot berjalan menuju tsrget tertentu. Dengan adanya akselerasi, diharapkan robot dapat lebih seimbang saat berjalan mendekati target. penambahan algoritma akselerasi pada robot dilakukan pada proses input kecepatan berjalan. algoritma akselerasi dan deselerasi pada program robot berupa menambahkan proses inkremental dari parameter kecepatan algoritma berjalan robot yang sudah ada pada *framework* darwin. Diagram blok akselerasi dan deselerasi dapat dilihat pada gambar 9.

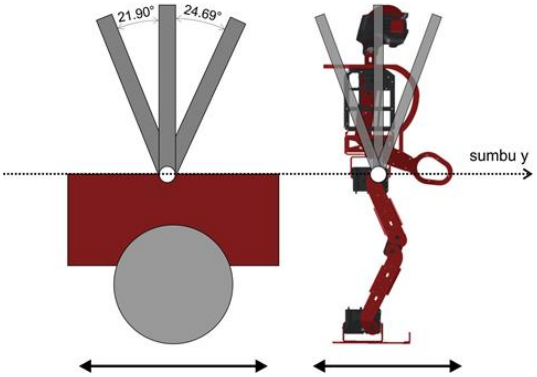


Gambar. 9. Diagram blok akselerasi dan deselerasi robot

B. *Permodelan Inverted Pendulum pada proses Akselerasi dan Deselerasi Robot*

Ketika robot melakukan gerakan dari diam ke berjalan dengan kecepatan tertentu, atau dari berjalan ke diam, atau dari nilai kecepatan tertentu ke nilai kecepatan lain, pasti ada gaya yang menyebabkan robot terhuyung. hal ini sesuai dengan Hukum Newton I dan III dimana suatu benda akan tetap mempertahankan posisinya jika tidak ada gaya yang bekerja, dan jika suatu benda diberi gaya tertentu maka akan timbul gaya reaksi yang arahnya berlawanan dan besarnya sama dengan gaya tersebut. Pada robot humanoid, misalnya ketika robot bergerak dari diam ke berjalan maju. Bagian yang melakukan gerakan adalah kaki, sementara badan dan bagi- an lain tetap diam. Kaki dan badan saling berhubungan sehingga saat kaki bergerak maju badan akan cenderung mempertahankan posisinya sehingga kondisi robot menjadi miring ke belakang, begitu pula sebaliknya pada saat robot berubah dari berjalan menuju berhenti, badan akan condong ke depan.

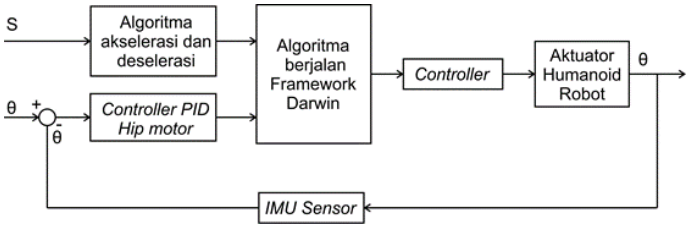
Dengan asumsi kaki robot humanoid adalah roda *chartwheel* dan badan merupakan batangnya pada model *inverted* pendulum *chartwheel*, maka join bagian hip robot humanoid adalah bagian yang dikontrol untuk menyeimbangkan badan. Berikut visualisasi permodelan *inverted* pendulum pada robot dapat dilihat pada gambar 10.



Gambar. 10. Analogi permodelan inverted pendulum.

C. *Sistem Keseimbangan pada Akelerasi dan Deselerasi Robot*

Sistem keseimbangan robot terpisah dari proses akselerasi dan deselerasi, namun proses akselerasi dan deselerasi mempengaruhi pengaturan parameter kontrol pada sistem keseimbangan robot. Keduanya ditambahkan ke dalam *script* algoritma berjalan pada *framework* darwin. Diagram blok Sistem keseimbangan robot dapat dilihat pada gambar 11.

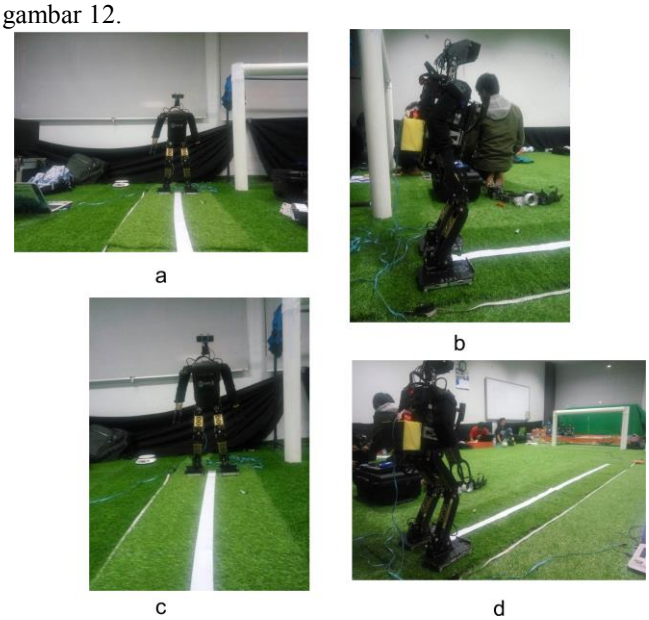


Gambar. 11. Diagram blok sistem keseimbangan robot.

IV. PENGUJIAN KESEIMBANGAN BERJALAN PADA AKSELERASI DAN DESELERASI ROBOT

Pada pengujian ini dilakukan pengujian keseimbangan robot tidak hanya dilakukan pada saat akselerasi dan deselerasi saja, namun juga dilakukan pengujian tanpa akselerasi dan deselerasi tanpa keseimbangan, pengujian akselerasi dan deselerasi tanpa keseimbangan, dan pengujian keseimbang- an tanpa akselerasi dan deselerasi. hal ini bertujuan untuk memperoleh data semua kondisi berjalan robot sehingga bisa dianalisisa apakah proses penambahan keseimbangan akselerasi dan deselerasi dapat berpengaruh pada performa robot.

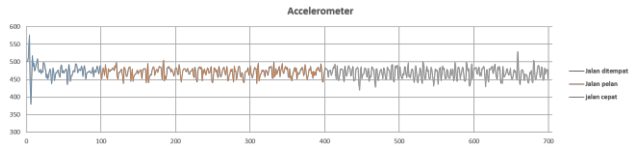
Pengujian dilakukan dengan cara menjalankan robot sepanjang 4 meter dengan menggunakan garis putih sebagai lintasannya. Robot akan berhenti berjalan bila garis sudah hilang dari pandangan kamera robot. sudut kepala dibuat tetap, tidak bergerak mengangguk atau menoleh. Setiap pengujian robot juga diukur kecepatan robot untuk menempuh jarak lintasan yang disediakan. Lintasan pengujian dapat dilihat pada gambar 12.



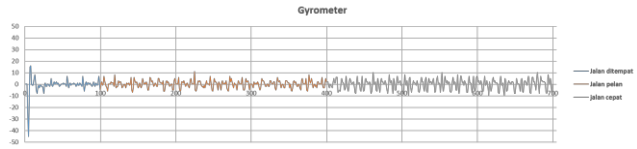
Gambar. 12. Pengujian sistem keseimbangan pada Akelerasi dan Deselerasi Robot.

A. Keseimbangan berjalan Robot tanpa Akselerasi, deselerasi, dan kontrol keseimbangan

Pengujian ini dilakukan untuk mengetahui data sensor pada kondisi awal robot tanpa akselerasi dan keseimbangan tambahan. Dengan adanya data ini maka akan dapat dibandingkan seberapa besar efek optimasi yang dilakukan. grafik hasil pengujian dapat dilihat pada gambar 13. dan gambar 14.



Gambar. 13. Data sensor *accelerometer* Robot berjalan tanpa Akselerasi, deselerasi, dan sistem keseimbangan.

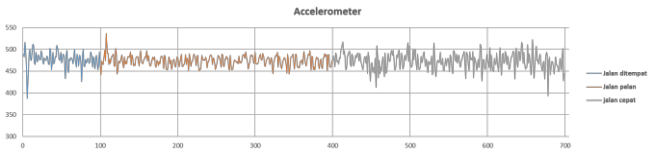


Gambar. 14. Data sensor *gyrometer* Robot berjalan tanpa Akselerasi, deselerasi, dan sistem keseimbangan.

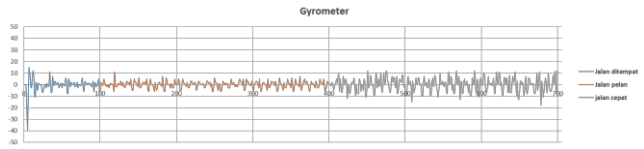
Berdasarkan pada gambar 13. dan gambar 14. dapat diketahui bahwa tanpa akselerasi, saat robot langsung di drive pada kecepatan 22 cm perdetik terdapat *overshoot* pada sensor accelerometer, sedangkan pada proses deselerasi terdapat osilasi sebelum akhirnya berhenti

B. Keseimbangan berjalan Robot dengan Akselerasi dan deselerasi tanpa kontrol keseimbangan

Dengan adanya akselerasi, robot diharapkan dapat melakukan gerakan berjalan lebih cepat pada lintasan yang disediakan.Pengujian ini dilakukan untuk mengetahui data sensor saat robot berjalan dengan akselerasi dan deselerasi tanpa adanya keseimbangan tambahan agar dapat diketahui bagaimana efek akselerasi dan deselerasi yang ditambahkan pada keseimbangan robot. Grafik dapat dilihat pada gambar 15. dan gambar 16.



Gambar. 15. Data sensor *accelerometer* Robot berjalan dengan Akselerasi dan deselerasi tanpa sistem keseimbangan..

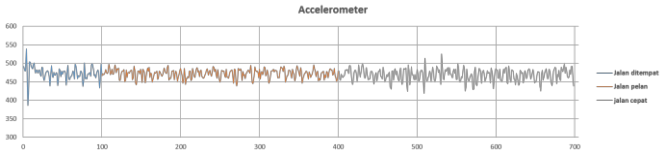


Gambar. 16. Data sensor *gyrometer* Robot berjalan dengan Akselerasi dan dese- lerasi tanpa sistem keseimbangan.

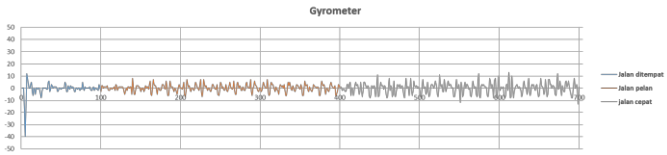
Dari data pada gambar 15. dan gambar 16. dapat dilihat bahwa proses akselerasi mengurangi *overshoot* pada awal sensor accelerometer sehingga robot lebih stabil pada awal robot.

C. Keseimbangan berjalan Robot tanpa Akselerasi dan deselerasi dengan kontrol keseimbangan

Tujuan dari pengujian ini adalah untuk mengetahui efek sistem keseimbangan yanh ditambahkan pada robot jika robot berjalan dengan normal tanpa melakukan akselerasi dan deselerasi. Sistem keseimbangan diapli- kasikan pada *joint* hip robot pada sumbu X (*pitch*). Data grafik sensor dapat dilihat pada gambar 17. dan 18.



Gambar. 17. Data sensor *accelerometer* Robot berjalan tanpa Akselerasi dan dese- lerasi dengan sistem keseimbangan.

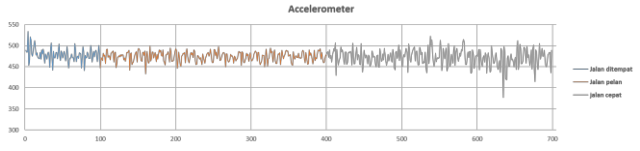


Gambar. 18. Data sensor *gyrometer* Robot berjalan tanpa Akselerasi dan dese- lerasi dengan sistem keseimbangan.

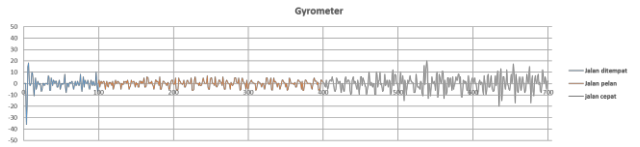
Data pada gambar *17.* dan *18.* menunjukkan bahwa apabila robot berjalan normal tanpa akselerasi dan deselerasi diberi sistem keseimbangan tambahan pada aktuator robot bagian hip dapat memperhalus awal berjalan robot namun saat pergantian kondisi dari berjalan ke berhenti robot menjadi lebih tidak stabil

D. Keseimbangan berjalan Robot dengan Akselerasi, deselerasi, dan kontrol keseimbangan

Pengujjian ini merupakan tujuan utama dari pengujian sistem, dengan mengaplikasikan sistem keseimbangan pada gerakan berjalan robot yang target. penambahan algoritma akselerasi pada robot dilakukan telah ditambahkan. Data yang didapatkan kemudian dibandingkan dengan pengujian lainnya apakah performa robot lebih baik atau lebih buruk. Grafik data sensor dapat dilihat pada gambar 19 dan 20



Gambar. 19. Data sensor *accelerometer* Robot berjalan dengan Akselerasi dan deselerasi dengan sistem keseimbangan.



Gambar. 20. Data sensor *gyrometer* Robot berjalan dengan Akselerasi dan de- selerasi dengan sistem keseimbangan.

Berdasarkan data diatas, pada gambar 19 dan 20 terlihat bahwa sistem keseimbangan yang diaplikasikan pada robot yang berjalan dengan akselerasi dan deselerasi dapat mengurangi *overshoot* ketika awal berjalan dan mengurangi osilasi ketika akan berhenti karena tambahan deselerasi

V. KESIMPULAN/RINGKASAN

Pada penelitian ini telah berhasil dilakukan implementasi Permodelan *inverted* pendulum pada keseimbangan robot saat berjalan dengan proses akselerasi. Implementasi dilakukan pada robot Humanoid Ithaaro. Impelentasi tersebut telah digunakan dalam lomba robot sepak bola tingkat regional. Dari tugas akhir ini, didapatkan kesimpulan sebagai berikut:

1. Desain mekanik robot Ithaaro berhasil diimplementasikan. Berdasarkan hasil pengujian didapatkan bahwa kecepatan robot berjalan maju sebesar 23 cm perdetik, kecepatan berjalan ke samping 7,5 cm perdetik, sedangkan kecepatan berputar robot 30 derajat er langkah
2. Proses Akselerasi dan deselerasi berhasil ditambahkan pada gerakan berjalan robot sehingga jalan robot lebih stabil untuk kecepatan tinggi.
3. Permodelan *inverted* pendulum pada keseimbangan robot saat berjalan dengan proses akselerasi dapat membuat robot lebih stabil saat berjalan..

UCAPAN TERIMA KASIH

Penulis ingin menyampaikan terimakasih kepada: Keluarga penulis, khususnya bapak dan ibu yang senantiasa memberikan dukungan baik dukungan material maupun non material. Bapak Ardyono Priyadi, S.T., M.Eng., Dr.Eng . selaku Ketua Jurusan Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember.Bapak Dr. Ir. Djoko Purwanto, M.Eng. dan Bapak Muhtadin, ST., MT. selaku dosen pembimbing yang selalu memberikan saran serta bantuan dalam penelitian ini. Bapak ibu dosen pengajar Jurusan Teknik Elektro, khususnya bidang studi Elektronika. Dan tidak lupa seluruh teman-teman beserta dosen pembimbing Tim Robotika ITS, khususnya seluruh anggota tim ICHIRO yang telah membantu dalam pengerjaan robot.

REFERENSI

[1] A. H. Dahlan, “Deteksi fitur dan penentuan lokasi robot pemain se- pak bola berbasis penanda yang tidak unik,” in The 2nd Symposium On Robot Soccer Competition, (Yogyakarta), 2014. No citations.

[2] RoboCup Soccer Humanoid League Rules and Setup. (Dikutip pada halaman 7).

[3] Peraturan dan Persiapan Kontes Robot Sepak Bola Indonesia. No citations.

[4] I. Ha, Y. Tamura, and H. Asama, “Development of open platform humanoid robot DARwIn-OP,” Advanced Robotics, vol. 27, pp. 223– 232, Feb. 2013. (Dikutip pada halaman 8).

[5] I. Budiono, “Tim ichiro dalam kontes robot sepak bola indonesia 2014,” in The 2nd Symposium On Robot Soccer Competition, (Yogyakarta), 2014. No citations.

[6] J. Zannatha, R. Cisneros Limon, A. Gomez Sanchez, E. Her- nandez Castillo, L. Medina, and F. Lara Leyva, “Monocular vi- sual self-localization for humanoid soccer robots,” in Electrical Communications and Computers (CONIELECOMP), 2011 21st International Conference on, pp. 100–107, Feb 2011. No citations.

[7] C. Hsien Hsia, J. Shiun Chiang, and S. Hung Chang, “Adaptive Vision-Based Self-Localization System for Humanoid Robot of Ro- boCup,” vol. 9 Number 3, Mar. 2013. No citations.

[8] M. Andung Muntaha, Muhtadin, and M. Hery Purnomo, “Self-localization for Humanoid Soccer Robots Based on Triangulation Method,” in INDONESIAN SYMPOSIUM ON ROBOT SOCCER COMPETITION, (SEMARANG), DIAN NUSWANTORO UNIVERSITY, June. No citations.

[9] S. Thrun, Probabilistic robotics. Intelligent robotics and autonomous agents, Cambridge, Mass: MIT Press, 2005. No citations.

[10] C. Forster, M. Pizzoli, and D. Scaramuzza, “SVO: Fast semi-direct monocular visual odometry,” pp. 15–22, IEEE, May 2014. No citations.

[11] C.-H. Chang, S.-C. Wang, and C.-C. Wang, “Vision-based cooperative simultaneous localization and tracking,” pp. 5191–5197, IEEE, May 2011. No citations.

[12] H. Schulz and S. Behnke, “Utilizing the structure of field lines for efficient soccer robot localization,” Advanced Robotics, 2012. No citations.

[13] S. Kohlbrecher, A. Stumpf, and O. v. Stryk, “Grid-Based Occupancy Mapping and Automatic Gaze Control for Soccer Playing Humanoid Robots,” in Proc. 6th Workshop on Humanoid Soccer Robots at the 2011 IEEE-RAS Int. Conf. on Humanoid Robots, (Bled), Oct. 2011. No citations.

[14] C. Y. Low, N. Aziz, M. Aldemir, M. Mellado, R. Dumitrescu, and H. Anacker, “Principle Solution for Designing Collaborative Humanoid Soccer Robots,” Procedia Engineering, vol. 41, pp. 1507–1515, Jan. 2012. No citations.

[15] S. Gtz, M. Leuthuser, J. Reimann, J. Schroeter, C. Wende, C. Wilke, and U. Amann, “A Role-Based Language for Collaborative Robot Applications,” in Leveraging Applications of Formal Methods, Verification, and Validation (R. Hhnle, J. Knoop, T. Margaria, D. Schreiner, and B. Steffen, eds.), pp. 1–15, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. No citations.

[16] L. Shuhua and G. Gaizhi, “The application of improved hsv color space model in image processing,” in Future Computer and Communication (ICFCC), 2010 2nd International Conference on, vol. 2, pp. V2–10–V2–13, May 2010. No citations.

[17] P. Hart, "How the hough transform was invented [dsp history]," Signal Processing Magazine, IEEE, vol. 26, pp. 18–22, November 2009. No citations.

Implementasi Sistem Handling dan Dribbling pada ERSOW (EEPIS Robot Soccer on Wheels)

M. Irfan Mas'udi*, Fernando Ardilla, Iwan Kurnianto Wibowo, Rizki Taufik Ulil A,
Bagas Mahardika, Muhammad Jauharul M

Laboratorium Robotika Politeknik Elektronika Negeri Surabaya
Kampus PENS, Jalan Raya ITS, Sukolilo Surabaya 60111
Surabaya, Indonesia
irfanmasoudi@gmail.com

***Abstrak*— Sistem *handling* dan *dribbling* merupakan salah satu hal yang paling mendasar didalam sepak bola. Maka dari itu, mekanisme sistem *handling* dan *dribbling* pada *middle size robot soccer* menjadi bagian yang sangat penting. Implementasi sistem *handling* dan *dribbling* bertujuan agar robot dapat menggiring bola dan mengikuti manuver robot seperti halnya pemain sepak bola yang melakukan teknik *dribbling,* untuk melakukan *handle* bola tetap pada penguasaan sampai dengan melewati lawan. Tidak hanya itu, implementasi sistem ini juga memiliki tujuan agar robot dapat bergerak lebih dinamis dan fleksibel. Pengunaan metode kontrol PID pada sistem *handling* dan *dribbling* sebagai pengatur respon *dribbler* dapat diterapkan untuk menjaga posisi bola agar tetap ideal. Dengan demikian, robot dapat melakukan proses *handling* dan *dribbling* bola ketika robot melakukan pergerakan translasi dan rotasi.**

Kata Kunci—dribbling; handling; middle size robot soccer

I. PENDAHULUAN

Pertandingan antara robot dan tim juara dunia dalam permainan sepakbola yang diperkirakan akan dilaksanakan pada tahun 2050 adalah cita- cita dari pengembang Robocup. Para peneliti robot *soccer* terus melakukan berbagai penelitian untuk meningkatkan kemampuan dari robot agar mampu melakukan permainan sepakbola seperti halnya manusia. Kompetisi Robocup, bidang robot soccer dibagi menjadi 2 divisi yaitu *humanoid robot soccer* dan *mobile robot soccer*. Divisi *humanoid* dikonsentrasikan dalam membangun sebuah mekanik robot pemain sepakbola layaknya seperti manusia. Sedangkan pada divisi mobile robot soccer lebih berkonsentrasi pada kecerdasan dan penerapan strategi pada robot *soccer*.

Pada pertandingan robot *soccer,* setiap robot selalu mempunyai kemampuan untuk bermain sepak bola, tetapi tidak semua memiliki kecerdasan yang sama untuk menyelesaikan sebuah tugas dan membuat sebuah goal, karena banyak faktor yang mempengaruhi hal tersebut. Seperti dalam hal pergerakan robot yang kurang stabil, mekanik yang tidak stabil atau bahkan sistem kecerdasan yang terlalu rumit sehingga menyebabkan lama dalam penyelesaian sebuah perintah.

Dalam penelitian ini, difokuskan pada desain dan kontrol *dribbler* pada *mobile robot soccer*. Penggunaan *dribbler* dimaksudkan untuk mempermudah robot dalam menguasai bola dan melakukan pergerakan, dengan desain *dribbler* yang baik maka pergerakan robot akan terlihat natural dan lebih fleksibel dalam melakukan manuver, karena tidak terpengaruh oleh *feedback* yang dihasilkan oleh pergerakan bola. Dalam mendesain *dribbler* pada robot *soccer* disesuaikan dengan peraturan *Middle Size Robot Soccer* antara lain selama pertandingan bola tidak boleh masuk ke area cekungan robot lebih dari seper tiga diameter bola, kecuali pada saat menangkap bola. Dan pada saat menghentikan bola, bola tidak boleh masuk lebih dari setengah dari diameter bola. Dalam melakukan kontrol terhadap kecepatan motor maka digunakan sensor yang dapat membaca sudut pada poros mekanik *dribbler*, sehingga pada penelitian ini input dari sistem adalah nilai yang didapatkan dari potensiometer sebagai sensor sudut. Dari pembacaan sensor tersebut maka dapat diketahui posisi dari bola, yang kemudian dilakukan kontrol kecepatan motor pada *dribbler*. Dengan memperhatikan posisi robot terhadap dribler kanan dan kiri maka akan ditentukan keputusan yang tepat agar bola dapat berada diposisi yang ideal.

II. DASAR TEORI

A. Robot Soccer

Robot *soccer* adalah salah satu jenis robot yang mempunyai tugas untuk melakukan permainan sepakbola layaknya manusia. Ada dua jenis robot sepakbola yaitu berupa humanoid dan juga ada yang berupa mobile robot. Dalam perkembangannya, robot soccer memiliki daya tarik tersendiri, sehingga berbagai perlombaan diadakan untuk robot jenis ini seperti KRSBI yang setiap tahun diadakan di Indonesia dan juga sebagai persiapan untuk mengikuti event internasional RoboCup. Event RoboCup memiliki cita-cita, yaitu diharapkan pada tahun 2050 mampu mencetak tim sepak bola robot yang mampu melawan tim juara dunia sepak bola.



Gambar 1. *Middle size robot soccer*

B. PID Controller

PID *controller* merupakan salah satu jenis pengaturan yang banyak digunakan. Selain itu sistem ini mudah digabungkan dengan metode pengaturan yang lain seperti Fuzzy dan Robust. Sehingga akan menjadi suatu sistem pengaturan yang semakin baik. PID *Controller* sebenarnya terdiri dari 3 jenis cara pengaturan yang saling dikombinasikan, yaitu *Proporsional*, *Derivative*, dan *Integral*. Masing-masing memiliki parameter tertentu yang harus di set agar sistem dapat bekerja dengan baik, yang disebut dengan konstanta. Setiap jenis, memiliki kelebihan dan kekurangan masing-masing, hal ini dapat dilihat pada tabel 1.

TABLE I. RESPON CONTROLLER TERHADAP PERUBAHAN KONSTANTA

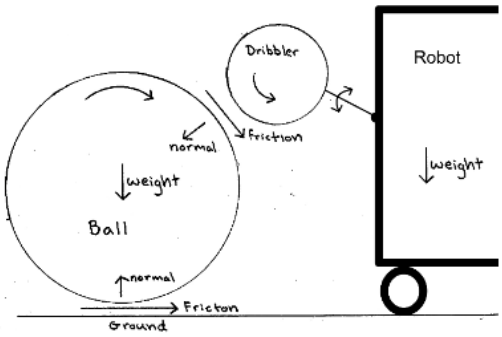
Closed-Loop Response	Rise Time	Overshoot	Settling Time	SS Error
Kp	Decrease	Increase	Small change	Decrease
Ki	Decrease	Increase	Increase	Eliminate
Kd	Small change	Decrease	Decrease	Small change

Parameter-parameter tersebut, tidak bersifat independen, sehingga salah satu nilai konstanta dirubah, maka sistem tidak akan bereaksi seperti yang diinginkan. Tabel diatas hanya dipergunakan sebagai pedoman jika akan melakukan perubahan konstanta, untuk merancang suatu PID *controlle*r, dengan menggunakan metode *trial & error*. Sehingga perancang harus mencoba mengkombinasikan pengaturan beserta konstantanya untuk mendapatkan hasil terbaik yang paling sederhana.

C. Dribbler System

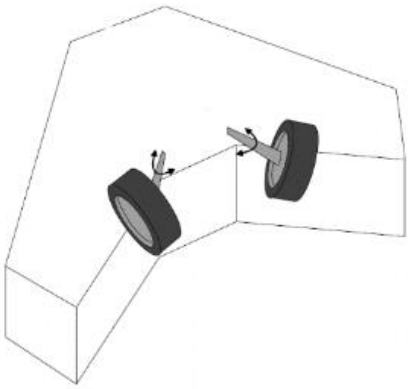
Konsep penggunaan motor dc untuk melakukan kontrol pada bola sangat mudah untuk dipahami. Jika bola melakukan kontak dengan bidang yang memiliki gaya tarik yang melebihi gaya gesek antara bola dan lantai, maka bola akan mengikuti bidang tersebut. Jika bidang tersebut adalah roda yang digerakkan dengan motor untuk mengatur rotasi dari roda tersebut, maka akan menyebabkan bola ikut berputar. Namun hal tersebut hanya dapat terjadi jika motor memiliki torsi yang cukup untuk membuat bola berputar dan memberikan gaya yang lebih besar di bandingkan dengan gaya yang dihasilkan oleh bola dan lantai.

Dalam peraturan yang diterapkan di Robocup, bola hanya sepertiga bagian yang boleh masuk ke dalam *dribbler*. Jika peletakan roda seperti pada Gambar 2, maka gaya yang berlaku tidak terlalu besar, namun hal tersebut masih akan memberikan efek pada gerak dari bola.



Gambar 2. Konsep *active dribbler*

Kontak antara bola dan *dribbler* sangat berpengaruh pada sistem pergerakan dari robot. Roda *dribbler* harus memiliki kecepatan sudut yang sangat tinggi, sehingga bola dapat tertahan pada jarak yang dapat dijangkau oleh *dribbler*. Terdapat beberapa contoh konsep *dribbler* yang diterapkan di Robocup, meskipun jumlah roda dan konsep yang diterapkan hampir sama. Penggunaan dua roda untuk memberikan titik kontak yang lebih besar sehingga dapat memberikan gaya yang lebih besar pula dapa bola, dan sistem pergerakan asimetris akan memberikan kemampuan untuk memutar bola dan menempatkan bola pada posisi yang sesuai dengan yang diharapkan. Gambar 3 merupakan desain dari konsep penggunaan dua roda pada sistem handler atau dribbler pada *middle size robot soccer.*



Gambar 3. Konsep dua roda pada *active dribbler*

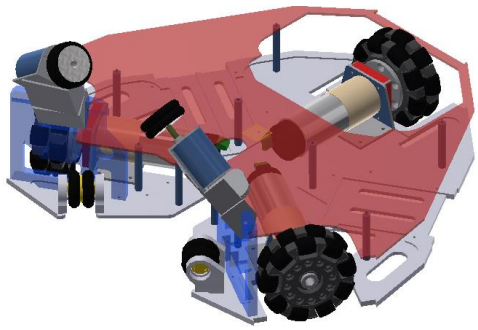
Untuk melakukan kontrol pada bola, maka arah rotasi dari roda *dribbler* harus disesuaikan. Karena pergerakan roda tersebut akan mempengaruhi gaya yang akan menyebabkan rotasi pada bola. Dari pergerakan bola tersebut akan memberikan umpan balik pada sistem, umpan balik yang diberikan berupa posisi bola terhadap robot.

III. DESAIN SISTEM

A. Desain Mekanik

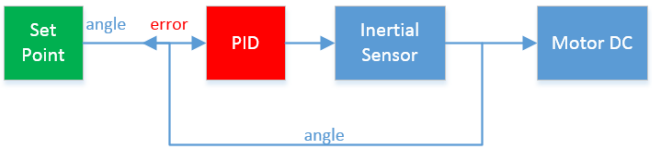
Desain mekanik *handling* dan *dribbling* pada ERSOW

memiliki kemampuan untuk melakukan kontrol pada bola, sehingga bola akan berada pada posisi ideal. Mekanisme *handling* dan *dribbling* menggunakan motor dc sebagai penggerak yang digunakan untuk mengatur kecepatan putaran dan arah roda. Untuk mendapatkan posisi dari bola, diperlukan sensor untuk membaca sudut pada poros mekanik *handling* dan *dribbling* menggunakan potentiometer. Desain mekanik *handling* dan *dribbling* pada ERSOW ditunjukkan pada Gambar 4.



Gambar 4. Desain mekanik *handling* dan *dribbling* ERSOW

Mekanisme *handling* dan *dribbling* diperlukan kontrol kecepatan dan arah putar roda *dribbler* sehingga bola dapat bergerak sesuai dengan pergerakan robot. Kecepatan dan arah putar motor diperlukan umpan balik dari potensiometer untuk mengetahui posisi kemiringan dari sudut poros. Posisi kemiringan sudut poros akan menetukan kecepatan dan arah bola ketika robot bergerak yang dikontrol menggunakan kontrol PID (*Proportional Integral Derivative*). Diagram sistem kontrol pada mekanisme *handling* dan *dribbling* terdapat pada Gambar 5.



Gambar 5. Sistem Kontrol pada mekanisme *handling* dan *dribbling*

Set Point yang diberikan pada sistem ini adalah nilai tegangan dari potensiometer ketika sepertiga dari bagian bola masuk pada robot, seperti pada Gambar 6.



Gambar 6. Posisi sepertiga bola masuk pada bagian robot

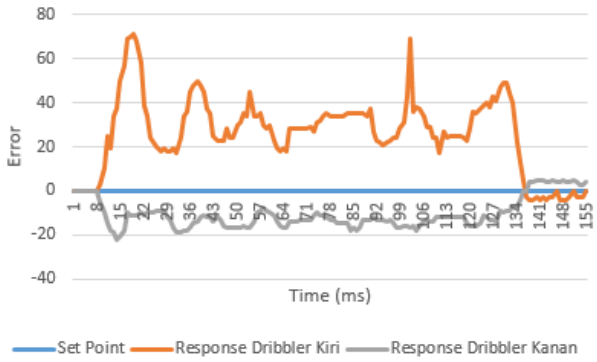
Nilai error yang didapatkan dari perubahan tegangan potensiometer digunakan sebagai umpan balik dari kontrol PID pada sistem ini. Hasil dari keluaran kontrol PID adalah respon dari roda *dribbler* sehingga dapat menentukan arah dan kecepatan putaran motor.

IV. HASIL DAN ANALISA

Pengujian dan analisa dimaksudkan untuk mengetahui sejauh mana sistem yang dibuat dalam penelitian ini dapat bekerja sesuai dengan yang diharapkan serta melakukan analisa pada setiap tahapan proses dan hasil secara keseluruhan.

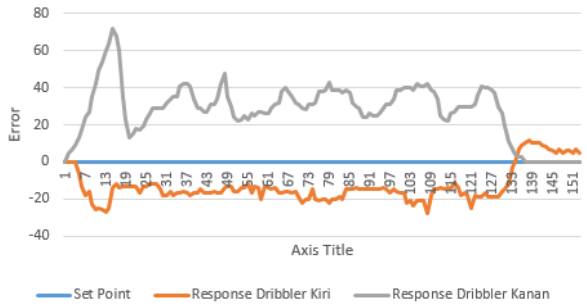
A. Respon Sistem Terhadap Pergerakan Robot Translasi

Pada pengujian respon sistem terhadap pergerakan robot secara translasi dilakukan dengan melakukan pergerakan robot terhadap sumbu x dan y. Gambar 7 menunjukkan respon sistem pada saat pergerakan robot ke kanan atau bergerak pada sumbu x positif.



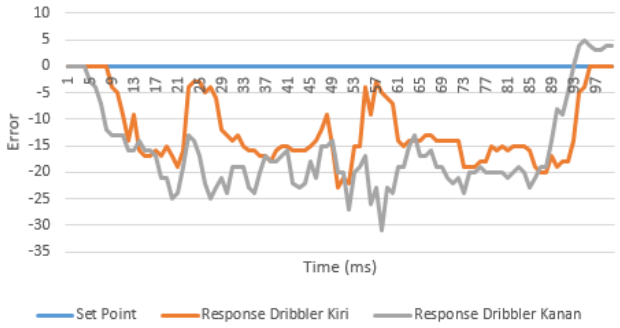
Gambar 7. Respon sistem pada saat pergerakan robot ke kanan

Berdasarkan Gambar 7 dapat dilihat bahwa pada saat pergerakan robot ke kanan, maka beban atau nilai *error* pada *dribbler* kiri menjadi lebih besar di bandingkan dengan *dribbler* kanan. Gambar 8 menunjukkan respon sistem pada saat pergerakan robot ke kiri atau bergerak pada sumbu x negatif.



Gambar 8. Respon sistem pada saat pergerakan robot ke kiri

Dari Gambar 8, respon *dribbler* pada saat pergerakan robot ke kiri mengakibatkan beban atau nilai *error* pada *dribbler* kanan menjadi lebih besar dibandingkan dengan *dribbler* kiri. Pergerakan robot translasi terhadap sumbu x mengakibatkan respon *dribbler* untuk memutar bola sesuai dengan pergerakan robot. Untuk dapat melakukan putaran terhadap bola diperlukan respon *dribbler* kanan dan kiri yang berlawanan. Kemudian pada Gambar 9 dan 10 menunjukkan respon sistem pada saat robot bergerak maju dan mundur atau bergerak pada sumbu y positif dan negatif.



Gambar 9. Respon sistem pada saat pergerakan robot maju

Dari grafik pada Gambar 9 dapat menunjukkan bahwa pada saat pergerakan robot maju, maka beban atau nilai *error* pada *dribbler* kanan dan kiri berimbang. Pada Gambar 10 menunjukkan respon sistem pada saat pergerakan robot mundur atau bergerak pada sumbu y negatif.

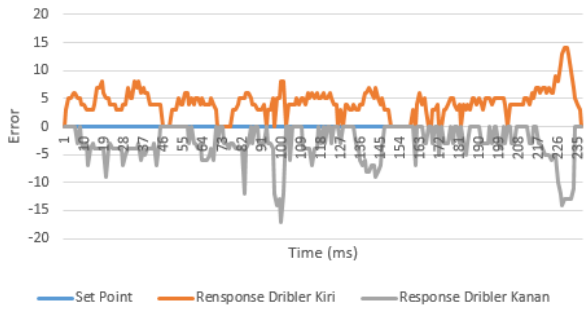


Gambar.10. Respon sistem pada saat pergerakan robot mundur

Berdasarkan pada Gambar 10, bahwa pada saat pergerakan robot maju, maka beban atau nilai *error* pada *dribbler* kanan dan kiri berimbang. Dari grafik respon *dribbler* ketika robot melakukan pergerakan maju dan mundur diperlukan respon *dribbler* kanan dan kiri yang seimbang, sehingga dapat memberikan putaran terhadap bola agar robot dapat mempertahankan posisi bola.

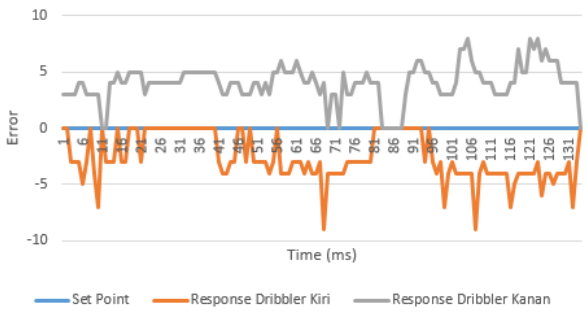
B. Respon Sistem Terhadap Pergerakan Robot Berotasi

Pengujian respon sistem terhadap pergerakan robot ketika melakukan pergerakan rotasi, yang menjadi titik pusat rotasi adalah titik tengah bola. Sehingga respon *dribbler* akan mempertahankan bola agar berada pada titik pusat rotasi ketika robot berotasi. Pengujian ini dilakukan dengan melakukan pergerakan robot ketika berotasi ke kanan dan ke kiri . Pada Gambar 11 menunjukan respon *dribler* ketika robot berotasi ke kanan. Respon *dribbler* kanan dan kiri berusaha mempertahankan bola dengan memutar bola ke arah kiri. berlawanan dengan pergerakan robot.



Gambar.11. Respon sistem pada saat pergerakan rotasi ke kanan

Berdasarkan Gambar 12, ketika robot berotasi ke kiri, respon *dribbler* kanan dan kiri memutar bola berlawanan dengan pergerakan robot. Dari Gambar 11 dan 12 maka dapat terlihat bahwa respon *dribbler* akan menjaga bola berada pada titik pusat rotasi.



Gambar.12. Respon sistem pada saat pergerakan rotasi ke kekiri

V. PENELITIAN SELANJUTNYA

Penelitian selanjutnya adalah memperbaiki desain dan kontrol pada handler robot sehingga pergerakan bola lebih natural. Selain itu penambahan parameter masukan seperti kecepatan motor sehingga output dari sistem lebih maksimal dan sesuai dengan yang diharapkan.

REFERENSI

[1] Hoogendijk, R."Design of a Ball Handling Mechanism for Robocup". DCT 2007.018

[2] Santos Serafim, M. "Ball Handling Mechanism for Mobile Robot". 2013

[3] Li, Xiang., Wang, Maosen., Zell, Andreas. "Dribbling Control of Omnidirectional Soccer Robots".

[4] V. Ciesielski and S.Y.Lai,"Developing a dribble-and-score behaviour for robot soccer using neuro evolution", in *Workshop on Intelligent and Evolutionary Systems*, Duniedin, New Zealand, 2001, pp.70-78.

Deteksi Bola Putih Menggunakan OpenCV pada Robot Penyerang dengan Mini PC Odroid XU4

Yoga Sukma Pradana*, David Christanto, Tunggal Manda Ary T., San Sayidul Akdam A.
Fakultas Ilmu Komputer
Universitas Brawijaya
Malang, Indonesia
yogapradana@gmail.com

Abstrak—*Di dalam Kontes robot sepak bola Indonesia, deteksi objek bola sangatlah penting. Di setiap pertandingan, tim pasti memiliki robot penyerang. Robot ini harus mampu mendeteksi bola putih yang ada di lapangan. Pada jurnal ini menggunakan Odroid XU4 sebagai mini pc dan opencv sebagai library untuk mendeteksi bola putihnya. Penggunaan odroid xu4 berdampak pada waktu pengambilan gambar yang semakin cepat. Sehingga robot dapat mendeteksi dan menuju bola dengan cepat. Eksekusi motion menendang pun dapat lebih cepat.*

Kata Kunci — ***ball recognizing; odroid XU4; openCV; robot***

I. PENDAHULUAN

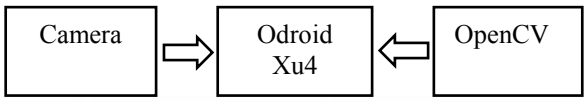
Kontes Robot Sepakbola Indonesia Humanoid (KRSBI Humanoid) merupakan salah satu divisi lomba yang ada di Kontes Robot Indonesia 2017. KRSBI Humanoid sekaligus menjadi ajang kualifikasi nasional untuk mewakili Indonesia dalam RoboCup yang merupakan kompetisi robot sepak bola resmi tingkat dunia dibawah organisasi RoboCup [1].

Robot di KRSBI Humanoid tiap tahunnya terus mengalami peningkatan kemampuan. Hal ini dikarenakan peraturan atau rule yang tiap tahun juga berubah menjadi lebih kompleks. Salah satu peraturan yang mengalami perubahan di Kontes Robot Indonesia 2017 ini adalah penggunaan bola yang tidak lagi berwarna orange. Bola yang digunakan di tahun ini berwarna putih.

Robot penyerang adalah salah satu robot di KRSBI Humanoid. Robot ini harus mampu memiliki beberapa kemampuan dasar, salah satunya adalah deteksi bola. Bola pada KRSBI Humanoid yang digunakan saat ini adalah bola berwarna putih. Tentunya menggunakan bola berwarna putih ini, akan menjadi tantangan tersendiri. Robot harus mampu membedakan bola dengan garis lapangan dan gawang yang juga berwarna putih. Oleh karena itu, pada jurnal ini akan dibahas bagaimana robot penyerang dapat mendeteksi bola putih dengan menggunakan *library* OpenCV.

II. METODE

Pada robot ini menggunakan Odroid Xu4 sebagai main *controller* yang digunakan sebagai pengolah gambar dan menggunakan *library* OpenCV sebagai pengolah citra digital pada bola putih.



Gambar. 1 : Blok diagram



Gambar. 2. Raspberry Pi 2

Odroid XU4

Odroid kembali merilis mini-PC-Xu4 Odroid terbaru. XU-4 membawa solusi HMP (Heterogen Multi-Processing). Seri XU-4 ini adalah seri XU yang paling popular.

Spesifikasi Odroid XU4

- CPU: Samsung Exynos544 Cortex-A15 + Korteks A7 2 GHz (Octacore 64 bit).

- GPU: Mali-T628 MP6 (OpenGL ES 3.0/2.0/1.1 and OpenCL 1.1 full profile).
- RAM: 2GB LPDDR3 SDRAM.
- GPIO: 30 + 12 GPIO.
- Ports: 2x USB 3.0, 1x USB 2.0, Gigabit Ethernet, HDMI 1.4a, microSD dan konektor eMMC.

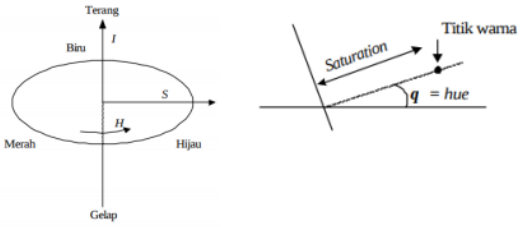
ODROID-XU4 adalah generasi baru perangkat komputasi dengan perangkat keras yang lebih powerfull dan hemat energi serta bentuk yang lebih kecil.

Menawarkan dukungan open source, board ini bisa menjalankan berbagai jenis Linux, termasuk Ubuntu 16.04 terbaru dan Android 4.4 KitKat, 5.0 Lollipop dan 7.1 Nougat.

Dengan menerapkan interface eMMC 5.0, USB 3.0 dan Gigabit Ethernet, ODROID-XU4 menawarkan kecepatan transfer data yang luar biasa, fitur yang semakin dibutuhkan untuk mendukung daya proses lanjutan pada perangkat ARM.

Atribut Warna HSV

Selain RGB, warna juga dapat dimodelkan berdasarkan atribut warnanya. Setiap warna memiliki 3 buah atribut, yaitu intensity (I), hue (H), dan saturation (S). h*ue* disini merepresentasikan nama dari suatu warna seperti merah, biru, hijau dan sebagainya, v*alue* menentukan gelap terangnya suatu warna dan s*aturation* berhubungan dengan cerah atau suramnya suatu warna [2].



Gambar. 3. Model HSV

Hue dikuantisasi dengan nilai dari 0 sampai 255, dimana 0 menyatakan warna merah, lalu memutar nilai-nilai spektrum tersebut kembali lagi ke 0 untuk menyatakan merah lagi. Ini dapat dilihat sebagai sudut dari 0° sampai 360°. Jika suatu warna mempunyai saturation = 0, maka warna tersebut tanpa hue, yaitu dibuat dari warna putih saja. Jika saturation = 255, maka tidak ada warna putih yang ditambahkan pada warna tersebut.

Saturation dapat digambarkan sebagai panjang garis dari titik pusat lingkaran ke titik warna.

Intensity nilainya dari gelap sampai terang (gelap = 0, terang = 255). Intensity dapat digambarkan sebagai garis vertikal yang menembus pusat lingkaran.

Circular Hough Transform

Pada tahap ini kami bermaksud untuk memperkirakan lingkaran yang terlihat pada *image*. Ruang parameter untuk CHT adalah matriks 3-D dengan sumbu x dan y menjadi piksel sepanjang lebar dan tinggi sumbu gambar dan z yang menunjukkan radius yang berbeda [3].

Untuk setiap garis pada lingkaran dengan jari-jari yang berbeda terpusat pada titik itu ditarik dalam bidang z-bidang yang sesuai dari ruang parameter, yaitu untuk titik (i, j) atau (baris, kolom) dari peta tepi dan r radius, lingkaran pada Tingkat parameter ruang yang terpusat pada (i, j). Proses ini diulang untuk jari-jari lainnya. Lingkaran ini ditambahkan di ruang parameter, artinya jika satu titik dimiliki beberapa lingkaran dan setiap nilainya akan bertambah. Dalam metode kami, penjumlahan ini digantikan oleh pertumbuhan eksponensial untuk menghitung intensitas setiap titik sebagai berikut:

$$I_n = I_o \times c \quad (1)$$

Dimana I_n dan I_o mewakili intensitas baru dan lama dari sebuah titik di ruang parameter, masing - masing dan c adalah konstanta positif yang lebih besar dari satu kesatuan.

OpenCV

OpenCV merupakan *library yang* dikembangkan oleh Intel *Corporation* dan disediakan untuk komputasi pada aplikasi berbasis *realtime* [4]. OpenCV pada umumnya banyak digunakan untuk proses pengolahan citra digital. OpenCV dapat berjalan pada berbagai platform, salah satunya adalah Linux. Penggunaan OpenCV antara lain pada seni interaktif, inspeksi tambang, menampilkan peta di web melalui teknologi robotik. Modul pada OpenCV sangat kompleks dan implementasi dari library ini juga sangat banyak. Jadi, dalam pengembangan suatu aplikasi lebih mudah untuk proses membangun sebuah aplikasi realtime.

Pengolahan Citra Digital

Pengolahan citra adalah pemrosesan citra atau mengolah nilai piksel pada citra untuk memperbaiki sebuah citra, mendefinisikan citra maupun mengklasifikasikannya [5]. Pada umumnya, operasi-operasi pada pengolahan citra diterapkan pada citra apabila :

1. Perbaikan atau memodifikasi citra perlu dilakukan untuk meningkatkan kualitas penampakan atau untuk menonjolkan beberapa aspek informasi yang terkandung di dalam citra,
2. Elemen di dalam citra perlu dikelompokkan , dicocokkan, atau diukur,
3. Sebagian citra perlu digabung dengan bagian citra yang lain.

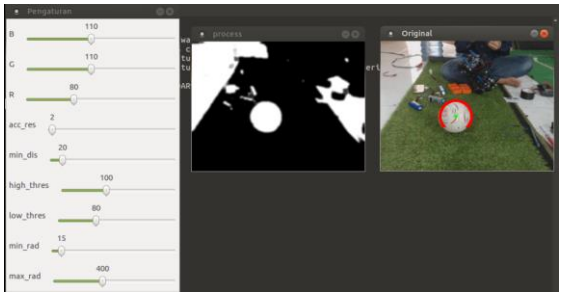
Thresholding merupakan proses segmentasi pada gambar yang memisahkan obyek yang

diperlukan dengan yang tidak diperlukan. Dalam paper ini *threshold* digunakan untuk mengurangi noise pada citra sehingga mempermudah dalam mendefinisikan suatu citra.

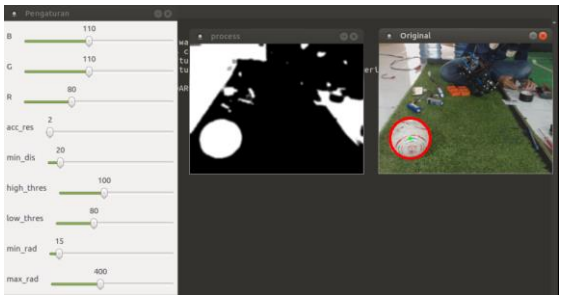
III. DISKUSI

Dalam uji coba pengambilan gambar. Bola diletakan pada beberapa tempat yang banyak kemungkinan terdapat banyak noise yang dapat mempengaruhi robot mengenali bola.

Hasil pengujian pengambilan gambar menggunakan metode tresholding dan circular hough transform untuk mengenali bola.

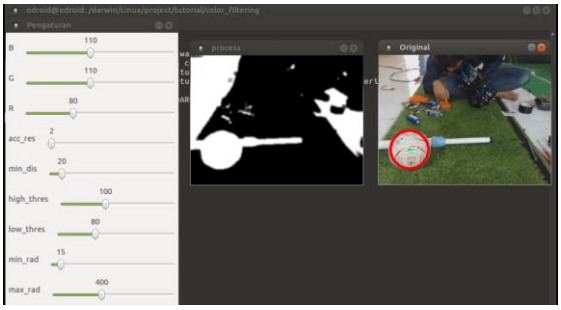


Gambar. 4. Pengujian 1

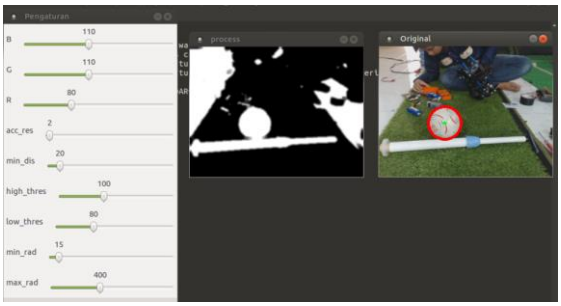


Gambar. 5. Pengujian 2

Hasil pengujian pengambilan gambar menggunakan metode circular hough transform dengan diberikan sebuah noise.



Gambar. 6. Pengujian 3



Gambar. 7. Pengujian 4

IV. KESMPULAN

Hasil dari uji coba dalam pengambilan gambar dengan menggunakan metode houghcircle, robot dapat mengetahui posisi bola dan dapat menentukan apa yang dilihat tersebut adalah bola atau bukan. Penggunaan odroid xu4 juga berdampak pada waktu pengambilan gambar yang semakin cepat. Sehingga robot dapat mendeteksi dan menuju bola degan cepat. Eksekusi motion menendangpun dapat lebih cepat.

DAFTAR PUSTAKA

[1] Pitowarno, M.Eng, Dr.Ir. Endra; K, Ertanto Budi;. (2015). KONTES ROBOT SEPAK BOLA INDONESIA (KRSBI) 2015. Jakarta: Kementerian Pendidikan dan Kebudayaan Nasional (KEMDIKBUDNAS).

[2] Helmiriawan. (2012). Rancang Bangun dan Analisis Sistem Pemantau Lalu Lintas Menggunakan OpenCV dengan Algoritma Canny dan Blob Detection. Depok: Universitas Indonesia.

[3] Hossein Mohammad, Rezaie Jamshid; 2011. An Innovative Implementation of Circular Hough Transform using Eigenvalues of Covariance Matrix for Detecting Circles . International Symposium ELMAR-2011.

[4] Shengze, Peng; Yongge, Wen;. (2010). Research Based on the HSV Humanoid Robot Soccer Image Processing. IEEE, 2-3.

[5] Putra, D. (2010). Pengolahan Citra Digital. Yogyakarta: Penerbit Andi.

Dual-camera Perception System on Soccer Robot

Dhimas Bintang Kusumawardhana*, Kusprasapta Mutijarsa

School of Electrical Engineering and Informatics
Bandung Institute of Technology
Bandung, Indonesia
dbkusumawardhana@students.itb.ac.id

***Abstract*—The perception system is one of the major parts of the autonomous mobile robot. The perception system performs a sense to know and recognize objects on its working environment. In soccer robots, perception systems use cameras to recognize the field along with its features, such as goalpost and ball. A camera-equipped perception system has limitations in capturing object images on the field. This paper discusses the results of the combined two cameras, namely front camera and omnidirectional camera, to detect the position of objects on the field, the position of the ball and goalpost, in a better way. The image captured by the camera is processed by using digital image processing techniques such as image thresholding, Gaussian blur, Canny edge detector and Hough transform. The techniques are implemented by using the OpenCV library. Object information obtained from the image processing is forwarded to the robot coordination system, so then the robot is able to control the movement and strategy during the game. Testing has been done to prove that the perception system built is able to extract information from the exist objects on the field, in accordance with the specifications of sensors and processors used. Overall, the perception system with two cameras is able to extract the information needed by soccer robots to recognizing the environment. Integration testing of the whole soccer robot system shows that the robot is able to recognize, approach, then herd and kick the ball toward the goal.**

Keywords— Canny edge detector, Front camera, Gaussian blur, Hough transform, Image thresholding, Omnidirectional camera, Soccer robot

I. INTRODUCTION

Various research has been done to improve the development in the field of robotics. Among various types of robots, soccer robots tend to get more attention than other types of robots. This is because soccer robot is more interesting to develop. In addition, the soccer robot must have cognitive abilities that support the game strategy, therefore it seems more challenging.

Implementation of the perception system on robots serves to detect objects that are located on the field. Without a perception system, it is impossible for a robot to be able to carry out its mission of playing soccer by following the general rules of soccer robot. In other words, the existence of a perception system on a soccer robot is a mandatory requirement for a soccer robot platform.

The focus of this final project is to develop soccer robot perception system so that the robot is able to recognize the objects that exist in the vicinity. The object defined is limited to the ball and the goalpost only. Derived from these criteria, the basic ability that must be possessed by the perception system is ability which makes the robot can recognize those objects. The first function is to recognize the goalpost and its direction. The second function is to detect the existence of the ball from a distance. And, the last function is to get the position of a close ball around the robot.

The perception system of two cameras is applied by using two cameras that work simultaneously, i.e. front camera and omnidirectional camera. The front camera is able to capture image from a distance, while the viewing angle is limited to one side. This camera is used to detect the goal and the ball away. Besides, omnidirectional camera capable of capturing images from various angles at once, yet experiencing a lot of distortion to capture the image information of distant objects. This camera is used to detect the ball that is around the robot.

II. PROPOSED ALGORITHMS

A. Gaussian Blur

In image processing, Gaussian blur is an image blurring by using Gaussian function. This method is widely used in graphics software to reduce noise and obscure details on images. The visual effect of this technique is the image with a smooth blur with a flat distribution. Gaussian blur is commonly used in the early stages of digital image processing to improve the structure of the image on a certain scale.

Mathematically, the application of Gaussian blur to the image is identical to applying the convolution to the image and Gaussian function. Since the Fourier transform of the Gaussian function is a Gaussian function as well, the application of the Gaussian blur affects high-frequency component removal. In other words, Gaussian blur is a low pass filter.

The Gaussian function for one dimension is formulated by

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-} \qquad \text{(Eq. 1)}$$

The Gaussian function equation for two dimensions is the multiplication of each Gaussian function in one dimension.

$$G(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2}{2}} \qquad \text{(Eq. 2)}$$

With x is the distance from the origin on the horizontal axis, y is the distance from the origin on the vertical axis, and is the standard deviation of the Gaussian distribution. When applied in two dimensions, this equation produces a circular contoured surface and is concentrated at the center of the circle following the Gaussian distribution. The value of the distribution is then used to form the convolution matrix applied to the original image [1] - [2].



Fig. 1. Example of a pair of original image (left) and the image after processed through Gaussian Blur [1]

B. Canny Edge Detector

Canny edge detector is an edge detection method that uses a multi-stage algorithm to detect the edges of objects in an image. Canny edge detector is able to extract the structural information of various objects while reducing the amount of data to be processed.

Among the various edge detection methods that have been developed so far, Canny edge detector algorithm is considered as the best method that provides detection results. Easy criteria and simple processes make this algorithm one of the most popular edge detection algorithms [3].

There are five stages in the Canny edge detector algorithm. The first stage is the use of Gaussian Blur to smooth the image. The second stage is the determination of the gradient in the image. The third stage is application of non-maximum suppression to the image to remove unwanted portions of the image. The fourth stage is the application of a double threshold to select a particular part of the remaining pixels [3] - [4], [6]. Finally, the last stage is the edge determination using hysteresis.

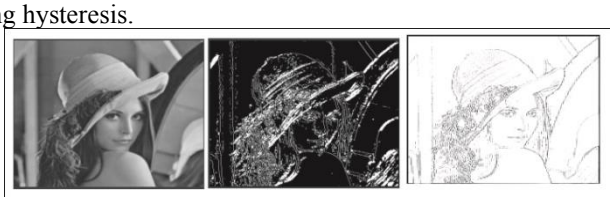


Fig. 2. Examples of images before (left) and after (right) processed through Canny edge detector [5]

C. Hough Transform

Hough Transformation is a feature extraction technique used in digital image processing analysis. The purpose of this technique is to find a particular shaped object through a voting procedure. The voting procedure is performed within a certain range of parameters, with the obtained candidate object referred as the local maximum on the accumulator space. The classical Hough transformation is concerned with the identification of the line in the image, but recently Hough transform has been extended to identify various shapes i.e. circles or ellipses [6].

Hough line transformation is applied to detect line-shaped objects in an image [7]. The main principle of this method is to transform the coordinates (x, y) into coordinates (R, θ). R shows the shortest distance from the line passing point (x, y) to the origin (0,0). Theta (θ) denotes the rotation angle to the X axis. The Hough transform will produce the Hough space, the voting result of the detected pixel and the line passing through the pixel [8].

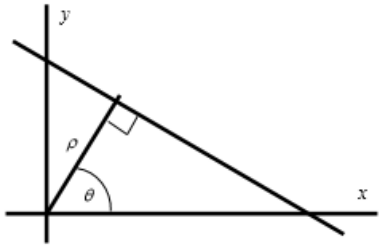


Fig. 3. Coordinate transformation from (x,y) to (R, θ) [8]

The image of the ball caught by camera may change from time to time depending on the environmental conditions and distortion on the camera. To detect ball features, it uses conic-fitting method. The algorithm used in this method is Lin algorithm that uses algebraic distance. The algebraic distance algorithm is denoted by

$$\epsilon^2(a) = \sum_{i=1}^{n} F(a, x_i)^2 = \|Da \quad \text{(Eq. 3)}$$

With $\|a\|^2 =$, andmatrix D is an n x 6 with x_i representing row.This function minimizes the objective function objectively to form an eigenvector.

$$E = \|Da\|^2 - \lambda(\|a\| - 1) = a^T D^T Da - \lambda(a^T a - 1) \quad \text{(Eq. 4)}$$

$$\nabla_a E = 0 \Leftrightarrow 2D^T Da - 2\lambda a = 0 \quad \text{(Eq. 5)}$$

with is Lagrange multiplier [9].

III. Perception System Design

A. Robot System Block Diagram

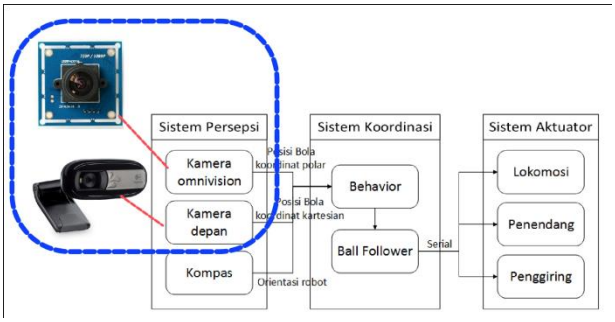


Fig. 4. Robot system block diagram, the surrounded part is the scope of this research

B. Goalpost Detection Using Front Camera

Goal detection by using the front camera works by looking for vertical and horizontal lines which is suspected to be the

line of goalpost. From these lines, the algorithm then searches for the intersection location between the horizontal line and the vertical line which is the top corner of the goalpost. Then check the vertical line from the pole down to find the bottom of the goalpost.



Fig. 5. Flowchart of goal detection algorithm using front camera

C. Distant Ball Detection Using Front Camera

Distant ball detection utilizes the CHT (Circle Hough Transform) method [6] to find the existence of a circular object within the image. Circle-shaped objects magnitude obtained then calculated to obtained the coordinates of the center. These coordinates are the spherical coordinates of the image frame.



Fig. 6. Flowchart of distant ball detection algorithm using front camera

D. Close Ball Detection Using Omnidirectional Camera

Detection of close ball around robot by using image thresholding method is purposed to distinguish orange ball from other objects on the field. The object detection process is ended after finding the contour of the object which is needed to describe the selected pixel as a new object.



Fig. 7. Flowchart of close ball detection algorithm using omnidirectional camera

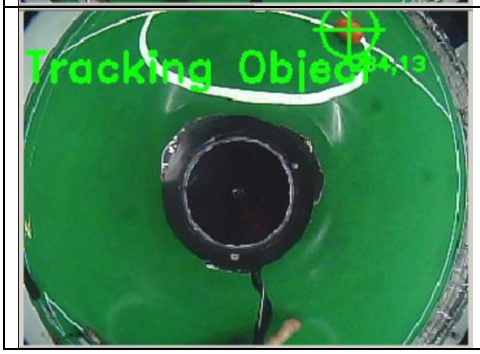
IV. Implementation and Testing Result

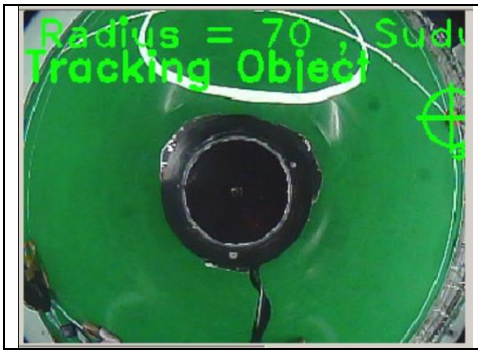
To find information that can be extracted by omnidirectional camera, observation is attempted on the result of digital image processing, which can be done through the interface that has been shown. Testing is done at four points around the robot by recording the distance and angle of the ball against the robot. Distances are given in centimeters, while angles are given in degrees. It is defined that 0 degrees is right in front of the robot, the angle in the direction of the clockwise direction is positive, and the angle in the counterclockwise direction is negative.

Angle feature extraction from the omnidirectional camera image gives incomplete results. Distortion appears in the image due to the camera's convex lens, so it impacts the ball readings in Cartesian coordinates. Distortion in the convex lens refers to the polar coordinates, whereas the determination of angles using Cartesian coordinates. As a result, the further distance the ball from the center of the robot (camera center), the greater the error resulted.

It can be seen from the picture in position 3 and 4, that the difference in distance between the two positions is in fact quite large (actually about 40 cm), but the extraction results show that the distance difference is only 20 cm. However, the use of omnidirectional camera is intended to detect a ball close to the robot so that the robot can maneuver to dribbling ball. In other words, errors that increase as the actual spacing of the ball increases with the robot has no major effect on the robot's behavior.

TABLE I. Ball Detection Result Using Omnidirectional Camera

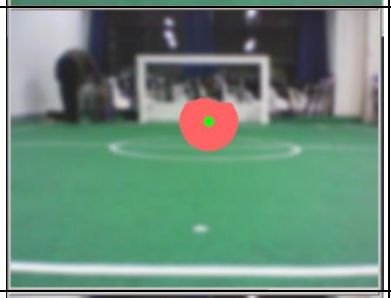
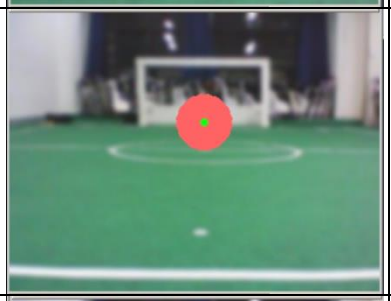
Image Resulted	Distance	Angle
	39 cm	144°
	56 cm	-44°
	50 cm	54°

	70 cm	34°

Different from omnidirectional camera, the front camera is aimed to find the ball in a long distance from the robot. Front camera function testing has been done by testing the ability to read the ball at different distances, with an increment of 1.2 meters. The feature resulted is a coordinate of spherical image of the camera in pixels' unit, with the original axis located in the top left corner. The x axis is positive to the right and the y-axis is positive.

The position of the ball against the camera is shown at the table, it is shown in Cartesian coordinates. The blurring provides a good effect on the CHT algorithm for detecting circle shaped object. As a result of the disguise performed, the noise around the ball object becomes reduced so the process of ball detection becomes easier. The 7.2 meters' sphere actually looks very small on the image, but with the CHT algorithm, a small circle object can be performed as a means as a ball. That way the robot can find a ball which is located at the opposite end of the field.

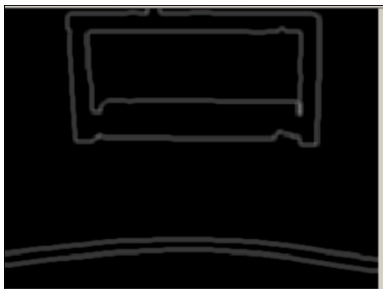
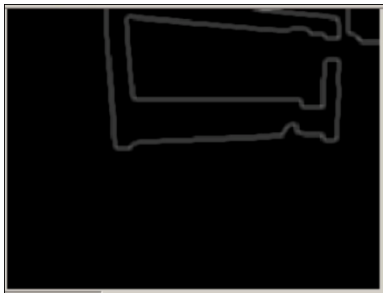
TABLE II. BALL DETECTION RESULT USING FRONT CAMERA

Distance	Image Resulted	Interface
1.2 m		X : 166.00 Y : 152.00
2.4 m		X : 168.00 Y : 118.00
3.6 m		X : 172.00 Y : 106.00
4.8 m		X : 168.00 Y : 100.00
6 m		X : 168.00 Y : 94.00
7.2 m		X : 156.00 Y : 90.00

To detect goalpost, Hough line algorithm that works by finding the line to be classified as a goal object is implemented. Hough line algorithm results are strongly influenced by the angle of image due to the difference in projection sizes of goalposts, thus is affects the process of image selection.

The goal detection process for two different shooting angles gives the same result, which is able to give the position of the goal relative to the camera. At color thresholding stage, all objects with white color are detected. But in the second stage, the Hough line algorithm selects all white objects and defines certain objects which is believed to be the goal. In the process, there is not always the two goalposts immediately found, but sometimes the algorithm makesa prediction when there is only one pole detected. As long as the pole detected, either just one or two, the program can always provide the pole position information.

TABLE III. GOALPOST DETECTION RESULT USING FRONT CAMERA

Image Resulted	Interface
	Kiri 146 14 248 Kanan 146 14 248 Kiri 147 11 246 Kanan 266 150 5 255 Kiri 153 10 245 Kanan 267 136 17 252 Kiri 146 8 245 Kanan 268 128 19 204 Kiri 145 6 244 Kanan 269 122 23 180 Gawang Kanan Points 172 704262 Hanya satu tiang terdeteksi 268 52 Kiri 148 35 109 Kanan 148 35 109 Tidak terdeteksi Prediksi Kanan
	83 4 Kiri 98 44 64 Kanan 98 44 64 Tidak terdeteksi Prediksi Kiri Points 81 18473 Hanya satu tiang terdeteksi 83 5 Kiri 109 70 76 Kanan 109 70 76 Tidak terdeteksi Prediksi Kiri Points 54 Points 75 Points 54 Points 92 165738 Dua tiang terdeteksi 88.693 67.8415 284.117 70.9364 Points 75 Points 69

V. CONCLUSION

The perception system is able to detect the existence of objects in the form of a ball and goal by utilizing two cameras that work simultaneously. The front cameras can be used to detect the presence of distant ball and goalpost, while the omnidirectional camera can be used to detect close ball around the robot.

ACKNOWLEDGMENT

Author wishes to thank Dago Hoogeschool Team and Dagozilla Team for their kindness to complete this project.

REFERENCE

[1] Z. Zhengwu, K. Eric, S. Anuj, "Gaussian Blurring-Invariant Comparison of Signals and Images" in Proc. IEEE TRANSACTIONS ON IMAGE PROCESSING, 2013, p. 3145.

[2] P. Timothy, C. Andrea, H. David, "ccurate and Efficient Method for Smoothly Space-Variant Gaussian Blurring" in Proc. IEEE TRANSACTIONS ON IMAGE PROCESSING, 2010, p. 1362.

[3] N. Abdullah Al, K. Yinan, H. Md Nazmul, "Performance Analysis of Canny's Edge Detection Method for Modified Threshold Algorithms", in Proc. ICEEE, 2015, India.

[4] L. Juseong, T. Hoyoung, P. Jongsun, "Energy Efficient Canny Edge Detector for Advanced Mobile Vision Applications" in Proc. IEEE Transactions on Circuits and Systems for Video Technology, 2016.

[5] T. Ashwini P, S. B. Bagal, "A Distributed Canny Edge Detector: Comparative Approach", Vishwakarma Institute of Technology, 2015.

[6] S. Sang-Woo, K. Myunggyu, "Efficient architecture for circle detection using Hough transform", ETRI, 2015.

[7] Y. Huashan, S. Guocan, W. Lina, Z. Min, "A new method based on Hough Transform for quick line and circle detection", Hubei University of Science and Technology, 2015.

[8] A. N. Fitriana, K. Mutijarsa, W. Adiprawita, "Color-Based Segmentation and Feature Detection for Ball and Goal Post on Mobile Soccer Robot Game Field", in Proc. ICITSI, 2015.

ICHIRO TEAM – Team Desciption Paper

Muhtadin*, Satria Hafizhuddin, Muhammad Arifin, Muhammad Reza Ar Razi, Dhany Satrio Wicaksono, Tommy Pratama, Sulaiman Ali, Deo Alfitra Ramazhoni, Naufal Ihza Revandhika, Agatha Putri Adwitya, Bada Maulidia Putri, Anas Mufid Nurrochman

Intelligence Robot Laboratory, Robotic Center Institut Teknologi Sepuluh nopember
Campus Institut Teknologi Sepuluh Nopember, Jalan Raya ITS Sukolilo
Surabaya, Jawa Timur 60111, Indonesia
muhtadin.s@gmail.com

Abstrak—. Paper ini menjelaskan mengenai desain pada Robot ICHIRO. Robot ICHIRO menggunakan platform robot Darwin yang telah mengalami banyak modifikasi mengenai hardware dan software mengikuti perkembangan aturan permainan dari robocup. Dengan mengimplementasikan beberapa metode pola trajectory planning dan optimasi algoritma, robot dapat berjalan di rumput sintetis dengan kemampuan kecepatan berjalan sebesar 28 cm/sec. Pada kemampuan vision, dilakukan optimasi algoritma sehingga robot dapat mendeteksi dan membedakan antara bola dan gawang yang memiliki warna yang sama.

Kata Kunci— ***Communication; Field line; Framework; Goalpost; Grid; ICHIRO Team; Self localization***

I. PENDAHULUAN

ICHIRO adalah salah satu tim Robot di ITS yang memiliki riset terhadap robot humanoid sepak bola. Tim ICHIRO dibentuk sejak tahun 2012 yang terdiri atas mahasiswa sarjana. Robot yang digunakan pada awal terbentuknya tim ini menggunakan platform robot Darwin. Platform Robot Darwin telah mengalami banyak modifikasi setiap tahunnya mengenai mekanik, elektronik dan program[3]. Setiap tahun, tim ICHIRO mengikuti Kontes Robot Sepak Bola Indonesia yang merupakan pertandingan sepak bola antar robot yang mengadopsi pertandingan Robocup. Tim ICHIRO selalu mendapatkan peringkat dua di Indonesia. Pada tahun 2016, tim mendaftarkan pada Robocup dan lolos dalam babak kualifikasi. Namun karena ada permasalahan dana dan waktu yang sangat dekat dalam persiapan robot dalam mengikuti KRSBI dan Robocup, maka tim memutuskan untuk mengundurkan diri pada Robocup. Pada tahun ini, tim telah melakukan pengembangan robot mengenai keseimbangan robot berjalan pada rumput sinteis, optimasi pendeteksian fitur objek gawang dan bola dan lokalisasi. Selain itu, tim melakukan pengembangan pembuatan platform robot baru, yaitu ICHIRO-2 yang memiliki kemampuan mekanik dan elektronik yang lebih tinggi daripada robot ICHIRO-1. Pada paper ini akan membahas mengenai hardware dan software pada Robot ICHIRO yang saat ini telah dikerjakan untuk persiapan dalam mengikuti KRI Nasional.

II. PENELITIAN SEBELUMNYA

Tim ICHIRO menggunakan platform robot Darwin untuk mengembangkan riset permainan robot sepak bola humanoid. Pada tahun 2014, robot dimodifikasi pada bagian kamera menggunakan kamera Logitech C920 dan penambahan sensor magnet untuk menentukan orientasi arah daerah musuh dan daerah sendiri. Pada pendeteksian bola, dilakukan pendeteksian dengan menggabungkan fitur dari warna lapangan untuk membedakan antara bola atau noise. Jika didapatkan warna yang sama dengan warna bola namun berada di luar lapangan, maka hal tersebut tidak akan dianggap bola, sedangkan jika bola berada disekitar warna hijau lapangan, maka hal tersebut akan dianggap bola. Selain itu, ditambahkan algoritma lokalisasi dengan mengenali fitur- fitur yang ada dalam lapangan berupa bola, gawang dan garis. Dengan mengetahui fitur tersebut, robot akan dapat mengetahui posisi dirinya dalam lapangan pertandingan[2]. Metode untuk mendapatkan posisi x dan posisi y robot menggunakan metode triangulasi sedangkan orientasi diperoleh menggunakan sensor magnet.

Pada tahun 2015 dilakukan penambahan algoritma kerjasama robot melalui koordinasi antar robot. Komunikasi untuk melakukan kerjasama antar robot menggunakan protokol UDP dengan membagi informasi state robot, orientasi dan pengetahuan tentang bola. Tipe komunikasi broadcast menggunakan UDP karena tidak adanya handshaking untuk memulai koneksi komunikasi sehingga memiliki kemampuan komunikasi yang cepat. Pengiriman data informasi antar robot dilakukan setiap satu detik dengan pertimbangan kondisi respon robot dan beban jaringan. Hasil yang didapat dari strategi kerjasama antar robot adalah robot yang tidak berada di sekitar area penglihatan bola akan secara pasti mengetahui posisi bola jika salah satu robot lain mengetahui bola tersebut[4]. Robot lain akan membagikan informasi lokasi posisi bola sehingga posisi bola akan diketahui oleh robot lain. Untuk mempermudah dalam melakukan kerjasama antar robot, maka lokalisasi lapangan dibagi dalam 24 grid seperti pada gambar 1[9]. Sehingga setiap robot akan saling berbagi informasi grid posisi robot, dan jika mendapatkan bola, maka obot lain bisa menuju ke posisi grid tersebut agar menjaga bola jika sewaktu-waktu bola tersebut keluar dari jangkauan robot.



Fig. 1. Pembagian grid lapangan untuk lokalisasi dalam kerjasama antar robot

III. HARDWARE

Tim ICHIRO memiliki 4 robot yang memiliki kemampuan untuk berjalan, melihat, dan bermain sepak bola. Ketiga robot yang digunakan dalam tim ini menggunakan platform robot Darwin OP yang telah mengalami banyak modifikasi[3]. Robot ini merupakan robot generasi pertama pada robot ICHIRO. Sedangkan robot lainnya merupakan generasi kedua yang disebut dengan robot ICHIRO-2 yang saat ini sedang dalam proses pengembangan. Dalam pemaparan hardware ini akan dibahas mengenai system mekanik dan system elektronik pada setiap robot.

A. Sistem Mekanik

Sistem mekanik pada robot ICHIRO-1 menggunakan platform robot Darwin OP dengan terdapat tambahan modifikasi pada bagian kamera robot dan dasar alas kaki robot. Pada bagian kamera robot menggunakan kamera Logitech C920 menggantikan kamera Logitech C905 yang merupakan kamera asli dari Darwin. Sedangkan pada dasar alas kaki robot ditambahkan 6 pole robot yang nantinya akan menyentuh pada bagian rumput sintetis. Penambahan pole robot ini membuat robot dapat berjalan dengan baik pada rumput sintetis. Sehingga pada bagian alas kaki ini tidak berbentuk datar karena akan kesulitan dalam berjalan dirumput sintetis. Dengan penambahan pole yang digunakan pada rumput sintetis ini akan mempengaruhi keseimbangan robot dalam berjalan.


Fig. 2. Robot ICHIRO-1 yang merupakan modifikasi dari robot Darwin

Pada robot ICHIRO-2 ini memiliki tinggi sebesar 84.394 cm dengan menggunakan 20 derajat kebebasan (dof) yang diantaranya 6 dof diletakkan pada kaki menggunakan servo Dynamixel MX-106T, 3 dof diletakkan pada lengan menggunakan servo Dynamixel MX-64T sedangkan 2 dof diletakkan pada kepala menggunakan servo Dynamixel MX-28T. Robot ICHIRO-2 masih dalam proses pengembangan dan akan direncanakan bermain nantinya pada KRI Nasional 2017. Desain robot ICHIRO-2 ada pada gambar 3.

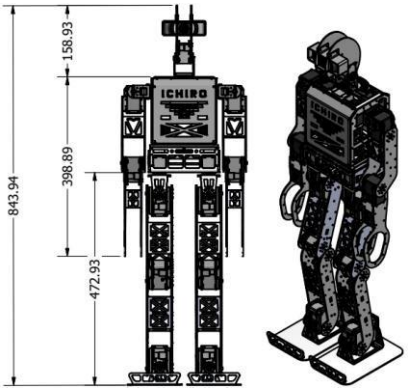


Fig. 3. Desain robot ICHIRO-2

B. Sistem Elektronik

Pada robot ICHIRO-1, proses komputasi utama robot menggunakan FitPC2i memiliki kemampuan proses 1.6GHz Intel Atom Z530 Single Core yang memiliki kapasitas RAM sebesar 1Gb. Pada bagian kontrol servo menggunakan STM32F103 ARM Cortex 32 Bit. Sensor yang digunakan pada robot ini menggunakan sensor Inertial Measurement Unit yang merupakan kombinasi dari 3 axis gyroscope, 3 axis accelerometer dan 3 axis digital kompas.

Pada robot ICHIRO-2, proses komputasi utama robot menggunakan NUC6i5SYH yang memiliki kemampuan proses 2.9GHz yang memiliki kapasitas RAM 4Gb. Pada bagian kontrol servo menggunakan STM32F103 ARM Cortex 32 Bit. Sensor yang digunakan pada robot ini sama seperti pada robot ICHIRO-1, yaitu menggunakan sensor Inertial Measurement Unit yang merupakan kombinasi dari 3 axis gyroscope, 3 axis accelerometer dan 3 axis digital kompas.

IV. PENGONTROLAN GERAKAN DAN CARA BERJALAN ROBOT

Pergerakan robot dalam robot humanoid sangat banyak, namun pada intinya gerakan dasar robot adalah kemampuan dalam berjalan, menendang dan bangun saat terjatuh. Gerakan menendang dan bangun memiliki algoritma yang berbeda jika dibandingkan dengan algoritma berjalan. Pada algoritma menendang dan bangun, *trajectory planning* dilakukan menggunakan *forward kinematics* dengan melakukan perubahan pada sudut-sudut tertentu di setiap sendi robot. Kemudian setiap perubahan disimpan dalam suatu step dan dibangkitkan setiap step tersebut dengan kecepatan dan waktu tertentu. *Trajectory planning* pada gerakan tersebut dibangkitkan dengan gerakan gaya open loop sehingga tidak ada umpan balik yang digunakan untuk memperbaiki kondisi perubahan yang tidak tepat. Hal ini dikarenakan gerakan menendang dan bangun tidak dilakukan secara gerakan terus-menerus.

Berbeda halnya dengan gerakan menendang dan gerakan bangun, pada gerakan berjalan, algoritma berjalan pada robot

ini dibangkitkan secara close loop dengan melihat kondisi robot saat pada *Single Support Phase* (SSP) dan *Double Support Phase* (DSP). *Trajectory Planning* pada gerakan berjalan dibangkitkan menggunakan inverse kinematics dengan melihat posisi keseimbangannya pada *Center of Mass* (CoM). Posisi CoM pada robot digunakan sebagai referensi untuk membangkitkan keseimbangan robot saat berjalan. Proses keseimbangan robot dibangkitkan pada gerakan kaki dan tangan robot saat melewati batas dari posisi keseimbangan robot. Pada penggunaan rumput sintetis ini, dapat dilihat karakteristik posisi CoM yang berubah kondisinya dikarenakan alas yang tidak rata. Hal ini dapat dijadikan sebagai referensi untuk optimasi mendapatkan gaya gerakan berjalan baru pada rumput sintetis. Melalui optimasi algoritma tersebut, robot dapat stabil berjalan pada rumput sintetis dan memiliki kecepatan berjalan sebesar 20 cm/detik.

V. PENGLIHATAN

Kemampuan vision menjadi hal utama dalam permainan sepak bola robot. Hal ini dikarenakan pada tahun ini, warna antara bola, gawang dan garis memiliki warna yang sama, yaitu berwarna putih[1]. Berbagai metode telah dicoba dan dioptimasi agar dapat membedakan antara bola, gawang dan garis.

Pada pengenalan bola, fitur pendeteksian bola menggunakan metode *Histogram of Oriented Gradient* (HOG). Karakteristik fitur HOG dari sebuah bola ditunjukkan oleh distribusi gradient yang berupa garis vector. Hasil fitur HOG di lakukan proses learning menggunakan *Support Vector Machine* (SVM) untuk menghasilkan sebuah model yang digunakan sebagai acuan deteksi. Sehingga proses pengenalan bola akan didapatkan[6].

Pada pengenalan gawang, karena warna dan bentuk gawang memiliki konstruksi yang sama dengan garis, maka fitur pengenalan gawang ditambahkan dengan warna dari lapangan. Dalam hal ini, gawang yang dideteksi akan selalu berada tegak lurus dengan garis gawang. Ketika garis gawang dan gawang saling bersinggungan akan membentuk huruf T yang berbentuk terbalik, lalu akan ditarik garis keatas sampai batas warna dari lapangan. Jika didapatkan data garis tersebut mencapai garis terluar dari warna lapangan, maka hal tersebut akan dijadikan sebagai acuan data gawang. Sehingga dalam hal ini, proses pengenalan gawang diawali dengan fitur pengenalan garis. Pengenalan garis menggunakan metode hough transform yang kemudian didapatkan dua perpotongan garis, garis tersebut kemudian dicari nilai posisi tengahnya menggunakan persamaan dibawah ini.

$$(P_x, P_y) = \left(\frac{(x_1y_2 - y_1x_2)(x_3 - x_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}, \frac{(x_1y_2 - y_1x_2)(y_3 - y_4) - (y_1 - y_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)} \right)$$

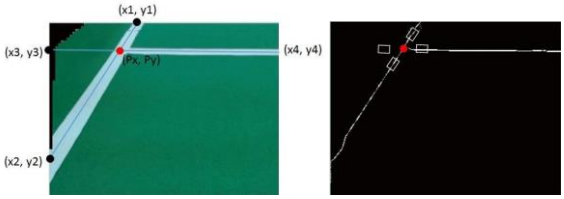


Fig. 3. Deteksi perpotongan garis dan pengenalan fitur

Ketika titik tengah dari perpotongan garis ditemukan, maka dilakukan pengecekan pada setiap sisi pada garis tengah. Jika didapatkan 2 titik, maka garis tersebut dikenal sebagai fitur garis L. Jika terlihat 3 titik, maka garis tersebut dikenal sebagai fitur garis T. Perpotongan garis L selalu berada pada ujung lapangan, sedangkan garis T berada pada tengah lapangan. Hal ini dapat dijadikan sebagai acuan untuk pengenalan posisi lokalisasi di setiap robot. Ketika posisi lokasi robot diketahui menggunakan fitur garis tersebut, maka posisi gawang dapat diprediksi[4].

VI. KESIMPULAN

Hasil pengujian pengembangan robot ICHIRO pada tahun ini meliputi perancangan pengembangan walking dan vision robot. Pada pengembangan walking, robot dapat berjalan secara stabil pada rumput sintetis dengan mengubah gaya trajectory planning dengan. Hasil pengujian berjalan pada rumput sintetis didapatkan kecepatan robot sebesar 20 cm/detik. Perancangan vision meliputi pendeteksian bola, gawang dan garis yang memiliki warna yang sama yaitu warna putih. Pendeteksian bola menggunakan metode HOG berhasil dideteksi, sedangkan pada deteksi gawang menggunakan metode deteksi berdasarkan posisi garis dan warna lapangan. Perancangan hardware saat ini sedang dalam tahap pengembangan robot ICHIRO-2 dan akan direncanakan untuk ikut bermain pada KRI Nasional 2017

REFERENSI

[1] RoboCup Soccer Humanoid League Rules and Setup, https://www.robocuphumanoid.org/

[2] Dahlan, A.H. Feature Detection and Humanoid Soccer Robot Localization in RoboCup Environment Based on Orientation Sensor and Non-unique Landmark. 2nd Symposium on Robot Soccer Competition. Yogyakarta, Indonesia (2014)

[3] Muhammad Arifin, Muhammad Ardi Pradana, Uti Solichah. ICHIRO Kid-Size 2016 Team Description Paper. Intelligence Robot Laboratory ITS, Surabaya. (2016)

[4] Budiono I. Collaborative Humanoid Soccer League on Darwin-OP Platform Based on Rule of KRSBI 2015. ITS, Surabaya. (2015)

[5] R. . Fabre, H. Gimbert, L. Gondry. Rhoban Football Club Team – Description Paper Humanoid KidSize League, Robocup 2016 Leipzig. Robocup 2016, Leipzig. (2016)

[6] Hafez Farazi, Phillip Allgeuer, and Sven Behnke. A monocular vision system for playing soccer in low color information environtmens. In Proceedings of 10th workshop on Humanoid Soccer Robots, IEEE-RAS Int. Conference on Humanoid Robots, Seoul, Korea. (2015)

[7] Reinhard Gerndt, Daniel Seifert, Jacky Hansjoerg Baltes, Soroush Sadeghnejad, and Sven Behnke. Humanoid robots in soccer: Robots versus humans in RoboCup 2050. IEEE Robotics and Automation Magazine, 22(3):147-154, 2015.(2015)

[8] Marcell Missura and Sven Behnke. Balanced walking with capture steps. In Robocup 2014: Robot World Cup XVIII, volume 8992 of Lecture Notes in Computer Science, pages 3-15, Springer. (2014)

[9] Kohlbrecher A. Stumpf, and O. v. Stryk. Grid-Based Occupancy Mapping and Automatic Gaze Control for Soccer Playing Humanoid Robots. Proc. 6th Workshop on Humonoid Soccer Robots at the2011 IEEE-RAS Int.Conf. on Humanoid Robots. (2011)

[10] A. Milanovic, S. Perfomance of udp and tcp communication on personal computers. *Electrotechnical Conference 2000. MELECON 2000. 10th Mediterranean*, vol. 1, pp.286- 289 vol.1. (2000)

Rancang Bangun Robot DEVATA NEW GENERATION 2017

Anthony Susanto*, Januar Chayadi, Danieal, Agus Budi Dharmawan
Universitas Tarumanagara
Jakarta, Indonesia
anthonysusanto27@yahoo.com

Abstrak – Para peserta KRSBI-H 2017 senantiasa dituntut untuk dapat mengembangkan kemampuan robotnya dalam mekanika, elektronika, hingga algoritma seiring dengan perkembangan teknologi yang diterapkan dalam pertandingan. Rancangan perangkat mekanik dan pemilihan perangkat sistem komunikasi akan mempengaruhi kemampuan robot dalam melakukan setiap algoritma dan perintah yang dihasilkan. Pada makalah ini, akan dibahas rancangan bangun robot DEVATA NEW GENERATION (DNG) 2017 yang merupakan pengembangan dari hasil rancangan robot DEVATA A. Tinggi DNG lebih besar dengan perkalian 1,4 dari tinggi robot DEVATA A yang akan mempengaruhi setiap titik gerak robot tersebut. Untuk pemilihan perangkat elektronik dan sistem komunikasi data pada robot DNG digunakan *main controller* Raspberry Pi 3 model B, dan *sub controller* CM-530 dan Arduino Nano dalam robot DNG. Teknik komunikasi data yang digunakan dalam robot DNG adalah komunikasi serial, komunikasi paralel 6 bit, dan komunikasi paralel 9 bit. Berdasarkan hasil pengujian yang dilakukan, sistem komunikasi data yang dipilih pada robot DNG dapat berfungsi sesuai dengan perintah-perintah yang dihasilkan dari setiap proses yang dilakukan.

Kata Kunci – Arduino Nano; CM-530; DEVATA NEW GENERATION; *Gyroscope and Accelerometer*; KRSBI-H 2017; Raspberry Pi 3 Model B; Sistem Komunikasi Data

I. PENDAHULUAN

Kontes Robot Sepak Bola Indonesia Humanoid (KRSBI-H) 2017 adalah penyelenggaraan KRSBI-H tahun kelima semenjak KRSBI-H diadakan pertama kali di tahun 2013. Cikal bakal KRSBI-H adalah Kontes Robot Cerdas Indonesia (KRCI) *Robo Soccer Humanoid League* (RSHL) tahun 2012 yang sebelumnya bernama KRCI *Expert Battle*[1]. KRSBI adalah sebuah kontes robot bermain sepak bola layaknya manusia. Para peserta KRSBI-H senantiasa dituntut untuk dapat mengembangi kemampuan robotnya dari tahun ke tahun. Seperti pada tahun ini, terdapat perubahan ukuran bola dari bola tenis menjadi bola berstandar FIFA dengan ukuran size 1, serta perubahan ketebalan rumput sintetis yang digunakan pada pertandingan.

Universitas Tarumanagara (UNTAR) kembali berpartisipasi dalam Kontes Robot Sepak Bola Indonesia Humanoid (KRSBI-H) 2017. Dalam kontes tahun ini, Tim KRSBI-H UNTAR merancang robot sebanyak 4 buah yang terdiri dari DEVATA NEW GENERATION sebanyak 1 buah sebagai robot penyerang, DEVATA A sebanyak 1 buah sebagai robot penyerang DEVATA A1 sebanyak 1 buah sebagai robot bertahan dan DEVATA C sebagai robot penjaga gawang.

Robot DEVATA NEW GENERATION (DNG) adalah robot *humanoid* sepak bola generasi terbaru yang dirancang oleh tim KRSBI UNTAR 2017. Robot ini memiliki perangkat keras (*hardware*) dan perangkat lunak (*software*) yang disatukan menjadi sebuah sistem untuk bekerja selayaknya sebuah robot sepak bola yang berperan sebagai penyerang.

Pemilihan sistem komunikasi data yang tepat sangatlah berpengaruh pada kemampuan robot yang dapat dihasilkan[2]. Hal ini sangatlah penting untuk diperhatikan karena dengan ketepatan pemilihan sistem komunikasi data, maka data yang akan dikirimkan antar kontroller dapat berfungsi dengan baik.

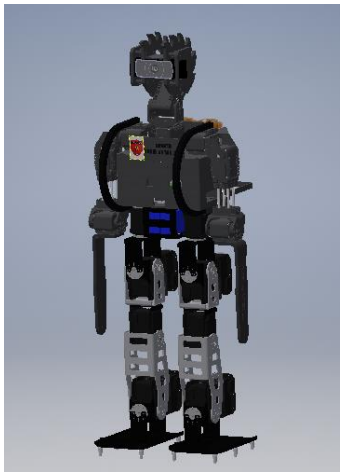
Pada makalah ini akan dibahas tentang perancangan mekanik kerangka robot DNG serta sistem komunikasi data yang berpengaruh dalam operasi robot tersebut saat dijalankan.

II. TINJAUAN PUSTAKA

DEVATA NEW GENERATION (DNG) merupakan robot *humanoid* dengan 20 DOF (*Degree of Freedom*) yang dirancangan untuk menjadi penyerang pada KRSBI-H 2017. DNG adalah robot pengembangan dari hasil rancangan robot DEVATA A[2]. Tinggi DNG lebih tinggi dengan perkalian 1,4 dari tinggi robot DEVATA A. Untuk manufaktur robot DNG, digunakan alumunium pada bagian kaki, dan *3D printing* dengan *filament* jenis HIPs (*High Impact Polystrene*) pada bagian badan dan kepala. Robot DNG memiliki spesifikasi sebagai berikut :

Aktuator	:	Kepala	: 2 Dynamixel AX-12A
		Tangan	: 2x3 Dynamixel AX-18A
		Kaki	: 2x6 Dynamixel MX-64
CPU	:	*Main controller*	: Raspberry Pi 3 Model B
		Sub Controller	: CM-530, Arduino Nano
Sensor	:	*Camera*	: Logitech C920
		Gyro,Accelero & Compass	: MPU9150
Power	:	*Battery* LiPo 3 *Cell* 2200mAh	

Dimensi : Tinggi : 63 mm
Telapak Kaki : 60 mm x 100 mm
Tangan : 197 mm
Lebar Tubuh : 195 mm



Gambar 1. Desain mekanik robot

Raspberry Pi 3 Model B adalah komputer yang memiliki ukuran sebesar kartu kredit dengan harga yang cukup terjangkau[3]. Raspberry Pi 3 Model B digunakan sebagai *main controller*. *Main controller* berfungsi sebagai pusat pengolah data dari *sub controller*, membaca dan mengolah citra yang diperoleh dari sensor kamera, serta menghasilan perintah untuk melakukan gerakkan badan robot dan kepala. Untuk melakukan pengolahan citra, digunakan Open CV yang beroperasi pada bahasa pemograman C++.



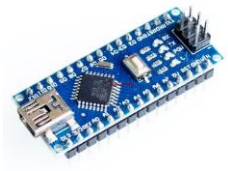
Gambar 2. Raspberry Pi 3 model B

CM-530 adalah suatu unit pengendali buatan ROBOTIS berbasis ARM 32-bit STM32F103RE[4]. CM-530 dapat digunakan untuk mengendalikan servo Dynamixel sebagai aktuator robot. CM-530 dapat mengendalikan servo AX, dan MX *series*. Sebagai *sub controller*, CM-530 akan menerima data dari *main controller* berupa perintah gerakan yang harus dilakukan robot.



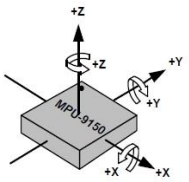
Gambar 3. CM-530

Arduino Nano adalah sebuah papan pengembangan *microcontroller* yang berukuran kecil. Arduino Nano yang digunakan adalah Arduino Nano berbasis Atmega328[5]. Arduino Nano yang akan digunakan berjumlah 2 buah dengan fungsi yang berbeda pada setiap perangkat tersebut. Kedua Arduino Nano tersebut berperan sebagai *sub controller* pada sistem tersebut. Arduino Nano 1 berfungsi untuk membaca sensor *gyroscope, accellerometer and compass* MPU9150 dan mengirimkannya kepada *main controller*, serta mengolah data sensor MPU9150 dan mengatur pergerakan servo tangan robot agar dapat menyeimbangkan badan secara mandiri. Sedangkan Arduino nano 2 berfungsi mengatur pergerakkan servo kepala sesuai dengan perintah yang diperoleh dari *main controller*



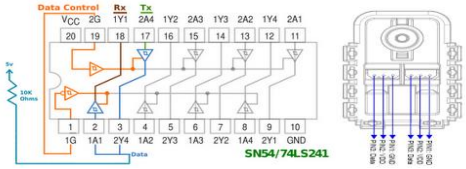
Gambar 4. Arduino Nano

Sensor *gyroscope* dan *accelerometer* yang digunakan terintegrasi dalam sensor MPU9150. Sensor ini merupakan kombinasi dari sensor MPU9150 yang memiliki *3-axis gyroscope, 3-axis accelerometer,* dan sensor *AK8975* yang memiliki *3-axis digital compass*[6]. Pembacaan sensor ini akan akan dikonversi ke dalam tiga sudut rotasi yaitu *pitch* (y), *roll (*x), dan *yaw* (z). Sudut rotasi tersebut dapat dilihat pada Gambar 5. Sensor ini digunakan untuk mendeteksi kondisi jatuh robot, dan arah sudut hadap robot dengan memperoleh sudut 0^o – 359^o. Nilai yang digunakan untuk mengetahui kondisi jatuh robot diperoleh dari kombinasi sudut rotasi *pitch* dan *roll*. Sedangkan sudut rotasi yaw digunakan untuk memperoleh arah sudut hadap robot.



Gambar 5. Kerangka rotasi dari sensor MPU9150

Untuk menggerakan servo Dynamixel kepala dan tangan, dibutuhkan IC 74LS241 untuk merubah komunikasi arduino secara *full duplex* ke half *duplex* sesuai cara komunikasi servo Dynamixel AX-12A[7]. Skema rangkaian dapat dilihat pada gambar 6.



Gambar 6. Skema rangkaian 74LS241 dan AX-12A

III. PERANCANGAN

Rancangan Robot DNG terbagi menjadi 2 bagian utama, yaitu bagian kerangka mekanik, dan sistem komunikasi data.

A. Kerangka Mekanik

Rancangan kerangka mekanik robot DNG mengacu pada konsep robot DARwIn-OP dan hasil rancangan robot DEVATA A. Tinggi DNG lebih tinggi dengan perkalian 1,4 dari tinggi robot DEVATA A. Sehingga kerangka dari robot tersebut akan menjadi lebih besar daripada rancangan DEVATA A. Akan tetapi, semua bagian kerangka pada robot DEVATA A tidak dapat dilakukan perkalian secara langsung dengan 1,4 agar dapat memperoleh kerangka baru robot DNG. Akan tetapi, nilai jarak antar titik gerak robot yang dilakukan perkalian dengan 1,4. Selain itu juga diperhatikan berbagai aturan yang terdapat pada peraturan pengukuran dimensi yang mengacu pada aturan Robocup, seperti ukuran panjang kaki, panjang tangan, dll. Hal ini dilakukan untuk menjaga keproposionalan dari setiap bagian robot DNG.

Sebagai contoh, pada robot DEVATA A memiliki tinggi 45cm. Maka tinggi pada robot DNG (Htop) adalah

$$Htop = (1{,}4 \times 45cm) = 63cm \quad (1)$$

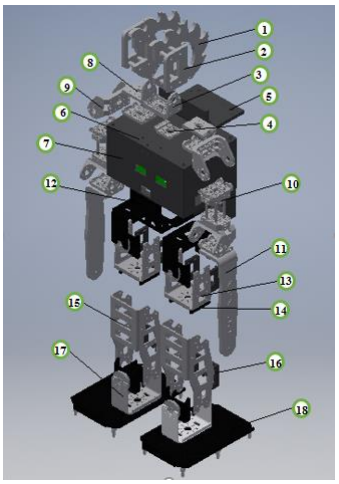
Untuk memperoleh jarak antar titik gerak pada betis (Jb) ataupun paha (Jp), maka digunanakan aturan pengukuran panjang kaki (0,5 x Htop) sehingga memperoleh persamaan sebagai berikut

$$Jb = [(0{,}5 \times Htop) - \text{dengan part yang lain}] / 2 \quad (2)$$

$$Jb = [(0{,}5 \times 63cm) - 9{,}9cm)] / 2 = 10{,}8cm \quad (3)$$

Sehingga diperoleh hasil jarak antar titik gerak pada betis adalah 10,8 cm. Nilai ini juga berlaku pada jarak antar titik gerak pada paha.

Aktuator yang digunakan pada robot DNG adalah 2x6 buah servo DYNAMIXEL MX-64 pada bagian kaki, 2x3 buah servo DYNAMIXEL AX-18A pada bagian tangan, dan 2 buah servo DYNAMIXEL AX-12A pada bagian kepala. Dikarenakan terdapat perbedaan pemilihan aktuator dengan robot DEVATA A, maka setiap ukuran kerangka mekanik harus diperhatikan kembali. Detail kerangka robot DNG dapat dilihat pada gambar 7 dan tabel I.



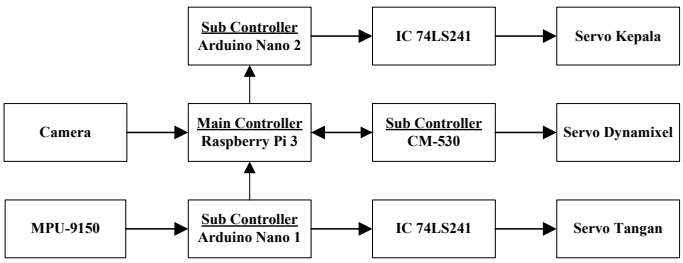
Gambar 7. Hasil rancangan kerangka robot DNG

TABEL I. KERANGKA ROBOT DNG

Nomor Kerangka	Nama Kerangka
1	Kerangka Kepala
2	Penahan Kamera
3	Bioloid FP04-F2
4	Bioloid FP04-F53
5	Tempat CM
6	Tutup Badan
7	Badan
8	Penghubung Dada
9	Bioloid FP04-F1
10	Bioloid FP04 - F10
11	Tangan
12	Tempat Baterai
13	Paha
14	Penghubung Paha
15	Betis
16	Penghubung Servo
17	Engkel
18	Alas Kaki

B. Sistem Komunikasi Data

Rancangan sistem pada robot DEVATA A dapat dilihat Pada gambar 8.



Gambar 8. Diagram alir komunikasi data pada robot DNG

Pada gambar 8, terdapat beberapa komunikasi data yang dapat dilihat, di antaranya :

1. Arduino Nano 1 ke *Main Controller*

Data yang dikirmkan dari Arduino Nano ke *Main controller* adalah data pembacaan sensor MPU-9150. Jenis pengiriman data yang digunakan adalah pengiriman data secara parallel 9 bit. Sehingga terdapat 2^9 = 512 kemungkinan kombinasi data yang diperoleh yaitu dari rentang 0 s/d 511. Data yang dikirimkan adalah hasil arah hadap robot pada rentang 0^o s/d 359^o, serta kondisi jatuh robot yang akan direpresntasi sesuai dengan tabel II.

TABEL II. KONDISI JATUH ROBOT

Kondisi Jatuh	*Pitch*	*Roll*	Data
Jatuh Depan	> 50		400
Jatuh Belakang	< -50		401
Jatuh Kanan		> 50	402
Jatuh Kiri		< -50	403

2. *Main Controller* ke Arduino Nano 2

Data yang dikirimkan dari *Main controller* ke Arduino Nano adalah data perintah untuk menggerakan servo pada kepala robot. Teknik pengiriman data yang digunakan dalam jenis ini adalah komunikasi serial. Pin UART pada *main controller* (Raspberry Pi) dihubungkan secara langsung ke pin 9 dan 10 pada Arduino Nano. Pada Arduino Nano teknik *serial* yang digunakan adalah *software serial*. Yaitu teknik pengiriman dan penerimaan data serial menggunakan pin biasa bukan pin UART. Digunakan *software serial* karena pin *serial* pada Arduino Nano digunakan untuk mengendalikan servo Dynamixel pada kepala robot.

3. Arduino Nano1 ke Servo Tangan

Data yang dikirimkan antara Arduino Nano dan servo Dynamixel adalah data perintah untuk menggerakkan servo. Teknik pengiriman data yang digunakan dalam jenis ini adalah komunikasi *serial*. Servo yang digerakkan pada servo tangan yang akan digunakan untuk *balancing* .

4. Arduino Nano2 ke Servo Kepala

Data yang dikirimkan antara Arduino Nano dan servo Dynamixel adalah data perintah untuk menggerakkan servo. Teknik pengiriman data yang digunakan dalam jenis ini adalah komunikasi *serial*. Servo yang digerakkan pada kepala robot adalah sebanyak 2 buah yang mewakili *pan* dan *tilt* kepala.

5. *Main Controller* ke CM-530

Data yang dikirimkan dari *Main controller* ke CM-530 adalah data perintah untuk menggerakkan badan robot. Teknik pengiriman data yang digunakan dalam jenis ini adalah komunikasi paralel 6 bit. Sehingga terdapat 2^6 = 64 kemungkinan kombinasi data yang diperoleh yaitu dari rentang 0 s/d 63. Dengan demikian, semua gerakan badan robot harus direpresentasikan ke dalam bentuk variabel motion. Data gerakan tersebut direpresentasikan sebagai berikut:

TABEL III. JENIS GERAKAN BADAN ROBOT DEVATA A

Motion	**Jenis Gerakan**
0	Diam
1	Jalan di tempat
2	Maju
3	Putar Kiri
4	Putar Kanan
9	Bangun Depan
10	Bangun Belakang
11	Tendang

IV. HASIL DAN PENGUJIAN

Berikut adalah robot DNG yang digunakan untuk mengikuti KRSBI 2017. Robot DEVATA A dapat dilihat sesuai dengan Gambar 9.



Gambar 9. Hasil realisasi robot

Untuk melakukan pengujian terhadap hasil setiap rancangan yang dilakukan, maka dilakukan pegujian pada kerangka mekanik dan sistem komunikasi data.

A. Kerangka Mekanik

Pada bagian kerangka mekanik, dilakukan pengujian terhadap perbandingan pengukuran antara rancangan berupa desain dengan realisasi rancangan yang telah diselesaikan pada robot DNG. Hasil pengukuran dapat dilihat pada tabel IV.

TABEL IV. HASIL PENGUKURAN DIMENSI ROBOT

Deskripsi	**Desain (cm)**	**Realisasi (cm)**
Tinggi Robot	63.5	64
Hcom/ Center of grafity	29.32	30
Luas Telapak Kaki	122.17	120.56
Lebar tubuh (tangan tidak mengembang)	25.94	25

Lebar tubuh (tangan mengembang)	78.16	78.3

Dapat dilihat bahwa tidak terdapat perbedaan yang cukup signifikan antar pengukuran dimensi pada desain dan realisasi rancangan.

B. Sistem Komunikasi Data

Untuk melakukan pengujian sistem komunikasi data robot DEVATA A, dilakukan beberapa percobaan sebagai berikut :

1. Arduino Nano 1 ke *Main controller*

Pengujian dilakukan dengan pengiriman data acak dari Arduino Nano 1 ke *Main controller* sebanyak 5 kali percobaan. Hasil pengujian dapat dilihat pada tabel V.

TABEL V. PENGUJIAN SISTEM ARDUINO NANO KE *MAIN CONTROLLER*

Percobaan ke	Data Dikirm	Data diterima	Kesimpulan
1	11	11	Berhasil
2	506	506	Berhasil
3	321	321	Berhasil
4	211	211	Berhasil
5	415	415	Berhasil

2. *Main controller* ke Arduino Nano 2

Pengujian dilakukan dengan pengiriman data acak dari *Main controller* ke Arduino 2 sebanyak 5 kali percobaan. Hasil pengujian dapat dilihat pada tabel VI.

TABEL VI. PENGUJIAN SISTEM *MAIN CONTROLLER* KE ARDUINO NANO 2

Percobaan ke	Data Dikirm	Data diterima	Kesimpulan
1	A	A	Berhasil
2	C	C	Berhasil
3	9	9	Berhasil
4	#	#	Berhasil
5	!	!	Berhasil

3. *Main controller* ke CM-530

Pengujian dilakukan dengan pengiriman data acak dari *Main controller* ke Arduino 2 sebanyak 5 kali percobaan. Hasil pengujian dapat dilihat pada tabel VII.

TABEL VII. PENGUJIAN SISTEM *MAIN CONTROLLER* KE CM-530

Percobaan ke	Data Dikirm	Data diterima	Kesimpulan
1	3	3	Berhasil
2	10	10	Berhasil
3	15	15	Berhasil
4	12	12	Berhasil
5	9	9	Berhasil

4. Pengujian keseluruhan sistem komunikasi

Pengujian keseluruhan sistem robot DNG dilakukan dengan memerintahkan robot untuk berjalan, mencari bola, serta menendang bola. Hal ini dilakukan untuk menguji seluruh sistem komunikasi yang berjalan pada robot. Hasil pengujian keseluruhan sistem dapat dilihat pada gambar 10.



(a) (b) (c)

Gambar 10. Pengujian seluruh sistem robot DNG (a) menghampiri bola, (b) persiapan menendang bola, (c) bola telah ditendang

V. KESIMPULAN

Untuk mendapat kerangka mekanik yang lebih besar, tidak dapat dilakukan perbandingan secara langsung, akan tetapi harus dilakukan perbandingan pada setiap jarak antar titik gerak setiap robot. Hal ini bertujuan agar tetap menjaga keproporsional bentuk hasil rancangan robot DNG.

Berdasarkan hasil percobaan, pemilihan sistem komunikasi data secara *serial* ,*paralel* 6 bit ,dan *paralel* 9 bit sangat cocok digunakan untuk komunikasi data antar *controller* yang digunakan robot DNG. *Controller* yang digunakan robot DEVATA DNG, yaitu Raspberry Pi 3 Model B, CM-530, dan Arduino Nano sebanyak 2 buah.

REFERENSI

[1] Buku Panduan KRSBI Humanoid 2017 versi 21.12.16. Direktorat Kemahasiswaan, Direktorat Jenderal Pembelajaran dan Kemahasiswaan, Kementerian Riset, Teknologi dan Pendidikan Tinggi. Jakarta. Indonesia. 2016. pp. 1-19

[2] Danieal, J. Chayadi, K. William, dan A.B. Dharmawan. "Rancang bangun sistem komunikasi data robot DEVATA A dalam KRSBI 2016", The 4th Indonesian Symp. on Robot Soccer Competition, PENS, Surabaya, Juni 2016. pp. 119-125

[3] Raspberry Pi Foundation. (2016). What is a raspberry pi?[online]. Available: https://www.raspberrypi.org/

[4] ROBOTIS. (2010). ROBOTIS e-manual v.1.29.00 CM-530[online]. Avaiable: http://support.robotis.com/en/product/controller/cm530.htm

[5] Arduino. (2017). Arduino Nano[online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardNano

[6] MPU-9150 Product Spesification Revision 4.3. InvenSense Inc. California. USA. 2013. pp. 6-10

[7] Y.Y. Joefrie, A. Yuliansyah, W.R. Nugraha, Felisyia, and R.A.Silva, "Sistem pergerakan servo dynamixel AX series berbasis Arduino pada robot NAMAEKA", in The 3rd Indonesian Symp. on Robot Soccer Competition, UMY, Yogyakarta, Juni 2015. pp. 130-133

Implementasi Pusat Tekanan (CoP) Untuk Kontrol Keseimbangan Postur Pada Robot Humanoid

Muhammad Arifin[1], Djoko Purwanto[2], Muhtadin[3], Anas Mufid Nurrochman[4]
Departemen Teknik Elektro, Fakultas Teknologi Elektro[1,2,4]
Departemen Teknik Komputer, Fakultas Teknologi Elektro[3]
Institut Teknologi Sepuluh Nopember
Jl. Arief Rahman Hakim, Surabaya 60111
muh.arifin.300cc@gmail.com

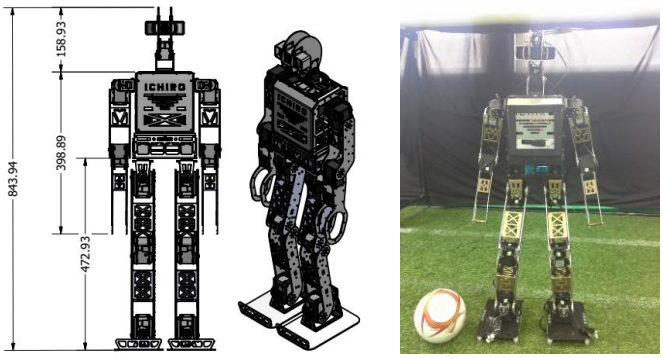
Abstrak—Dalam paper ini akan membahas mengenai perancangan dan implementasi pusat tekanan untuk mengontrol keseimbangan postur pada robot humanoid. Robot humanoid yang dirancang memiliki jumlah derajat kebebasan sebanyak 20 buah dan ukuran tinggi robot sebesar 843.94 mm. Hal dasar pada robot setelah dilakukan perancangan adalah pada postur berdiri robot dan keseimbangan robot. Postur berdiri robot yang tidak tepat akan menyebabkan permasalahan pada perencanaan gerakan selanjutnya. Dalam mencari postur berdiri dan keseimbangan yang tepat dapat dilakukan dengan mengetahui nilai dari Zero Moment Point (ZMP) pada robot. Salah satu cara untuk mencari ZMP adalah dengan mencari pusat tekanan (CoP) dengan menempatkan sensor tekanan pada alas kaki robot. Data informasi tersebut digunakan untuk menentukan postur robot dan kontrol keseimbangan robot. Pengujian dalam penelitian ini untuk mengetahui respon robot dalam mempertahankan postur dari lingkungan sekitar. Pada pembentukan awal postur berdiri robot didapatkan data posisi pusat tekanan berada di koordinat (-56,24). Dengan adanya penambahan θpengaturan pada pusat tekanan, didapatkan postur baru dengan posisi pusat tekanan di koordinat (-5,4) mendekati pusat robot yang seimbang. Pada pengujian kemiringan robot, robot dapat bertahan berdiri sampai pada sudut kemiringan 50°.

Kata Kunci*— keseimbangan; pusat pekanan; *robocup*; robot humanoid; *zero moment point

I. PENDAHULUAN

Seiring berkembangnya teknologi saat ini, tidak akan menutup kemungkinan bahwa semua kegiatan nantinya akan digantikan oleh semua robot baik itu dalam kegiatan industri, rumah tangga ataupun dalam hiburan. Salah satu organisasi dari Robocup memprediksikan dan mempunyai mimpi bahwa pada tahun 2050 nantinya, sepak bola robot dapat mengalahkan tim juara dunia sepak bola manusia dengan aturan permainan dari FIFA[1]. Dengan melihat tujuan tersebut, maka pihak organisasi FIFA melakukan pertandingan sepak bola robot dengan aturan yang berubah setiap tahunnya[1]. Hal tersebut menuntut robot untuk memiliki bentuk dan kemampuan seperti manusia yang dapat bermain sepak bola baik dalam strategi penyerangan maupun pertahanan secara otomatis. Lapangan pertandingan robot sepak bola saat ini mulai menggunakan rumput sintetis, sehingga robot diharuskan dapat bermain pada kondisi alas yang tidak tetap.

Hal dasar pada robot sebelum robot humanoid bermain sepak bola adalah pada kondisi postur berdiri robot. Kondisi berdiri robot yang tidak tepat pada pusat massa robot akan menyebabkan permasalahan pada perencanaan gerakan selanjutnya seperti gerakan berjalan atau gerakan menendang bola. Selain pada kondisi berdiri, robot yang tidak dilengkapi dengan keseimbangan akan mudah sekali jatuh jika berada di kondisi lantai yang tidak tetap atau di bidang miring. Untuk mengatasi hal tersebut, robot harus diperhitungkan kondisi bentuk tubuhnya agar mendapatkan jumlah gaya yang aktif bernilai nol diberbagai posisi yang dapat disebut dengan *Zero Moment Point*(ZMP). Perhitungan ZMP dapat dilakukan dengan cara memproyeksikan titik tumpu ujung bagian robot kearah alas dasar robot pada kondisi *Double Support Polygon* (DSP) yang disimplifikasi menggunakan *inverted pendulum* model [2]. Namun, hal ini masih terdapat nilai yang error karena terdapat perubahan pertambahan massa akibat dari mur, baut, kabel atau penambahan massa dari kondisi luar. Metode lain untuk mencari ZMP dapat dilakukan dengan menentukan Pusat Tekanan (CoP) dengan penambahan sensor tekanan pada alas kaki robot [3].



Gambar 1. Desain mekanik dan realisasi robot humanoid

II. DESAIN ROBOT

A. *Mekanik Robot*

Robot humanoid yang dirancang dalam hal ini memiliki jumlah dof sebanyak 20 dof, diantaranya 12 dof ada pada kaki, 6 dof di tangan, dan 2 dof di kepala. Dalam perancangan mekanik robot ini, bentuk dan fungsinya disesuaikan untuk kemampuan dalam kecerdasan bermain sepak bola. Ukuran bentuk dan konstruksi robot dari ukuran telapak kaki, panjang tangan dan luas alas kaki mengikuti aturan dari organisasi robocup. Desain robot yang dibuat menggunakan referensi dari robot Nimbro - OP yang merupakan tim robot dari Jerman. Desain robot tidak sepenuhnya sama, terdapat beberapa modifikasi pada bagian kepala, badan, tangan dan kaki. Hal ini dikarenakan keterbatasan bahan dan proses pemotongan yang tidak mampu dilakukan. Selain itu juga, robot dilakukan skala pengecilan dimana robot Nimbro-OP yang berukuran 95 cm dilakukan desain ulang sehingga mendapatkan tinggi robot sebesar 84.4 cm. Pada ukuran tersebut, robot dapat bermain pada tingkat kid-size dan teen-size pada pertandingan robocup. Bahan yang digunakan dalam robot ini menggunakan *aluminium alloy* tipe 5052 dengan ketebalan 2mm dan 3mm yang dipotong menggunakan mesin CNC laser. Proses pemotongan dan bending aluminium dilakukan dengan presisian yang rendah, sehingga part robot dapat mengakibatkan masalah tersendiri untuk keseimbangan robot. Penggerak robot dalam rancangan ini menggunakan servo Dynamixel tipe MX-106T untuk kaki, MX-64T untuk tangan, MX-28T untuk kepala. Penentuan jenis servo tersebut disesuaikan dengan kebutuhan robot dalam membutuhkan gaya yang besar dalam setiap pergerakan. Desain mekanik dan realisasi robot humanoid ada pada Gambar 1.

B. *Elektronik Robot*

Robot dilengkapi kontroler dengan menggunakan PC Intel NUC Core i5 yang memiliki kemampuan proses hingga mencapai 2.7GHz dan memiliki kapasitas RAM sebesar 4Gb serta penyimpanan menggunakan SSD 120 Gb. Kecerdasan robot dan penentuan berbagai aksi akan dilakukan pada kontroler ini, sehingga kontroler ini disebut *main controller*. Kontroler kedua yaitu CM-730 yang disebut dengan *sub controller*. CM-730 akan menerima data perintah dari main controller untuk menggerakan joint pada setiap servo robot. *Main controller* dan *sub controller* terhubung menggunakan jalur komunikasi serial sedangkan *sub controller* terhubung dengan servo menggunakan jalur komunikasi TTL Half Duplex. Maple mini terhubung kedalam *main controller* menggunakan jalur komunikasi serial. Maple mini akan membawa informasi mengenai data pusat tekanan robot yang menapak pada alas kaki robot untuk diolah pada *main controller* dalam sistem keseimbangan robot.



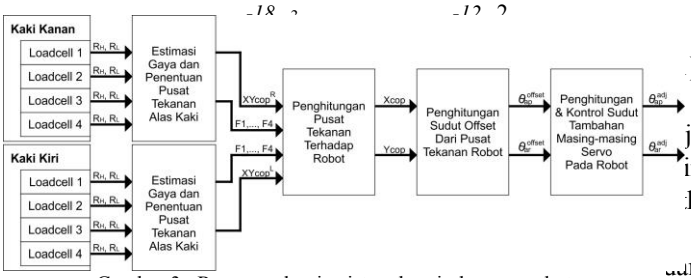
Gambar 2. Desain peletakkan sensor tekanan pada alas kaki robot

III. DESAIN SISTEM

Sensor tekanan yang diterapkan di robot humanoid menggunakan loadcell model YZC-133 dengan berat maksimal yang dapat terukur yaitu 10Kg. Loadcell tersebut akan memberikan informasi data tekanan yang memijak pada dasar robot yang kemudian diolah untuk mendapatkan pusat tekanan. Loadcell ini diletakkan pada ujung dari setiap alas kaki. Peletakan loadcell yang ada pada ujung alas kaki ini akan memberikan informasi bagaimana kondisi tekanan robot saat berdiri. Desain peletakkan sensor tekanan pada alas kaki robot ada pada Gambar 2 sedangkan desain rancangan pusat tekanan untuk pembangkitan postur dan keseimbangan ada pada gambar 3.

A. *Estimasi Gaya dan Penentuan Pusat Tekanan Alas Kaki*

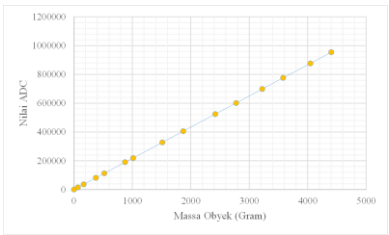
Pada estimasi gaya dan penentuan pusat tekanan alas kaki, data setiap sensor tekanan akan diolah dan didapatkan informasi pusat tekanan dalam koordinat kartesian pada setiap masing-masing kaki. Sebelum mendapatkan estimasi gaya, data keluaran dari loadcell dilakukan kalibrasi dan pengambilan data yang kemudian dibuat grafik. Hasil grafik tersebut ada pada Gambar 4. Dari grafik tersebut, dilakukan pendekatan dengan *polynomial regresi orde 3*, sehingga didapatkan persamaan 1 dengan y merupakan massa dan x adalah nilai adc yang didapatkan dari loadcell.



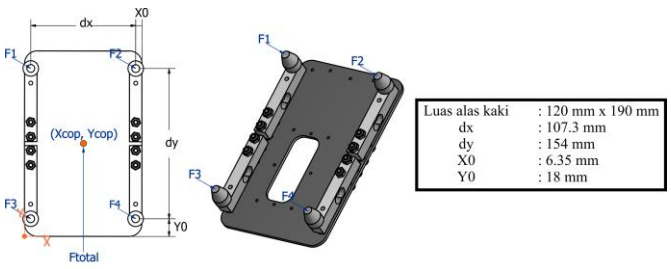
Gambar 3. Rancang desain sistem keseimbangan robot

jadi
ing-
kan
dan 3
erluar

dalam koordinat x dan Yo adalah jarak antara loadcell dengan alas kaki terluar dalam koordinat y, dx dan dy adalah jarak antar loadcell.



Gambar 4. Grafik massa objek dengan nilai adc yang didapatkan



Gambar 5. Penentuan pusat tekanan pada satu alas kaki

$$Xcop = X_0 + \frac{(F2+F4)*dx}{Ftotal} \quad (2)$$

$$Ycop = Y_0 + \frac{(F1+F2)*dy}{Ftotal} \quad (3)$$

B. Perhitungan Pusat Tekanan Robot

Pada perhitungan pusat tekanan terhadap robot, data pusat tekanan pada setiap alas kaki akan dikirim ke badan robot lalu diperhitungkan posisi pusat tekanan yang diproyeksikan pada kedua kaki robot dalam koordinat kartesian. Dari data tersebut, maka akan didapatkan persamaan 4 dan 5 dengan fw merupakan lebar kaki dan df adalah jarak antar kaki kanan dan kaki kiri.

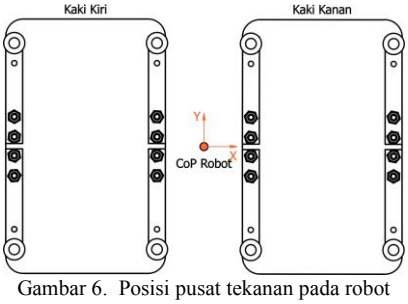
$$Xcop = (fw + \frac{df}{2}) + (Xcopl - Xcopr) \quad (4)$$

$$Ycop = (Ycopl + Ycopr)/2 \quad (5)$$

C. Penghitungan Sudut Kondisi Dari Pusat Tekanan

Posisi pusat tekanan robot dalam kondisi seimbang yaitu berada pada posisi koordinat (0, 0). Pada kondisi tersebut robot berada pada kondisi titik pusat massa dan gravitasi robot. Sehingga titik koordinat (0, 0) merupakan nilai setpoint pada robot. Gambar 6 menunjukkan letak koordinat pusat untuk pusat tekanan robot.

Saat posisi pusat tekanan robot tidak berada di posisi koordinat (0, 0), maka robot akan berada di kondisi yang tidak seimbang. Sehingga perlu dilakukan penentuan θkondisi agar robot kembali pada posisi seimbang. Untuk mendapatkan θkondisi dapat diperhitungkan dengan menggunakan kontrol PID agar respon θkondisi yang diberikan menjadi lebih halus untuk pergerakan robot. Posisi pusat tekanan yang terbaca akan dikurangi dengan koordinat (0, 0) yang merupakan setpoint pada robot, sehingga didapatkan nilai error untuk kontrol PID. Kontrol PID akan menentukan θkondisi yang kemudian akan ditambahkan dengan θasli untuk kontrol selanjutnya.



Gambar 6. Posisi pusat tekanan pada robot

D. Pembangkitan Postur Berdiri

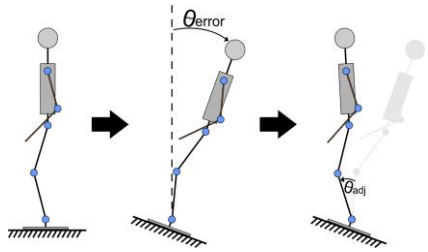
Pembuatan Pembuatan postur berdiri robot merupakan hal dasar untuk pergerakan pada robot, karena berbagai tindakan aksi dari robot akan dimulai dengan postur berdiri robot dan diakhiri dengan postur berdiri lagi. Jika postur berdiri robot yang dibuat tidak berada pada kondisi yang tepat, maka akan menyebabkan kesulitan dalam mengerjakan aksi lainnya seperti kondisi berjalan ataupun menendang dan akan mudah terjatuh. Dalam mendapatkan postur berdiri robot dapat dilakukan dengan dua tahap.

Tahap pertama yaitu pembangkitan postur berdasarkan pusat massa robot. Pembentukan postur dilakukan dengan metode *forward kinematics*. Nilai sudut pada setiap servo ditentukan nilainya hingga terbentuk posisi berdiri.

Tahap kedua yaitu pembangkitan postur dengan kontrol PID dari data posisi pusat tekanan. Sehingga akan didapatkan postur baru dengan kondisi berdiri yang seimbang yang berada pada koordinat (0, 0) untuk posisi pusat tekanan.

E. Sistem Keseimbangan Robot

Keseimbangan menjadi hal terpenting dalam robot untuk menjaga kondisi robot dalam mempertahankan posisi yang di inginkan. Robot harus mengetahui posisi tubuh robot apakah sedang dalam kondisi aman atau berada dalam kondisi akan jatuh. Jika robot tidak dapat mengetahui kondisi robot saat ini, maka saat ada gangguan gaya dari luar atau dari lantai yang miring, robot akan mudah untuk jatuh. Untuk mengetahui kondisi robot apakah sedang dalam kondisi aman atau tidak dapat diketahui dengan penggunaan sensor tekanan. Dengan menggunakan informasi data posisi pusat tekanan robot, maka akan diketahui kondisi robot apakah berada dalam kondisi normal dengan nilai posisi tekanan robot berada dalam koordinat (0, 0) atau berada dalam kondisi nilai error yang jauh, sehingga robot dapat menyeimbangkan kondisi robot untuk tidak jatuh. Nilai error yang didapatkan pada pusat tekanan akan dihitung θkondisi sehingga didapatkan θpengaturan yang merupakan penambahan dari θasli. Dikarenakan adanya perbedaan θpengaturan dengan θsebenarnya, maka akan didapatkan nilai error yang digunakan untuk kontrol PID agar didapatkan respon keseimbangan yang cepat dan mencapai *steady state*. Gambar 7 merupakan kondisi robot untuk menyeimbangkan diri saat lantai miring.



Gambar 7. Sistem keseimbangan robot saat di lantai miring

IV. HASIL PENGUJIAN

Pengujian yang dibahas dalam hal ini yaitu pengujian massa pada masing-masing loadcell untuk kaki kanan, pengujian pusat tekanan pada alas kaki dan terhadap robot, pengujian postur berdiri robot dan pengujian keseimbangan robot.

Tabel 1 menunjukkan hasil pengujian massa pada setiap loadcell di kaki kanan dengan diberikan variasi obyek dengan massa yang berbeda. Dari hasil pengujian dalam Tabel 1, didapatkan bahwa galat rata-rata yang diperoleh pada masing-masing loadcell dibawah 0.5. Hal ini menunjukkan bahwa data yang didapatkan dari pengukuran dengan data yang terukur sebenarnya memiliki akurasi yang tinggi.

Tabel 2 menunjukkan hasil pengujian pusat tekanan pada satu alas kaki kanan. Pengujian pusat tekanan pada satu alas kaki dalam hal ini untuk membandingkan data pusat tekanan yang diperoleh dengan data pusat tekanan yang didapatkan melalui perhitungan pada Persamaan 2 dan Persamaan 3. Pengujian dalam bagian ini dilakukan dengan meletakkan obyek pada masing-masing loadcell yang telah diketahui tekanannya. Dari hasil pengujian dalam Tabel 2 didapatkan bahwa galat rata-rata yang diperoleh sebesar 0.661 untuk koordinat x dan 1.276 untuk koordinat y pada alas kaki kanan. Hasil tersebut dapat disimpulkan bahwa hasil pusat tekanan memiliki tingkat akurasi yang baik.

Pengujian selanjutnya yaitu pengujian pusat tekanan dua alas kaki. Pengujian pusat tekanan pada dua alas kaki memiliki pengujian yang sama dengan pengujian pada satu alas kaki.

Tabel I. Pengujian Massa Pada Masing-Masing Loadcell Kaki Kanan

Massa Sebenarnya	Massa Terukur (gr)			
	Loadcell 1	Loadcell 2	Loadcell 3	Loadcell 4
51	51	51	51	51
99	99	98	101	99
151	150	151	153	151
206	206	206	207	206
258	257	257	259	257
502	501	500	502	500
553	553	553	553	551
994	994	995	994	992
1045	1046	1046	1044	1044
1495	1495	1494	1493	1494

Tabel II. Hasil Pengujian Pusat Tekanan Pada Kaki Kanan

Titik Uji (N)				Hasil Perhitungan		Hasil Pengujian	
F1	F2	F3	F4	x	y	x	y
0	0	0	0	0	0	0	0
0.78	0	0	0	6	172	6	171
0	0.68	0	0	114	172	113	171
0	0	0.73	0	6	18	6	17
0	0	0	0.79	114	18	113	17
0.77	0.64	0	0	55	172	55	171
0.78	0	0.78	0	6	95	6	96
0.78	0	0	0.81	61	94	60	93
0	0.65	0.65	0	60	95	60	95
0	0.65	0	0.72	114	91	113	92
0	0	0.66	0.66	60	18	60	18
0.73	0.63	0.6	0	41	125	41	124
0.72	0.66	0	0.65	76	123	74	124
0	0.64	0.67	0.61	76	69	77	70
0.68	0.65	0.63	0.6	59	98	58	98

Tabel III. Hasil Pengujian Postur Berdiri Robot

Postur Berdiri Robot	Tanpa θ Pusat Tekanan		Dengan θ Pusat Tekanan	
	Xcop	Ycop	Xcop	Ycop
	-56	24	-5	4

Namun pada kondisi ini, obyek diletakkan pada kedua alas kaki dan pengukuran pusat tekanan yang didapatkan, dibandingkan dengan perhitungan pusat tekanan menggunakan Persamaan 4 dan Persamaan 5. Obyek yang diletakkan dalam pengujian ini memiliki massa yang sama. Dari hasil pengujian, didapatkan galat rata-rata sebesar 0.733 untuk koordinat x dan 0.267 untuk koordinat y. Dari hasil pengujian tersebut, dapat disimpulkan bahwa pengujian pusat tekanan yang diperoleh melalui hasil percobaan dengan hasil perhitungan memiliki tingkat akurasi yang tinggi.

Tabel 3 menunjukkan kondisi postur berdiri robot yang menapak di lantai yang tidak beraturan. Lantai yang tidak beraturan dalam pengujian ini menggunakan rumput sintetis yang memiliki panjang rumput sebesar 30 mm sesuai standart aturan robocup 2017 saat ini. Robot diletakkan pada rumput sintetis dalam kondisi berdiri lalu dibaca data hasil pusat tekanan saat robot menapak di rumput sintetis. Dalam tabel tersebut, terlihat bahwa kondisi berdiri robot yang tanpa adanya penambahan θpengaturan dari posisi pusat tekanan akan berdiri dengan postur yang tidak seimbang, sedangkan dengan penambahan θpengaturan dari posisi pusat tekanan, robot berada di koordinat (-5,4) mendekati kondisi seimbang.

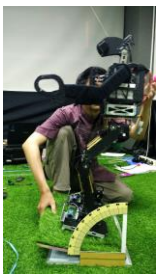
Tabel 4 menunjukkan hasil data pengujian perbandingan kondisi robot dalam postur berdiri dan lantai dimiringkan. Tujuan hal ini untuk mengetahui kemampuan berdiri dan respon robot saat dimiringkan.

Dari tabel tersebut, terlihat bahwa robot yang tidak menggunakan keseimbangan akan langsung jatuh saat dimiringkan pada sudut kemiringan 10°. Robot yang menggunakan keseimbangan IMU dapat berdiri sampai sudut kemiringan 35°, sedangkan robot yang menggunakan keseimbangan dari informasi pusat tekanan dapat berdiri

Tabel IV. Hasil Data Pengujian Keseimbangan Robot

Sudut Kemiringan	Kondisi		
	Tanpa Keseimbangan	Keseimbangan IMU	Keseimbangan Pusat Tekanan
10°	Jatuh	Tidak Jatuh	Tidak Jatuh
20°	Jatuh	Tidak Jatuh	Tidak Jatuh
30°	Jatuh	Jatuh	Tidak Jatuh
40°	Jatuh	Jatuh	Tidak Jatuh
50°	Jatuh	Jatuh	Tidak Jatuh

Tabel V. Respon Keseimbangan Robot

Tanpa Keseimbangan	Keseimbangan IMU	Keseimbangan Pusat Tekanan
		

sampai kemiringan 50°. Pada sudut kemiringan selanjutnya robot masih dapat berdiri, namun karena sudut tersebut terlalu miring mengakibatkan kaki robot tergelincir ke bawah, namun tidak jatuh ke belakang. Tabel 5 Menunjukkan respon robot saat dimiringkan dari depan.

V. KESIMPULAN

Dari hasil pengujian dalam implementasi sensor tekanan pada robot humanoid, maka didapatkan kesimpulan sebagai berikut :

1. Penggunaan loadcell sebagai sensor tekanan memiliki pembacaan akurasi tinggi dengan galat sebesar 0.133 dengan menggunakan pendekatan polynomial regresi orde 3,
2. Pembacaan pusat tekanan memiliki galat sebesar 0.733 untuk koordinat x dan 0.267 untuk koordinat y,
3. Robot dapat membentuk kondisi postur berdiri baru robot dengan memperhatikan posisi pusat tekanan hingga didapatkan posisi berdiri di koordinat (-5,4),
4. Robot dapat berdiri dan menyeimbangkan diri saat berada di bidang miring sampai pada sudut kemiringan 50 dengan menggunakan informasi pusat tekanan robot.

DAFTAR PUSTAKA

[1] R. Gerndt, D. Seifert, J. H. Baltes, S. Sadeghnejad, and S. Behnke, “Humanoid robots in soccer: Robots versus humans in robocup 2050,” IEEE Robotics Automation Magazine, vol. 22, pp. 147–154, Sept 2015.

[2] M. Vukobratovi´c and B. Borovac, “Zero-moment point - thirty five years of its life,” International Journal of Humanoid Robotics, vol. 1, no. 1, pp. 157–173, 2005.

[3] G. G. Muscolo, C. T. Recchiuto, C. Laschi, P. Dario, K. Hashimoto, and A. Takanishi, “A method for the calculation of the effective center of mass of humanoid robots,” in 2011 11th IEEE-RAS International Conference on Humanoid Robots, pp. 371–376, Oct 2011.

[4] I. Ha, Y. Tamura, H. Asama, J. Han, and D. W. Hong, “Development of open humanoid platform darwin-op,” in SICE Annual Conference 2011, pp. 2178–2181, Sept 2011.

[5] Muhammad Arifin, Muhammad Ardi Pradana, Uti Solichah. “ICHIRO Kid-Size 2016 Team Description Paper”. Intelligence Robot Laboratory ITS, Surabaya. 2016.

[6] M. Schwarz, J. Pastrana, P. Allgeuer, M. Schreiber, S. Schueller, M. Missura, and S. Behnke. “Humanoid TeenSize Open Platform NimbRo-OP”. pp. 568–575. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[7] G. Passault, Q. Rouxel, L. Hofer, S. N’Guyen, and O. Ly, “Low-cost force sensors for small size humanoid robot,” in 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), pp. 1148–1148, Nov 2015.

[8] R. Fabre, H. Gimbert, L. Gondry. “Rhoban Football Club Team – Description Paper Humanoid KidSize League, Robocup 2016 Leipzig”. Robocup 2016, Leipzig.

[9] Jung-Hoon Kim, Jung-Yup Kim and Jun-Ho Oh. “Adjustment of Home Posture of a Biped Humanoid Robot Using an Inertial Sensor and Force Torque Sensors”. International Conference on Intelligent Robots and Systems, San Diego, CA, USA, Nov 2007.

[10] Pranav Kuber, Sahil Kolhe, Pawan Kulkarni, Prajyot Kulkarni. Estimation of Zero Moment Point using Centre of Gravity based Method. ,”in International Journal of Engineering Research & Technology Vol. 5, issue 10, Oct 2016.

[11] Basilio, J.C. dan Matos, S.R., “Design of PI and PID Controllers with Trancient Performance Specification”, IEEE Transaction on Education, vol. 45 – no.4, pp. 364-370, November, 2002.

Increasing Stabiliy of Shooting Motion On EROS (EEPIS Robot Soccer) Using Joint Trajectory Controller

Adi Sucipto*, Achmad Subhan Kh., Ali Husein A., Dimas Pristovani, Imam Fajar F., Aulia Khilmi Rizgi, Ryan Satria Wijaya, Krisna Adji Syahputra, Ilham Fakhrul Arifin, M.Ayuf Basthomi, Ahmad Habib Almutawakkil, Anhar Risnumawan
Laboratorium Robotika Politeknik Elektronika Negeri Surabaya
Kampus PENS, Jalan Raya ITS, Sukolilo Surabaya 60111
Surabaya, Indonesia
master.chess@gmail.com

***Abstract*— This paper explains about Increasing stability of shooting which applied in EROS humanoid robot soccer. The main goal is robot did not fall when robot doing shooting motion. This control is generating the new trajectory system with including the IMU`s data condition. The robot will have a new pose for shooting depend on the filtering data inertial measurement unit (IMU) on it hand. To reduce the vibration effect caused by the swing robot steps. The system also uses vibration control. Vibration control read the angular velocity reaction in its body and compensate it with adjusting the torque value in pelvis joint (actuator). To support them, robot equipped IMU which installed in center of the body. The joint trajectory control is used in single support phase (SSP), double support phase (DSP) and center of gravity (COG) shifting condition in shooting motion.**

Keywords— COG Shifting condition; DSP (Double Support Phase); IMU; Joint trajectory control; SSP (Single Support Phase)

I. INTRODUCTION

The Competition between the robot and the world champions in a soccer game that is expected to take place in 2050 is the aspiration of the robocup developers. Researches of humanoid robot soccer continue to conduct research to improve the ability of robots like human beings.In the competition Humanoid robot division is concentrated in building a robot like human robot player. One of the ability of football player is shooting.In this paper we design the model of shooting like human being. Instability is one of the major defects in humanoid robots. Recently, various methods on the stability and reliability of humanoid robots have been actively studied. Many of the existing methods for humanoid shooting control are constrained by the need to regulate the arm positions without any consideration for navigation of the humanoid within a complex environment. To achieve more robust shooting control in these environments, the robot must be able to adjust its shooting state (center of mass motion).The goal of the present paper is to provide humanoids with Increasing Stabiliy of Shooting Motion on EROS (EEPIS Robot Soccer) Using Joint Trajectory Controller.The main goal of this research is robot did not fall when robot doing shooting motion.When the robot swing,mostly the robot can not stabilize the position.So,it make the robot fall.

II. CONTROL SHOOTING MODEL

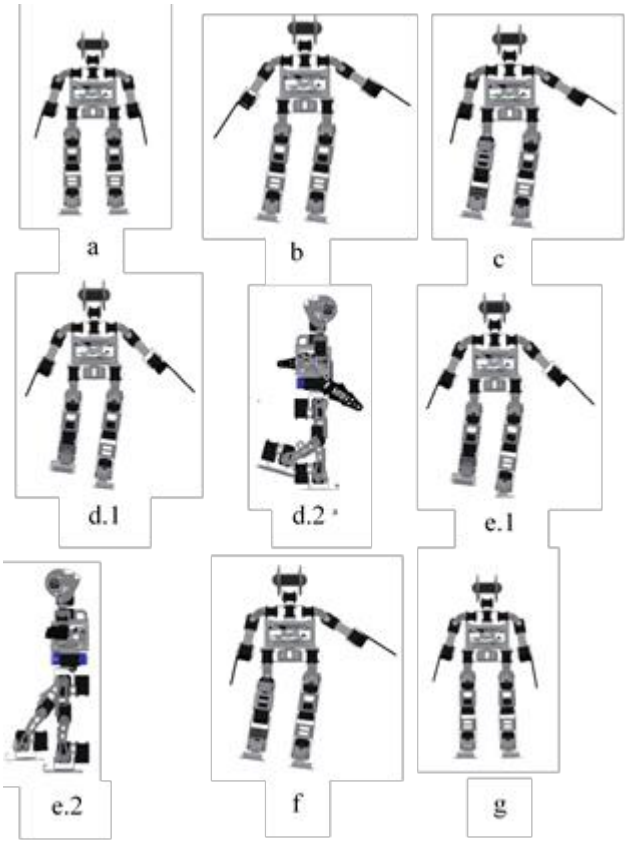


Fig. 1. EROS Control Shooting Model

The first step (a) is stands phase. In this case, there is nothing filter control from the hand because it is just stand and no need

the data filter from the IMU to the its hand. The second step (b) is DSP Shifting COG. This function is to make the center of gravity from the robot to be equal in this position like stand up before. The control from IMU is activated. It happens because of balancing of this body. The third step (c) is SSP Shifting COG. This function is to make the robot stabilize before swinging its foot. The robot lifts it's one of foot. Then the control is activated for balancing by the left hand move up. The fourth step (d.1 and d.2) is back swing. This step is the main problem when the robot shooting. Because it has error system. This is internal disturbance from robot when swinging. So, in this paper we add the control to solve this problem. The output of this control is moving up the arm to get the stable condition.

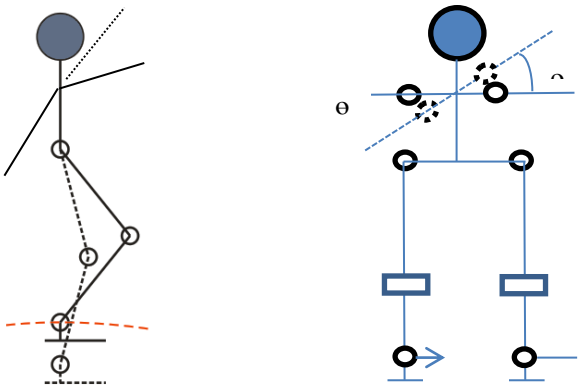


Fig. 2. (a) Single Support Phase. (b) The arm's control using trajectory controller

We manipulated variables obtained by the PID controller using equation:

$$B_{yt} = K_{p(y)}.\, e_{y(t)} + k_{i(y)}.\, {}_0\int^t e_\tau\, d\tau + \frac{k_{d(y)}.\,(e_{y(t)}-e_{y(t-1)})}{dt} \qquad (1)$$

$$B_{xt} = K_{p(x)}.\, e_{x(t)} + k_{i(x)}.\, {}_0\int^t e_\tau\, d\tau + \frac{k_{d(x)}.\,(e_{y(t)}-e_{y(t-1)})}{dt} \qquad (2)$$

We must make variable to declare the equations:

B_{yt} = control of the hand in Y-axis
B_{xt} = control of the hand in X-axis
$K_{p(y)}$ = variable of Proportional control in Y-axis
$K_{p(x)}$ = variable of Proportional control in X-axis
$k_{i(y)}$ = variable of Integral control in Y-axis
$k_{i(x)}$ = variable of Proportional control in X-axis
$e_{x(t)}$ = error value in X-axis
$e_{y(t)}$ = error value in Y-axis
$e_{y(t)}$-$e_{y(t-1)}$ = different error between present error and last error

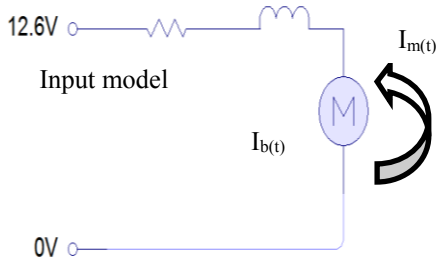


Fig. 3. Electric Model

$$F = B\,.\,l\,.\,I_b \qquad (3)$$
$$I_{m(t)} = k_t\,.\,I_{b(t)}$$
$$I_b = I_{m(s)} / k_t$$

$$V_b = B\,.\,l\,.\,\dot{\varphi} \qquad (4)$$
$$Vb(s) = k_b\,.\,\theta_{m(s)}$$

$$R_b\,.\,I_{b\,(s)} + l_b\,.\,S_{lb(s)} + Vb_{\,(s)} = E_{b\,(s)} \qquad (5)$$

Subtitute 3,4 and 5:

$$\frac{(R_b+L_{bs})I_m}{Kt} + K_{b(s)}\,.\,\theta_{m(s)} = E_{b\,(s)}$$

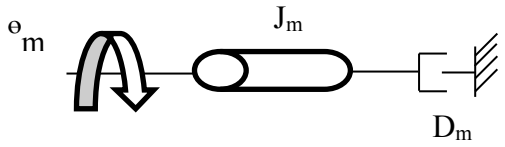


Fig. 4. Output model

$$(J_m s^2 + D_m s)\,.\,\theta = I_m \qquad (6)$$

Then, equation 5 entered to equation 6:

$$\frac{\theta_{m(s)}}{E_{b(s)}} = \frac{K_t / R_b J_m}{S\,[S + Jm\,(\,Dm + \frac{Kb\,)}{Ra}}$$

Then, the transfer function is

$$G_{1(s)} = \frac{\theta_{m(s)}}{E_{b(s)}} = \frac{k_m}{s(s+\alpha)} \qquad (7)$$

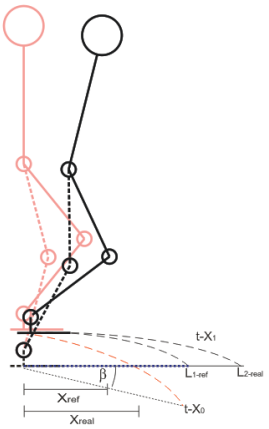


Fig. 5. Plant of Swinging on Shooting

when robot is shooting, resulting the acceleration. On the inverted pendulum on the vertical direction, resulting angular acceleration $\ddot{\theta}_g = (\frac{g}{L})\sin\theta$, and angular acceleration on horizontal direction is $\ddot{\theta}_z = -(\dot{v}/L)\cos\theta$. We can get transfer function $G_{(s)}$:

$$\ddot{\theta} = \ddot{\theta}_g + \ddot{\theta}_z = (g/L)\sin\theta - (\dot{v}/L)\cos\theta.$$

$$L\ddot{\theta} - g\theta = -\ddot{x}$$

$$G(s) = \frac{\theta(s)}{x(s)} = \frac{-s^2}{Ls^2 - g} = \frac{-s/g}{(\tau_L s + 1)(\tau_L s + 1)} \quad (8)$$

The (e) step is SSP Shifting COG. This function is to make the robot stabilize after swinging its foot. The Robot lift its one of foot. Then the control is activated for balancing by the left hand move up. The Last step is Stands.in this case, there is nothing filter control from the hand. Because it is just stand and no need the data filter from the IMU to the its hand. From the (a) to (g) figures to get approaching the system shooting model like human being based on Newton`s 1st law [5], we design the proportional Gyroscope`s angle measure as shown as the table.

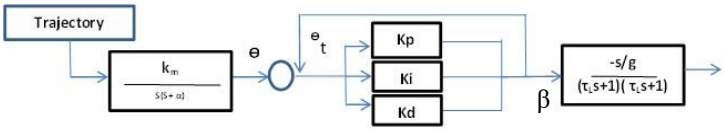


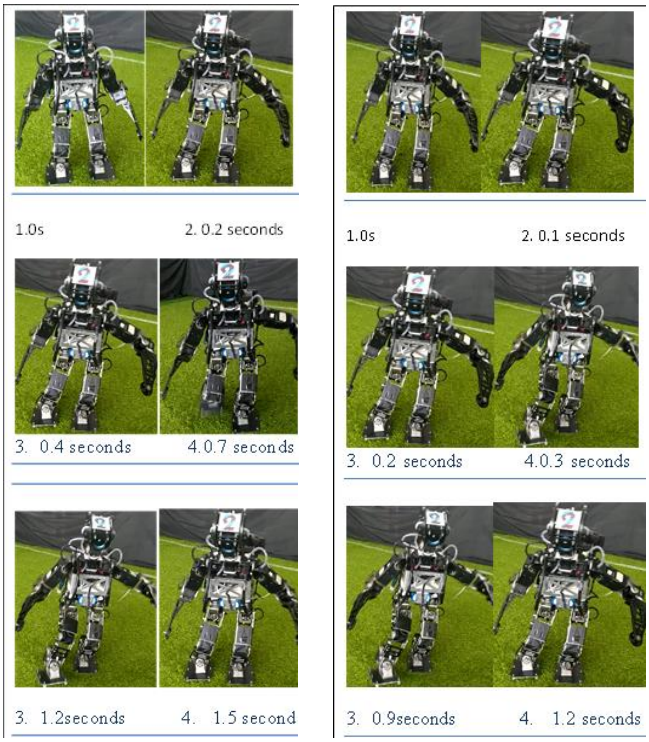
Fig. 6. Plant of Stability Trajectory

TABLE I. REFERENCE OF IDEAL ANGLE ON SHOOTING CONTROL

Step	1	2	3	4	5	6	7
Angle	8°	20°	30°	71°	61°	35°	8°

We get the 8° data when we calibrate the robot stand. Our approach is based on [1] who use a P-controller with the measured data of a gyroscope as input to compensate for external disturbances. The manipulated variables of the approach are directly used to control the servo motors used for balancing. Since gyroscopes are measuring the angular velocity independently from body tilt and roll, they can only be used to keep an angular velocity. As extension to the method introduced in [1], our approach uses a PID controller instead of a P controller. The PID controller is distinct from the P controller as it is able to eliminate the entire control deviation and it is consequently more reliable in this context. Furthermore, the approach of [1] suggests that stabilization is given when the rotational velocity of the body is kept by 0°. This is not applicable to our approach as the body has to move during the COM balance stabilizing. It has to be distinguished between desired angular velocity and undesired angular velocity to allow the two balancing components to work together.

III. RESULT



(a) (b)

Fig. 7. (a) Without Control. (b) With Control

From (a) to (g) fig. 1., it is shown that EROS need 1.5s to reach the state of shooting. Besides that, we get the Gyroscope`s data. The data change unrapidly (8 ;16 ;25 ;30 ;35 ;41 ;52 ;68;66;61;50;35;26;9). In this case, we do not give IMU`s control to the hand. We did the experiment 10 times. The result is 5 succeeded and 5 failed. We take the experiment in different place at the field. We get the result as shown table below:

TABLE II. RESULT OF EXPERIMENT

Experiment	Result	
	With Control	Without Control
1	√	X
2	√	X
3	√	√
4	√	√
5	X	√
6	√	X
7	√	X
8	√	X
9	X	√
10	√	√

From (a) to (g) fig. 1., it is shown that EROS need 1.2s to reach the state of shooting. Besides that, we get the Gyroscope`s data. The data change rapidly (8 ;23 ;47 ;91 ;92 ;71 ;9 ;2;7). In this case, we give IMU`s control to the hand. We did the experiment 10 times. The result is 8 succeeded and 2 failed. We take the experiment in the experiment in different place at the field.

The Graph Comparison between Shooting without control, shooting with control and design reference shooting:

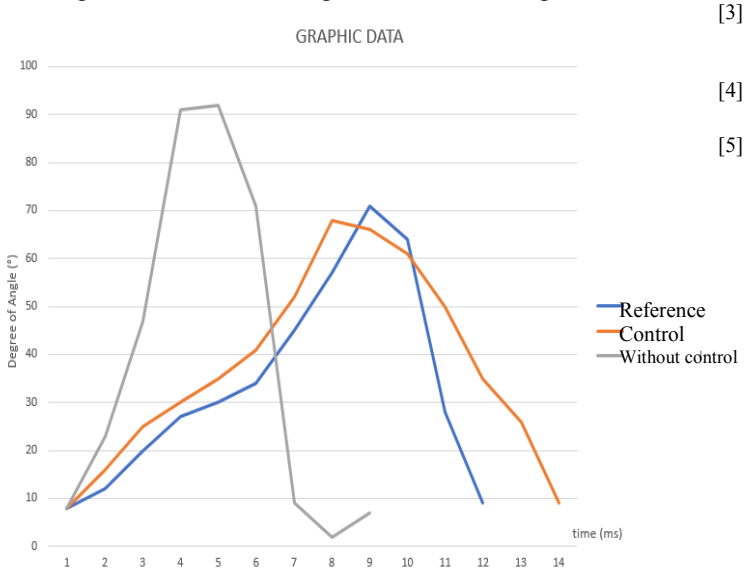


Fig. 8. The graph between with control,without control and reference

From the graph shows that Shooting with IMU`s control has 0-1.5s and shooting without IMU`s control has 0.1.2s. We can analyze that the robot with IMU`s control is faster in doing shooting. Besides that, we can see that the Gyroscope`s measure of robot with IMU`s control is more approximate with the reference of Gyroscope`s angle. The robot with IMU`s control has the change data gyroscope`s angle rapidly. It means that the robot is more stable than the robot without IMU`s control. As much the gyroscope`s data changing, it shows that the robot is unstable.

IV. CONCLUSION

Increasing control shooting motion with filtering IMU`s data Proposed to support the stability of EROS`s shooting. The shooting trajectory is controlled by the Hand`s balancing. By using this system, Trajectory shooting of robot can adapt the position depending on the center of gravity and center of pressure from the robot. This Filter IMU`s data has many affected for shooting, because it can stabilize the robot from internal disturbance (Error System) by robot when its shooting.

V. ACKNOWLEDGEMENT

The authors would like to thank for Ministry of Research, Technology and Higher Education and Student Affairs of PENS.

REFERENCE

[1] Akin, H.L., Mericli, T., ¨Ozkucur, K.C., G¨okce, B. “Cerberus 2010 team description paper” .2009

[2] Dimas Pristovani R, “Basic Walking Trajectory Analysis in FLOW Robot” in 2016 International Electronics Symposium (IES), pp.333-338, September 2016.

[3] Azhar Aulia Saputra, Achmad Subhan Khalilullah, Indra Adji Sulistijono, Naoyuki Kubota. “Adaptive motion pattern generation on balancing of humanoid robot movement”.CCECE 2015: 1479-1484

[4] M¨uller, J., Laue, T., Rofer, T., “Kicking a Ball – Modeling Complex Dynamic Motions for Humanoid Robots”.2011

[5] Hayward, J., “Newton`s Laws of system Dynamics”.2015

Pengembangan Pola Berjalan dan Stabilisasi pada *Humanoid Robot Soccer* DARDROID (DARWIN-ODROID)

Yohanes Fridolin Hestrio, Nicholas, Athina Maria Angelica, M. Arifin Julian
Universitas Indonesia, Kampus UI Depok 16424
Depok, Indonesia
io-ui@ui.ac.id

Abstrak—Salah satu permasalahan utama pada robot humanoid adalah motion, yaitu berkaitan dengan kestabilan robot ketika berjalan. Masalah ini dapat diatasi dengan menggunakan metode static walking dan dynamic walking serta aplikasi dari metode zero moment point dan invers kinematics yang didukung oleh inertial measurement unit dengan penerapan metode Kalman filter dan PD controller. PD controller menggunakan feedback dari IMU untuk memperbaiki posisi servo yang didapatkan dari perhitungan zero moment point dan invers kinematics selama robot berjalan.

Kata Kunci— ***Kalman Filter; Kinematika Terbalik; Least Square; PD controller, Zero Moment Point***

I. PENDAHULUAN

Masalah dan tantangan yang sering kali timbul dari robot humanoid adalah perancangan dan pengendalian kestabilan pergerakan jalan. Pada robot humanoid, untuk mencegah robot terjatuh karena ketidakstabilan, dibutuhkan instruksi detil yang didapatkan dari perhitungan yang akurat.

Penelitian mengenai robot humanoid sudah dilakukan di Universitas Indonesia, namun fokus penelitiannya hanya pada desain mekanik bracket sehingga timbul beberapa masalah pada *autobalance system* karena adanya ketidaktepatan posisi servo pada bracket yang mengakibatkan error dalam pergerakan servo. Karena adanya masalah tersebut, maka dibutuhkan riset lanjutan dari penelitian sebelumnya untuk mengoptimalisasi joint trajectory control, balancing control, dan motion planning. Penelitian ini akan membahas pengendalian zero moment point menggunakan forward dan invers kinematics.

II. SISTEM PERGERAKAN ROBOT

Agar robot dapat bergerak dengan baik maka dibutuhkan suatu bentuk kerja sama antara sistem mekanik, elektrik, dan *artificial intelligence* robot dengan masing-masing sistem memiliki peran tersendiri yang menopang sistem pergerakan robot secara keseluruhan. Model jalan robot humanoid dapat dibagi menjadi dua fase gerak utama yaitu *Single Support Phase* (SSP), dimana hanya satu kaki yang melakukan kontak dengan tanah untuk menopang robot, dan *Double Support Phase*, dimana kedua kaki melakukan kontak dengan tanah.

A. Kalman Filter

Agar dapat berjalan dengan baik dan efisien, robot humanoid memerlukan suatu sistem yang menghitung pergerakan servo agar robot tetap stabil saat berganti dari SSP ke DSP dan sebaliknya, salah satunya yaitu metode Kalman filter. Metode Kalman filter sendiri adalah teknik perhitungan matematis (algoritma) yang memberikan metode perhitungan efisien dalam proses mengestimasi kebenaran suatu data dengan cara meminimalisir rata-rata kuadrat galat (*Mean Squared Error/MSE*).

Kalman Filter digunakan untuk mengetahui *zero moment point* (ZMP) dan *center of mass* (CoM) pada robot ketika bergerak secara dinamis maupun statis. sehingga dapat mendukung keseimbangan dinamis ketika berjalan dan keseimbangan statis ketika berhenti bergerak baik dalam fase SSP maupun DSP. Kalman filter digunakan pada proses yang dapat dinyatakan dalam bentuk persamaan *linear state* seperti pada persamaan berikut.

$$x_{k+1} = A_k x_k + B_k u_k + w_k$$

Persamaan diatas dapat diobservasi dengan model pengukuran yang memetakan *state x* ke keluaran *y* seperti pada persamaan berikut.

$$y_k = H_k x_k + v_k$$

Noise proses (w) dan noise pengukuran (v) merupakan noise yang saling bebas. Nilai estimasi *state* $\dot{x_k}$ pada Kalman filter ditentukan dari estimasi posteriori $\dot{x_k}$ serta selisih antara pengukuran sebenarnya $\dot{y_k}$ dan estimasi pengukuran $H_k x_k$ seperti pada persamaan berikut.

$$\dot{x} = \dot{x_k} + K_k(y_k - H\dot{x_k})$$

Selisih nilai antara pengukuran sebenarnya $y_k$dan estimasi pengukuran disebut sebagai nilai *residual*. Jika nilai *residual* adalah nol, maka itu menunjukkan bahwa hasil estimasi sama dengan hasil pengukuran. Nilai K_k adalah faktor gain pada Kalman filter.

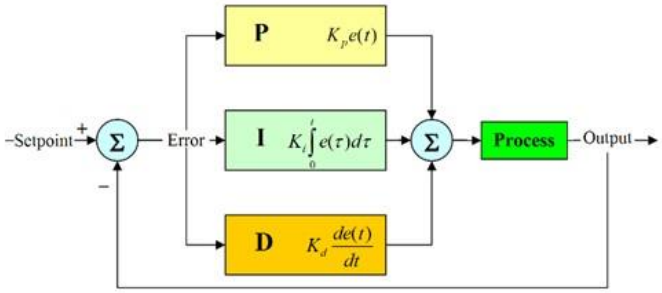
B. PID Controller

Selain menggunakan Kalman filter untuk memfilter data yang digunakan, diperlukan juga suatu sistem kontrol untuk mengatur pergerakan servo. Untuk ini digunakan *proportional-integral-derivative controller* (PID *controller*) yang merupakan sistem kontrol yang memanfaatkan *data feedback* untuk menjaga kestabilan sistem dengan terus menerus menghitung nilai kesalahan sebagai perbedaan nilai setpoint yang diinginkan dengan variabel proses yang terukur. PID *controller* akan

meminimalisir nilai kesalahan pada setiap waktu dengan penyesuaian *control variable*. Nilai PID *controller* dapat dihitung dengan menggunakan model persamaan berikut

$$u(t) = P(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

dengan Kp adalah *proportional gain,* Ki adalah *integral gain*, Kd adalah *derivative gain*, e adalah *error* dari setpoint dikurang variable proses, t adalah waktu, τ adalah variable integrasi dari waktu nol hingga t.



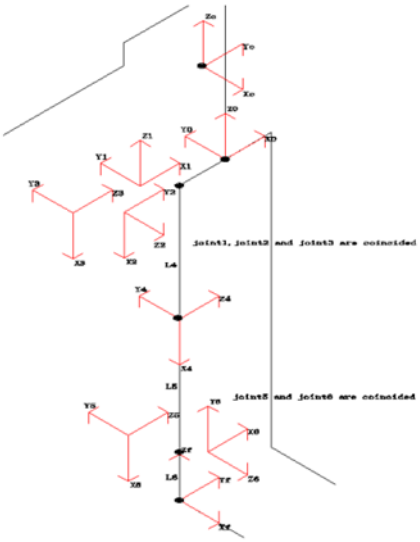
Gambar 1. Desain model PID *controller*

Pada model ini, P bertanggung jawab untuk mengolah dan meminimalisir nilai kesalahan saat ini, I bertanggung jawab untuk nilai kesalahan sebelumnya dan D bertanggung jawab untuk kemungkinan nilai kesalahan mendatang, berdasarkan rate perubahan tiap waktu. Model PID controller ini diperlukan untuk menjaga nilai parameter kesetimbangan robot baik saat statis maupun dinamis.

III. Sistem Artificial Intelligence Motion Robot

Sistem artificial intelligence yang mengendalikan motion pada robot humanoid terbagi menjadi 3 bagian utama: *Homogeneous Transformation Matrix* yang menjadi landasan bagi *invers kinematics* dan *forward kinematics* sebagai dasar dari pola gerakan, *Zero Moment Point* sebagai landasan kendali keseimbangan, dan IMU *Data Filtering* dan akuisisi data IMU sebagai feedback kendali keseimbangan.

Agar dapat merancang sistem *artificial intelligence* yang mampu mengendalikan motion robot dengan baik, maka dibutuhkan suatu metode untuk menjelaskan hubungan antara tiap-tiap bagian robot sebagai dasar pergerakan. Metode ini kemudian disebut sebagai Denavit-Hartenberg (DH) Parameters. Metode ini hanya dapat dipergunakan jika sisi kanan dan kiri robot simetris sehingga persamaan pada salah satu bagian robot dapat dipergunakan pada pasangan simetrisnya dengan melakukan pencerminan matriks.



Gambar 2. Bentuk robot secara Koordinat untuk 6 DOF kaki kanan (Secara 3 Dimensi)

Berdasarkan pada gambar 2, maka dengan metode DH parameter didapatkan sumbu tiap servo pada 6 DOF kaki kanan[13].

Dengan mengetahui parameter keenam DOF, DH parameter, dan limitasi sudut pada kaki kanan, maka untuk mendapatkan nilai DH parameter dan limitasi sudut pada bagian kaki kiri hanya perlu melakukan pencerminan matriks terhadap sumbu putarnya. Sementara itu, dengan menggunakan persamaan matriks pada kaki kanan ${}^{c}_{f}T$, menghitung nilai sudut $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$ dapat menggunakan geometri parsial dan aljabar parsial. Hal ini dikarenakan ${}^{6}_{f}T$ dan ${}^{c}_{0}T$ konstan, sehingga dapat disederhanakan menjadi:

$${}^{0}_{6}T = [{}^{c}_{0}T]^{-1} {}^{c}_{f}T \left[{}^{6}_{f}T\right]^{-1}$$

IV. Akuisisi dan Filtering Data

Pada penelitian ini, data yang diperoleh dari IMU difilter menggunakan metode Kalman filter. Metode Kalman filtering digunakan untuk memprediksi nilai yang paling mungkin karena ketika IMU bergerak bersama robot nilai yang dihasilkan tidak pasti dan sulit untuk diprediksi. Kalman Filter menggunakan sistem kepercayaan dengan mengambil nilai dari gyro sebagai nilai referensi Kalman filter, sedangkan variabel yang difilter adalah accelerometer dan kompas.

Sementara itu akuisisi data dalam penelitian ini menggunakan metode *least square* karena *least square* merupakan metode untuk menentukan fungsi pendekatan polinomial y = f(x) yang paling mendekati data (x1, y1) hingga (xn, yn) dengan persamaan fungsi polinomial f(x) sebagai berikut

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \cdots + a_a x + a_o$$

Untuk memproses data dari gyro, dibutuhkan nilai tengah data sebagai faktor normalisasi data dengan nilai tengah tersebut adalah 512.

Lalu untuk mengendalikan servo pada kaki agar posisi CoM tidak melebihi zero moment point robot, digunakan persamaan percepatan

$$\frac{d}{dt}\dot{x} = u_x$$

dengan menggunakan persamaan dasar dari zmp

$$Tzmp = mg(x - p_x) - m\ddot{z}_c = 0$$

Sehingga persamaan sebuah kendali optimal dengan zero moment point dapat dinyatakan sebagai berikut

$$J = \sum_{i=k}^{\infty} Q_e(p_d(i) - p(i))^2 + \Delta X^T Q_x \Delta X + R\Delta u(i)^2$$

dimana Q_e merupakan pengali dari referensi error ZMP, Q_x merupakan pengali error dari kondisi sistem dan R adalah pengali error input.

Dengan mengikuti hokum sistem kendali, maka didapatkan

$$u(k) - K_s \sum_{i=0}^{k} \big(p(i) - p_d(i)\big) - K_x X(k) - \sum_{i=k}^{N_L} G(i) p_d(k+i)$$

dimana K_s adalah gain error, K_x berisikan gain dari kondisi sistem dan $G(i)$ adalah variable fungsi yang dapat menunjukkan optimal preview gain untuk N_L preview steps.

Kemudian PD controller dalam penelitian ini adalah sistem PD controller untuk membuat sistem kendali berbasis IMU. PD controller ditgunakan pada bagian kaki dan tangan karena kedua bagian tersebut mempengaruhi postur badan robot saat robot bergerak. Sistem kendali ini dapat dinyatakan dengan persamaan

$$\Delta\theta = K_p(\theta_{des} - \theta_m) + K_d(\dot{\theta}_{des} - \dot{\theta}_m)$$

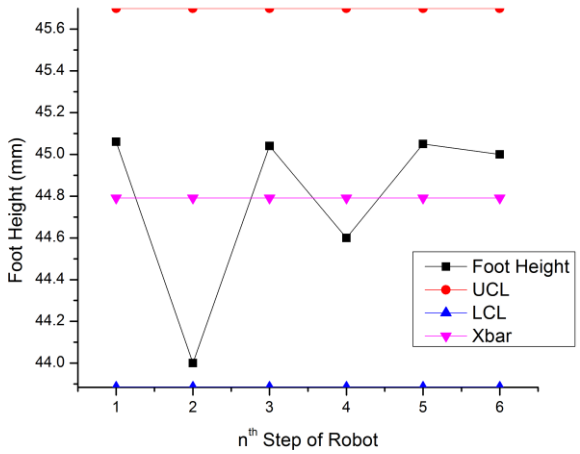
Dimana Kp dan Kd merupakan proportional dan derivative gain pada PD controller, (θdes, φdes) merupakan sudut torsi pada sumbu x dan y, dan ($\dot{\theta}$ *des*, $\dot{\varphi}$ *des*) merupakan kecepatan angular rata-rata sehingga (θdes, φdes) dan ($\dot{\theta}$ *des*, $\dot{\varphi}$ *des*) menunjukkan sudut torsi dan kecepatan angular. Nilai tersebut didapatkan dari *feedback control* IMU yang telah difiltering serta akuisisi data dengan metode Kalman filter dan *least square*.

V. HASIL PENGUJIAN ROBOT

Pengujian robot humanoid yang digunakan terdiri dari 3 bagian: pengujian parameter jalan, pengujian keseimbangan robot, dan pengujian kecepatan robot.

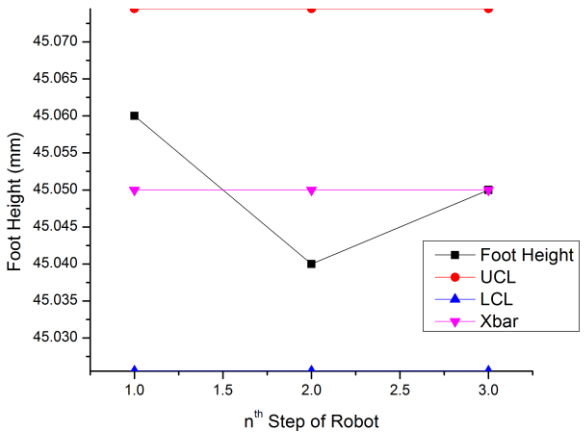
1) Hasil Pengujian Parameter Jalan

Pengujian ini dilakukan dengan kelima pameter yang diuji yaitu *step forward*, *Foot height*. *Y offset*, *Hip pitch offset dan period time*. Pengujian dilakukan dengan membandingkan hasil perhtiungan invers kinematika dengan hasil yang terjadi pada saat robot berjalan parameter yang diberikan adalah foot height 45 mm.



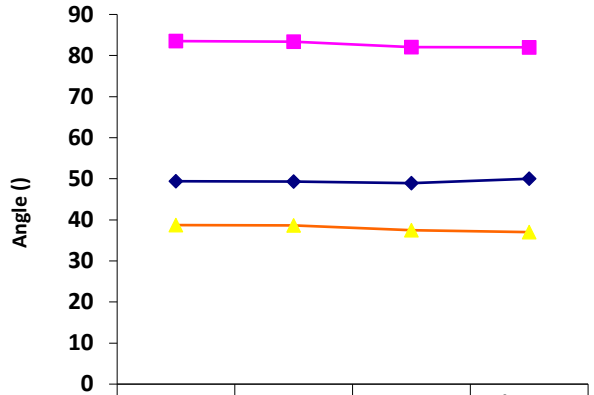
Gambar 3. Pengujian tinggi langkah kaki pada rumput 3 cm

Pengujian tinggi pada langkah kaki robot pada gambar 3 dan 4 menggunakan bagan bagan kendali (*control chart*) dengan batas rentang antara UCL (*Upper Control Limit*) dan LCL (*Lower Control Limit*) dengan menggunakan 2 kali standar deviasi data. Data yang diambil berupa langkah kaki sebanyak 6 langkah pada rumput dengan tinggi 3 cm. Pada gambar 4.1 yaitu pada rumput 3 cm dapat dilihat bahwa semua data uji masih berada dalam batas kendali.



Gambar 4. Pengujian tinggi langkah kaki kanan pada rumput 3 cm

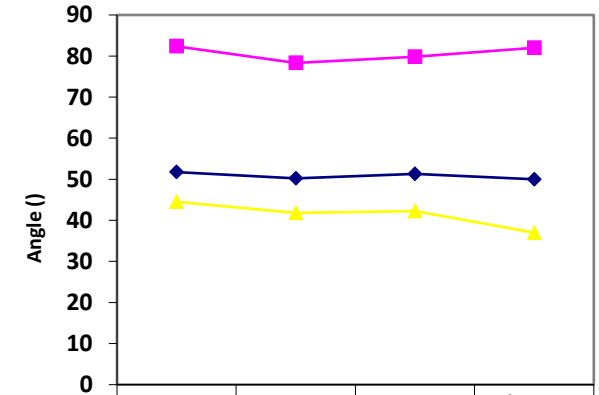
Pada gambar 4 terlihat bahwa terjadi adanya penyimpangan sebesar 0.00667 mm. Nilai tersebut merupakan rata-rata dari 3 kali pengujian pada posisi SSP. Dimana seharusnya tinggi kaki yang terjadi adalah 45mm, namun hal ini tidak demikian pada pengujian. Oleh karena itu, perlu dianalisis setiap sudut dan membandingkan dengan perhitungan sudut kinematika terbalik. Sehingga data yang perlu dibandingkan adalah servo ID pada kaki kanan yaitu ID 11, 13, dan 15.



	Pengujian 1	Pengujian 2	Pengujian 3	Invers Kinematika
ID 11	49.368	49.28	48.928	50
ID 13	83.512	83.336	82.016	82
ID 15	38.72	38.632	37.488	37

Gambar 5. Grafik pengujian servo kaki kanan

Dari gambar tersebut terlihat bahwa terjadi penyimpangan sebesar 0.49 mm secara rata-rata karena pengujian dilakukan 3 kali pada pose SSP. Tinggi kaki searusnya yang terjadi adalah 45mm. Maka, perlu dicek setiap data sudut dan membandingkannya dengan perhitungan invers kinematik. Dalam perhitungan invers kinematik didapatkan sudut-sudut yang terbentuk seharusnya. Data yan dibandingkan adalah ID 11 13 15 karena data yang digunakan adalah data dari kaki kanan robot.



	Pengujian 1	Pengujian 2	Pengujian 3	Invers Kinematika
ID 12	51.744	50.248	51.28	50
ID 14	82.368	78.32	79.856	82
ID 16	44.52	41.8	42.304	37

Gambar 6. Grafik pengujian sudut servo pada kaki kiri

Pada gambar 6 terlihat adanya penyimpangan-penyimpangan yang masih bisa ditoleransi, namun pada ID 16 terdapat *error* yang cukup signifikan karena perbedaan sudut hasil perhitungan dan sudut servo pada robot sangat jauh. Hal ini terjadi karena cuplikan yang diambil kemungkinan besar belum mencapai titik puncak, terlihat dari setiap tinggi angkat kaki yang tidak menyentuh angka 45. Hal ini diakibatkan oleh masalah pada konektor kabel sehingga ada keterlambatan sinyal yang didapatkan. *Error* ini juga terlihat ketika langkah robot saat berjalan robot yang terlihat sedikit tertatih di beberapa waktu. Hal ini juga dapat dibuktikan dengan penyebaran titik-titik ZMP pada telapak kaki robot.

2) Hasil Pengujian Trayektori Robot

Trajectory merupakan pola atau jejak robot dalam melakukan proses berjalan. Pengujian Trajectory dilakukan untuk dapat mengetahui secara perhitungan matematis kecepatan robot saat berjalan [4].

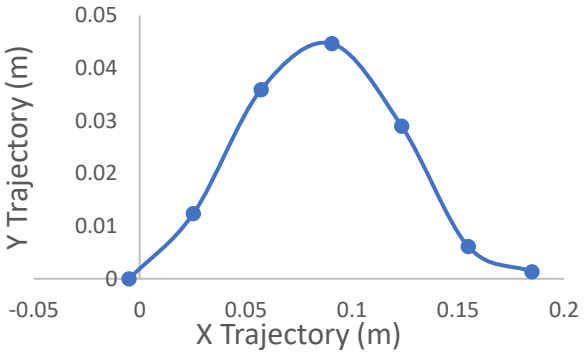
$$X(t) = (b+f)\left(\frac{t-t1}{t2-t1} - \frac{1}{2\pi}\sin\left(2\pi\frac{t-t1}{t2-t1}\right)\right) - b$$

dimana b adalah besar langkah kaki awal (m) dan f adalah besar langkah kaki akhir (m), t adalah variable waktu, t1 adalah waktu awal langkah pertama berjalan (s), t2 adalah waktu akhir langkah terakhir berjalan (s) dan X adalah trajectory arah sumbu x.

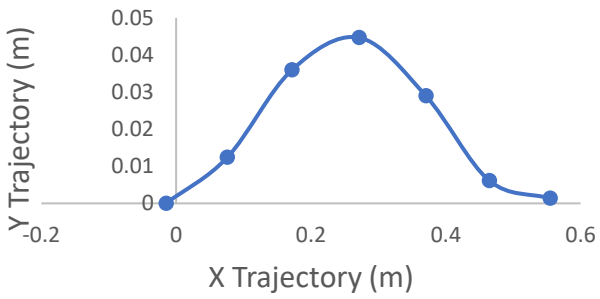
Selain itu juga, diperlukan trajectory pendukung yaitu trajectory arah sumbu y untuk mengetahui pola pergerakan arah sumbu y pada robot dengan menggunakan persamaan berikut [4] :

$$Y(t) = \left(2 - \left(\cos\left(2\pi\frac{t-t1}{t2-t1}\right) + 1\right)\right)\frac{h}{2}$$

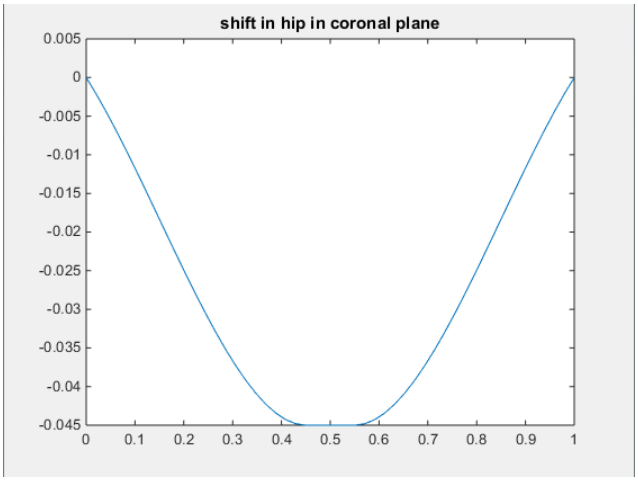
dimana h adalah tinggi langkah kaki robot ketika berjalan (m). Dari persamaan-persamaan di atas maka didapatkan grafik-grafik berikut



Gambar 7. Trayektori langkah engkel robot dengan besar langkah 5mm

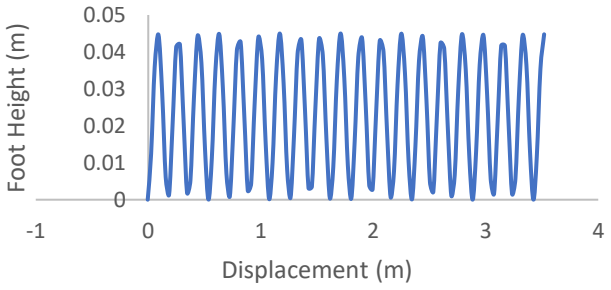


Gambar 8. Trayektori langkah engkel robot dengan besar langkah 15mm

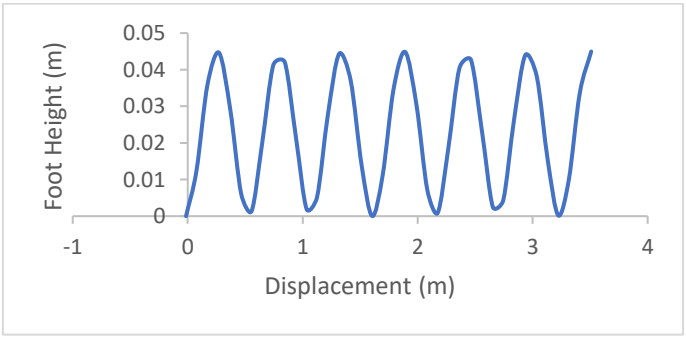


Gambar 9. Trayektori langkah engkel robot dengan besar langkah 5mm pada MATLAB

Jika pada pengujian jarak yang ditempuh adalah sebesar 3.5 m dengan jarak 1 langkah sebesar 0.03 m maka akan membutuhkan 117 langkah. Namun, apabila jarak 1 langkah sebesar 0.01 m maka akan membutuhkan 350 langkah. Maka untuk mendapatkan kecepatan awal dari suatu robot diperlukan periode satu fasa berjalan dan jarak langkah, untuk mendapatkan percepatan yang diperlukan keseluruhan langkah berjalan.



Gambar 10. Trayektori langkah robot menempuh 3.5 m dengan besar langkah 5mm



Gambar 11. Trayektori langkah robot menempuh 3.5 m dengan besar langkah 15mm

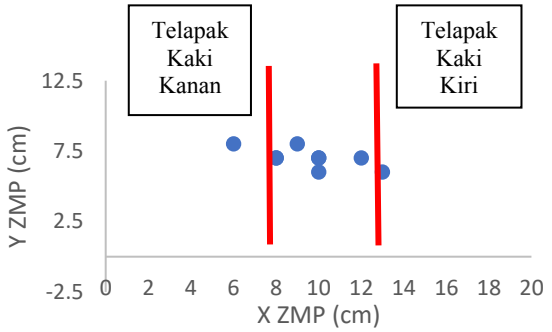
Kecepatan dan percepatan secara teoritis bisa didapatkan dengan rumus

$$\dot{X}(t) = \frac{1.2}{0.315}\left(1 - \cos\left(\frac{2\,\pi\,t}{0.315}\right)\right)$$

Dimana $\dot{X}$ adalah kecepatan pada x-axis (cm/s). jika diketahui bahwa waktu sudah dilakukan pada percobaan maka dapat diketahui kecepatan maksimalnya dalam fase tersebut ktika ketika dalam fasa 90 derajat. Artinya pada detik ke 0.0787 adalah kecepatan maksimum.

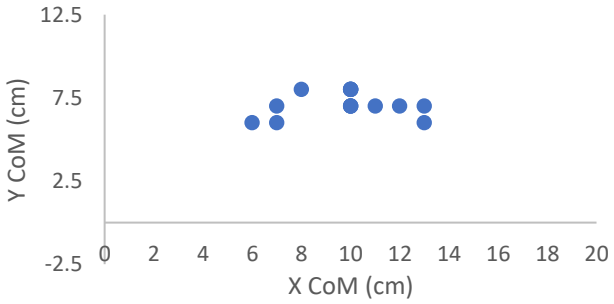
3) Hasil Pengujian Keseimbangan Robot

Pengujian kestabilan robot diukur menggunakan stability margin pada robot. Pengukuran nilai stability margin dilakukan pada mode berjalan. Dengan menggunakan metode Zero Moment Point didapatkan nilai (Xcom, Ycom, Zcom) sebesar (10, 7, 26.5) cm. Nilai Xzmp dan Yzmp tersebut terjadi pada saat posisi awal robot, sehingga tidak adanya pengaruh akibat percepatan dan kecepatan yang bekerja. Namun, ketika pada tahap dimana kondisi adanya percepatan yang berbeda beda sehingga adanya penyebaran titik ZMP seperti pada grafik berikut.

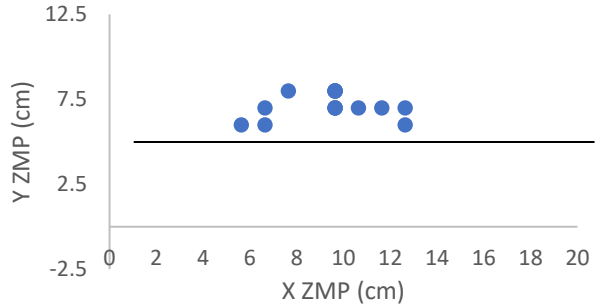


Gambar 12. Penyebaran titik ZMP pada saat jalan ditempat

Dari gambar 11 terlihat bahwa titik-titik zmp saat robot jalan ditempat masih berada didalam *support polygon* artinya robot masih dalam keadaan stabil.



Gambar 13. Penyebaran titik Centre of Mass pada saat berjalan



Gambar 14. Penyebaran titik ZMP pada saat berjalan

Pada gambar 14 terlihat bahwa titik-titik zmp berada didepan kaki robot. Hal ini dikarenakan posisi berjalan cenderung membungkuk dengan tujuan ketika robot dalam kondisi bergerak moment inersia pada bagian atas badan robot berada pada titik zmp dan com robot ketika jalan ditempat. Hal ini masih dapat dikategorikan aman karena zmp pada masih pada saat berjalan berada di *support polygon* robot. Oleh sebab itu, perlu dilihat penyebaran ZMP pada saat Single Support Phase (SSP) dan Double Support Phase (DSP) secara terpisah.

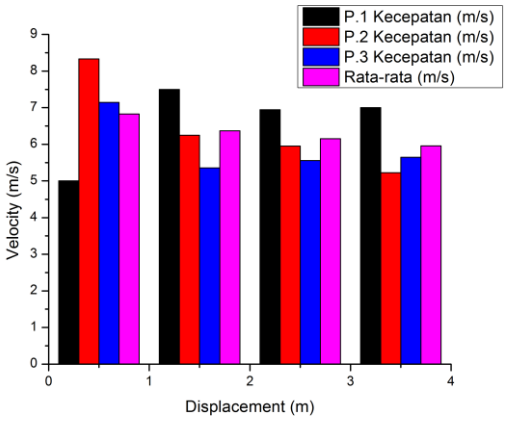
Badan Robot cenderung membungkuk kedepan sehingga hasi scattering terlihat bahwa titik-titik zmp berada didepan kaki robot. Hal ini jug dilakukan karena memang robot berjalan kedepan sehingga titik zmp berada cenderung dibagian depan telapak kaki robot. Pada saat berjalan masih dalam keadaan aman karena masih didalam *support polygon* . Oleh karena itu, perlu dilihat penyebaran ZMP dari keadaan *Single Support Phase* (SSP) dan DSP secara terpisah.

4) Hasil Pengujian Keseimbangan Robot

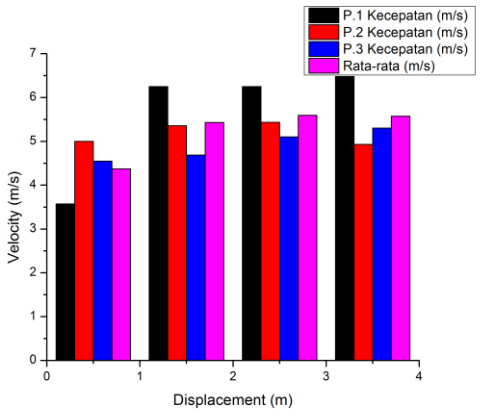
Pengujian Kecepatan Robot dilakukan pendekatan dengan 2 kecepatan yaitu kecepatan maksimal dan kecepatan minimal dengan jarak tempuh tertentu dan 2 besar langkah yaitu maksimal dan minimal [4]. Untuk kecepatan robot dapat dinyatakan dengan persamaan berikut:

$$V = 2\,\frac{b}{T}$$

dimana V adalah kecepatan Robot (m/s), b adalah besar langkah kaki (m) dan T adalah periode dari langkah kaki robot (s). Dari pengujian parameter jalan diketahui periode jalan robot sebesar 0.6s, dengan jarak langkah minimal 0.005m dan jarak langkah maksimal 0.015m maka didapat kecepatan minimal sebesar 0.01667 m/s dan kecepatan maksimal sebesar 0.05m/s. Namun perlu dilakukan pengujian dengan beberapa percobaan berjalan robot dengan menggunakan stopwatch pada lapangan rumput dengan tinggi rumput 1,5 cm dan 3 cm.



Gambar 15. Grafik kecepatan maksimal robot pada rumput 3 cm

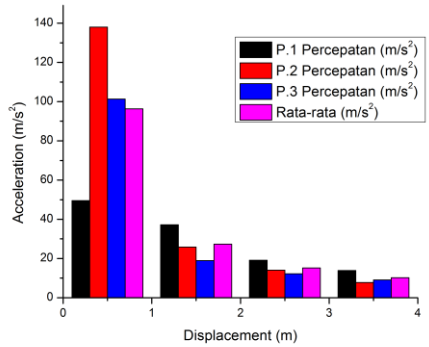


Gambar 16. Grafik kecepatan minimal robot pada rumput 3 cm

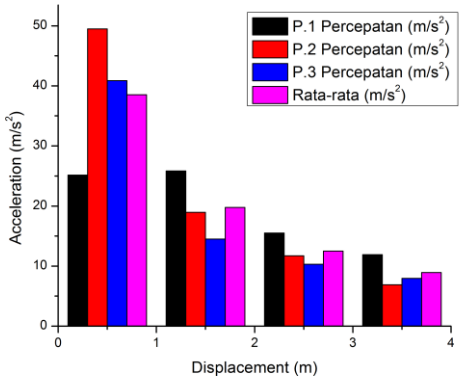
Grafik pada gambar 15 dan 16 diatas merupakan grafik kecepatan terhadap jarak yang ditempuh, masing-masing dilakukan dengan kecepatan maksimal dan minimal yang dapat dilakukan oleh robot. Pada grafik terjadi kenaikan yang tidak terlalu terlihat, namun ketika melihat rata-rata dari setiap waktu dalam 3 kali percobaan maka dapat dikatakan kecepatan mengalami kenaikan. Jika pada perhitungan kecepatan minimal sebesar 0.01667 m/s dan kecepatan maksimal sebesar 0.05m/s, maka pada pengujian ini didapatkan kecepatan minimal rata-rata sebesar 5.24m/s, sedangkan kecepatan maksimal rata-rata sebesar 6.33m/s. Jika dianalisis lebih lanjut, hal ini dapat terjadi akibat adanya kenaikan kecepatan, dengan demikian dapat diketahui percepatan yang terjadi dengan persamaan sebagai berikut.

$$a = \frac{V(t)\,T - 2b}{T\,t}$$

Sehingga didapatkan hasil percepatan seperti pada grafik berikut.



Gambar 17. Grafik Percepatan Maksimal Robot Pada Rumput 3 cm



Gambar 18. Grafik Percepatan Minimal Robot Pada Rumput 3 cm

Grafik pada gambar 17 dan 18 diatas merupakan grafik percepatan terhadap jarak yang ditempuh, masing-masing dilakukan dengan percepatan maksimal dan minimal yang dapat dilakukan oleh robot. Dari grafik dapat dilihat bahwa pada saat menempuh jarak yang relatif pendek robot memiliki percepatan yang cukup besar. Hal ini dikarenakan pada jarak pendek robot dikatakan masih memiliki momentum kecepatan yang cukup besar akibat dari mulainya fase berjalan pada robot. Namun, seiring jarak yang semakin jauh percepatan robot mulai menurun. Jika dianalisis lebih jauh nilai momentum dari kecepatan awal ketika mulai bergerak semakin menurun akibat jarak tempuh robot semakin jauh dan juga dipengaruhi oleh gangguan gerak robot akibat tingginya lapangan rumput yang ada. Dari hasil pengujian ini didapatkan percepatan minimal rata-rata sebesar 19.92m/s^2 dan percepatan maksimal rata-rata sebesar 37.25m/s^2.

VI. KESIMPULAN

1. Dalam penelitian ini diperoleh kestabilan jalan robot rata-rata sebesar 10mm dengan memanfaatkan sistem auto-balance dari IMU dan metode Kalman filter serta akuisisi data least square dengan margin error maksimal 16%.
2. Kombinasi dan desain material yang dilakukan dapat menaikan CoM sehingga dapat meningkatkkan kecepatan robot. Kestabilan ditingkatkan dengan penambahan luas telapak kaki.
3. Kecepatan minimal rata-rata robot sebesar 5.24m/s pada rumput 3 cm, sedangkan, kecepatan maksimal rata-rata robot sebesar 6.33m/s.
4. Percepatan minimal rata-rata robot sebesar 19.92m/s^2 pada rumput 3 cm, sedangkan, percepatan maksimal rata-rata robot sebesar 37.25m/s^2.
5. Simulasi dengan matlab dapat dikembangan untuk simulasi kestabilan dan kecepatan jalan robot dengan input data berupa dimensi robot dan kecepatan yang diinginkan.

REFERENSI

[1] G. Eason, B. Noble, and I.N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (*references*)

[2] K. Elissa, “Title of paper if known,” unpublished.

[3] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[4] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

Sistem Koordinasi Berbasis Perilaku pada Robot Sepak Bola

Muhammad Isnain Hartanto, Kusprasapta Mutijarsa*, dan Widyawardana Adiprawita
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
kusprasapta.mutijarsa@itb.ac.id

***Abstrak*— Robot sepak bola merupakan bidang riset yang sangat populer. Kompleksitas rancang bangun robotnya membuat pengembangan robot sepak bola semakin menantang. Makalah ini membahas hasil rancang bangun sistem koordinasi robot sepak bola sebagai integrasi dari informasi sistem persepsi ke dalam strategi permainan. Pengolahan strategi permainan menghasilkan informasi ke sistem lokomosi, penggiring, dan penendang bola. Sistem koordinasi antar subsistem diimplementasikan menggunakan sistem cerdas berbasis perilaku. Pemilihan ini didasari pada perancangan sistem cerdas berbasis perilaku dilakukan dengan cara *bottom-up*, dimulai dari perilaku sederhana. Sehingga memudahkan peneliti untuk mengembangkannya secara berkelanjutan. Implementasi sistem ini menggunakan *Finite State Machine* (FSM) dengan bantuan logika *fuzzy* sebagai dasar pengambilan keputusan. Sistem koordinasi robot sepak bola ini telah diimplementasikan pada platform robot Devara. Dari hasil pengujian di lapangan, robot dapat mengenali bola, bergerak menuju bola, menggiring, dan menendang bola ke arah gawang. Waktu rata-rata robot untuk mencari bola hingga memasukkan gol ke gawang adalah 13,2 detik.**

Kata kunci – robot sepak bola, sistem cerdas berbasis perilaku, FSM, logika fuzzy.

I. INTRODUCTION

Perkembangan robot sepak bola sudah menjadi daya tarik bagi peneliti untuk terus mengembangkannya. Salah satu topik yang menjadi perhatian adalah sistem koordinasi robot dalam menentukan strategi dalam permainan sepak bola. Sistem koordinasi ini bertujuan untuk memberikan kemampuan robot untuk mencetak gol secara konsisten dalam waktu yang cepat dalam pertandingan sepak bola robot yang berdurasi 2x10 menit [1].

Dalam penentuan strategi permainan, terdapat beberapa metode yang biasa digunakan. Dalam 20 tahun terakhit, dua metode yang menjadi topik pengembangan adalah menggunakan *Artificial Intelligent* (AI) dan robot berbasis perilaku (*behavior based robot*) [2].

Metode AI bertujuan untuk menghasilkan mesin yang mampu menampilkan kecerdasan tingkat manusia. Mesin semacam itu diharuskan memiliki kemampuan untuk beralasan, membuat rencana untuk tindakan masa depan, dan untuk melakukan tindakan saat ini [3]. Metode ini memiliki keunggulan dalam kondisi permainan yang dinamis karena robot memiliki kemampuan untuk beradaptasi dengan lingkungan. Namun, metode ini memiliki hambatan karena kecerdasan tingkat manusia tentu sangat kompleks. Sehingga metode AI dapat diimplementasikan untuk *hardware* yang memiliki spesifikasi *processing* yang cukup tinggi.

Robot berbasis perilaku (*behavior based robot*) merupakan perilaku cerdas robot yang membagi aksi robot berdasarkan perilaku yang didefinisikan. Sistem ini dibangun dengan cara *bottom-up*, dimulai dari perilaku sederhana, banyak di antaranya mungkin berjalan bersamaan di otak robot tertentu, memberikan saran mengenai tindakan robot mana yang harus dilakukan [4]. Pada tingkat yang lebih tinggi, perilaku cerdas (*intelligent behavior*) dikembangkan sebagai kemampuan untuk mencapai suatu tujuan, di lingkungan yang tidak terstruktur. Lingkungan yang tidak terstruktur adalah lingkungan yang berubah dengan cepat dan tak terduga.

Pada riset ini, digunakan metode robot berbasis perilaku cerdas. Hal ini disebabkan implementasi dari riset ini menggunakan robot dengan kemampuan *processing* yang terbatas. Ditambah lagi, permainan sepak bola robot memiliki intensitas dan tempo permainan yang tinggi. Dengan keterbatasan ini, metode AI kurang cocok untuk digunakan karena sangat kompleks dan pengambilan keputusannya membutuhkan waktu yang cukup lama untuk kondisi *processor* yang rendah. Dalam implementasi robot berbasis perilaku, digunakan metode *Finite State Machine* (FSM) sebagai pengatur aliran perilaku robot dan logika *fuzzy* sebagai pengatur pengambilan gerakan robot [5].

Makalah ini memiliki garis besar sebagai berikut. Bab II membahas platform robot Devara yang digunakan dalam melakukan implementasi sistem. Bab III mendeskripsikan rancangan perilaku robot dan implementasinya dalam bentuk FSM dan logika fuzzy. Bab IV membahas hasil pengujian sistem koordinasi robot pada platform robot Devara. Terakhir, bab V menjelaskan kesimpulan yang diperoleh dari riset ini.

II. ROBOT PLATFORM

Platform robot yang digunakan pada implementasi riset ini adalah platform robot Devara yang dikembangkan oleh tim Dagozilla dari Institut Teknologi Bandung. Robot ini memiliki dimensi panjang x lebar x tinggi sebesar 40 cm x 48 cm x 65 cm dengan berat sebesar 14,7 kg.



Gambar 1. Platform Robot Devara

A. *Sistem Aktuator*

Robot menggunakan konfigurasi 4 motor lokomosi sebagai penggerak robot. Masing-masing motor memiliki spesifikasi torsi maksimum sebesar 110 oz-in dan kecepatan putar maksimum 350 rpm. Keempat motor ini telah dilakukan pengendalian sehingga robot dapat melakukan gerakan maju, mundur, geser kanan, geser kiri, dan berputar secara stabil.

Selain motor lokomosi, robot juga memiliki penggerak bola berupa penggiring dan penendang. Penggiring yang digunakan berupa penggiring aktif yang mampu memutar bola sesuai arah putar alaminya. Sehingga ketika robot bergerak, bola tidak tertahan / slip. Penendang yang digunakan memiliki sistem aktuator solenoida. Rata-rata tendangan yang dapat dilakukan robot adalah 5 meter.

Semua sistem actuator ini dikendalikan dalam *board* STM32 Nucleo F446RE. *Board* ini bertindak sebagai *sub controller* yang mengatur pergerakan sistem aktuator. *Sub controller* ini berkomunikasi dengan sebuah *main controller* sebagai pusat kendali robot dengan protocol komunikasi serial.

B. *Sistem Persepsi*

Robot menggunakan sistem persepsi berupa kamera *omnivision* yang diletakkan di atas robot, kamera depan, dan kompas. Ketiga sensor ini digunakan sistem koordinasi dalam pengambilan keputusan.

Kamera *omnivision* berguna untuk melihat obyek yang terletak dekat dengan robot. Kamera ini memiliki batasan radius deteksi sebesar 2 meter dari pusat robot. Output dari kamera ini adalah posisi bola di kamera dalam sistem koordinat *polar* (radius dengan nilai 0 pada pusat robot dan *theta* dengan nilai 0 pada orientasi depan robot). Sehingga untuk mengatasi posisi bola yang jauh, digunakan kamera depan yang hanya bisa menangkap bola pada satu arah hadap robot. Output dari kamera ini adalah posisi bola di kamera dalam sistem koordinat *kartesian* (XY dengan nilai 0,0 pada kiri atas). Kamera ini memiliki batasan jarak sebesar 7,2 meter.

Kedua kamera ini diproses menggunakan *library* openCV. Pemrosesan citra kedua kamera telah dirancang dan diimplementasikan dengan telah melewati proses pengujian sebelumnya.

Sensor kompas yang digunakan adalah modul cmps11. Modul ini memiliki output berupa nilai sudut hadap robot terhadap orientasi mata angin. Nilai sudut ini juga telah dilakukan pemrosesan untuk mengompensasi interferensi magnet yang ditimbulkan aktuator robot. Pemrosesan ini juga telah melewati proses pengujian sebelumnya.

C. *Sistem Main Controller*

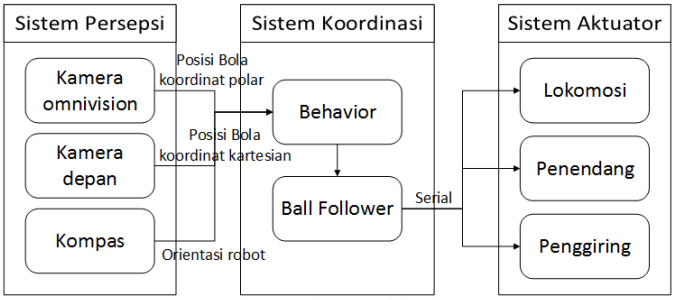
Main controller yang digunakan robot adalah *Single Board Computer* (SBC) Odroid XU4. SBC ini berfungsi sebagai pusat implementasi sistem koordinasi robot. Odroid XU4 memiliki spesifikasi sebagai berikut [6].

TABEL I. SPESIFIKASI ODROID XU4

Komponen	Spesifikasi
CPU	Samsung Exynos 5 Octa 2 GHz
GPU	ARM Mali-T628
RAM	2 GHz DDR3L
OS	Ubuntu Mate 16.04

III. RANCANGAN DAN IMPLEMENTASI PERILAKU ROBOT

Sistem koordinasi pada robot sepakbola merupakan sistem yang mengatur dan mengintegrasikan sistem yang dimiliki oleh robot sepakbola. Sistem koordinasi ini mencakup strategi robot dalam melakukan permainan sepakbola. Sistem yang dimiliki robot sepakbola ini berupa sistem persepsi sebagai sensor robot dalam mengenali lingkungan dan sistem aktuator sebagai penggerak robot.

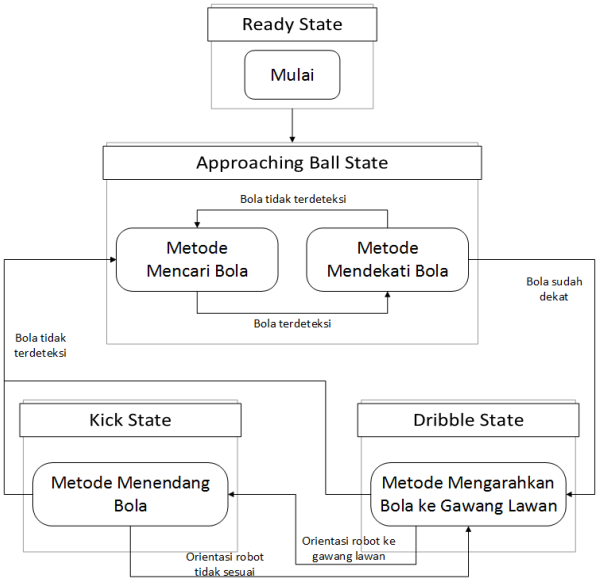


Gambar 2. Diagram Blok Sistem

A. *Behavior*

Strategi permainan sepakbola yang dilakukan robot menggunakan paradigma robot berbasis perilaku. Pada paradigma ini, robot memiliki *state* yang merepresentasikan perilaku yang harus dilakukan. Perilaku robot ini didefinisikan dalam state yang diatur oleh sebuah FSM. Adapun dalam melakukan penentuan gerakan yang dilakukan robot, digunakan metode logika *fuzzy*.

Implementasi dari FSM robot berupa modul *behavior* sebagai pusat perilaku robot. Model FSM yang digunakan adalah *Mealy machine*, sehingga aliran *state* dipengaruhi oleh nilai variable dan state nya setiap saat. Oleh karena itu, nilai dari masing-masing *variable* ini mempengaruhi *state* dari robot. Berikut *state* robot pada modul *behavior*.



Gambar 3. Diagram Blok Perilaku Robot

Masing-masing *state* dari modul *behavior* ini saling terhubung. Kondisi ini memastikan setiap posisi bola dan kompas dapat berjalan sesuai dengan state yang seharusnya. Selanjutnya, metode yang dilakukan robot pada setiap *state* didefinisikan pada modul *Ball Follower*.

Pada FSM ini, setiap *state* selalu melakukan pengecekan kondisi bola di kamera. Setiap bola hilang dari deteksi baik kamera depan maupun kamera *omnivision*, maka state dikembalikan ke *approaching ball state*. Metode robot dalam mencari bola juga ditempatkan di *approaching ball state*, tidak menjadi *state* berbeda. Hal ini dijelaskan pada pembahasan sebagai berikut.

1) Robot menggunakan kamera *omnivision* yang dapat menjangkau pandangan 360° dari robot. Dapat diasumsikan bahwa bola yang berada di seluruh radius jarak pandang kamera *omnivision* pasti dapat dideteksi oleh robot. Sedangkan kamera depan hanya dapat mendeteksi untuk bola pada jarak jauh dengan jarak minimal tertentu pada arah depan robot. Ketika bola tidak terdeteksi oleh kedua kamera, maka dapat disimpulkan bahwa bola berada cukup jauh dari robot hingga tidak terdeteksi kamera depan dan tidak berada pada arah hadap robot.

2) Metode mendekati bola berada pada *approaching ball state*. Sehingga dengan kesimpulan sebelumnya, maka metode mencari bola tidak dapat dilepaskan dari metode mendekati bola. Hal ini disebabkan kedua kondisi ini sering terjadi pertukaran kondisi dimana kamera depan kehilangan bola karena pergerakan bola dan robot yang cepat. Penjelasan ini yang mendasari penempatan metode mencari bola pada *approaching ball state*.

B. Ball Follower

Modul *Ball Follower* berfungsi sebagai implementasi *path planning* yang harus dilakukan robot dari *state Behavior*. Untuk dapat melakukan *path planning*, *ball follower* menggunakan nilai posisi bola di kamera dan nilai kompas sebagai acuan nilai gerakan yang harus dilakukan. Adapun untuk mengonversi nilai posisi bola dan kompas menjadi gerakan, digunakan metode logika *fuzzy*. Kontroler ini memiliki acuan error jarak dan sudut bola terhadap robot dari hasil kamera *omnivision*. Nilai error ini yang menentukan nilai gerakan yang dilakukan.

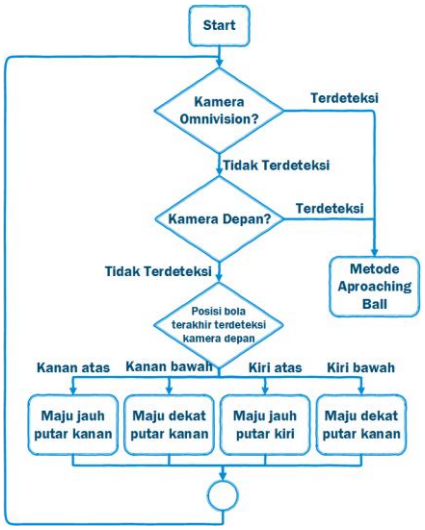
Untuk dapat melakukan gerakan, terdapat tiga elemen gerakan dasar yang diberikan kepada modul lokomosi. Adapun aturan nilai kecepatan ini sebagai berikut.

1) $v_x > 0$ arah maju, $v_x < 0$ arah mundur
2) $v_y > 0$ arah kiri, $v_y < 0$ arah kanan
3) $v_a > 0$ arah putar clockwise, $v_a < 0$ arah putar counter clockwise

Berikut proses *Ball Follower* yang dilakukan.

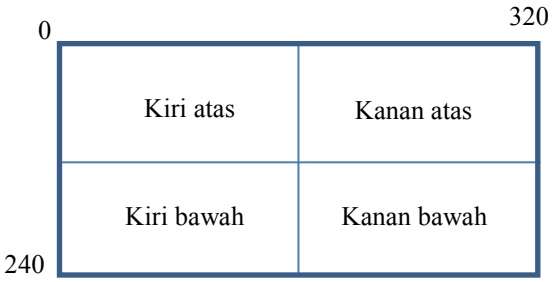
1) Proses Mencari Bola

Pada metode ini, dilakukan *path planning* robot dalam usaha mencari bola yang tidak dapat terdeteksi. Hal ini dapat disebabkan bola berada pada radius di luar kemampuan *vision* robot ataupun bola tertutupi oleh robot lain. Untuk melakukan *searching ball* ini, metode yang dilakukan adalah sebagai berikut.



Gambar 4. Diagram blok metode mencari bola

Dalam menentukan arah putar orientasi, digunakan logika *fuzzy* dengan pembagian logika berdasarkan terakhir kali kamera depan mendeteksi bola.



Gambar 5. Diagram arah putar orientasi dengan logika *fuzzy*

Aturan *fuzzy* kamera depan dibagi menjadi 4.

a) X < 160 dan Y < 120 daerah kiri atas

b) X > 160 dan Y < 120 daerah kanan atas

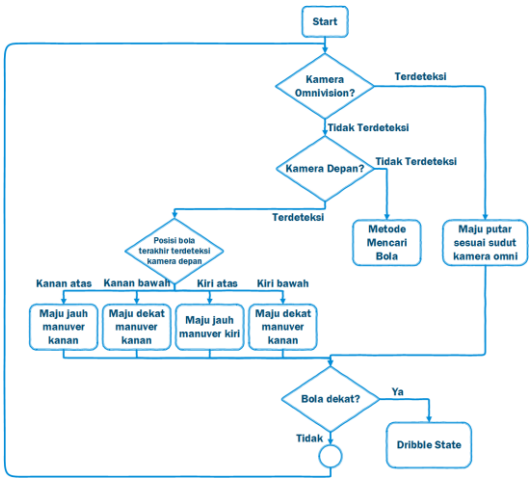
c) X < 160 dan Y > 120 daerah kiri bawah

d) X > 160 dan Y > 120 daerah kanan bawah

Untuk menghasilkan gerakan putar orientasi robot, digunakan parameter v_x dan v_y. Terakhir ketika bola telah terdeteksi oleh salah satu kamera, robot memanggil metode mendekati bola.

2) Proses Mendekati Bola

Pada metode ini, dilakukan *path planning* robot dalam usaha mendekati bola yang telah terdeteksi. Untuk mendekati bola, metode yang dilakukan adalah sebagai berikut.



Gambar 6. Diagram blok metode mendekati bola

Adapun aturan *fuzzy* yang digunakan sama dengan aturan *fuzzy* mencari bola. Untuk menghasilkan gerakan manuver, digunakan tiga komponen kecepatan sekaligus dengan aturan sebagai berikut.

a) Maju jauh manuver kanan: v_x positif, v_a positif, v_y positif

b) Maju dekat manuver kanan: v_x positif, v_a positif, v_y positif

c) Maju jauh manuver kiri: v_x positif, v_a negatif, v_y negatif

d) Maju dekat manuver kiri: v_x positif, v_a negatif, v_y negatif

Kemudian ketika bola sudah terdeteksi kamera *omnivision*, maka robot maju dengan memutar orientasi. Adapun nilai putaran robot bergantung pada sudut bola di kamera *omnivision*. Terdapat aturan *fuzzy* sebagai berikut.

a) Sudut < 5°, setpoint = MaxRLTurn

b) Sudut > 5° dan Sudut < 14°, setpoint = 100

c) Sudut > 14° dan Sudut < 25°, setpoint = 200

d) Sudut > 25° dan Sudut < 36°, setpoint = 250

e) Sudut > 36° dan Sudut < 45°, setpoint = 300

f) Sudut > 45°, setpoint = 350

Setiap kategori *fuzzy* di atas memiliki besaran kecepatan putar yang berbeda. Nilai setpoint ini digunakan untuk mengonversi sudut kamera menjadi sudut putar. Adapun persamaan konversi ini sebagai berikut.

$$v_a = \frac{\theta}{180} x(FollowMaxRLTurn - setpoint)$$

Nilai sudut kamera dibagi dengan sudut putar maksimum yang bernilai 180° (karena sistem sudut dari -180° sampai 180°) dikalikan dengan nilai kecepatan putar maksimum robot (MaxRLTurn) dikurangi dengan *setpoint*. Adapun nilai *setpoint* yang digunakan berdasarkan aturan *fuzzy* di atas. Penentuan nilai setpoint ini diperoleh dengan metode *tuning*.

3) Proses Menggiring Bola

Pada metode ini, dilakukan path planning robot dalam usaha menggiring bola ke arah gawang lawan. Ketika menjalankan metode ini, bola dalam keadaan dekat dengan robot dan telah masuk ke penggiring robot. Untuk melakukan *dribble*, robot melakukan *pivot* dengan besar kecepatan putar sudut orientasi kompas untuk gawang (*variable* cmps_error). Adapun untuk mengonversi nilai sudut kompas menjadi kecepatan putar, digunakan persamaan berikut.

$$v_a = \frac{\theta}{180} xKp$$

Nilai sudut kompas dibagi dengan sudut putar maksimum yang bernilai 180° (karena sistem sudut dari -180° sampai 180°) dikalikan dengan Kp. Adapun nilai Kp yang digunakan adalah 1.5. Nilai Kp ini diperoleh dengan metode *tuning*.

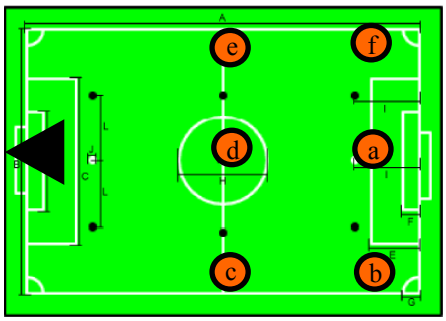
4) Proses Menendang Bola

Pada metode ini, robot melakukan tendangan. Sama seperti metode *dribble*, robot mengecek kondisi bola masih dalam penggiring atau tidak. Ketika bola keluar penggiring, maka robot mengubah *state* menjadi *approaching ball*. Kemudian robot mengecek orientasi robot ke gawang lawan. Apabila tidak sesuai, maka robot mengubah *state* menjadi *dribble state*. Apabila orientasi ini telah sesuai, maka robot memanggil penendang untuk mengaktifkan tendangan.

IV. KONFIGURASI DAN HASIL PENGUJIAN

A. *Kondisi Pengujian*

Pengujian dilakukan pada platform robot Devara seperti yang telah dijelaskan pada Bab II. Adapun pengujian dilakukan pada lapangan yang sesuai dengan standar Robocup, yaitu karpet berwarna hijau dengan ukuran 7.2 m x 4.8 m. Pengujian dilakukan dengan berbagai kondisi untuk mengetes kemampuan pergerakan robot sebagai berikut.



Gambar 7. Konfigurasi robot dan bola pada pengujian integrasi sistem

B. *Hasil Pengujian*

Pengujian pertama dilakukan untuk pengujian proses mendekati bola. Pengujian dilakukan dengan meletakkan bola pada salah satu posisi di lapangan di atas dan menjalankan robot untuk mendekati bola secara otomatis. Dilakukan pengamatan hasil deteksi kamera atas ketika robot mendekati bola.



Gambar 8. Konfigurasi robot dan bola pada pengujian integrasi sistem

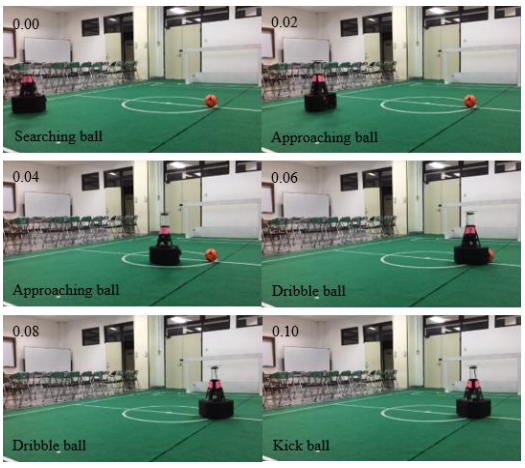
Dari hasil di atas, diperoleh bahwa ketika robot mendekati bola dan kamera atas mulai mendeteksi bola, nilai sudut error sebanyak 22^o. Kemudian robot memperbaiki error sudut ini hingga bola dapat masuk ke penggiring. Nilai error sudut yang terjadi terus mendekati 0 dengan osilasi nilai error yang relatif halus.

Pengujian kedua dilakukan untuk integrasi sistem koordinasi secara keseluruhan. Pengujian menggunakan konfigurasi pada poin A seperti yang dijelaskan sebelumnya.

TABEL II. HASIL PENGUJIAN SISTEM KOORDINASI

Titik	Gol	Waktu (s)
a	Ya	11.11
b	Ya	12.63
c	Ya	15.31
d	Ya	12.21
e	Ya	14.57
f	Ya	13.37
Rata-rata		13.2

Berikut timestamp dari perilaku robot ketika melakukan pengujian ketika bola berada di titik c.



Gambar 9. Timelapse pengujian sistem koordinasi robot

Dari hasil pengujian di atas, diperoleh bahwa robot dapat mencetak gol sesuai algoritma yang telah diimplementasikan. Adapun rata-rata waktu yang dibutuhkan untuk mencetak gol dari kondisi mencari bola adalah 13.2 detik.

KESIMPULAN

Pada riset ini, sistem koordinasi berbasis perilaku telah berhasil diimplementasikan pada robot sepak bola. Metode perilaku diimplementasikan dalam bentuk *Finite State Machine* dan Logika *Fuzzy*. Untuk memperoleh hasil gerakan yang halus, diperlukan pengaturan parameter gerakan robot. Berdasarkan pengujian yang dilakukan pada platform robot Devara, robot dapat melakukan gerakan mendekati bola dengan halus dan robot dapat mencetak gol dalam waktu rata-rata 13,2 detik.

Pada riset yang dilakukan juga memiliki konstrain yang membatasi. Ukuran lapangan yang digunakan adalah 7,2 m x 4,8 m. Apabila ukuran lapangan diperbesar, diperlukan perilaku proses mendekati bola yang berbeda untuk dapat mengatasi semua kondisi bola di lapangan.

UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih kepada tim Dagozilla ITB yang telah bekerja sama dan memberikan izin untuk pengembangan platform robot Devara.

REPERENSI

[1] M. Asada, T. Balch, A. Bonarini, A. Bredenfeld, and S. Gutmann, "Middle Size Robot League Rules and Regulation", in International MSL Workshop, Kassel, Germany, 2016.

[2] Ronald C. Arkin, "Behavior-based Robotics", Mobile Robot Laboratory, College of Computing, Georgia Institute of Technology, United States, 1998.

[3] E. Mansury, A. Nikookar, M. Salehi, "Artificial Bee Colony of ferguson splines for soccer robot path planning", First RSI/ISM International Conference on Robotics and Mechatronics, Iran, 2013.

[4] M.Wahde, "Behavior-based Robotics", Applied Mechanics Chalmers University of Technology, Sweden, 2007.

[5] Z. Pi, Z. Tang, Z. Pei, J. Xue, and S. Yao, " Role assignment for Soccer Robot using fuzzy inference system", Seventh International Conference on Fuzzy Systems and Knowledge, China, 2010

[6] Hardkernel, "Odroid-XU4 User Manual", Korea, 2017.

Rancang Bangun Sistem Lokomosi, Penggiring, dan Penendang pada Robot Sepak Bola

Faiz Anhar Widodo*, Kusprasapta Mutijarsa
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
faizaw@s.itb.ac.id

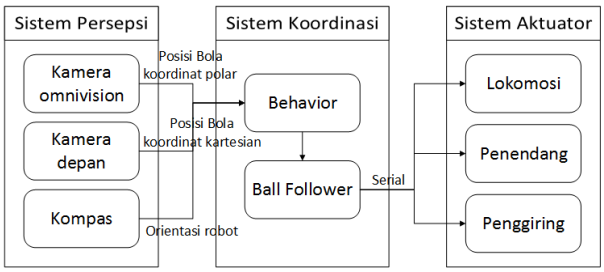
***Abstrak* – Sistem aktuator pada robot sepak bola terdiri atas sistem lokomosi, sistem penggiring, dan penendang. Keseluruhan sistem lokomosi, penggiring dan penendang pada robot sepak bola berhasil dibangun dan diintegrasikan pada robot sepak bola. Berdasarkan hasil uji fungsi robot keseluruhan dalam bermain bola, diperoleh sistem lokomosi dapat melakukan gerakan sesuai dengan instruksi yang diberikan dengan galat rata-rata kurang dari 5%. Sistem penggiring bola dapat menjaga bola selalu di depan robot dan berputar sesuai dengan gerak alami bola. Sistem penendang dapat menendang bola dengan jarak rata-rata 6.15 meter.**

Kata kunci: mobile robot, robot sepak bola, lokomosi, penggiring, penendang

I. PENDAHULUAN

Perkembangan robot saat ini menjadi sebuah topik yang sangat hangat untuk riset. Salah satu topik yang terus dikembangkan dalam riset robot adalah robot sepak bola. Robot sepak bola sendiri terdiri atas dua macam yaitu robot sepak bola humanoid dan robot sepak bola beroda. Pada penelitian ini akan dibahas mengenai robot sepak bola yang menggunakan pergerakan sistem roda.

Pada robot sepak bola beroda sendiri terdapat berbagai hal yang dapat diteliti, di antaranya yaitu sistem mekanik sasis robot, sistem persepsi, sistem penggerak/akuator, sistem *intelligence* dan koordinasi. Sistem penyusun robot sepak bola diperlihatkan pada Gambar 1.



Gambar 1. Diagram blok sistem robot sepak bola

Pada penelitian ini dilakukan rancang bangun sistem robot sepak bola dengan penekanan pada sistem lokomosi, sistem penggiring dan sistem penendang. Pembahasan mencakup desain, implementasi, hingga pengujian sistem yang dibuat.

II. SISTEM LOKOMOSI ROBOT SEPAK BOLA

Lokomosi didesain agar robot sepak bola dapat bergerak dengan efektif dan efisien di lapangan sepak bola robot. Pertimbangan yang tidak kalah penting adalah ketahanan sistem lokomosi terhadap benturan dengan objek di lapangan, yaitu bola, gawang, dan robot lain. Lokomosi juga dirancang dengan memperhatikan ketersediaan komponen di pasaran, untuk menjamin ketersediaan untuk penggantian komponen cadangan.

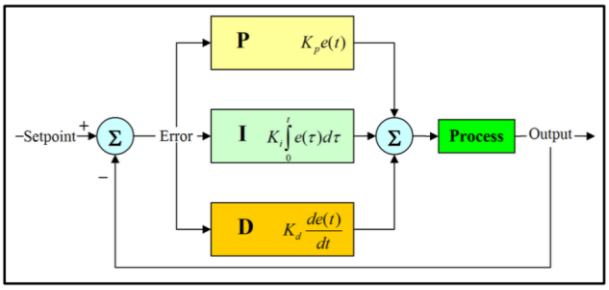
Robot sepak bola terdapat berbagai variasi, dari robot berbentuk humanoid, robot menggunakan roda berukuran kecil (*micro robot*) hingga berukuran besar. Pada robot sepak bola beroda terdapat beberapa mekanisme pergerakan, antara lain: *fixed wheel, centered orientable wheel, off-centered orientable wheel / caster wheel*, dan *omnidirectional wheel* yang masing-masing memiliki keunggulan dan kelemahan.

Pada robot sepak bola ini, sistem lokomosi menggunakan roda jenis *omnidirectional*. Robot dengan roda *omnidirectional* dapat bergerak ke segala arah dengan kecepatan dan akselerasi yang sama. Hal ini akan sangat berguna mengingat posisi bola relatif terhadap robot akan sangat variatif..

Desain sistem *omnidirectional* yang memungkinkan untuk diimplementasikan pada robot adalah dengan konfigurasi empat roda, yang masing-masing roda dipasang berbeda 90 derajat. Dengan desain ini, didapatkan akselerasi yang sama ke segala arah yang diinginkan. Roda yang digunakan pada sistem ini diharuskan untuk dapat berotasi baik tegak lurus maupun pararel dengan arah geraknya. Gerak rotasi tersebut akan dapat dicapai dengan menggunakan *omni-wheel*.

A. *Kendali PID*

Sistem kendali *proportional integral derivative* (PID) merupakan salah satu sistem kendali yang paling umum diimplementasikan untuk berbagai sistem yang ada. Bentuk umum dari kendali PID adalah sebagai berikut.



Gambar 2. Diagram blok sistem PID

Kontrol PID dapat dirumuskan pada persamaan berikut: [5]

$$\boldsymbol{u(t)= Kp(e(t)+(1/\tau i)\int e(t)dt+\tau d(de(t)/dt))} \quad (1)$$

dimana,

u(t) : perintah ke proses/ sistem

e(t) : perbedaan antara nilai keluaran dengan perintah/ acuan

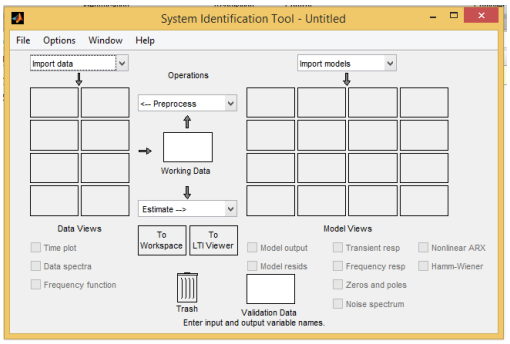
Kp : konstanta proporsional

τi : waktu integral, dimana konstanta integral (Ki) = Kp/τi

τd : waktu derivative, dimana konstanta derivative (Kd) = Kp*τd

dt : waktu pencuplikan data

Proses kendali PID pada robot sepak bola diawali dengan pencarian fungsi transfer motor. Kepada semua motor diberikan *step response* dan data dari *encoder* dicatat pada *spreadsheet*. Kemudian dari data input dan output tersebut, dicari modelnya menggunakan MATLAB® System Identification Tools® .

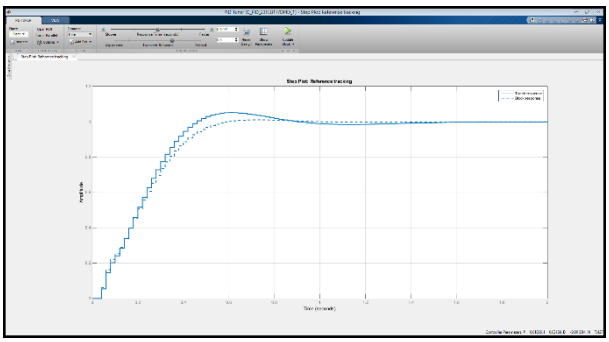


Gambar 3. Tampilan MATLAB® System Identification Tools®

Setelah mendapatkan estimasi fungsi transfer dari motor, maka selanjutnya dapat dilakukan pemodelan sistem *closed loop* dengan pengendali PID pada MATLAB® SIMULINK®. Dengan simulasi tersebut, dilakukan optimasi dari konstanta Proporsional, Integral, dan Derivatif sehingga didapatkan sistem dengan spesifikasi sesuai dengan yang diinginkan. Dalam hal ini diinginkan sistem dengan *overshoot* tidak lebih dari 20%, *rising time* dan *settling time* kurang dari 0.8 detik.



Gambar 4. Diagram blok untuk keempat motor



Gambar 5. Tampilan PID Tuner

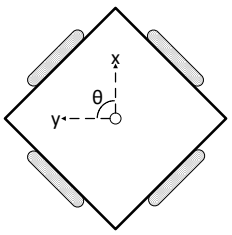
TABEL I. KONSTANTA SISTEM PID

Motor	1	2	3	4
Kp	1.399	4.345	-0.1574	8.104
Ki	1.7623	18.1802	-0.101	15.318
Kd	1.5687	5.8424	-0.1529	9.5837
N	1.9114	14.387	-0.1542	11.2567

Setelah konstanta tiap motor didapatkan, kemudian konstanta tersebut diimplementasikan pada *source code*.

B. *Kinematika*

Robot sepak bola akan diatur strateginya oleh fitur *main controller* (yang tidak dibahas pada riset ini). *Main controller* akan mengirimkan perintah berupa kecepatan skalar yang terdiri atas 3 kecepatan yaitu Vx, Vy, dan Vrot.



$$v_x(t) = \frac{dx(t)}{dt} \quad (2)$$

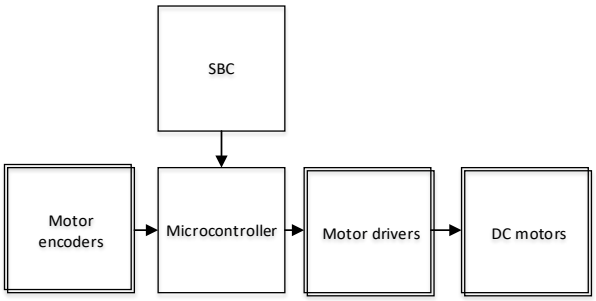
$$v_y(t) = \frac{dy(t)}{dt} \quad (3)$$

$$\omega(t) = \frac{d\theta(t)}{dt} \quad (4)$$

Gambar 6. Kinematika 4 *omniwheel*

Kemudian sistem lokomosi mengubah kecepatan yang diberikan tersebut menjadi kecepatan tiap motor menggunakan formulasi sebagai berikut[1].

$$\begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \\ v_4(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & d \\ -1 & 0 & d \\ 0 & -1 & d \\ 1 & 0 & d \end{bmatrix} \cdot \begin{bmatrix} v_x(t) \\ v_y(t) \\ \omega(t) \end{bmatrix} \quad (5)$$

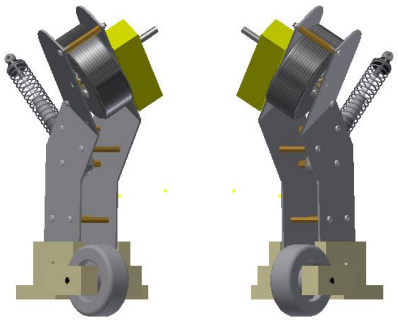


Gambar 7. Diagram blok sistem lokomosi

III. SISTEM PENGGIRING ROBOT SEPAK BOLA

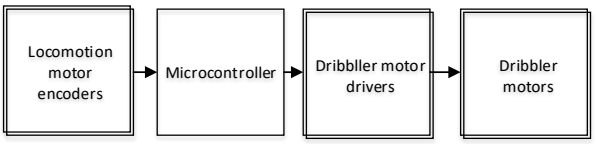
Penggiring pada *Soccer Robots (*Robot Sepakbola) yang didesain merupakan penggiring aktif. Maksud penggiring aktif di sini adalah penggiring tersebut menggunakan aktuator. Aktuator yang digunakan pada penggiring bola adalah *DC motor* 12 V. Penggiring didesain agar bola yang telah didapat oleh robot tidak mudah direbut oleh lawan. Penggiring di sini hanya boleh memegang sepertiga bagian bola agar permainan tetap menarik. Penggiring juga di desain agar dapat menggiring bola sesuai dengan arah alami bola saat robot bergerak.

Untuk dapat melakukan giringan yang sedemikian maka diperlukan sistem penggiring yang andal. Modul penggiring didesain terdiri atas chasis penggiring, roda penggiring, motor penggiring, dan roda omni. Motor penggiring dan roda penggiring akan mengenai bola bagian atas dan roda omni akan menyentuh bola bagian bawah. Casis penggiring dibutuhkan untuk memasang roda dan motor pada *base* robot. Bentuk dari arah roda penggiring bola ini berbentuk V agar dapat menangkap bola yang mengenai penggiring dan agar bola tidak lepas saat roda melakukan giringan[4].



Gambar 8. Desain CAD sistem penggiring

Penggiring yang diinginkan adalah penggiring yang dapat mengatur arah putaran bola sesuai dengan arah gerak alami bola saat robot bergerak. Oleh karena itu, kontrol motor penggiring akan bergantung pada lokomosi. Kontrol motor penggiring mengambil kecepatan dan arah lokomosi sebagai inputnya, dan mengeluarkan output untuk diteruskan ke modul *motor driver* penggiring.

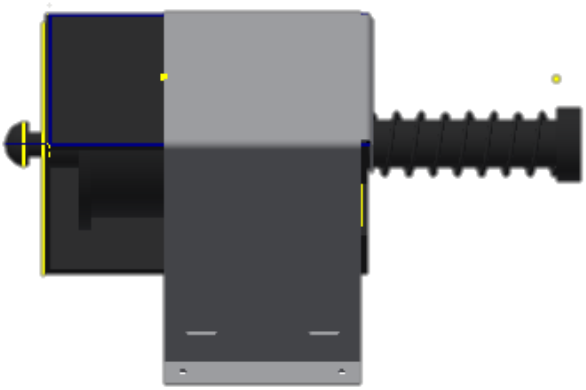


Gambar 9. Diagram blok sistem penggiring

IV. SISTEM PENENDANG ROBOT SEPAK BOLA

Robot sepakbola dirancang agar dapat menendang bola sehingga robot dapat membuat gol ke gawang lawan dengan lebih cepat daripada harus menggiring bola hingga ke gawang lawan. Terdapat beberapa jenis aktuator yang dapat dijadikan sistem penendang. Beberapa di antara itu adalah mekanisme pegas, mekanisme pneumatik, dan mekanisme solenoida. Dari ketiga mekanisme ini memiliki karakteristik masing-masing seperti kekuatan tendangan, berat, ruang kosong yang dibutuhkan, lama durasi tiap tendangan, keamanan, jumlah tendangan yang bisa dilakukan, dan biaya pembuatan [2].

Mekanisme yang dipilih pada sistem penendang bola robot sepakbola adalah mekanisme solenoida. Mekanisme solenoida memiliki kompleksitas yang cukup tinggi. Mekanisme solenoida ini memanfaatkan gaya Lorentz. Batang besi akan diletakkan di tengah kumparan tembaga. Saat kumparan tersebut diberikan tegangan maka besi akan bergerak masuk ke kumparan [3].

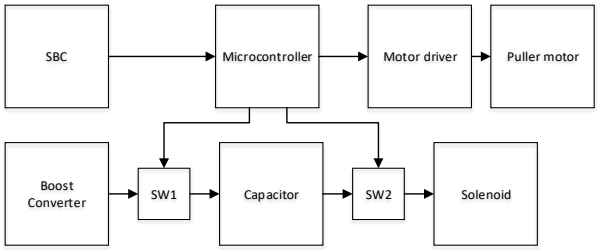


Gambar 10. Desain CAD sistem penendang

Jumlah tendangan yang dapat dilakukan oleh mekanisme ini hanya bergantung pada energi yang dimiliki oleh baterai. Kekuatan tendangan pada mekanisme solenoida dipengaruhi oleh jumlah energi yang dimiliki solenoida. Gaya yang dapat dikeluarkan oleh mekanisme solenoida dapat kita lihat menggunakan formula Lorentz.

$$F = B \times l \times I$$

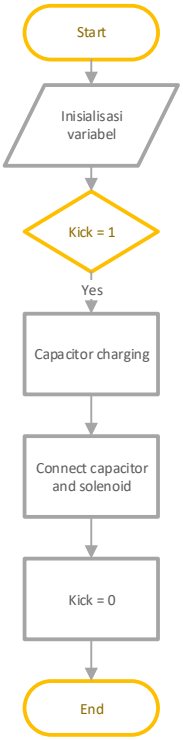
B adalah *flux density* [T], l adalah panjang solenoid [m], dan I adalah arus yang dialirkan pada solenoida tersebut [A].



Gambar 11. Diagram blok sistem penendang

Modul penendang akan menerima input dari *main controller* kapan saat robot butuh tendangan. Mikrokontroler akan mengartikan komunikasi serial yang diberikan dari main controller untuk diteruskan kepada modul penendang. Untuk melakukan sebuah tendangan, terdapat tiga hal yang perlu dilakukan.

Hal pertama yang dilakukan adalah mengisi kapasitor. Hal ini dilakukan dengan menutup rangkaian antara *boost converter* dan kapasitor. Hal kedua yang yang dilakukan adalah memberikan daya dari kapasitor ke solenoida. Hal ini dilakukan dengan menutup rangkaian antara kapasitor dan solenoida. Hal ketiga yang dilakukan adalah melakukan tarikan pada *plunger* / batang besi pada solenoida. Hal ini dilakukan dengan memberikan sinyal PWM positif pada motor penarik *plunger*. Setelah *plunger* ditarik, sinyal PWM negatif diberikan agar *plunger* solenoida siap untuk melakukan tendangan berikutnya.

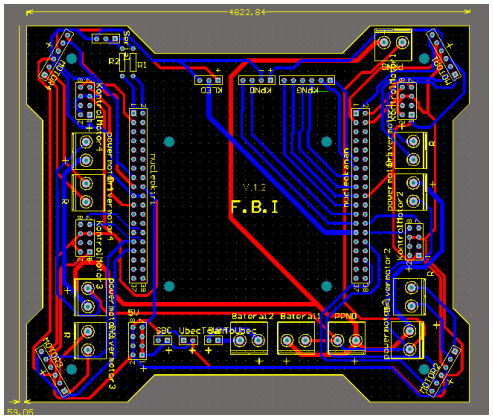


Gambar 12. Diagram alir sistem penendang

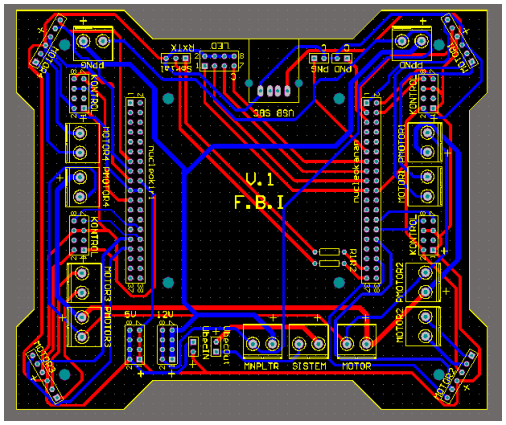
V. MIKROKONTROLER

Pada robot sepak bola, modul ini merupakan modul yang menghubungkan modul SBC ke modul lokomosi, modul penggiring, dan modul penendang. Modul ini akan menerima perintah-perintah dari modul *single-board-computer*, menerjemahkan perintah-perintah yang diterima, kemudian mengirim sinyal kontrol hasil komputasi ke motor lokomosi, sistem penggiring, dan sistem penendang.

Modul ini menggunakan microkontroler Nucleo-64 STM32F446RE berbasis ARM® mbed 32 bit. Nucleo-64 menggunakan mikroprosesor ARM® Cortex®-M4 CPU dengan frekuensi mencapai 180 MHz. Nucleo-64 dapat menampung memori sebesar 512 KB untuk Flash dan 128 KB untuk SRAM. Mikrokontroler ini memiliki *timer* berjumlah hingga 17 buah.



Gambar 13. Desain PCB dengan *route* pada *top layer*

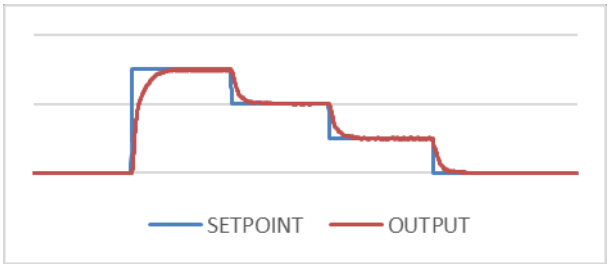


Gambar 14. Desain PCB dengan *route* pada *bottom layer*

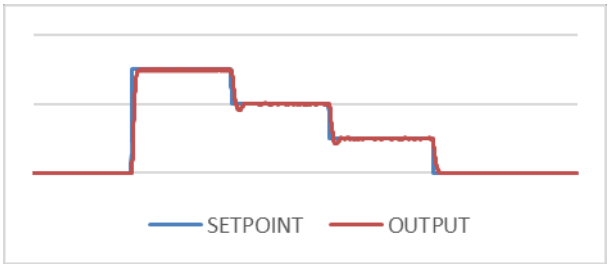
VI. PENGUJIAN SISTEM

A. *Sistem Lokomosi*

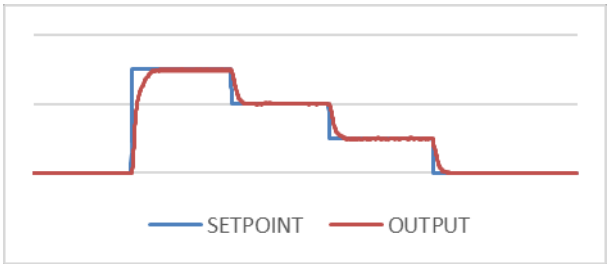
Terdapat dua hal yang diuji pada sistem lokomosi yaitu hasil kendali PID dan hasil gerakan dasar. Pengujian pertama dilakukan dengan memberikan empat setpoint dengan jangka waktu 3 detik untuk masing-masing setpoint. Empat setpoint tersebut yaitu kecepatan 100%, 75%, 50%, dan 0%.



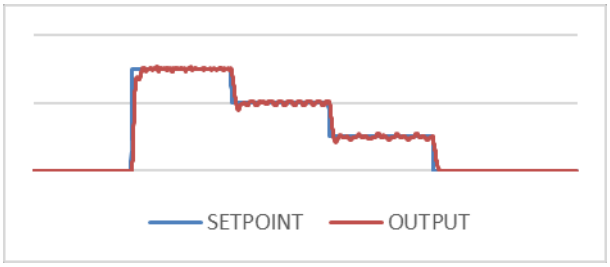
Gambar 15. Hasil pengujian motor 1



Gambar 16. Hasil pengujian motor 2



Gambar 17. Hasil pengujian motor 3



Gambar 18. Hasil pengujian motor 4

Hasil menunjukkan bahwa kendali PID telah berhasil dilakukan dan sesuai spesifikasi yang diinginkan yaitu *rise time* dan *settling time* kurang dari 0.8 detik serta overshoot kurang dari 20%.

Percobaan berikutnya yaitu robot diberikan perintah untuk melakukan gerakan dasar seperti maju, mundur, gerak ke kanan dan gerak ke kiri. Hasil pengujian ditunjukkan pada tabel berikut.

TABEL II. HASIL PENGUJIAN GERAKAN DASAR

Gerakan	**Jarak Diinginkan (m)**	**Jarak Ditempuh (m)**	**Galat Miring (cm)**
Maju	4.75	4.70	8
Mundur	4.75	4.72	5
Kanan	4.75	4.70	10
Kiri	4.75	4.69	4

Hasil menunjukkan bahwa gerakan dasar dapat dilakukan dengan galat kurang dari 5%.

B. Sistem Penggiring

Pengujian penggiring dilakukan dengan menaruh bola di depan robot, kemudian robot diberikan perintah untuk melakukan pergerakan ke beberapa arah. Hasil pengujian ditunjukkan pada tabel berikut.

TABEL III. HASIL PENGUJIAN SISTEM PENGGIRING

Gerakan Robot	Bola Lepas atau Tidak	Arah Putaran Bola
Maju 4 meter	Tidak	Maju
Mundur 4 meter	Tidak	Mundur
Kanan 4 meter	Tidak	Kanan
Kiri 4 meter	Tidak	Kiri
Rotasi Searah Jarum Jam	Tidak	Kanan
Rotasi Berlawanan Arah Jarum Jam	Tidak	Kiri

Hasil pengujian menunjukkan bahwa sistem penggiring dapat berfungsi dengan baik pada robot sepak bola. Dengan ini, maka bola yang telah didapat oleh robot saat permainan tetap dalam kendali robot dan menjadi lebih sulit bagi tim lawan untuk merebut.

C. Sistem Penendang

Pengujian sistem penendang diawali dengan menguji sistem *capacitor charging* apakah telah berfungsi dengan baik. Hasil menunjukkkan bahwa sistem dapat memberikan daya kepada kapasitor hingga 77.5 Volt.



Gambar 19. Hasil pengujian pengisian kapasitor

Pengujian berikutnya adalah pengujian fungsi dan jarak tendangan. Sistem penendang yang telah terintegrasi dengan robot diberikan perintah untuk menendang bola yang telah diposisikan di depan robot. Hasil pengujian ditunjukkan oleh tabel berikut.

TABEL IV. HASIL PENGUJIAN SISTEM PENENDANG

Tendangan Ke-	Berhasil	Jarak Lurus (m)	Jarak Samping (cm)	Penarik
1	Ya	6.5	20	Berfungsi
2	Ya	7.2	9	Berfungsi
3	Ya	5.1	330	Berfungsi
4	Tidak	-	-	Berfungsi
5	Ya	5.8	50	Berfungsi

Hasil pengujian menunjukkan bahwa sistem penendang telah dapat berfungsi walaupun sempat terjadi 1x gagal tendang. Hal ini dikarenakan bug yang terjadi saat sinyal *switch* antara solenoida dan kapasitor ditutup. Jarak rata-rata tendangan adalah 6.15 meter.

VII. KESIMPULAN

Berdasarkan pengujian dan analisis terhadap proses yang dibuat pada rancang bangun sistem lokomosi, penggiring, dan penendang pada robot sepak bola ini didapatkan kesimpulan sebagai berikut.

1) Tiap motor pada sistem lokomosi memiliki *rise time* dan *settling time* kurang dari 0.8 s serta *overshoot* kurang dari 20%.
2) Gerakan dasar lokomosi telah berfungsi dengan galat kurang dari 5%.
3) Sistem penggiring dapat membuat bola selalu di depan robot.
4) Sistem penggiring dapat membuat gerakan bola selayaknya gerakan giringan alami saat robot bergerak.
5) Sistem penendang dapat melakukan tendangan dengan jarak rata-rata 6.15 meter.

DAFTAR PUSTAKA

[1] Helder P. Oliveira, Armando J. Sousa, A. Paulo Moreira, Paulo J. Costa. “Modeling and Assessing of Omni-directional Robots with Three and Four Wheels”. Universidade do Porto.

[2] S.Hamidreza Mohades Kasaei, S. Mohammadreza Mohades Kasaei, S.Alireza Mohades Kasaei. “Design and Implementation Solenoid Based Kicking Mechanism for Soccer Robot Applied in Robocup-MSL”. International Journal of Advanced Robotic Systems. 2010.

[3] Fernando Ribeiro, Gil Lopes, Bruno Martins, Pedro Silva, Ivo Moutinho, Paulo Braga, Nino Pereira. “Magnetically Impelled Kicker for Robotic Football in MSL RoboCup”. Universidade do Minho. 2016.

[4] Shota Chikushi, Tharindu Weerakoon, Takashi Sonoda, Kazuo Ishii. “Ball Dribbling Control for RoboCup Soccer Robot”. International Conference on Artificial Life and Robotics. 2017

[5] Kiam Heong Ang, G. Chong, Yun Li. “PID Control System Analysis, Design, And Technology”. IEEE Transactions on Control Systems Technology. 2005

URT-ROSO

Daniyah*, La Ode Muhamad Idris
Department of Electrical Engineering
Islamic University of Sultan Agung
Semarang, Indonesia
Email: deluca@std.unissula.ac.id

***Abstract*— This paper details the URT-ROSO design as a Soccer Robot, created in preparation for Kontes Robot Indonesia 2017 in Bandung, Indonesia. Some improvements made from the kicker to the ball tracker. The main focus in the discussion is the solution of the concept of a kicker uses a solenoid to get a power that capable for kicking a ball. Then secondary focus will discuss about ball tracking using android app as media and how the robot using three wheels as its propulsion can recognize its position.**

Keywords—soccer robot; solenoid; ball tracking; android; robot positioning

I. INTRODUCTION

URT-ROSO is the KRSBI division of the robotics team in Islamic University of Sultan Agung. URT-ROSO is structured to follow the Kontes Robot Indonesia where the robot soccer using wheels (such as in RoboCup MSL) is a new division in Kontes Robot Indonesia. URT-ROSO underwent several changes after the league in the Kontes Regional III, the changes made are not too significant. For example some change are in the kicking mechanism. This paper will discuss about URT-ROSO with focus on mechanical explanation, kicker, and also how image processing done by robot in detecting ball.

II. ROBOT MECHANISMS

A. Basic Mechanic

On URT-ROSO, for striker robot uses a design with three wheels for the movement or called omnidirectional which has three degrees of freedom while for kipper using a four-wheel design. As a driving movement using a planetary motor PG 45 where this motor can move if the input voltage is 12V - 24V which will produce a rotation speed 90rpm with a current 1.61A. And this motor is quite capable for carrying the robot in the desired stable position. These three wheels has the algorithm that will be different from the robot that uses four-wheel. These three wheels have the better ability to maneuver than the four-wheel so it is suitable to serve as a striker who needs more movement in the pursuit of the ball than the robot kipper. URT-ROSO uses the omni wheel, this wheels have good ability to move in every direction and slip less so it helps the robot to have a good movement. Here is the overall mechanical design of the URT-ROSO striker robot.

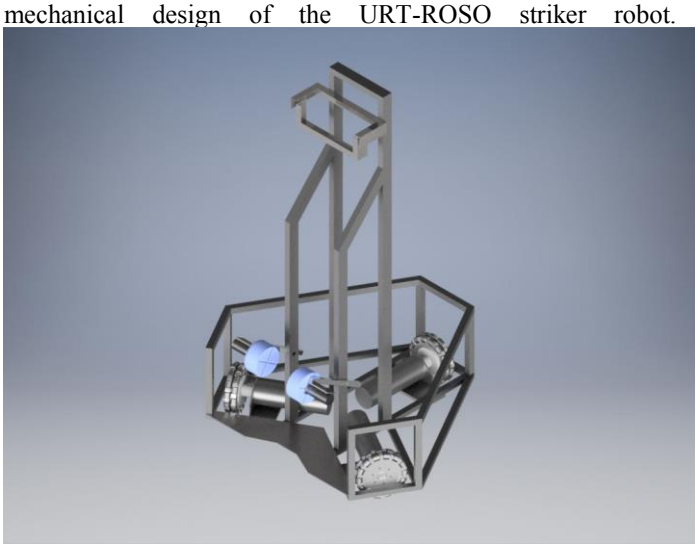


Figure 1. Basic Mechanic of URT-ROSO

B. Kiwi Drive

On URT-ROSO movement uses three omni wheels called Kiwi Drive. Kiwi Drive is where there are three omni wheels mounted in an equilateral triangle configuration.

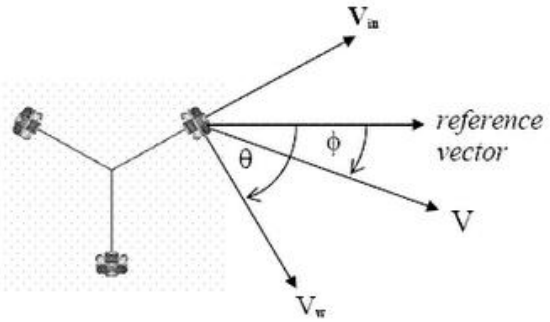


Figure 2. Kiwi Drive Sketch

By using this kiwi drive makes the robot has a better maneuver, also has a more simple control. In kiwi drive, the movement of the wheels must rotate in rhythm. Here is the preview of kiwi drive wheels movement.

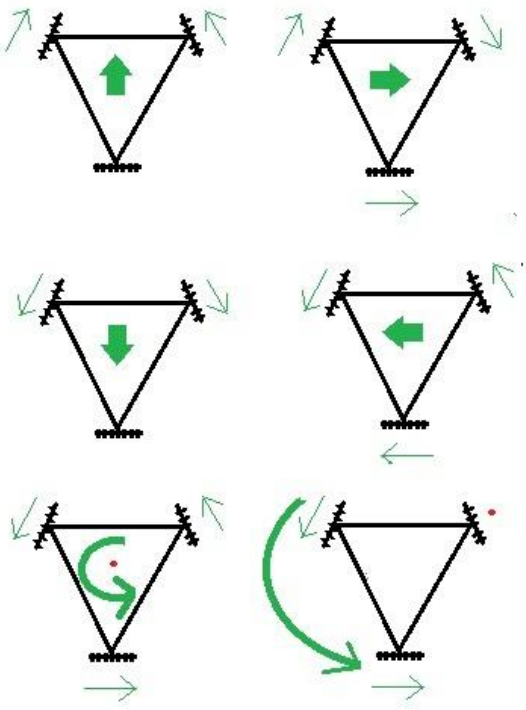


Figure 3. Kiwi Drive Movement

To move properly, in kiwi drive we must calculate its propulsion, degree, even its vector. The algorithm used in this kiwi drive is calculated as follows

$$\begin{aligned}\left[\vec{v}_{\parallel}\right] &= \vec{v}\cdot\hat{u} \\ &= \left(v_x\hat{\imath}+v_y\hat{\jmath}\right)\cdot\left(-\frac{1}{\sqrt{2}}\hat{\imath}+\frac{1}{\sqrt{2}}\hat{\jmath}\right) \\ &= \left(-\frac{1}{\sqrt{2}}v_x+\frac{1}{\sqrt{2}}v_y\right)\end{aligned} \quad (1)$$

This calculation can resolve the velocity of a perpendicular parallel component. So it can easily know the wheel speed with the formula as below

$$\begin{aligned}\left[\vec{v}_{w}\right] &= \left[\vec{v}_{\parallel}\right] \\ &= \left(-\frac{1}{\sqrt{2}}v_x+\frac{1}{\sqrt{2}}v_y\right)\end{aligned} \quad (2)$$

C. Dribbler

URT-ROSO has the ability to grab the ball and it is necessary when the robot moves. Dribbler coupled with a dc motor and rubber wheels. This motor dc has 700rpm will turns the wheel that has been arranged to keep the ball, where in accordance with the rules of Kontes Robot Indonesia 2017 that the ball must roll naturally, so if the robot is going forward then the ball must also spin forward and vice versa, also in the rules there is a limit that the grips of the ball only ⅓ from the surface of the ball.

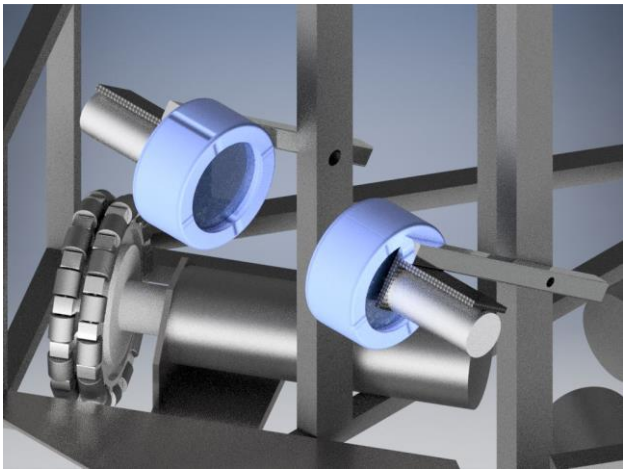


Figure 4. Ball Dribbler

III. KICKER MECHANIC

Earlier at the time of participating in the Kontes Regional, URT-ROSO uses a simple kicker driven by a PowerWindow dc motor coupled by a kicking mechanic. When using this motor kicker the force of the ball is not as good as we expect. The ball only can rolling not parabolic bouncing so it can pass the rival kipper. Then when prepare to compete at the national URT-ROSO decided to use the solenoid as its kicker. Where the solenoid works on large currents in order to produce large power as well. To get a large current in the manufacture of solenoid winding required an appropriate calculations so it will produce the desired strength. For the calculation of the winding then use the following formula

$$\mathcal{L} = \frac{(d^2\cdot n^2)}{18d+40\ell} \quad (3)$$

Where d the diameter of the desired solenoid in inches and n is the number of turns of the solenoid while ℓ is the length of the solenoid in inches. The more winding, the resistance will be increase. So URT-ROSO uses a wire with a diameter of 1 mm and wrapped around a PVC pipe with a diameter of 23mm with length of 125mm and the coils have 1187 turns (12 layers). But using solenoid is more dangerous than using the other kicker style because solenoid needs capacitor bank that produce big voltage. To improve the solenoid, we control the magnetic field radiated form and try to avoid the magnetic effect on other parts in robot such as compass HMC5883L that uses in robot to recognize the robot position. URT-ROSO uses black tape as its magnetic shield.

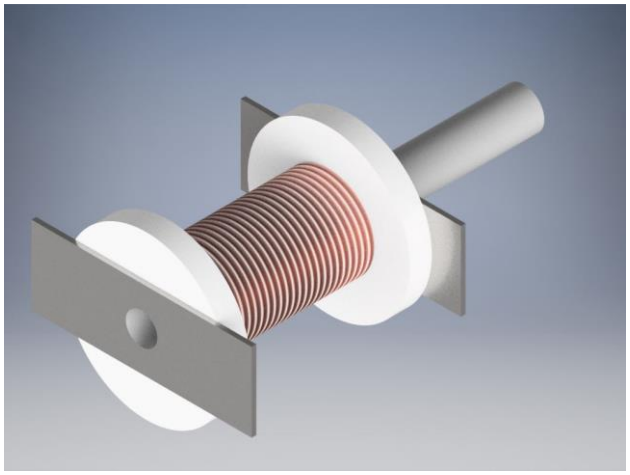


Figure 5. Solenoid

A. Solenoid Mechanic

Solenoid mechanics use PVC pipe with length 125mm. For URT-ROSO kicker mechanics use nylon and iron as the driving force. With nylon length a little shorter than the length of iron. The solenoid design should suitable with the space in the robot, so we designed the solenoid as minimalist as it can be. The nylon which is attached with iron shaft is non-ferromagnetic material so this nylon will shoots outwards and hit the ball.

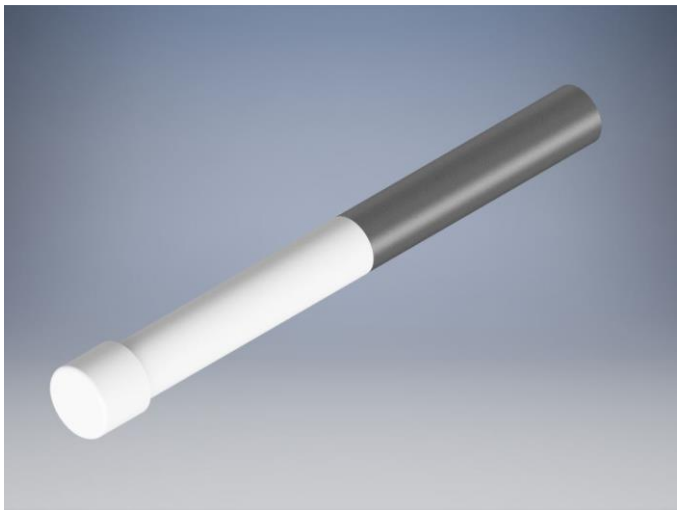


Figure 6. Solenoid Shaft

In order to know the force that this solenoid produce, use this formula below

$$\frac{1}{2}m_{planger}v_{planger}^{\ 2} = \frac{1}{2}m_{ball}v_{ball}^{\ 2} + \frac{1}{2}\mathcal{J}_{ball}w^2 \quad (4)$$

$$\mathcal{J}_{ball} = \frac{2}{3}m_{ball}r_{ball}^{\ 2}$$

$$= 363 \times 10^{-3}[Kg \cdot m^2]$$

$$w = \frac{v_{ball}}{r}[rad/s]$$

B. Electronics

To get a large voltage on the solenoid requires an electronic circuit boost converter and black box contains a series of capacitor bank. Output of this boost converter circuit is 450V which is capable of kicking the ball with the power of more than 320J with the expected speed is 9m / s. Here is a schematic used on URT-ROSO

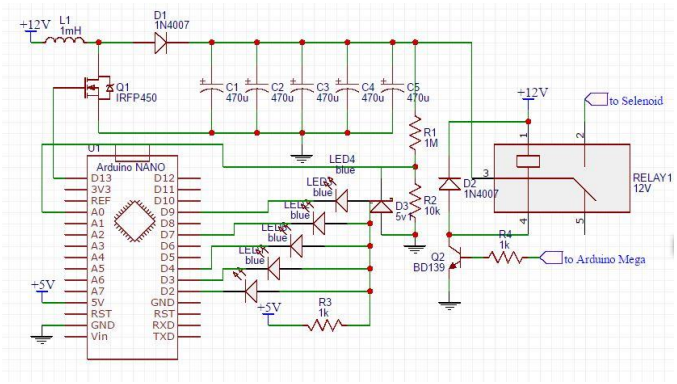


Figure 7. Solenoid Schematic

Using this circuit the obtained voltage is 450V with an input voltage of 12V this can occur due to a boost converter circuit using a bank capacitor of 470uF which is paralleled by five capacitors to produce the desired output voltage. In charging to achieve an output voltage of 450V it will take approximately 14 seconds.

With this boost converter, the output voltage will higher than the input voltage. So it is make sense that the voltage and current ratio will be like this formula below

$$\frac{I_{out}}{I_{in}} = \frac{V_{in}}{V_{out}} \quad (5)$$

IV. Image Processing

In support the movement of the robot then URT-ROSO equipped with pixy camera and android that perform image processing to be able to detect the ball. Image processing done by android camera is more accurate than the pixy camera. The research team made their own android application that can detect the ball by processing on the color data, can also know the location of the ball according to the x and y axis by integrating on the compass which is owned by android. And by

using Bluetooth the android and the micro will connected. The preview of android app interface is down below



Figure 8. Android Program

V. CONCLUSION

Using three wheels with kiwi drive as URT-ROSO propulsion is the best choice for robot soccer movement because with this kiwi drive, robot is easier to control and has a better maneuver than the two-wheels or even four-wheels.

URT-ROSO decided to change the motor kicker style with solenoid kicker style because with solenoid the robot is more powerful to kick the ball. Using solenoid kicker style, the ball will bounce with parabolic so it increases the ball opportunity to pass the rival kipper. But using solenoid kicker style is harmful so it will need more concern in safety. By applying advance calculating when arrange the solenoid is the key to have safety solenoid kicker style.

ACKNOWLEDGMENT

URT-ROSO would like to thank our sponsors, as well as the Islamic University of Sultan Agung, specifically the Faculty of Industrial Technology for Electrical Engineering Major. And all of senior in Robotics Team that always teaches us and support us so URT-ROSO can be made. Without their continued support, developing our URT-ROSO and competing at Kontes Robot Indonesia would not be possible.

REFERENCES

[1] Kasaei, S. M., Kasaei, S. M., & Kasaei, S. M. (2010). Design and Implementation Solenoid Based Kicking Mechanism for Soccer Robot Applied in Robocup-MSL. International Journal of Advanced Robotic Systems.

[2] Serafim, M. S. (2013). Ball Handling Mechanisms for Mobile Robots. *Master of Science Thesis*.

[3] Zandsteeg, C. (2005). Design of a RoboCup Shooting Mechanism.

Rancang Bangun Robot KRSBI Beroda dengan Raspberry Pi Pada Tim ISOLA_F

Muhammad Renaldy*, Dicky Mardiansyah, Budi Afwan N, dan Didin Wahyudin Ph.D.
Departemen Pendidikan Teknik Elektro
Universitas Pendidikan Indonesia
Bandung, Indonesia
muhammad.renaldy@student.upi.edu

***Abstrak*— Penelitian ini menjelaskan tentang desain dari robot sepak bola yang digunakan pada tim ISOLA_F untuk Kontes Robot Sepak Bola Indonesia divisi beroda yang mengacu pada peraturan *Middle Size League (MSL) Robocup.* Robot yang digunakan pada penelitian ini menggunakan biaya yang serendah mungkin tetapi dengan hasil yang dapat bersaing dengan robot lainnya. Pada penelitian ini sistem pencitraan yang digunakan menggunakan kamera *omnidirectional* dan sistem kontrol menggunakan *Raspberry Pi.* Untuk pemetaan robot ini menggunakan sensor *rotary encoder* untuk menentukan koordinat robot.**

Kata Kunci*— *Robot Sepak Bola, Raspberry Pi, Robocup, omnidirectional kamera, rotary encoder.

I. PENDAHULUAN

Robocup (*Robot World Cup)* merupakan sebuah program internasional yang dimulai pada tahun 1997 untuk mempromosikan penelitian *Multi Agent System* (MAS) dan *Distributed Artificial Intelligence* (DAI) untuk menyelesaikan berbagai masalah [1]. Salah satu divisi dari robocup yaitu MSL (*Middle Size League)* dimana robot menggunakan roda sebagai penggerak utama. Di Indonesia ada KRSBI (Kontes Robot Sepak Bola Indonesia) divisi beroda dimana mengacu pada peraturan MSL.

Tim ISOLA_F yang merupakan salah satu peserta KRSBI divisi beroda merancang sebuah robot dengan biaya yang serendah mungkin. Robot menggunakan kamera *omnidirectional* yang dibuat sendiri. Untuk *controller* utama menggunakan Raspberrry Pi dan OpenCV sebagai *library* utama. Untuk sistem penendang menggunakan pneumatik yang dikontrol melaui *solenoid valve*.

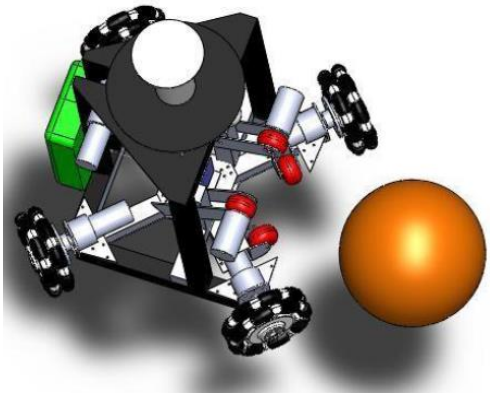
II. MATERIAL DAN METODE

Pembuatan robot sepak bola beroda memiliki 3 tahapan, tahapan pertama yaitu menganalisa kebutuhan dan perancangan desain. Pada penelitian ini komponen-komponen robot yang diperlukan dapat dilihat pada TABEL 1. Setelah menentukan komponen yang diperlukan selanjutnya merancang desain robot.

TABEL I. DAFTAR KOMPONEN ROBOT

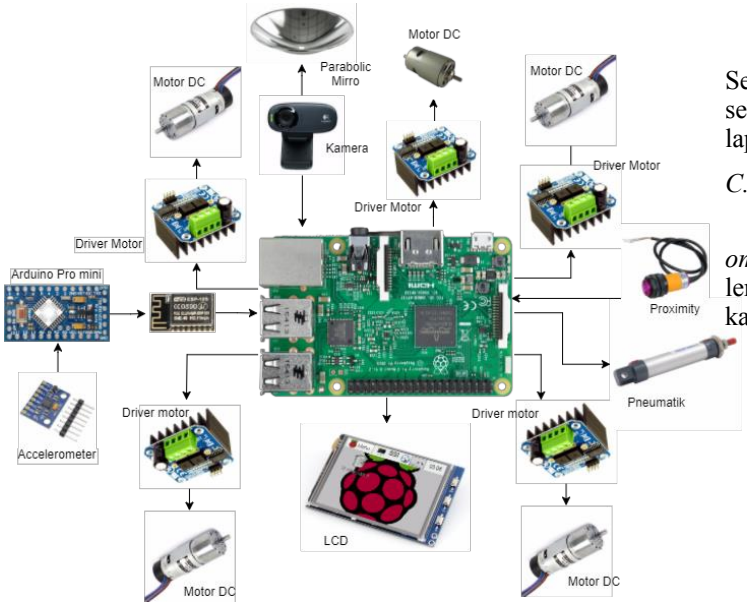
No	Komponen	Type	Kategori
1.	Raspberry Pi	Model 3	Prosessor
2.	Kamera	Webcam Logitech C270	Sensor
3.	IMU	MPU-6050	Sensor
4.	Motor DC	PG36 *with rotary encoder*	Aktuator
5.	*Driver* Motor	BTS7960	*Driver*
6.	Arduino	Pro Mini	Prosessor
7.	Baterai Lipo	4S 5500 mah	*Power*
8.	Wifi *Module*	ESP-12S	Komunikasi
9.	Selenoid Valve	24V DC	Aktuator
10.	Infrared Sensor	E18-D80NK	Sensor

Desain robot yang digunakan adalah sistem *omniwheel* dengan konfigurasi X, dimana konfigurasi ini menggunakan 4 roda yang disusun membentuk huruf X seperti pada gambar 1. Desain *omniwheel* dengan 4 roda ini memang agak sulit dikontrol, akan tetapi memungkinkan robot untuk akselerasi dengan kecepatan sama di segala arah[2]. Roda omniwheel memiliki beberapa kelebihan yaitu regangan yang rendah pada motor, memiliki akselerasi tinggi dan resiko motor terbakar lebih kecil. Selanjutnya setelah robot selesai dirancang proses pembuatan kerangka robot dapat dilakukan.



Gambar 1. Desain Robot .

Tahapan kedua adalah perakitan sistem hardware, pada tahapan ini motor dikendalikan melalui driver motor kemudian driver motor dihubungkan ke GPIO Raspberry Pi. Untuk kamera dapat lansung dihubungkan ke port USB Raspberry Pi. Untuk pembacaan sensor Accelerometer , sensor inframerah dan *rotary encoder* digunakan Arduino Pro Mini yang dikomunikasikan ke Raspberry Pi menggunakan komunikasi *USART*. Robot menggunakan ESP-12S untuk melakukan komunikasi dengan *base station*



Gambar 2. Blok diagram hardware

Tahapan ketiga yaitu pemograman. Pemograman Raspberry Pi menggunakan bahasa *python* dan pemograman pada arduino menggunakan bahasa C/C++.

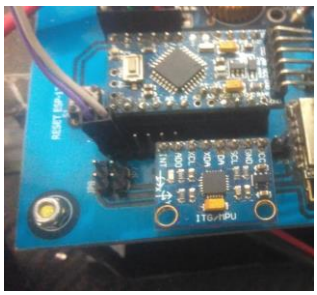
III. SYSTEM HARDWARE

A. *Raspberry Pi*

Raspberry Pi dalam penelitian ini digunakan sebagai kontroler utama. Raspberry Pi awalnya di-*install Operating System* (OS) Raspbian Jeesie. Untuk pemograman dapat menggunakan bahasa *python.* Raspberry Pi harus di-*install* OpenCV sebagai *library* untuk menggunakan *image processing.*

B. *Sensor IMU (Inertial Measurement Unit)*

Pada penelitian ini untuk menentukan orientasi robot menggunakan sensor MPU-6050. Dimana sensor tersebut dibaca menggunakan Arduino Pro Mini dan dikomunikasikan ke Raspberry Pi..

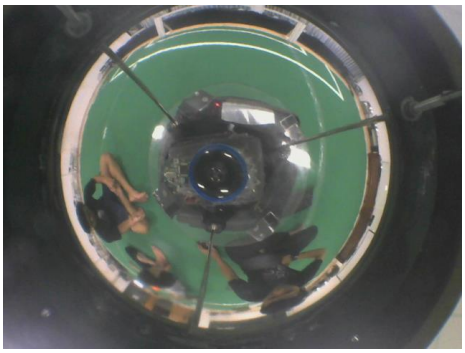


Gambar 3. Sensor MPU6050 pada robot

Dengan sensor ini robot dapat mengetahui orientasi robot. Sensor IMU akan dikombinasikan dengan *rotary encoder* sehingga robot ini dapat mengetahui posisi robot terhadap lapangan.

C. *Kamera Omnidirectional*

Pada robot ini untuk pencitraan menggunakan kamera *omnidirectional*. Pada robot ini digunakan kamera *webcam* dan lensa setengah bola. Kamera ini dirancang menggunakan kamera webcam Logitech C270 dengan resolusi 3MP.



Gambar 4. Hasil gambar kamera omnidirectional

D. *Arduino Promini*

Arduino pro mini digunakan untuk membaca sensor IMU dan rotary encoder. Dengan ini robot dapat mengetahui posisi robot dan orientasi nya terhadap lapangan. Data yang dihasilkan di kirimkan ke Raspberry Pi menggunakan komunikasi *USART.*

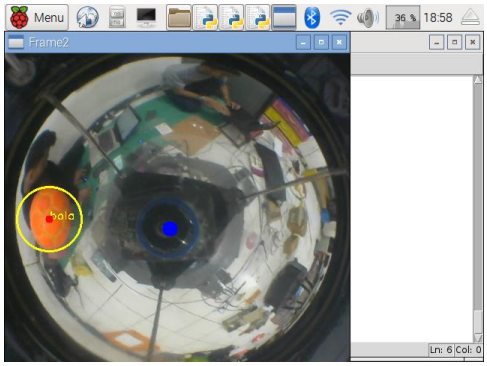
E. *ESP-12S*

Modul Wifi ESP-12S digunakan untuk melakukan komunikasi data dari *base stasion* ke robot sehingga robot dapat menerima perintah dari *Reference Box.*

IV. SISTEM SOFTWARE

A. *Image Processing*

Sistem pencitraan pada robot ini menggunakan *library* OpenCV dimana menggunakan *filter* HSV (*Hue, Saturation* dan *Value)* untuk menentukan posisi bola, musuh dan gawang.



Gambar 5. Contoh hasil *filter* gambar untuk menentukan bola

Untuk menentukan posisi bola pada gambar di kamera *omnidirectional* yang berbentuk lingkaran menggunakan rumus koordinat kutub yaitu :

$$\Theta = \arctan(y/x) \qquad (1)$$

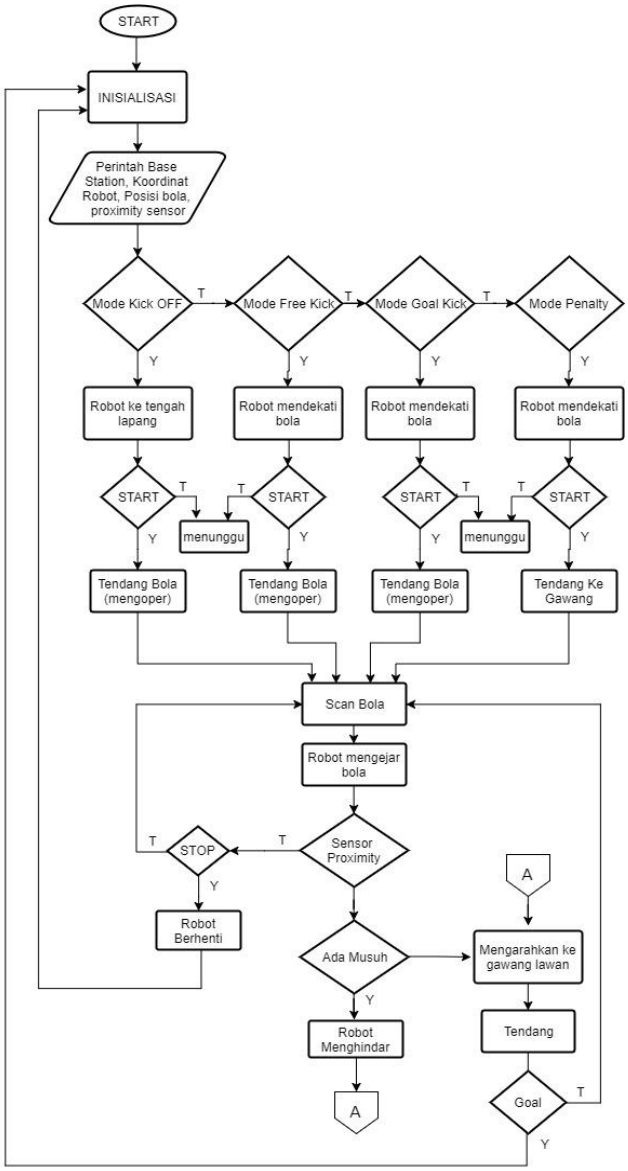
Sedangkan untuk jarak bola terhadap robot menggunakan rumus Pythagoras yaitu:

$$Jarak^2 = x^2 + y^2 \qquad (2)$$

Selain bola *image processing* juga digunakan untuk menentukan musuh dan posisi gawang. Dari hasil *filter* warna tersebut digabungkan dan sehingga selain dari *rotary encoder* dan sensor IMU robot dapat mengetahui posisi terhadap lapang menggunakan *system localization* dengan membandingkan gambar *template* dengan gambar robot sebenarnya di lapangan.

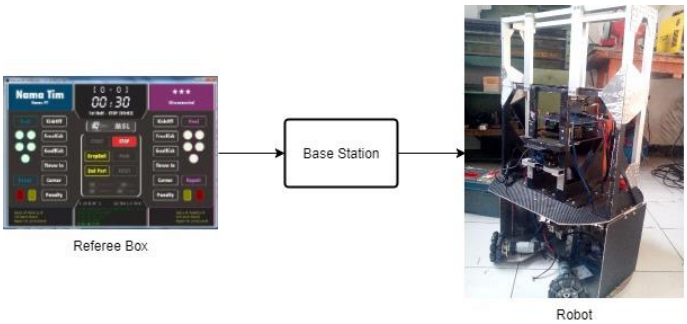
B. Algoritma Pemograman

Pada robot ini algoritama yang digunakan yaitu robot menerima *input* dari Arduino untuk menentukan posisi dan orientasi robot. Lalu menerima *input* dari ESP-12S menunggu perintah dari *base station* untuk mode yang digunakan. Untuk lebih jelas ada pada gambar 6 tentang *flowchart* pemograman dari robot ini.



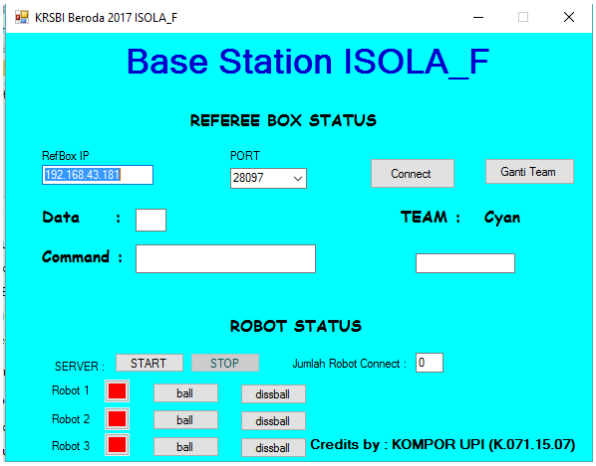
Gambar 6. *Flowchart* pemograman robot

C. Komunikasi Robot



Gambar 7. Sistem komunikasi robot

Untuk komunikasi robot ke *referee box* robot menggunakan *base stasion* dimana robot memakai komunikasi *wifi*. Sistem *base stasion* menggunakan komunikasi TCP/IP antara *base stasion* dan *referee box*.



Gambar 8. Sistem base stasion ISOLA_F

V. HASIL PERCOBAAN

A. *Posisi sudut bola*

Berdasarkan persamaan 1 untuk menemukan sudut bola terhadap robot dibutuhkan koordinat dari posisi bola. Dari hasil percobaan yang kami lakukan didapat hasil pembacaan koordinat bola oleh robot pada TABEL II.

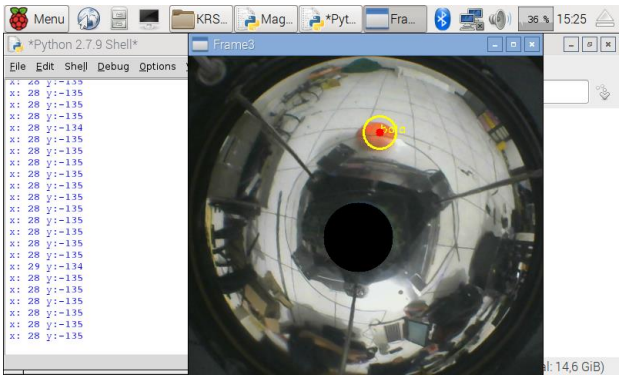
TABEL II. HASIL KOORDINAT TERHADAP SUDUT SESUNGGUHNYA

Sudut (*Real*)	**Koordinat X**	**Koordinat Y**
0	146	-2
45	98	98
90	1	117
135	-76	78
180	-110	1
225	-92	-92
270	-1	-135
315	104	-104

Dari hasil tersebut kami mendapatkan *error* antara sudut yang dibaca oleh robot dengan sudut sesungguhnya pada TABEL III.

TABEL III. PERBANDINGAN SUDUT SESUNGGUHNYA DENGAN SUDUT YANG TERBACA OLEH ROBOT

Sudut (*Real*)	**Sudut terbaca**	**Error (%)**
0	0	0
45	45	0
90	89,51	-0,005441
135	135	0
180	179,57	-0,0023578
225	225	0
270	269,52	-0,0017746
315	315,74	0,0023622



Gambar 9. Pembacaan Koordinat Bola

B. *Posisi Jarak*

Untuk posisi jarak bola digunakan persamaan 2 dimana juga menggunakan koordinat pembacaan. Dari hasil percobaan didapatkan hasil pada table 4.

tantangan untuk posisi robot dan cahaya untuk pembacaan posisi bola kami dapat membuat pembacaan posisi bola dengan akurat.

VI. KESIMPULAN

Dalam penelitian ini didapatkan bahwa rancangan robot ISOLA_F dengan menggunakan kamera *omnidirectional* yang dirancang sendiri mendapatkan hasil yang memuaskan. Kesalahan yang dibaca oleh robot kurang dari 0.5%. Hal tersebut masih wajar dimana masih sangat kecil untuk mempengaruhi pergerakan robot. Walaupun masih ada

REFERENCES

[1] R. Junkai, X. Chenggang, X. Junhao, H. Kaihong, and L. Huimin, "A Control System for Active Ball Handling in the RoboCup Middle Size League," pp. 4396–4402, 2016.

[2] N. Lynch, "Conceptual Design of a Robot for the RoboCup Competition," *Lehigh.Edu*

Design and Implementation Software in Indonesian Traditional Dancing Humanoid Robot Development Platform

Desiana Dien Nurchalifah*, Kusprasapta Mutjiarsa, dan Widyawardana Adiprawita
Electrical Engineering Departement, School of Electrical Engineering and Informatics
Bandung Institute of Technology
Bandung, Indonesia
desiana.nurchalifah@gmail.com

***Abstract* – This paper discussed the results of design and implementation software and user interface used in building dance movements of humanoid robot. Sequences of dancing movements are created from several basic set of movements which had been planted in this software. Basic dance movements including hand, legs, neck, and hip in the humanoid robot with 25 degree of freedom (DOF). This development platform software of dancing movements built in low level and high level. On the side of low level, it is designed ability to control every servo motor in humanoid robot using forward kinematic. On the high-level side, graphical user interface is built as a composer of dancing movements set practically. Set of dancing movements in which had created in the software platform on the computer, then sent to robot by communication module that had been made. Subsequently, humanoid robot would dance based on sequences recieved. In the implementation, dancing movements created are two traditional dances in Indonesia, which are Topeng Betawi dance and Gending Sriwijaya dance. Based on results, development software platform created could compose and control six sets of Gending Sriwijaya dancing movements to dance in the duration of four minutes. This software platform created to alleviate humanoid robot dances developer to build new dance movements. Dance movements is not to be programmed based on text.**

Keywords—Software development platform, humanoid robot, traditional dancer robot, GUI, low level controller, high level controller, interface, communication.

I. INTRODUCTION

Tourism is one of the most important sectors in many countries in general, and Indonesia in particular. Divided into 17,504 islands and 34 provinces, making Indonesia has unique and variety of art and culture that shapes into an Indonesian characteristic. The manifestation of copyrights and works is shaping the identity of the Indonesian nation. Culture can be viewed as an individual identity in a nation that reflects the diversity of the nation by providing insight and meaning to national development. With the development of technology in Indonesia, the next generation prefer to leave the conservation of local culture, so the culture in Indonesia is extinct. One form of technological development that is renewable humanoid robot. Humanoid robot is a robot designed to resemble humans. Humanoid robot is made with a design to perform certain functions, one of which is dancing. Based on BPS data in August 2015, the tourism sector in Indonesia continues to increase [1]. This challenges researchers to combine technology and the culture value of Indonesia in developing especially in the tourism sector.

Programming on the humanoid robot that has been used until now has run well in controlling the robot, but the complexity of programming caused limitations to whom using the humanoid robot. This could decrease the development of education about humanoid robots in certain communities. The complex system of humanoid robots requires in-depth knowledge to operate the robot, so it requires an intermediary between the humanoid robot and the user. One alternative is to create humanoid robot control software to perform traditional Indonesian dance movements.

Software designed using secure shell connection between user and robot. With software, the complexity of robotic programming that previously needed to understand low-level language is represented as a button containing the basic movements of traditional Indonesian dance. Software created has a class that controls the framework of the robot in order for the robot to execute movements or the intended action in accordance with commands sent by the user. The robot framework runs forward kinematics so that the robot can move 25 servo pieces in accordance with traditional dance moves. In addition, coordination between dancers is required by communication between two dance robots. For example, to synchronize movements between robots, communication process of waiting the other robot is needed to make the movements created in harmony with each other.

II. LITERATURE STUDIES

A. Humanoid Robot

The development of humanoid robots such as ASIMO [2] and HUBO [3] has various research in the field of robotics for example motion planning, communication, manipulation, and walking. Various platforms are developed to achieve the ideal state of research, which concludes the manufacture of humanoid robots to focus on modifiable systems, high performance, easy repairs, and also has a common

environment at an affordable price. One of the platforms that can be used is the DARwIn-OP platform.

DARwIN-OP (Dynamic Anthomorphic Robot with Intelligence - Open Platform) is a platform developed by Virginia Tech's RoMeLa (Robotics & Mechanism Laboratory) laboratory in collaboration with University of Pennsylvania, Purdue University, and South Korean company called Robotis. The developed platform has high computing capabilities. Due to the development of dynamic robot capabilities, this platform is well suited for research and education.

This platform has a total of 20 degrees of freedom, with a distribution of 12 pieces on the foot, 6 on the hand, and 2 on the neck and head. This platform has much in common with the Bioloid GP platform, but it has more DOFs and has a more robust frame with aluminum usage as a whole frame material. This platform uses single board computer technology (SBC) as the main data processor, so this system has the capability of qualified to perform data processing with good performance.

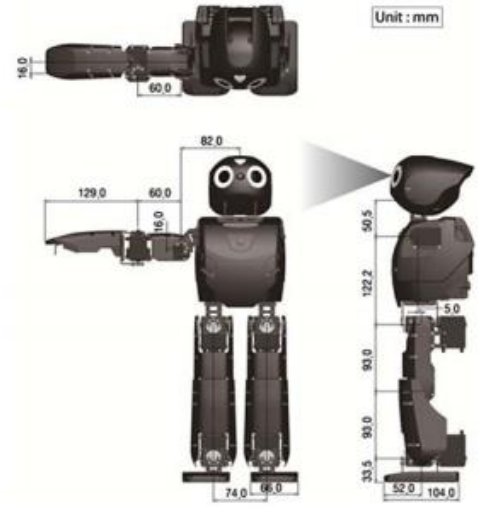


Figure 1. DARwIn-OP Platform[4]

B. Object Oriented Programming

Object-oriented programming is organizing program to combine data and functionality into an object. In a development platform that requires a program with complex structures, it is necessary to use object oriented programming techniques. In object-oriented there are two important aspects, which are class and object. The class creates a new type with the object as an instance. Classes have attributes and methods (function that exist in the class). Methods within the class can be private, public or protected. [5] While objects can store data using common variables. Objects can have functionality by using functions in a class (method). Objects in a class can be used as an input or output parameters from the use of library. [6] In the development platform, the main program able to use the library in variety amount, so that objects in the library could be related to each other in accordance with the main purpose of programming. Examples of object-oriented programming can be done using C ++ and Python.

C. TCP/IP Communication Protocol

TCP provides services with establishing connection. A logical connection is established between the original station and transmitter. Features on TCP connections such as error and flow control make TCP a connection with the preference of data transfer. TCP receives data from the application layer, adds the data needed to get a reliable operation, then sends it to the IP. TCP connects port numbers with specific applications, generates the number of each byte in the data stream, calculates the timeout to ensure undetected frames, and uses certain messages to create and stop duplex communications. TCP IP is a point-to-point connection with communication from a computer on a network to another computer. Connection using TCP is a non-state connection due to a client's request at a given time has no correlation with any other time. The process stack using TCP/IP described as follows.

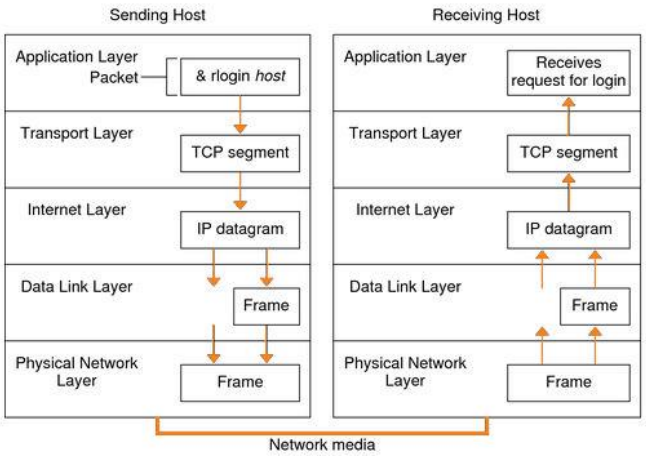


Figure 2. TCP/IP Protocol Diagram[7]

III. IMPLEMENTATION AND DESIGN

The design and implementation consisted of Graphical User Interface and communication module which results as below

- Implementation and Design of Graphical User Interface (GUI)

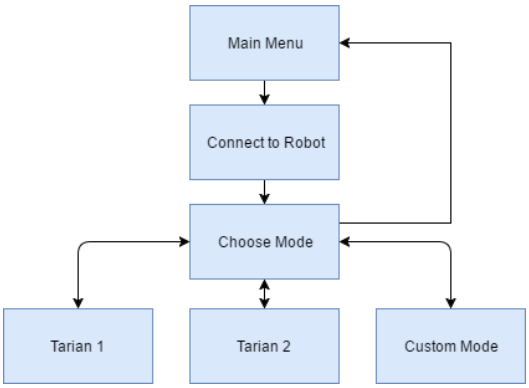


Figure 3. GUI Block Diagram

The created application has six windows that are main menu, connect to robot, selection of desired mode, dance example 1, dance 2 example, and free mode.

• Design and Implementation of communication modules

TCP communication system is used in communication design between two humanoid robots that are used. One humanoid robot sends data packets using IP addresses. One of the humanoid robot is made into a server of TCP connection and the other robot become the client on the same connection. The design procedure performed by the humanoid robot on server is shown in the diagram as follows

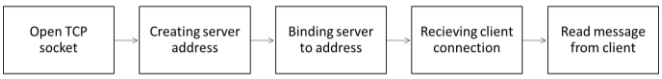


Figure 4. Server Procedure

While the procedure design performed on client is shown in the diagram as follows



Figure 5. Client Procedure

IV. RESULTS

Graphical user interface robot test results are as follows

• Secure Shell Connection Testing

In this test is performed performance testing of the software by measuring the speed of the connection through the router under normal conditions. The results of this test can be a reference to evaluate the software. Test results are as follows.

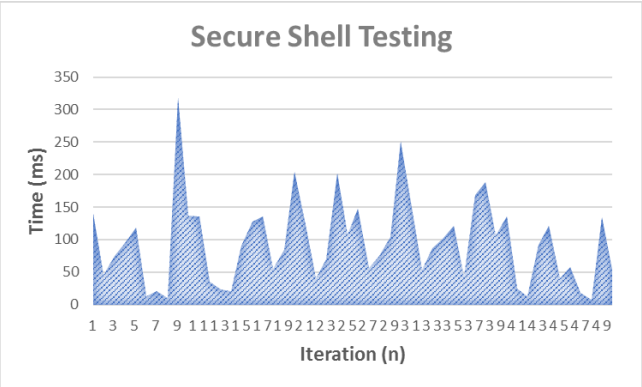


Figure 6. Secure Shell Connection Testing

On connection test performed has been successful with the average length of time required in the connection is 95.54ms

• Secure Shell Speed Test

In this test, performance testing of the software is done by measuring the speed opening a Secure Shell connection. The results of this test could be a reference to evaluate the software. Test results are as follows.

```
================ RESTART: D:\apps\buatTA\results\ref\ssh_3.py ================
Connecting to server on ip 192.168.124.4.
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 3.10.103 armv7l)

 * Documentation:  https://help.ubuntu.com/

Last login: Tue Jan  1 00:19:36 2013 from 192.168.124.8
odroid@odroid:~$
$
 Waktu yang diperlukan untuk koneksi adalah :
1.8348820209503174
detik
```

Figure 7. SSH Speed Testing 2

From the test results with Secure Shell, the user interface was able to successfully connect with the duration of 1-2 seconds after several data logging to test the connection.

• Testing Low Level Program Executions

This test is measuring the software performance by testing the speed of executed low level program in humanoid robot using Secure Shell. The results of the test could be reference to the software evaluation. The results are as follows.

```
 Waktu yang diperlukan untuk eksekusi program adalah :
7.033968925476074
detik

>>> [H[J
[Action Editor for DARwIn v1.00]

 *Terminal screen size must be 80(col)x30(row).
 *Current terminal has 80 columns and 24 rows.


Press any key to start program...
[H[JID: 1(R_SHO_PITCH)[    ]

ID: 2(L_SHO_PITCH)[    ]                              Page Number:
ID: 3(R_SHO_ROLL) [    ]                               Address:
ID: 4(L_SHO_ROLL) [    ]                                Play Count:
ID: 5(R_ELBOW)    [    ]                                 Page Step:
ID: 6(L_ELBOW)    [    ]                                Page Speed:
ID: 7(R_HIP_YAW)  [    ]                                Accel Time:
ID: 8(L_HIP_YAW)  [    ]                              Link to Next:
ID: 9(R_HIP_ROLL) [    ]                              Link to Exit:
```

Figure 8. Low Level 1 Testing

```
 Waktu yang diperlukan untuk eksekusi program adalah :
7.036706924438477
detik

>>> [H[J
[Action Editor for DARwIn v1.00]

 *Terminal screen size must be 80(col)x30(row).
 *Current terminal has 80 columns and 24 rows.


Press any key to start program...
[H[JID: 1(R_SHO_PITCH)[    ]

ID: 2(L_SHO_PITCH)[    ]                              Page Number:
ID: 3(R_SHO_ROLL) [    ]                               Address:
ID: 4(L_SHO_ROLL) [    ]                                Play Count:
ID: 5(R_ELBOW)    [    ]                                 Page Step:
ID: 6(L_ELBOW)    [    ]                                Page Speed:
ID: 7(R_HIP_YAW)  [    ]                                Accel Time:
ID: 8(L_HIP_YAW)  [    ]                              Link to Next:
ID: 9(R_HIP_ROLL) [    ]                              Link to Exit:
ID:10(L_HIP_ROLL) [    ]
ID:11(R_HIP_PITCH)[    ]
```

Figure 9. Low Level 1 Testing

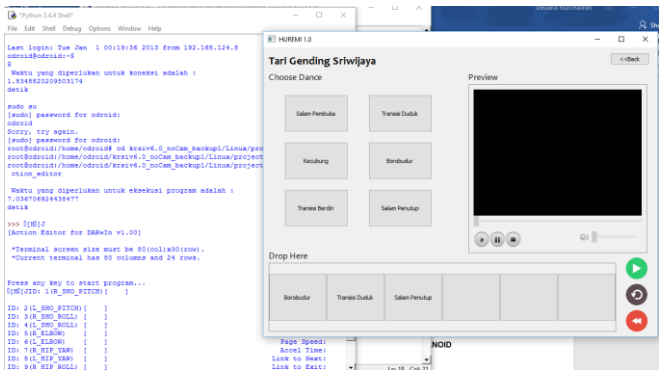


Figure 10. Command Testing

From the test results, the user interface was able to successfully measure the speed of program execution on the humanoid robot with the required length of time is in the range 7-8 seconds. This execution causes time delays on commands sent by the software.

• Testing Every Button on The Interface

Testing each software button produces as follows

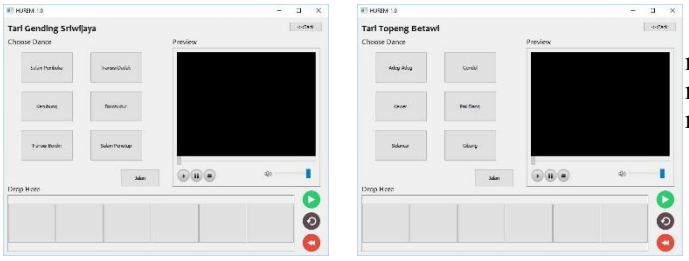


Figure 11. Gending Sriwijaya (left) and Topeng Betawi (right)

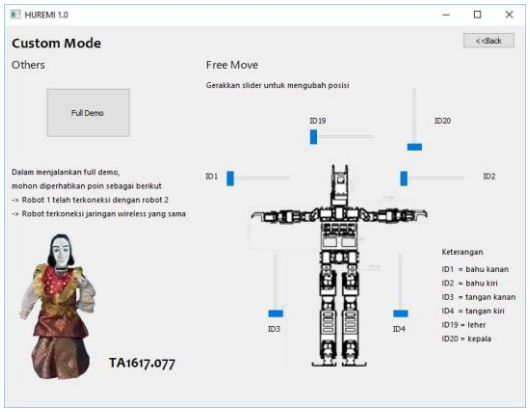


Figure 12. Custom Window



Figure 13. Robot Dancing Results

From the test results, the software can perform its function in giving commands to the robot.

In addition to testing the software, communication module testing is also done with the following results

• Packet loss testing

The test results of the humanoid robot communication module is the long-lasting wireless connection using wireless module and the time required for communication between robots with the following results

```
64 bytes from 192.168.124.4: icmp_seq=88 ttl=64 time=1314 ms
64 bytes from 192.168.124.4: icmp_seq=89 ttl=64 time=395 ms
64 bytes from 192.168.124.4: icmp_seq=90 ttl=64 time=30.8 ms
64 bytes from 192.168.124.4: icmp_seq=91 ttl=64 time=18.5 ms
64 bytes from 192.168.124.4: icmp_seq=92 ttl=64 time=28.1 ms
64 bytes from 192.168.124.4: icmp_seq=93 ttl=64 time=24.5 ms
64 bytes from 192.168.124.4: icmp_seq=94 ttl=64 time=20.9 ms
64 bytes from 192.168.124.4: icmp_seq=95 ttl=64 time=1033 ms
64 bytes from 192.168.124.4: icmp_seq=96 ttl=64 time=42.3 ms
64 bytes from 192.168.124.4: icmp_seq=97 ttl=64 time=33.6 ms
64 bytes from 192.168.124.4: icmp_seq=98 ttl=64 time=1013 ms
64 bytes from 192.168.124.4: icmp_seq=99 ttl=64 time=21.3 ms
64 bytes from 192.168.124.4: icmp_seq=100 ttl=64 time=19.8 ms

--- 192.168.124.4 ping statistics ---
100 packets transmitted, 98 received, 2% packet loss, time 99209ms
rtt min/avg/max/mdev = 17.513/320.427/2678.101/538.230 ms, pipe 3
```

Figure 14.7 Robot 1 Testing Communication Data

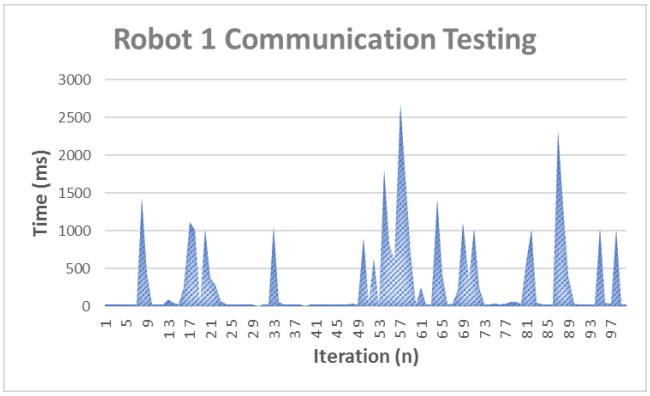


Figure 15. Robot 1 Testing Communication Graph

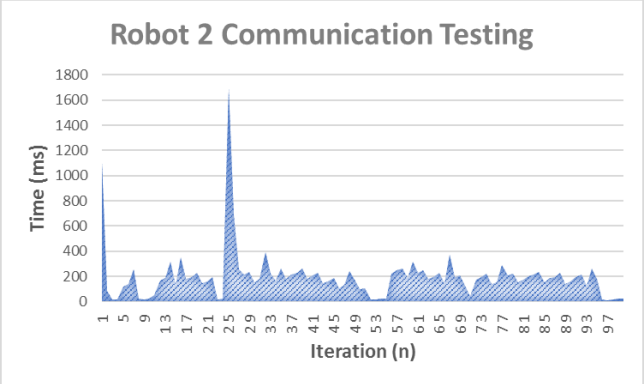


Figure 16. Robot 2 Testing Communication Graph

On successful testing the communication between the two robots with the average time required is 257.5 ms in a single command.

• Testing communication between robots

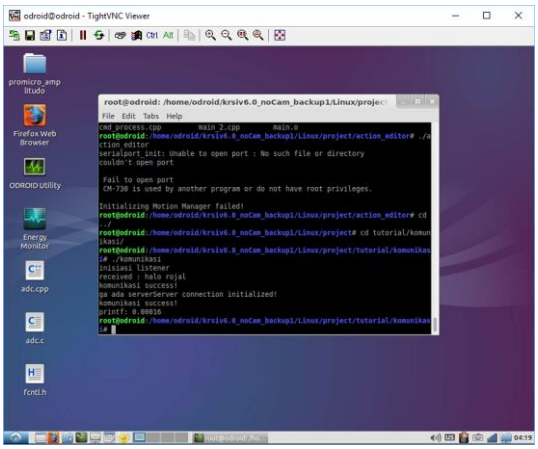


Figure 17. Communication Command Speed 1

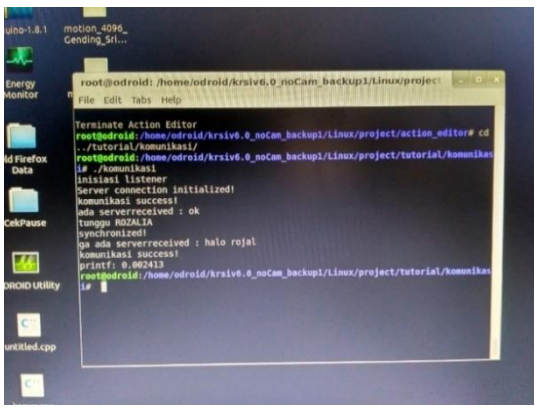


Figure 18. Communication Command Speed 2

From the test results, communication between robots directly in giving orders or receipts generate time in the range 0-2 seconds.

V. CONCLUSION

This paper discussed the design and implementation software to develop dance moves on traditional Indonesian humanoid robot dancer. Some conclusions that can be drawn are as follows:

1) Software development platform in humanoid robot dancer had designed and implemented using GUI. The interface had transformed from text-based into GUI which has succeeded in reducing complexity in developing humanoid robot moves, specifically in creating traditional Indonesian dance moves.
2) Software is created using PyQt which is cross-platformed, so that the software can be widely used on computer with different background of operating systems
3) Software is created to control movements of the robot on the sub-controller by using forward kinematics
4) Communication protocol between humanoid robot can be done with TCP protocol. The total delay time on the measured TCP communication protocol is 2 seconds

ACKNOWLEDGEMENTS

This paper is created solely for the author's Final Documentation. Acknowledgments are conveyed by the authors to Team TA 161701077 and Team Dago Concordia who participated in the manufacture and development of the traditional dancing humanoid robot platform used in this paper.

REFERENCES

[1] https://www.bps.go.id/linkTabelStatis/view/id/1387, (diakses 24 April 2017)

[2] M. Hirose and K. Ogawa, *Honda Humanoid Robots Development,* Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 365, issue 1850, pp. 11-19, 2007.

[3] I.W. Park, J. Y. Kim, J. Lee, and J. H. Oh, Mechanical Design of Humanoid Robot Platform KHR-3 (KAIST Humanoid Robot – 3:HUBO),International Conference on Humanoid Robots, pp. 321-326, 2005

[4] https://support.robotis.com/en/product/darwin-op.htm, (diakses 24 April 2017)

[5] J. Farrell, Object-Oriented Programming Using C++, 4th ed. Boston: Course Technology, Cengage Learning, 2009, pp.455–467.

[6] M. Lutz, Learning Python, 4th ed. Boston: O'Reilly Media, 2009, pp.533-540.

[7] Sun Microsystems, *System Administration Guide: IP Services*, California : Network Circle Santa Clara, 2002, p.40

Rancang Bangun *Development* Platform Robot *Humanoid* Penari Tarian Tradisional Indonesia

Ahmad Zaki Haninut Tuqo, Kusprasapta Mutjiarsa*
Program Studi Teknik Elektro, Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
soni@stei.itb.ac.id

***Abstrak*— Indonesia memiliki kekayaan budaya yang berlimpah; kaya akan kesenian, kaya akan tarian daerah tradisional. Tercatat sekurang-kurangnya 300 tarian daerah yang tersebar di 34 provinsi di Indonesia. Setiap tarian memiliki ciri-ciri dan ke-khasan tersendiri. Keberagaman tarian ini merupakan kekayaan yang tak ternilai harganya, sehingga perlu selalu dijaga dan dilestarikan. Teknologi adalah salah satu perangkat yang digunakan untuk membantu manusia. Penggunaan teknologi untuk pelestarian budaya adalah suatu keniscayaan. Pengembangan teknologi robot humanoid yang dapat menari tarian tradisional adalah salah satu usaha untuk melestarikan budaya Indonesia. Penggunaan teknologi robot humanoid yang dapat secara otomatis menari tarian tradisional di berbagai tempat umum atau tempat wisata juga bermanfaat untuk memperkenalkan budaya tarian daerah ke wisatawan, baik domestik maupun mancanegara. Robot penari yang dibuat adalah robot humanoid dengan 25 derajat kebebasan (Degree of Freedom - DOF), yang dibangun dari 12 motor servo pada kaki (kanan dan kiri), 10 motor servo pada tangan, 2 motor servo pada leher, dan 1 motor servo pada badan. Perhitungan gerak robot menggunakan kinematika balik (inverse kinematic) dan kinematika maju (forward kinematic) untuk menyusun gerakan dasar tarian. Suatu set gerakan tari disusun dari urutan gerakan-gerakan dasar. Untuk sinkronisasi gerakan tari dengan musik pengiringnya, robot dilengkapi dengan kemampuan mendeteksi musik. Komunikasi audio dilakukan antara perangkat sumber musik dengan robot menggunakan modul audio bluetooth. Dengan demikian robot dapat mendengarkan dengan baik ketika musik sedang dinyanyikan atau dimatikan, tanpa terganggu oleh suara lain di lingkungan sekeliling robot. Dalam implementasinya, robot dikembangkan untuk menarikan tarian daerah Palembang, yaitu tarian Gending Sriwijaya. Pada proses pengujian, platform robot penari humanoid ini dapat menarikan urutan gerakan tarian Gending Sriwijaya selama 4 menit 10 detik, dimulai dari gerakan masuk, kecubung, sembah duduk, stupa, tolak balak, ulur benang hingga hormat akhir. Robot dapat mulai menari saat musik pengiring diputar dan dapat berhenti ketika musik pengiring dimatikan.**

Kata Kunci— degree of freedom, deteksi musik, tarian dasar, gerakan jalan, Robot humanoid, robot penari tarian tradisional, development platform, sistem gerak.

I. PENDAHULUAN

Platform robot penari *humanoid* merupakan robot berjenis *bipedal* yang memiliki fungsi utama untuk melakukan gerakan tari tradisional Indonesia. Untuk menjalani fungsi utamanya, yaitu melakukan gerakan tari tradisional Indonesia, robot ini dilengkapi dengan spesifikasi-spesifikasi yang menunjang. Seperti penggunaan 25 *Degree of Freedom* (DOF), adanya modul untuk mendeteksi suara, serta dilengkapi dengan sensor inersia yang berguna sebagai sensor keseimbangan pada robot.

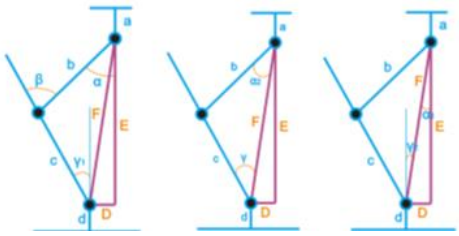
Dengan fungsi utama untuk melakukan gerakan tari seperti manusia, robot penari *humanoid* juga harus dapat melakukan gerakan yang menyerupai manusia sempurna. Terutama pada saat robot melakukan gerak jalan. Sehingga, perlu dilakukan pertimbangan mengenai distribusi tinggi dan berat robot yang dibuat dengan menjadikan perbandingan distribusi tinggi dan berat manusia yang memiliki proporsi tubuh ideal menjadi acuan distribusi tinggi dan berat komponen robot.[1]

Robot penari *humanoid* ini juga harus dapat melakukan adaptasi pada tipe lantai yang berbeda-beda. Sehingga dibutuhkan desain yang baik pada bagian kaki dengan mempertimbangkan kompleksitas dan tingkat adaptasi robot pada jenis lantai-lantai tertentu.[1]

Menurut Fred, Sias, dan Zheng (1990) untuk mendapatkan tingkat adaptasi yang tinggi pada keadaan lantai yang tidak rata, diperlukan sejumlah 7 DOF pada rancangan kaki tiap robot *humanoid* yang tersebar pada pinggul (3 buah), lutut (1 buah), dan pergelangan kaki (3 buah). Namun, karena tingkat kompleksitas yang tinggi dilakukan penyederhanaan dengan menghilangkan 1 DOF, yaitu pada bagian pergelangan kaki, yang tidak mempengaruhi tingkat adaptasi robot pada permukaan jalan yang tidak rata. Sehingga hanya diperlukan 6 DOF saja pada setiap rancangan kaki.[2]

Hal lain yang perlu diperhatikan dalam proses perancangan dan implementasi platform robot penari *humanoid* adalah metode yang digunakan untuk mendapatkan sudut dari masing-masing *joint*. Salah satu metode yang dapat digunakan adalah metode kinematika balik (*inverse kinematic*).

Metode kinematika balik merupakan metode untuk mendapatkan sudut dari masing-masing *joint* dengan berdasarkan kepada posisi akhir dari gerakan yang diinginkan. Metode ini membutuhkan data posisi akhir yang diinginkan serta panjang masing-masing *link*. Kemudian dilakukan komputasi hingga didapatkan sudut setiap *joint*.[3]



Gambar 1. Analisa kinematika pada bidang x-z

Komputasi dilakukan untuk mencari nilai-nilai sudut tiap *joint* dengan menggunakan persamaan sebagai berikut.

$$\alpha = \tan^{-1}\left(\frac{(Xgeser}{\sqrt{Ygeser2+(b\ +\ c-Zgeser)^2}}\right)+$$

$$cos^{-1}\left(\frac{b^2+Xgeser^2+Ygeser^2+(b+c-Zgeser)^2-c^2}{2b\sqrt{Xgeser^2+Ygeser^2+(b+c-Zgeser)^2}}\right)\ldots 1$$

$$\gamma_1 = \cos^{-1}\left(\frac{c^2+Xgeser^2+Ygeser^2+(b+c-Zgeser)^2-b^2}{2c\sqrt{Xgeser^2+Ygeser^2+(b+c-Zgeser)^2}}\right)$$

$$-tan^{-1}\left(\frac{(Xgeser}{\sqrt{Ygeser2+(b\ +\ c-Zgeser)^2}}\right)\ldots 2$$

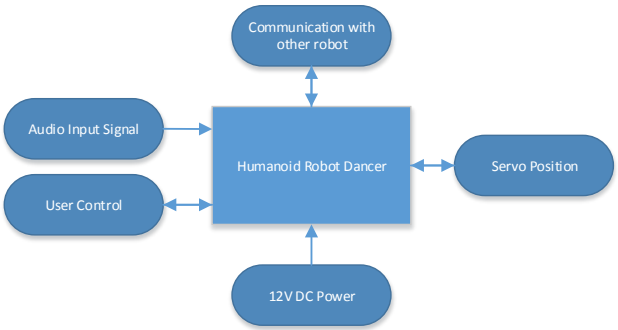
$$\beta = \cos^{-1}\left(\frac{c^2+Xgeser^2+Ygeser^2+(b+c-Zgeser)^2-b^2}{2c\sqrt{Xgeser^2+Ygeser^2+(b+c-Zgeser)^2}}\right)+$$

$$cos^{-1}\left(\frac{b^2+Xgeser^2+Ygeser^2+(b+c-Zgeser)^2-c^2}{2b\sqrt{Xgeser^2+Ygeser^2+(b+c-Zgeser)^2}}\right)\ldots 3$$

II. DESAIN

A. Desain Sistem

Platform robot penari *humanoid* dirancang untuk memenuhi fungsi dan spesifikasi yang telah ditentukan sebelumnya. Berdasarkan fungsi-fungsi yang harus dijalankan, sistem platform robot penari *humanoid* dapat dituliskan dalam bentuk diagram blok seperti pada Gambar 2.



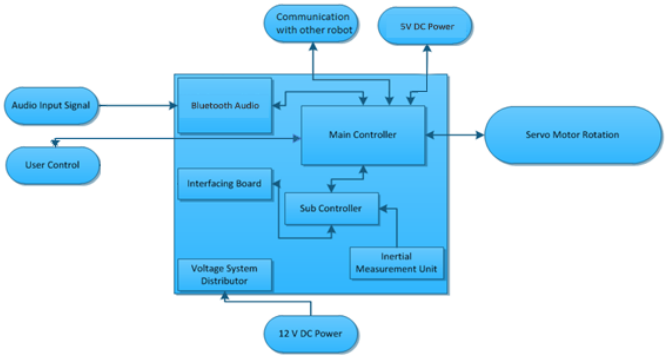
Gambar 2. Diagram blok level 0

Sistem platform robot penari *humanoid* memiliki input berupa kontrol user dan sinyal audio yang kemudian diproses untuk mendapatkan nilai-nilai posisi servo. Secara lebih detail, dengan mengamati fungsi-fungsi dari tiap modul yang digunakan, sistem platform robot penari *humanoid* dapat dituliskan dalam diagram blok level eperti tertera pada Gambar 3.

Pada tugas akhir ini, perancangan sistem berfokus pada sistem kerangka, sistem gerak perpindahan robot, dan sistem pendeteksi suara.

B. Desain Kerangka

Proses perancangan kerangka robot penari *humanoid* menggunakan berbagai pertimbangan seperti fungsi,

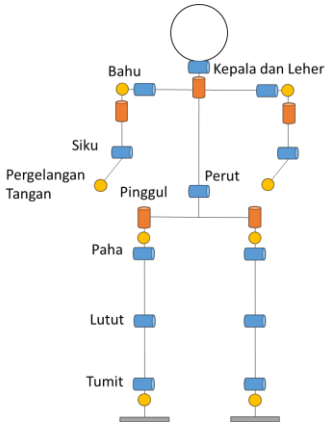


Gambar 3. Diagram blok level 1

kompleksitas, dan referensi platform lainnya. Referensi yang digunakan dalam proses perancangan adalah platform DARwin-OP dan platform bioloid Grand Prix.

Dengan mempertimbangkan fungsi utama untuk melakukan gerakan tarian, maka dilakukan perancangan untuk mendapatkan jumlah DOF sebanyak mungkin agar didapatkan gerakan yang tidak kaku. Sehingga ditentukanlah jumlah DOF sebanyak 25 buah dengan mempertimbangkan jumlah DOF pada setiap kaki berjumlah 6 buah,[2] jumlah DOF pada setiap tangan 5 buah, pada leher 2 buah, dan pada badan 1 buah seperti pada Gambar 4.

Selanjutnya dengan mempertimbangkan ketahanan struktur, maka dipilihlah bahan dasar kerangka utama menggunakan aluminium dan pada bagian tangan digunakan bahan dasar plastik ABS agar didapatkan bobot yang ringan, desain yang presisi (menggunakan cetak 3D), dan tingkat ketahanan yang tetap tinggi.

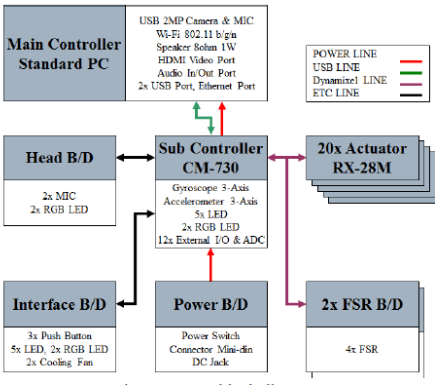


Gambar 4. Distribusi DOF pada robot

C. Desain Perangkat Keras

Desain perangka keras pada platform robot penari *humanoid* mengadaptasi dari desain perangkat keras pada *Dynamic Anthropomorphic Robot with Intelligence - Open Platform* (DARwin-OP) dengan melakukan beberapa perubahan untuk menyesuaikan fungsi dan fitur seperti penambahan *sub-controller*, perubahan jenis *main controller* yang digunakan, penambahan dan perubahan jenis beberapa *actuator*, serta penambahan komponen pendeteksi suara.

Darwin-OP sendiri merupakan salah satu platform robot *humanoid* yang dapat dikembangkan secara bebas baik pada bagian perangkat keras, maupun perangkat lunak. Platform ini juga telah menyediakan beberapa fitur yang dapat digunakan seperti *motion planning*, *walking*, *manipulating*, *vision processing*, dan *artificial intelligence*. [4]



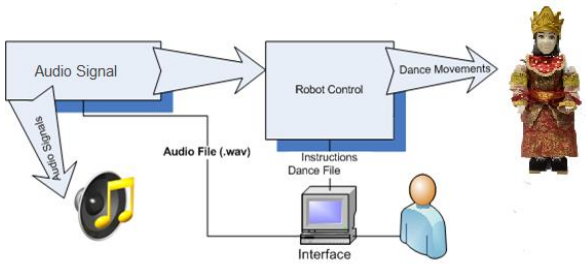
Gambar 5. Diagram blok DARwin-OP

Main Controller yang digunakan pada platform robot penari *humanoid* menggunakan jenis *single board computer* (SBC) dengan tipe Odroid-XU4 setelah melakukan perbandingan spesifikasi dan harga dengan tipe fitpc2i dan Raspberry-Pi. Kemudian dilakukan beberapa penyesuaian untuk mencapai fungsi dari platform robot penari *humanoid* dengan melakukan perubahan dan penambahan beberapa *actuator* bertipe XL320 dan penambahan *sub-controller* dengan tipe OpenCM 9.04c untuk mengendalikan *actuator* dengan tipe data dxl 2.0 dan level tegangan 7.4 V.



Gambar 6. Hubungan *main controller*, *sub-controller*, dan servo

Selain itu, dilakukan penambahan komponen untuk mendeteksi ada tidaknya musik pengiring yang sedang diputar. Komponen ini dihubungkan melalui *microcontroller* arduino yang hanya digunakan untuk memroses algoritma pendeteksian suara dan langsung mengirimkannya secara *real-time* kepada *main controller*. Komponen yang digunakan adalah *bluetooth audio module*. Komponen ini berguna untuk menghindari terganggunya suara yang diterima sistem oleh suara yang terdapat dilingkungan luar sistem.

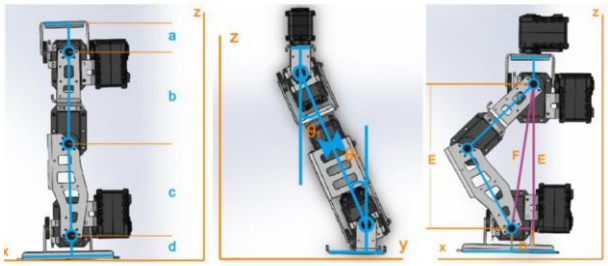


Gambar 7. Diagram kerja modul pendeteksi musik

D. Desain Perangkat Lunak

Desain perangkat keras pada platform robot penari *humanoid* mengadaptasi dari desain perangkat lunak pada platform DARwin-OP. Adaptasi dilakukan dengan mengambil *framework* dari platform tersebut, kemudian penggunaan dan modifikasi pada *framework* tersebut difokuskan pada bagian sistem penggerak. Tepatnya pada *source code* bagian *walking*.

Perancangan perangkat lunak bagian sistem penggerak, tepatnya bagian *walking* robot menggunakan metode kinematika balik. Dengan metode ini, ditentukan beberapa posisi akhir kaki yang kemudian dapat dilakukan komputasi untuk mendapatkan sudut tiap *joint*. Selain posisi akhir dari kaki, dengan metode kinematika balik juga diperlukan panjang-panjang tiap-tiap link yang digunakan. Kemudian dilakukan pencari nilai sudut-sudut tiap *joint* dengan persamaan **1, 2, dan 3**.



Gambar 8. Konfigurasi kaki robot pada bidang x-z dan y-z

III. IMPLEMENTASI

A. Kerangka Platform Robot Penari Humanoid

Implementasi rangka dari platform robot penari *humanoid* telah mengikuti desain yang telah ditentukan sebelumnya. Hasil dari implementasi rangka telah sesuai dengan harapan awal karena didapatkan ketelitian yang cukup tinggi pada proses pemotongan, pembengkokan, maupun pencetakan komponen 3D



Gambar 9. Salah satu bagian hasil implementasi rangka

B. Modul Penggerak (Gerakan Jalan Robot)

Implementasi modul penggerak bagian jalan dilakukan dengan mencari posisi akhir dari kaki, kemudian dilakukan perhitungan tiap-tiap *joint* pada kaki robot dengan menggunakan persamaan **1, 2, dan 3** dengan sebelumnya mengukur panjang tiap *link* pada robot seperti yang tertera pada Gambar 8.



Gambar 10. Implementasi gerakan jalan pada robot

C. Modul Pendeteksi Suara

Implementasi modul pendeteksi suara dilakukan dengan terlebih dahulu membaca nilai awal saat ada dan tidak ada lagu. Selanjutnya dapat ditentukan nilai ambang batas, yaitu nilai terkecil yang merepresentasikan titik 0 yang digunakan. Di mana pada saat nilai terbaca lebih kecil dari nilai ambang, maka akan dianggap tidak ada suara. Berlaku juga sebaliknya.

Selanjutnya, untuk mendapatkan hasil yang lebih presisi disusun algoritma untuk mengecek rentang kondisi. Sehingga jika terdapat suatu anomali nilai yang berubah tiba-tiba, nilai tersebut tidak merepresentasikan perubahan kondisi ada tidaknya musik yang diputar.



Gambar 11. Hasil deteksi musik dengan algoritma yang diterapkan

IV. PENGUJIAN

Pengujian dilakukan untuk membuktikan bahwa desain yang telah diimplementasikan memenuhi spesifikasi awal yang ditentukan. Modul-modul yang akan dilakukan pengujian pada tugas akhir ini meliputi kerangka robot, modul penggerak, modul pendeteksi suara, dan integrasi sistem.

A. Kerangka Platform Robot Penari Humanoid

Pengujian rangka dilakukan dengan cara melakukan pengukuran terhadap rangka yang telah diimplementasikan. Hasil dari pengujian tersebut adalah sebagai berikut.

- Tinggi robot : 56,0 cm
- Rentang tangan : 52,6 cm
- Rentang kaki : 57,0 cm
- Massa robot : 2,2 kg

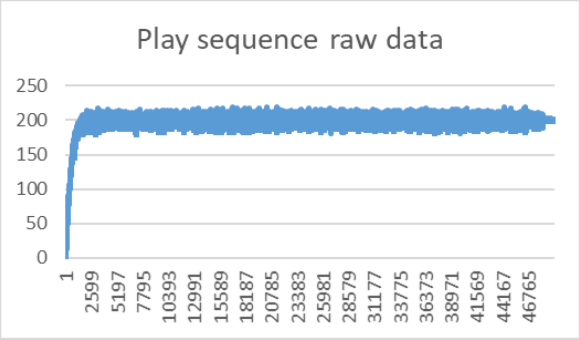
Selain melakukan pengujian dengan melakukan pengukuran, pengujian rangka juga dilakukan dengan cara membuat robot menari sampai robot lemas. Pada pengujian bagian ini didapatkan waktu maksimum robot menari adalah 4 menit 39 detik.

B. Modul Penggerak (Gerakan Jalan Robot)

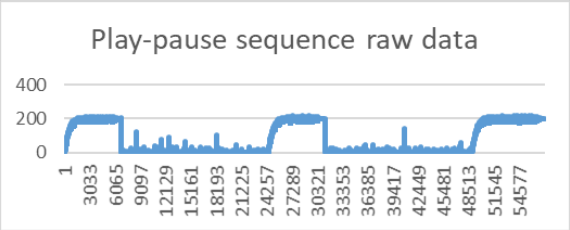
Pengujian modul penggerak dilakukan dengan melakukan pengesetan parameter-parameter yang telah terdapat pada *source code* platform DARwin-OP sehingga didapatkan robot bergerak lurus. Parameter-parameter ini dapat mengubah posisi kaki saat berjalan, tinggi langkah kaki, jarak kaki melangkah dan lain sebagainya. Salah satu parameter yang sangat mempengaruhi kondisi berjalan adalah *hip_pitch_offset*. Parameter ini mengatur arah kecondongan robot (ke depan atau belakang)

C. Modul Pendeteksi suara

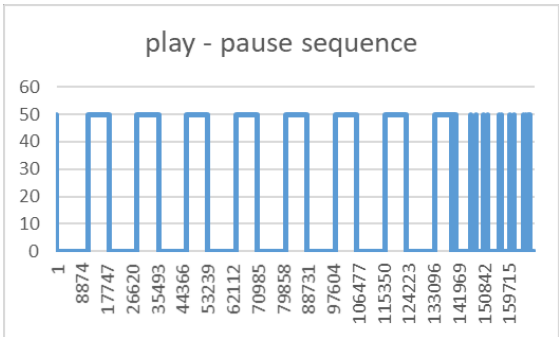
Pengujian modul pendeteksi suara dilakukan dengan melakukan pengecekan nilai awal saat musik diputar, kemudian dilakukan pengecekan pada kondisi musik diputar dan diselingi *pause* ditengah-tengah. Selanjutnya dilakukan pengecekan hasil akhir dengan menggunakan algoritma yang telah diimplementasikan.



Gambar 12. Hasil pembacaan awal saat musik dimainkan



Gambar 13. Nilai hasil pembacaan awal dengan mode play-pause



Gambar 14. Bacaan keadaan ada tidaknya musik dengan algoritma yang telah diimplementasikan

Hasil ini menunjukan kondisi saat ada tidaknya musik dengan menggunakan algoritma yang telah diimplementasikan. Pada pengujian diberikan beberapa jeda untuk mengamati kesesuaian kondisi. Hasil pengujian yang didapatkan sudah baik. Yaitu, sudah dapat membedakan antara ada dan tidaknya musik saat musik dimainkan ataupun dimatikan.

D. Pengujian integrasi

Pada pengujian integrasi robot penari *humanoid*, didapatkan poin-poin hasil pengujian sesuai dengan lingkup hasil pengujian sebagai berikut.

- UI telah dapat melakukan komunikasi dengan robot

Pada pengujian integrasi, dilakukan pengujian UI untuk menjalankan perintah demo pada robot. Perintah ini terdapat di dalam mode *Custom* pada UI. Pada perintah demo, telah dilakukan penggabungan seluruh modul pada robot.

- Robot melakukan gerakan tarian dasar dengan sinkronisasi gerakan antar robot

Pada pengujian gerakan tarian dasar, dilakukan pengujian menggunakan tarian Gending Sriwijaya dengan tambahan aksesoris pada robot untuk membuat robot terlihat lebih menarik. Pada pengujian ini dilakukan sinkronisasi gerakan antar robot yang dilakukan pada proses pengujian.



Gambar 15. Pengujian integrasi robot

Robot dapat menarikan gerakan tarian dasar (gerakan yang diuji merupakan gerakan dari tari gending sriwijaya) dipadukan dengan gerakan jalan di antara gerakan tarian. Pada pengujian integrasi ini juga dilakukan pengujian sinkronisasi gerakan dengan melakukan komunikasi antar robot. Dari hasil pengujian didapatkan gerakan robot telah tersinkronisasi.

- Robot mendeteksi ada tidaknya musik

Pada pengujian integrasi robot dilakukan pengujian sistem deteksi suara dalam integrasi gerakan robot. Hasil pengujiannya, didapatkan bahwa robot dapat berhenti bergerak ketika musik berhenti walaupun gerakan robot belum selesai dijalankan seluruhnya.

V. SIMPULAN

Kerangka robot telah diimplementasikan dengan baik. Yaitu telah memenuhi spesifikasi awal dengan tinggi total robot sebesar 56,0 cm dan berat robot sebesar 2,2 kg dengan kemampuan robot untuk beroperasi dengan kerangka ini selama 4 menit 39 detik. Kemudian robot telah dapat berjalan lurus pada kondisi lantai yang rata dengan parameter tertera pada Tabel 1.

Pada proses pengujian integrasi, robot telah dapat membedakan kondisi saat ada dan tidak ada suara. Pada kondisi musik di *pause* di tengah tarian, maka robot akan berhenti menari. Robot penari *humanoid* juga telah diuji untuk menarikan tarian Gending Sriwijaya selama 4 menit 10 detik.

Kesulitan yang terjadi pada saat pengerjaan proyek ini adalah minimnya referensi dan tutorial yang tersedia untuk melakukan modifikasi pada *framework* bawaan DARwin-OP. Sebagian besar referensi di luar hanya tata cara melakukan modifikasi sederhana dan sebagian lainnya hanya cara-cara menggunakan framework DARwin-OP.

REFERENSI

[1] Xia, Z., Liu, L., Xiong, J., & Chen, K. (2007). Design aspects and development of humanoid robot THBIP-2. *Robotica*, 1-8.

[2] Fred, R., Sias, J., & Zheng, Y. F. (1990). How Many Degrees-of-freedom Does a Biped Need. Intelligent Robots and Systems (pp. 297-302). Ibaraki: IEEE.

[3] Izma Azizah, Hanna, Silvia Anandita, dan Tio Dwi Laksono, "*Robot Humanoid Penari Tarian Daerah Indonesia* (Perancangan)", Dokumen B300, Institut Teknologi Bandung, 2014

[4] Ha, I., Tamura, Y., Asama, H., Han, J., & Hong, D. W. (2011). Development of Open Humanoid Platform DARwIn-OP. SICE Annual Conference (pp. 2178-2181). Tokyo: Waseda University.

Thermal Vision pada Manusia dengan Pengaruh Terhadap Warna Pakaian

Mona Arif M. B*, Rizki Alandani, dan Gusti Made Arya
Teknik Elektro
Universitas Lampung
Bandar Lampung, Indonesia
mona.batubara@eng.unila.ac.id

***Abstrak*—Machine vision dapat diaplikasikan pada banyak bidang, salah satunya adalah pencitraan manusia. Pencitraan manusia dapat dilakukan dengan menggunakan kamera thermal dengan pendeteksian radiasi panas yang dipancarkan oleh tubuh manusia. Panas tubuh manusia diserap oleh pakaian yang dikenakan sehingga mempengaruhi hasil pencitraan kamera inframerah. Pada penelitian ini akan dilihat bagaimana pengaruh perbedaan warna pakaian terhadap penyerapan panas dari tubuh manusia. Dengan menggunakan variasi warna pakaian yang berbeda maka akan dihasilkan citra dengan pola panas tubuh berbeda bergantung pada kecerahan warna pakaian.**

Kata Kunci—thermal vision; kamera inframerah; colour temperature; human recognition

I. PENDAHULUAN

Teknologi pengolahan citra terus dikembangan baik pada bidang robotik hingga industri. Pada bidang robotik, teknologi *machine vision* diterapkan untuk meningkatkan fungsi. Proses pendeteksian objek saat ini telah beralih menggunakan *sensor array* yang tertanam dalam kamera. Lebih jauh lagi, penggunaan kamera thermal semakin marak untuk mendapatkan kemampuan tertentu yang tidak dapat dilakukan oleh kamera optik. Pada kondisi gelap (kondisi dimana intensitas cahaya rendah), maka kamera optikal akan sulit mendeteksi keberadaan objek. Pada kondisi tersebut kamera thermal lebih baik dalam pendeteksian karena kamera thermal menangkap radiasi panas dari objek yang akan dideteksi. Manusia memiliki panas tubuh yang dapat dengan mudah dicitrakan oleh kamera inframerah. Aplikasi pendeteksian manusia dengan menggunakan kamera inframerah sudah banyak diterapkan, seperti pada pemeriksaan bandara, *nightvision*, alat militer, *security*, dsb.

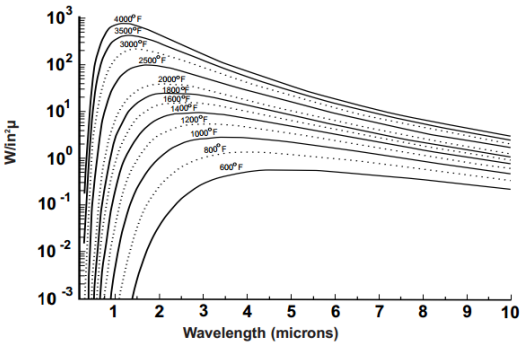
Penelitian ini merupakan implementasi dan penggunaan sensor *thermal* untuk penjajakan panas tubuh manusia[1], [2] yang variatif terhadap warna pakaian yang digunakan. Penelitian ini juga termasuk bagian dari pengembangan terhadap riset yang telah kami lakukan terhadap sensor infrared dan thermal pada *Unmanned Aerial Vehicle*.

II. TINJAUAN PUSTAKA

A. *Radiasi Thermal*

Energi panas dapat berpindah dengan tiga metode yaitu konduksi, konveksi dan radiasi. Panas atau *thermal* adalah energi yang dihasilkan dari gejolak elektron, atom, ion, dan molekul dalam jumlah besar. Jika diberikan kepada material maka sejumlah energi tersebut akan ditransfer dan bergerak mengakibatkan temperatur lokal meningkat. Perpindahan pada zat padat adalah konduksi panas, pada zat cair adalah konveksi, dan tanpa perantara disebut dengan radiasi. Radiasi oleh alam telah diperkenalkan oleh Maxwell sebagai *electromagnetic waves*, terbentang dari X-*rays* hingga *microwaves* melalui cahaya tampak dan *infrared*.

Radiasi *thermal* yang tampak pada mata manusia yaitu pada panjang gelombang 0.4 hingga 0.8 um (gelombang cahaya tampak). Setiap objek yang memancarkan radiasi *thermal* memiliki temperatur yang lebih besar dari *absolute zero* (temperatur dimana tidak ada *vibrational motion* pada objek atom; temperatur -460-deg F). Getaran atom yang tinggi ada pada objek dengan sifat panas, sebaliknya objek dingin memiliki getaran yang rendah. Reaksi partikel elektrik positif (proton) dan negatif (elektron) membuat medan dan gelombang elektromagnetik yang sepadan dengan besar pancaran radiasi *thermal*. Ketika radiasi gelombang elektromagnetik menggapai obejek yang dingin, medan elektromagnetik dan medan listrik bereaksi pada partikel atom objek dingin. Pada kondisi ini objek dingin akan menyerap energi elektromagnetik dari objek panas. Energi ditransfer dari objek panas ke objek dingin tanpa kontak fisik dan tanpa medium diantaranya. [3]



Gambar 1. Skala linier grafik kurva Planck

Kebanyakan radiasi energi *thermal* dari objek yang memiliki panas berada pada spektrum *infrared*, yaitu dari 0.1 hingga 100 microns (1 microns = 10^{-6} m). Jumlah energi yang diradiasikan oleh objek panas and gelombang elektromagnetik ditentukan oleh temperatur objek. Hubungan antara energi radiasi dan gelombang elektromagnetik ditunjukkan pada grafik yang disebut "*Planck's Curve*" –lihat gambar 1 – yang ditemukan oleh Max Planck. Dari grafik tersebut kita dapat melihat bahwa objek panas meradiasikan energinya di atas spektrum gelombang. Dapat dilihat pula semakin tinggi temperatur heater maka peak energi gelombang akan semakin mengecil. Gelombang energi puncak dan temperatur heater dan dikalkulasi menggunakan hukum perpindahan Wien:

$$\text{Peak Energy Wavelength (microns)} = \frac{5269 \text{ microns } °R}{\text{Temp.}(°F) + 460} \quad (1)$$

Ada parameter fisik yang harus diperhatikan saat menghitung berapa banyak heat energy objek [4], yaitu:

Stefan-Boltzman *Equation* – digunakan untuk menghitung jumlah daya radiasi dari permukaan objek (*blackbody*) pada temperatur tertentu, ditunjukkan pada formula

$$E = \sigma T^4 \qquad (2)$$

E = energi per unit area (W m^{-2})

σ = konsanta (5.67 x 10^{-8} W m^{-2} K^{-4})

T = temperatur permukaan (^oK)

View Factor – menjelaskan prosentase energi pancaran objek yang mengenai produk. Stefan-Boltzman equation hanya menghitung radiasi dari sumber panas, namun tidak menghitung energi radiasi yang mengenai objek. Terdapat hubungan geometrik antara ukuran heater dan objek penerima, dan jarak diantara keduanya. Semakin dekat heater dengan objek penerima, maka akan semakin efisien transfer panas tersebut.

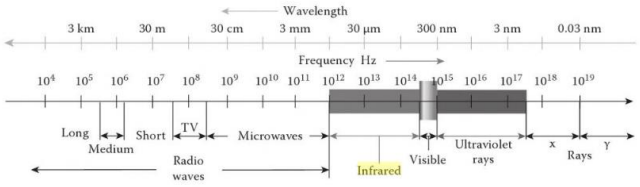
Emmisivity - menjelaskan bagaimana energi radiasi yang diserap oleh produk. Tidak semua energi yang menggapai benda itu diserap. Sebagian akan di refleksi dan sebagian lain akan diteruskan benda tersebut.

Total Incident Energy = Absorbed + Reflected + Transmitted

Ini adalah sifat fisik produk yang menentukan seberapa baik objek menyerap energi radiasi yang mengenainya. Sifat fisik ini menentukan seberapa baik sebuah permukaan memancarkan energi radiasi. *Emissivity* adalah nilai numerik dengan skala antara 0 dan 1 yang menunjukkan seberapa baik kemampuan objek untuk menyerap atau memancarkan energi radiasi. Objek yang dengan sempurna menyerap energi radiasi disebut dengan *blackbody* (memiliki *emissivity* = 1).

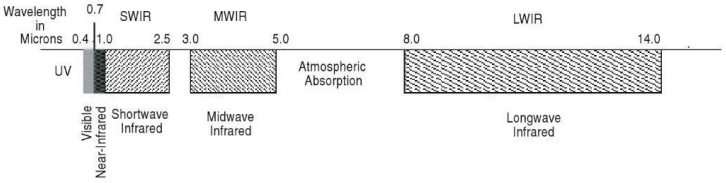
B. Spectrum Infrared *pada Gelombang Elektromagnetik*

Gelombang elektromagnetik adalah gelombang yang tidak memerlukan medium transmisi dan dideskrisikan dengan *velocity*, frekuensi, dan *wavelength* (lihat gambar 2). Berdasarkan perbedaan panjang gelombang dan frekuensi gelombang elektromagnetik terdiri dari sinar gamma (<0.01 nm), sinar-X (0.01-10nm), ultraviolet (10-400 nm), cahaya tampak (390-750 nm), *infrared* (0.75-1 mm), microwave (1 mm – 1 m), dan gelombang radio (1 mm - km) seperti ditunjukkan pada gambar 3. Gelombang *infrared* meliputi semua radiasi elektromagnetik yang lebih dari gelombang cahaya tampak, tapi lebih pendek dari *microwaves*.



Gambar 2. Spektrum gelombang elektromagnetik [5]

Spektrum-spektrum *infrared* dibagi menjadi beberapa spektrum diantaranya *Near Infrared* (NIR), *Shortwave Infrared* (SWIR), *Midwave Infrared* (MWIR), dan *Longwave Infrared* (LWIR). Panjang gelombang NIR memiliki bentangan antara 0.7-1.0 um, gelombang SWIR dengan bentangan panjang gelombang antara 1.0-2.5 um, MWIR pada bentangan panjang gelombang antara 3.0-5.0 um, dan panjang gelombang LWIR adalah pada bentangan 8.0-14.0 um. Bentangan spektrum gelombang *infrared* dapat dilihat pada gambar 4.



Gambar 3. Spektrum gelombang Inframerah dengan jenis-jenisnya (NIR, SWIR, MWIR, LWIR) [5]

Pada panjang gelombang 3 um atau lebih tinggi, radiasi yang dicitrakan dari benda-benda memancar berdasarkan temperatur secara signifikan, karenanya sering disebut dengan *thermal infrared*. Spektrum *thermal infrared* dibagi menjadi dua spektrum utama yaitu MWIR dan LWIR. Diantara kedua spektrum tersebut terbentang pancaran atmosfir dengan panjang gelombang antara kurang lebih 5 dan 8 um. Pada spektrum tersebut, pencitraan termal sangat sulit untuk dilakukan karena begitu kabur dengan termal udara. Range spektrum dengan panjang gelombang lebih dari 14 um diistilahkan dengan *Very Longwave Infrared* (VLWIR)

dengan aplikasi yang lebih tinggi, namun pada penelitian ini hanya dibahas sampai spektrum LWIR.

Nilai radiasi yang dipancarkan bergantung pada temperatur dan *emissivity* dari material. Sebagai contoh pada tubuh manusia dengan suhu rata-rata (37 deg C), melalui pengukuran empiris sedikitnya nilai *emissivity* yaitu 0.91 pada MWIR dan sekurang-kurangnya 0.97 pada LWIR. Objek dengan efisiensi tinggi memantulkan cahaya, maka semakin sedikit efisiensi memancarkan energi termalnya (*emissivity*) [5]. Nilai *emissivity* maksimal 1.0 (biasanya e = 0.99 atau lebih besar) artinya objek secara sempurna menyerap dan memancarkan radiasi, maksimum *emissivity* biasa diistilahkan dengan *blackbody*, dan *emissivity* yang lebih rendah biasa disebut graybody. Pada banyak kasus, deteksi objek dengan *thermal infrared* lebih baik menggunakan LWIR karena emisi LWIR lebih tinggi dari MWIR. Gambar 5 dapat dilihat perbandingan hasil pencitraan pada spektrum *infrared* yang berbeda.



(a) (b)

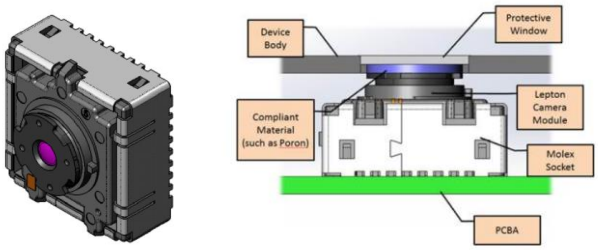
Gambar 5. Citra wajah yang ditangkap secara simultan pada (a) spektrum cahaya tampak 0.4-0.7 um dengan hasil *grayscale*, (b) *longwave infrared* 8.0-14.0 um dengan *colormap*

C. *Deteksi Infrared dengan Kamera LWIR (Lepton FLIR)*

Material yang digunakan untuk deteksi *infrared* salah satunya adalah jenis *bolometer*, yang dapat mengukur kekuatan radiasi elektromagnetik berdasarkan panas material dengan hambatan listrik yang bergantung suhu. Lebih spesifik lagi, material jenis *microbolometer* digunakan sebagai detektor pada kamera *thermal* tanpa pendingin yang biasa disebut *uncooled microbolometer*. Penelitian menunjukkan bahwa *microbolometer* memiliki performa yang mendekati ideal untuk sensor *infrared* suhu ruang dan ditujukan untuk konstruksi pencitraan inframerah tanpa pendingin yang rendah biaya [6]. Microbolometer *array* ditempatkan pada kamera untuk pendeteksian radiasi panas (*infrared*) pada objek dengan panjang gelombang 7.5-14 um.

Kebanyakan kamera thermal digunakan untuk *remote-sensing* adalah *cooled system*. *Cooled system* sangat sensitif dan akurat dalam pengindraan *thermal*, sistem ini juga besar, mahal, dan memakan banyak *power*. Karena ukurannya dan konsumsi *power* yang besar, maka *system cooled thermal* camera tidak digunakan pada *small unmanned aerial vehicle*. Bagaimanapun, *microbolometer thermal camera* tidak sesensitif dan seakurat *cooled system* [7].

Lepton FLIR merupakan salah satu sistem kamera *infrared* yang terintegrasi dengan LWIR microbolometer sensor array, perakitan fixed-focus lens, dan elektronika signal-processing. Kamera Lepton FLIR telah tertanam fungsi-fungsi pengolahan citra *thermal* digital diantaranya kompensasi otomatis *thermal* lingkungan, *noise filter*, koreksi *non-uniformity*, dan *gain control* yang terintegrasi pada satu sistem.

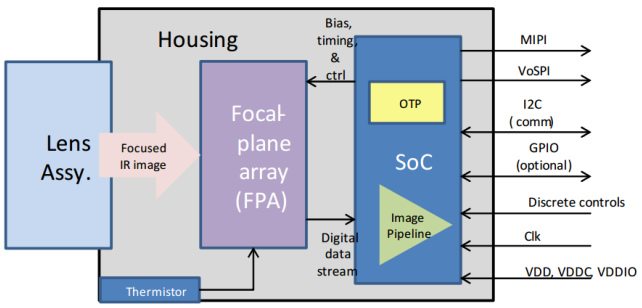


(a) (b)

Gambar 5. (a) Tampilan fisik (b) Struktur desain penyusun kamera FLIR Lepton [8]

Pendeteksian panas objek menggunakan teknologi sensor *Uncooled vanadium-oxide* (Vox) microbolometer yang mana temperaturnya akan berfluktuasi sesuai respon fluks insiden. Perubahan temperatur akan menyebabkan perubahan yang proposional pada masing-masing resistansi microbolometer. Material Vox *microbolometer* pada kamera memiliki *temperature coefficient of resistance* (TCR) yang tinggi dan *noise* yang rendah, menyebabkan *thermal* sensivity yang sangat baik dan uniformity yang stabil. Sensivitas deteksi *thermal* pada kamera ini kurang dari <50 mK atau 0,050 C. [8]

Bentuk *body* dan struktur fisik dari kamera FLIR Lepton dapat dilihat pada gambar 6. Dengan ukuran 8.5 x 11.7 x 5.6 mm (tanpa socket), 10.6 x 11.7 x 5.9 mm (dengan socket). Berdasarkan spesifikasinya, bagian lensa memiliki FOV-horizontal 51-deg, dan 63.5-deg diagonal dengan piksel aktif 80 (h) × 60 (v) digunakan untuk menangkap citra yang relatif lebih besar dari kamera *thermal* pada umumnya. Hasil pencitraan dari segmen lensa dikirim ke *Focal Plane Array* (FPA) untuk dikonversi dan menghasilkan arus yang periodik. Serial stream dari FPA diterima oleh perangkat *system on chip* (SoC) yang menyajikan pemrosesan sinyal dan *output formatting*. Diagram arsitektur sederhana dari modul kamera Lepton ditunjukkan pada gambar 7.



Gambar 7. Arsitektur kamera Lepton. [8]

Kamera *thermal* merupakan salah satu instrumen untuk mengukur dan mencitrakan panas suatu objek melalui radiasi *infrared* yang dipancarkannya. Sensor *array* pada kamera akan bekerja dengan memetakan radiasi tersebut sehingga menghasilkan informasi distribusi suhu dan *thermal* dari suatu objek.

III. METODE PENELITIAN

A. *Perancangan Alat dan Sistem*

Pencitraan objek dilakukan dengan menggunakan kamera thermal inframerah untuk melihat visualisasi panas yang tersebar dalam objek panas. Objek panas dalam peneitian ini adalah pakaian yang sedang dipakai. Tubuh manusia memiliki panas dan akan mengalirkan panas ke pakain yang sedang digunakannya. Dalam penelitian ini akan dilihat distribusi panas pada pakaian yang sedang digunakan dan pengaruhnya terhadap warna pakaian.

Beberapa alat yang digunakan pada penelitian ini dapat dilihat pada TABEL I.

TABEL I. JUSTIFIKASI PENGGUNAN KOMPONEN

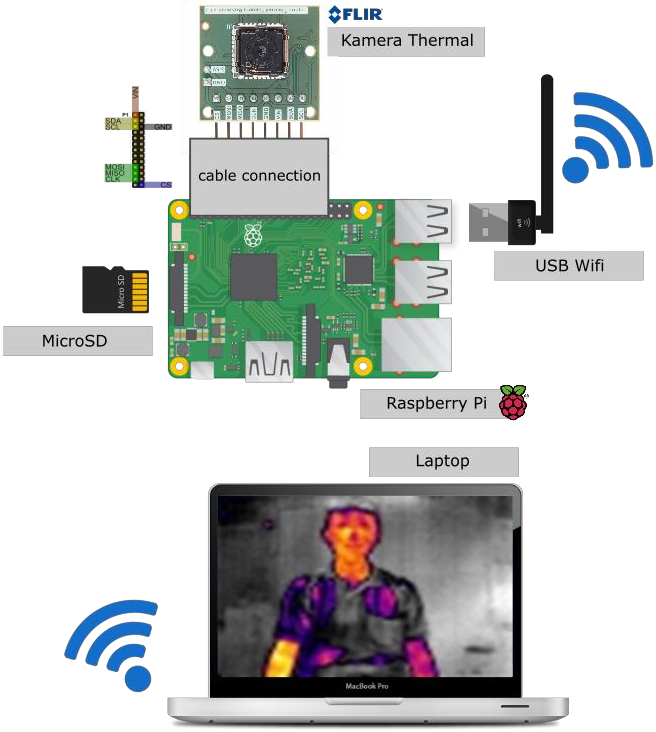
No	Nama Komponen	Justifikasi Penggunaan
1	Kamera LWIR (*Long Wave Infra-Red*) FLIR Lepton	Menangkap radiasi panas dari pakaian serta mencitrakannya secara digital
2	Raspberry Pi 3	Komputer pemrosesan utama untuk pengolahan dan penyimpanan citra digital
3	Raspbian Linux-*based Operating System*	Sistem operasi yang digunakan berbasis linux untuk memudahkan komunikasi antar device
4	*Software OpenCV* dan *Eclipse IDE C++*	Perangkat lunak untuk pengolahan citra digital berbasis coding dengan bahasa C++
5	Laptop Corei3 2,4 Ghz 4GB RAM	Laptop sebagai user interface untuk melihat hasil pengolahan citra inframerah pada pakaian

Sebuah sistem memiliki piranti penyusun yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*). Sistem secara sederhana terdiri dari masukan (*input*), proses, dan keluaran (*output*). Pada sistem pendeteksian panas objek, elemen-elemen penyusunnya diklasifikasi sebagai berikut:
Elemen masukan dari sistem adalah perangkat keras berupa sensor atau *transducer* yang mengirimkan sinyal informasi kepada *prosesor* utama (*main processor*). Sensor yang dipakai pada sistem pendeteksian panas ini adalah sensor *infrared* yang telah dipaketkan dengan sistem kamera digital. Pada penelitian ini digunakan kamera *longwave infrared* LEPTON FLIR yang mendeteksi radiasi pada panjang gelombang 8 – 14 microns.

Komponen proses pada sistem bekerja sebagai otak dan pusat kendali dari sistem tersebut. Pemrosesan utama pada sistem yang hendak dibangun pada penelitian ini menggunakan mini komputer Raspberry PI yang *portable* untuk pengolahan citra digital. Pengolahan citra digital membutuhkan kinerja prosesor yang lebih besar dibanding pemrosesan lainnya, karenanya pada penelitian ini menggunakan komputer Raspberry Pi 3 model B dengan *processor* ARMv8 CPU 1.2GHz 64-bit quad-core.

Keluran (*output*) dari suatu sistem adalah hasil dari pemrosesan. Keluaran yang dihasilkan dari pemroses dapat berupa tulisan (huruf, angka, simbol khusus), *image* (dalam bentuk grafik atau gambar), suara, dan bentuk lain yang hanya dapat dibaca oleh mesin (*machine-readable form*). Komputer Raspberry Pi akan menghasilkan keluaran berupa informasi digital mengenai citra distribusi panas pada pakaian. Informasi citra panas objek dapat dilihat melalui komputer pengguna secara grafis untuk dianalisis kemudian.

Gambar 1 berikut memperlihatkan perancangan sistem pencitraan deteksi panas pada objek yang ditangkap oleh kamera thermal.



Gambar 8. (a) Sketsa letak komponen-komponen utama pada Raspberry Pi, (b) Diagram perangkat keras internal Raspberry Pi

B. *Metode Pengambilan Data*

Kamera thermal LWIR dapat menangkap radiasi inframerah yang terpancar dari panas tubuh manusia. Panas tubuh manusia mengalir melalui proses konduksi ke pakaian yang sedang digenakan. Penelitian ini akan melihat hasil penyerapan panas pakaian terhadap tubuh manusia

berdasarkan warna pakaian tersebut. Objek yang akan dicitrakan adalah dua orang dengan jenis kelamin dan model pakaian yang berbeda. Seorang pria dengan mengenakan pakaian kaos oblong dan seorang wanita dengan mengenakan pakaian mukena duduk sebagai objek yang akan difoto oleh kamera inframerah seperti terlihat pada Gambar 9.

Secara teori warna akan berpengaruh pada penyerapan panas. Warna gelap akan cenderung menyerap panas lebih banyak daripada warna terang. Jenis warna yang digunakan yaitu warna putih, biru, dan hitam pada pakaian yang dikenakan masing-masing objek yang hendak dicitrakan. Dua orang sebagai objek tersebut berolahraga terlebih dahulu guna meningkatkan panas tubuhnya agar penyerapan panas ke pakaian dapat maksimal.

Pengolahan citra menggunakan library OpenCV pada IDE C++ dikonfigurasikan untuk menampilkan citra dengan range panas yang dinamis sesuai dengan radiasi terendah dan tertinggi yang terbaca oleh kamera LWIR.

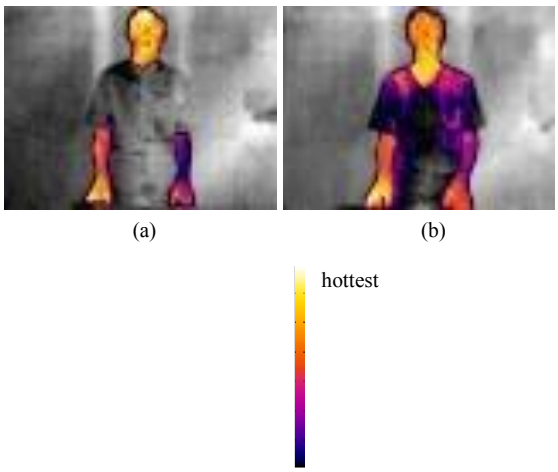


(a) (b)

Gambar 8. Objek pencitraan (a) pria dengan kaos, (b) wanita dengan pakaian tertutup (mukena)

IV. DATA HASIL DAN PEMBAHASAN

Hasil dari pencitraan terlihat pada Gambar 9 dan Gambar 10 sebagai citra yang dihasilkan kamera thermal LWIR. Gambar 9 memperlihatkan citra panas yang dihasilkan objek pria dengan mengenakan kaos lengan pendek dengan variasi warna yang berbeda. Citra pakaian warna putih (Gambar 9a) tidak memiliki panas yang sama dengan panas kulit manusia dengan tidak menyerap panas tubuh manusia dengan baik. Gambar 9b menjukkan ada penyerapan panas pada pakaian berwarna biru dari tubuh manusia. Warna hitam pada pakaian yang dikenakan ditampilkan pada citra dengan panas yang serupa dengan tubuh manusia menunjukkan penyerapan yang baik dari panas tubuh manusia seperti terlihat pada Gambar 9c.



(a) (b)



(c)

Gambar 9. Citra thermal objek pria dengan pakaian berwarna (a) putih, (b) biru, (c) hitam.

Pada objek wanita yang mengenakan pakaian lebih tertutup terlihat hasil pencitraan pada Gambar 10. Pakaian dengan warna putih tidak terlihat pada citra Gambar 10a menampakkan panas yang sama dengan tubuh manusia yang terbuka (telapak tangan dan wajah). Gambar 10b memperlihatkan visualiasi panas pakaian berwarna biru dengan penyerapan panas sebagian pada pakaian yang dikenakan. Gambar 10c menampilkan penyerapan panas dari tubuh manusia pada pakaian berwarna hitam yang lebih baik daripada warna biru dan putih. Citra pada pakaian berwarna hitam, sisi panasnya membentuk grafis lekuk tubuh manusia.





(c)

Gambar 10. Citra thermal objek wanita dengan pakaian berwarna (a) putih, (b) biru, (c) hitam.

Secara konsekuen, gambar yang dihasilkan kamera thermal pada pencitraan objek menampilkan hasil yang berbeda pada warna pakaian yang berbeda, baik pada objek pria maupun objek wanita dengan model pakaian yang digunakan masing-masing.

V. KESIMPULAN

Penelitian ini menunjukkan perbedaan penyerapan panas pada pakaian yang dikenakan tubuh manusia dengan variasi warna berbeda. Hasil citra yang dihasilkan kamera *thermal* menunjukkan perbedaan panas yang ditampilkan dengan tiga warna berbeda (putih, biru, dan hitam). Citra kamera *thermal* menampilkan pakaian dengan warna gelap menyerap panas dari tubuh lebih baik daripada pakaian dengan warna terang.

REFERENSI

[1] G. Cielniak, "People tracking by mobile robots using thermal and colour vision," 2007.

[2] M. Correa, G. Hermosilla, R. Verschae, and J. Ruiz-del-Solar, "Human Detection and Identification by Robots Using Thermal and Visual Information in Domestic Environments," *J. Intell. Robot. Syst.*, vol. 66, no. 1–2, pp. 223–243, Apr. 2012.

[3] X. Maldague, Ed., *Infrared methodology and technology*. Yverdon, Switzerland ; Langhorne, Pa., U.S.A: Gordon and Breach Science Publishers, 1994.

[4] "Radiant Heating With Infrared; A Technical Guide To Understanding And Applying Infrared Heaters." Watlow Electric Manufacturing Company, 1999.

[5] B. Bhanu and I. Pavlidis, Eds., *Computer vision beyond the visible spectrum*. London: Springer, 2005.

[6] P. W. Kruse, Ed., *Uncooled Infrared Imaging Arrays and Systems Volume 47*. Academic Press, 2011.

[7] A. M. Jensen, M. McKee, and Y. Chen, "Procedures for processing thermal images using low-cost microbolometer cameras for small unmanned aerial systems," in *2014 IEEE Geoscience and Remote Sensing Symposium*, 2014, pp. 2629–2632.

[8] "FLIR LEPTON Long Wave Infrared (LWIR)." FLIR, 2014.



KONTES ROBOT INDONESIA 2017

ISBN 978-602-72004-3-2

