



UMY

UNIVERSITAS
MUHAMMADIYAH
YOGYAKARTA
Unggul & Islami

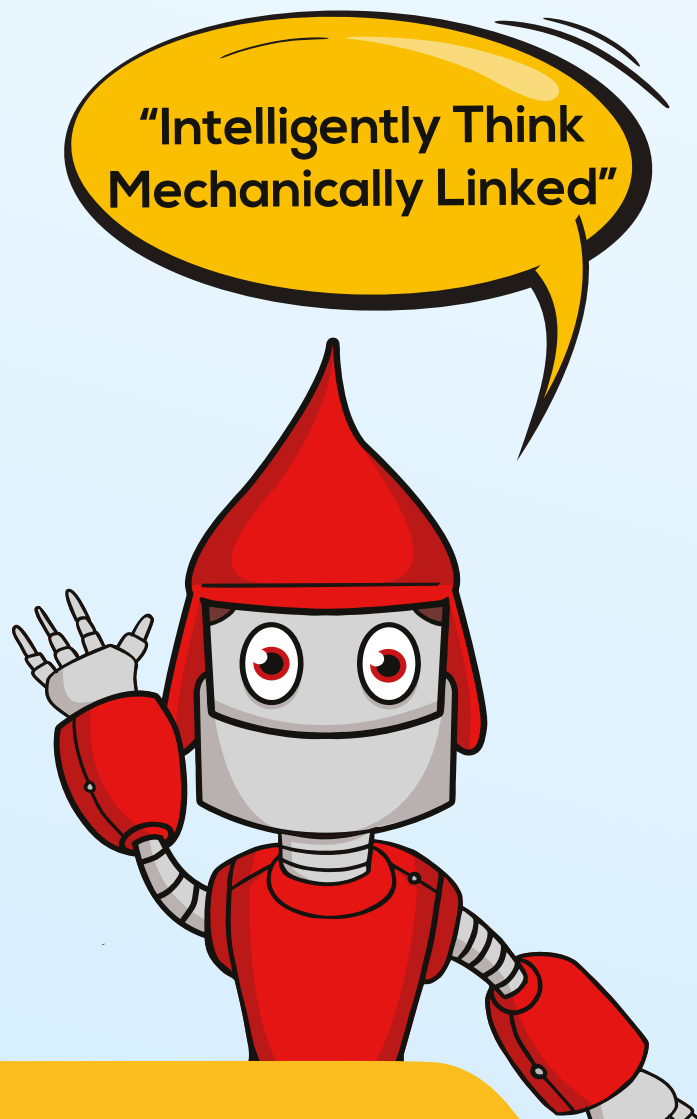


PROCEEDING

The 6th Indonesian Symposium
on Robotic Systems and Control
(ISRSC)

.....

10 JULI 2018 | UNIVERSITAS MUHAMMADIYAH YOGYAKARTA



PROCEEDING

The 6th Indonesian Symposium on Robotic Systems and Control (ISRSC)



ISBN 978-602-5450-43-3



SUSUNAN PANITIA

General Chair

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

Steering Committee

Dr. Ir. Gunawan Budiyo, M.P. (Universitas Muhammadiyah Yogyakarta)

Dr. Kusprasapta Mutijarsa (Institut Teknologi Bandung)

Hilman Latief, Ph.D. (Universitas Muhammadiyah Yogyakarta)

Ir Nafi Ananda Utama, M.S. (Universitas Muhammadiyah Yogyakarta)

Organizing Committee

Sri Atmaja P. Rosyidi, Ph.D., PE. (Universitas Muhammadiyah Yogyakarta)

Eko Purwanti, Ph.D. (Universitas Muhammadiyah Yogyakarta)

Achmad Fachruddin, S.E., M.Si. (Universitas Muhammadiyah Yogyakarta)

Sri Sudarsi, SS., M.Int. (Universitas Muhammadiyah Yogyakarta)

Taufik Akhbar, S.E., MBA. (Universitas Muhammadiyah Yogyakarta)

Triyana, A.Md. (Universitas Muhammadiyah Yogyakarta)

Reviewer

Ir. Heru Santoso BR, M.Eng., Ph.D (Universitas Gadjah Mada) – Chair

Prof. Dr. Benyamin Kusumoputro (Universitas Indonesia)

Dr. Ir. Djoko Purwanto, M.Eng. (Institut Teknologi Sepuluh Nopember)

Dr. Eril Mozef, MS., DEA. (Politeknik Bandung)

Ir. Gigih Prabowo, MT (Politeknik Elektronika Negeri Surabaya)

Dr. Ir. Indrawanto (Institut Teknologi Bandung)

Karisma Trinanda Putra, S. ST. MT. (Universitas Muhammadiyah Yogyakarta)

Prof. Dr. Mauridhi Hery Purnomo (Institut Teknologi Sepuluh Nopember)

Rama Okta Wiyagi, ST, M.Eng. (Universitas Muhammadiyah Yogyakarta)

Dr. Ramadoni Syahputra, ST., MT (Universitas Muhammadiyah Yogyakarta)

Dr. Ir. Wahidin Wahab, M.Sc (Universitas Indonesia)

Editor

Dr. Aris Widyo Nugroho (Universitas Muhammadiyah Yogyakarta)

Dr. Ramadoni Syahputra, ST., MT (Universitas Muhammadiyah Yogyakarta)

Desain Sampul dan Tata Letak

Budi Nugroho, S.IP

Penerbit

Lembaga Penelitian, Publikasi & Pengabdian Masyarakat Universitas Muhammadiyah Yogyakarta

Jl. Brawijaya, Kasihan, Bantul Yogyakarta 55183, Tel.: +62 274 387656 ext. 159, Fax: +62 274 387646

Email: lp3m@umy.ac.id

PRAKATA CHAIRMAN ISRSC

Assalaamu 'alaikum warahmatullahi wabarakaatuh,

Alhamdulillah,

kita bisa berjumpa kembali untuk bersilaturahmi dalam nuansa syiar teknologi robotika di 6th Indonesia Symposium on Robotic Systems and Control (ISRSC) 2018 di Yogyakarta ini. Sebuah kehormatan bagi saya selaku *chairman* simposium 6th ISRSC 2018 untuk menyambut Anda semua di acara yang merupakan rangkaian kegiatan Kontes Robot Indonesia (KRI) Tingkat Nasional 2018 ini.

ISRSC 2018 dapat dilaksanakan dengan baik oleh *host* kali ini, yaitu Universitas Muhammadiyah Yogyakarta berkat kontribusi *keynote speaker*, semua *author* makalah, *reviewer*, panitia pelaksana, dan tentu saja Direktorat Pembelajaran dan Kemahasiswaan Ristekdikti yang telah mendukung dan membiayai acara ini. Pada kesempatan ini saya sampaikan terimakasih yang sebesar-besarnya atas kontribusi Anda semua.

Tidak dapat dipungkiri bahwa secara perlahan, pasti dan berkesinambungan ISRSC 2018 yang tahun ini adalah tahun ke-6 sejak pertamakali diadakan di tahun 2013 telah mendongkrak popularitas kontes robot di Indonesia yang semula lebih sebagai ajang unjuk kebolehan rekacipta robot dalam suatu perlombaan atau pertandingan, menjadi ajang hasil unjuk prestasi riset dan teknologi yang merambah hampir seluruh aspek teknologi kekinian terutama di bidang robotika. Paling tidak hal ini nampak pada publikasi institusi peserta yang konsisten dalam pengembangan robotika dan para alumni ISRSC dan KRI yang dapat diterima dengan sangat baik di dunia industri, bahkan menjadi prioritas dan diperebutkan. Tentu hal ini sangat bernilai tambah dalam mendukung kemajuan iptek dan industri di tanah air, paling tidak ikut menyumbang sumber daya manusia yang handal, khas pejuang kontes robot.

Dengan hadirnya prosiding simposium yang lahir dari ajang Kontes Robot Sepak Bola Indonesia (KRSBI) 2018 ini, dan sekarang lebih luas karena mendukung seluruh isu riset di bidang robotika, sistem dan kontrol, maka kedepan diharapkan ISRSC makin dapat menjadi forum tukar menukar dan menyinergikan hasil-hasil penelitian sekaligus menjembatani komunikasi antar para peneliti keilmuan robotika di Indonesia.

Selamat berkarya,

Wassalaamu alaykum

Yogyakarta, 13 Juli 2018

Dr. Ir. Endra Pitowarno, M.Eng
Chairman ISRSC 2018

SAMBUTAN REKTOR UMY

Assalaamu'alaikum warahmatullahi wabarakaatuh.

Alhamdulillah, tahun 2018 ini, untuk yang kedua kalinya Universitas Muhammadiyah Yogyakarta (UMY) kembali diberikan amanah oleh Kementerian Riset, Teknologi dan Pendidikan Tinggi untuk menjadi tuan rumah dalam penyelenggaraan Kontes Robot Indonesia (KRI) tingkat nasional. Ini merupakan kehormatan bagi UMY untuk berkontribusi dalam pembinaan prestasi kemahasiswaan khususnya pengembangan iptek dalam bidang robotika.

Kontes Robot Indonesia merupakan salah satu wahana riil dalam mendorong eksistensi dan pengembangan teknologi robotika di kalangan mahasiswa. KRI juga memfasilitasi kegiatan temu ilmiah berupa simposium yang diberikan nama **The Indonesian Symposium on Robotic Systems and Control (ISRSC) 2018** bagi kalangan pengiat robotika Indonesia; yaitu para peneliti, praktisi, dan akademisi dalam bidang robotika untuk berbagi gagasan, hasil perancangan dan penelitian. **ISRSC 2018** adalah acara ilmiah tahunan bidang robotika yang diselenggarakan sebagai bagian dari rangkaian kegiatan Kontes Robot Indonesia Tingkat Nasional 2018.

Selain sebagai sarana diseminasi kepada para pegiat robotika, kami berharap prosiding yang lahir dari simposium tersebut juga meningkatkan penyebaran pengetahuan dan teknologi robotika kepada mahasiswa. Dengan demikian, hal itu dapat mendorong optimisme dan semangat generasi muda untuk melakukan percobaan dan penelitian dalam bidang ini.

Atas nama segenap civitas akademika UMY, saya mengucapkan terima kasih kepada Direktorat Kemahasiswaan Kemenristekdikti, Pemakalah dan Peserta Simposium, Reviewer dan Panitia Simposium serta berbagai pihak yang tidak bisa saya sebutkan satu per satu disini, atas kontribusinya dalam mensukseskan **ISRSC KRI 2018**.

Wassalaamu'alaikum warahmatullahi wabarakaatuh.

Yogyakarta, 13 Juli 2018

Dr. Ir. Gunawan Budiyo, MP.
Rektor UMY

KATA PENGANTAR EDITOR

Assalaamu 'alaikum warahmatullahi wabarakaatuh.

Alhamdulillahirobbil 'alamin. Segala puja dan puji hanya milik Allah *Subhanahu wa Ta'ala*, Rabb seluruh alam, tiada yang berhak disembah selain-Nya. Shalawat dan salam atas junjungan penghulu para nabi, Rasulullah Muhammad *shallallahu 'alaihi wa sallam*.

Para pembaca yang budiman, arus globalisasi yang disertai perkembangan teknologi yang semakin canggih, dunia kini memasuki era revolusi industri 4.0. Era ini menekankan pada pola *digital economy*, *artificial intelligence*, *big data*, dan *robotic*, dikenal dengan fenomena "*disruptive innovation*". Pola teknologi digital aras tinggi ini semakin mengukuhkan pentingnya aplikasi robot dalam berbagai bidang. Dalam rangka merespon fenomena ini, Pemerintah Indonesia melalui Kementerian Riset, Teknologi dan Pendidikan Tinggi (Kemenristekdikti) telah melakukan sosialisasi hingga kompetisi berupa kontes robot guna semakin meningkatkan penguasaan teknologi bidang robotika khususnya bagi dunia akademisi dan industri.

The Indonesian Symposium on Robotic Systems and Control (ISRSC) 2018 adalah acara temu ilmiah bidang robotika yang diselenggarakan sebagai bagian dari rangkaian kegiatan Kontes Robot Indonesia Tingkat Nasional. Simposium ISRSC menghadirkan penggiat, peneliti, praktisi, dan akademisi sebagai ajang bertukar informasi mengenai ilmu pengetahuan dan teknologi (Iptek) mutakhir di bidang robotika. Iptek mutakhir bidang robotika dalam bentuk karya ilmiah telah terhimpun dalam Prosiding ISRSC 2018 ini. Tim Editor Prosiding ISRSC 2018 merasa terhormat mendapat kesempatan menghimpun karya-karya anak bangsa yang sangat membanggakan. Dalam prosiding ini ditampilkan sebanyak 47 artikel ilmiah hasil penelitian dan metode-metode dalam menyelesaikan berbagai masalah serta tantangan pada bidang robotika. Prosiding ini dapat digunakan sebagai referensi penting khususnya bagi komunitas akademik, peneliti dan industri di bidang robotika. Namun demikian prosiding ini juga dapat dinikmati masyarakat umum sebagai wahana memperkaya pengetahuan dunia robotika dan perkembangannya.

Tim Editor menyampaikan terima kasih kepada Kemenristekdikti, UMY, para penulis artikel ilmiah, *reviewer*, panitia, dan seluruh pihak yang terlibat langsung maupun tidak langsung dalam Simposium ISRSC 2018. Tim Editor juga menghaturkan permohonan maaf atas segala kekurangan. Semoga karya ini memberikan kontribusi bagi pembangunan serta meningkatkan daya saing Indonesia. *Amiin*.

Wassalaamu 'alaikum warahmatullahi wabarakaatuh.

Yogyakarta, 13 Juli 2018

Tim Editor

DAFTAR ISI

SUSUNAN PANITIA	i
PRAKATA CHAIRMAN ISRSC	ii
SAMBUTAN REKTOR	iii
KATA PENGANTAR EDITOR	iv
DAFTAR ISI	v
Pengembangan Robot Sepak Bola Beroda Versi 2 Dagozilla untuk Kompetisi <i>Middle Size League (MSL)</i>	1
Development of Middle-Sized Mobile Robot FUKURO for Robot Soccer Competition	6
Penentuan Posisi Robot Menggunakan <i>Odometry Omniwheel</i>	11
Penerapan Metode <i>Color Filtering</i> HSV untuk Pendeteksian Bola pada Robot KRSBI Beroda	14
Read TCP/IP Menggunakan <i>platform</i> Matlab pada Robot Sepak Bola Beroda	19
Implementasi Jaringan Saraf Tiruan untuk Meningkatkan Akurasi Sensor Kompas HMC5983 pada Robot KRSBI Beroda	25
Sistem Kerja Robot Penjaga Gawang (<i>Keeper</i>) Divisi KRSBI Beroda Tim ROTA	30
Perancangan Sistem Gerak pada Robot Kiper R2C-WARRIOR	34
Sinkronisasi Pergerakan Robot Sepak Bola Beroda Menggunakan Metode PID dan <i>Odometry</i>	38
Implementasi Pergerakan <i>Omnidirectional</i> pada <i>Three-Omniwheeled Robot</i>	43
Sistem Pergerakan Robot pada Robot Bola Beroda Tim CHAKRAMAKARA UI	47
Mekanisme Penendang Menggunakan Solenoid pada Robot Sepak Bola Beroda	50
Sistem Deteksi Objek dan Posisi pada Robot Sepak Bola Beroda <i>Middle Size</i> pada Sistem Kamera <i>Omni Vision</i> dengan <i>Scan Lines</i>	54
Teknik Pengambilan Warna untuk Deteksi Objek Berbasis OpenCV	59
Lokalisasi Robot Sepak Bola terhadap Lapangan	62
Optimasi Deteksi Bola pada Robot Sepak Bola Beroda Menggunakan <i>Region-based Segmentation</i>	66
Penggunaan Kamera Global untuk Menentukan Koordinat Robot pada Lapangan	72
Implementasi Protokol TCP dan UDP dalam Sistem Komunikasi <i>Base Station</i> Robot Sepak Bola Beroda	76
Desain dan Implementasi Komunikasi <i>Control</i> Robot <i>Soccer</i> Beroda Menggunakan User Datagram Protocol (UDP)	82
Sistem Kendali Motor DC Menggunakan PID dan Komunikasi I2C Pada <i>Omni Direction</i> untuk Robot <i>Soccer</i> Ten-De (10-D)	89
ROBOSOC UM 2018: Deskripsi Tim Robot KRSBI Beroda Universitas Negeri Malang	94
Rancang Bangun Mekanik Ekspansi Robot <i>Goalkeeper</i> Sepak Bola Beroda Menggunakan Pneumatik	99
<i>Tracking</i> Warna Menggunakan Ruang Warna HSV pada Robot Scada Rt	104
Strategi Pergerakan Robot Sepak Bola Beroda Menggunakan Komunikasi TCP/IP	108
Metode Pendeteksi Bola dan Gawang pada Robot Sepak Bola Beroda	114

Penentuan Posisi Bola berdasarkan Sudut Arah Bola terhadap Robot pada Kamera <i>Omnidirectional</i>	117
Rancang Bangun Robot Sepak Bola Beroda	121
<i>Forward dan Inverse Kinematics</i> pada Robot Sepak Bola Beroda (<i>Robot Soccer Middle Size</i>)	126
Pendeteksian Bola Menggunakan Metode <i>YUV Color Filtering</i> pada Robot EWS Barracuda	130
Implementasi <i>Omnidirectional Vision Camera</i> Menggunakan Cermin Cembung	134
Centralized System PDI Controller for Balancing of Alfarobi Humanoid Soccer	140
Implementation of Stepping Strategy for Balance Recovery in EROS Humanoid Robot	144
Pola Gerak Dinamis pada Robot Krakatau FC	148
Sistem Penendang Menggunakan Pegas pada Robot Sepak Bola	152
Implementasi <i>Hough Circle, Contours and Convex Hull Method</i> untuk <i>Humanoid Robot Soccer</i> dalam Deteksi Bola Putih Berdasarkan Rule KRSBI Humanoid 2018	157
Prediksi Arah Datang Bola Menggunakan Triangulasi pada Robot Penjaga Gawang	162
Penerapan <i>Google's Speech API</i> sebagai Antarmuka Interaksi Robot dan Manusia	167
Dago Hoogeschool dalam Kontes Robot Sepak Bola Indonesia Humanoid 2018	171
Rancang Robot Humanoid Tim Devata 2018	176
Implementation of De Casteljau's Algorithm for Pattern Generation on BIOLOID Based Humanoid Robot's Leg Movement	181
GPU-Accelerated Monte Carlo Localization for Mobile Robot Soccer with Omnidirectional Camera	185
Segmentasi <i>Circle Hough Transform</i> untuk Deteksi Bola pada Robot Sepak Bola Beroda Menggunakan Kamera ov7670	189
Pengembangan Aplikasi Pendeteksi Objek Lingkaran Menggunakan Metode <i>Houghcircle</i> sebagai Media Pembelajaran Penginderaan Visual Robot	193
Penerapan <i>Web Base</i> sebagai Kontroller dan <i>NodeMCU 1.0 (Esp-12E)</i> sebagai Media Komunikasi pada Kendali Robot Soccer	202
Pemanfaatan Sensor Jarak untuk Mengidentifikasi Ruangan pada Map KRPAI (Studi Kasus: Robot KRPAI 2018 STIKOM DB)	208
Implementasi Interpolasi Linier pada Gerakan Tari Robot Humanoid	213
Rancang Bangun Sistem <i>Omnidirectional Vision</i> Cermin Spherical pada Robot KRSBI Beroda	219

Pengembangan Robot Sepak Bola Beroda Versi 2 Dagozilla untuk Kompetisi *Middle Size League* (MSL)

Azka Zakiyyatuddin, Abdullah Faqih Al Mubarak, Ahmad Wahrudin*, Annisa Zulfa Hidayah, Aulia Salsyabil, Aulia Widyaputra, Azis Adi Kuncoro, Dionesius Agung Andika Perkasa, Edgar Sutawika, Evan Febrianto, Faza Fahleraz, Irvan Maulana, Jeffry, Josephine Ariella, Muhammad Arbi Minanda, Muhammad Hilmi Asyrofi, Muhammad Kamal Nadjieb, Najmah Syahidah Al Ausath., Vidya Putri Priandiri, Yudhistira Yoga Semeru

Tim Dagozilla ITB, Unit Robotika ITB
Autonomous Vehicle Reseach Group Laboratory
Institut Teknologi Bandung
Bandung, Indonesia
krsbiberodaitb@gmail.com

Abstrak – Kontes Robot Indonesia (KRI) merupakan kompetisi robot nasional yang diselenggarakan setiap tahun guna mewadahi pengembangan keterampilan mahasiswa dalam bidang robotika. KRSBI (Kontes Robot Sepak Bola Indonesia) Beroda merupakan salah satu cabang yang diperlombakan pada KRI. Peraturan KRSBI Beroda mengacu pada kompetisi internasionalnya yaitu *RoboCup MSL (Middle Size League)* dengan beberapa penyesuaian sesuai kondisi di Indonesia. Robot dengan kemampuan lokomosi, menendang, menggiring, serta memainkan permainan secara otomatis dengan baik akan memberikan penampilan menariknya dalam pertandingan. Dalam paper ini dijelaskan mengenai pengembangan *platform* robot sepak bola beroda dari Tim Dagozilla ITB dalam satu tahun terakhir. Inovasi terutama dilakukan pada pembuatan *platform* robot versi baru (V.2) dan penggunaan (*Robot Operating System*) ROS.

Kata kunci: RoboCup MSL, KRSBI Beroda, Platform, ROS

I. PENDAHULUAN

Dagozilla merupakan Tim KRSBI Beroda dari Institut Teknologi Bandung. Tim ini terdiri dari mahasiswa S1 dengan berbagai macam latar belakang jurusan yang berbeda-beda yang saling berkontribusi untuk membangun *platform* robot sepak bola beroda yang dapat dipertandingkan pada cabang perlombaan *Middle Size League (MSL)*. *MSL* merupakan cabang perlombaan internasional dalam *RoboCup* yang mempertandingkan perlombaan sepak bola robot beroda antara dua tim yang berbeda [1].

KRSBI Beroda merupakan salah satu kompetisi di tingkat nasional yang dapat dijadikan ajang pengetesan dan pengujian robot yang telah dibangun. Pada tahun 2018, tim Dagozilla ITB mengembangkan robot versi terbaru yaitu versi 2 dari pengembangan tahun 2017 dengan pokok utama pada penggunaan tiga *omniwheel* dan *single*

vision serta penggunaan ROS sebagai *platform* utama dalam pengembangan *software*. Untuk memberikan performa yang baik, robot sepak bola beroda harus mempunyai kemampuan dasar berupa kemampuan lokomosi, menggiring, dan menendang bola yang baik serta dapat beroperasi secara otomatis. Paper ini berisi mengenai pengembangan versi 2 robot sepakbola beroda tim Dagozilla.

II. PLATFORM ROBOT V.2

Fitur utama yang dikembangkan dalam membangun robot versi 2 dititikberatkan pada sistem lokomosi, sistem penggiring, sistem penendang, sistem citra robot serta sistem perancangan *software*.



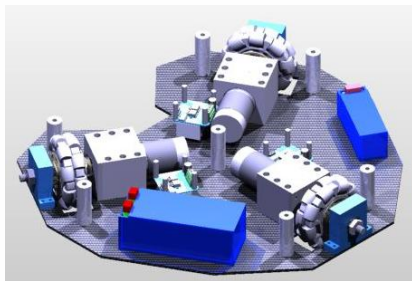
Gambar 1. Platform robot V.2

Pada bagian lokomosi dan penggiring, pengembangan terutama dilakukan pada motor dan sistem penunjang motor (rancang bangun). Sedangkan untuk sistem penendang pengembangan dilakukan terhadap peningkatan kemampuan solenoid untuk menendang bola termasuk dalam kontrol kecepatan tendangan. Pada sistem citra robot dikembangkan metode *omnivision* untuk mendapatkan gambar 360° dengan menggunakan

cermin katadioptrik sehingga dapat mempermudah dalam proses deteksi dan pemrosesan. Sistem vision dan penentuan keputusan strategi dilakukan dengan menggunakan *Robot Operating System (ROS)*.

A. Sistem Lokomosi

Pada pengembangan robot versi 2 tim Dagozilla menggunakan platform *omnidirectional* 3 roda *omniwheel* dan motor PG45 dengan *encoder* 7ppr sebagai penggerak utama sistem lokomosi. Sistem lokomosi ini ditunjang menggunakan material alumunium dural yang mempunyai sifat ringan tetapi kuat, sehingga diharapkan dapat tahan terhadap benturan dan masih memenuhi spesifikasi berat robot. Desain lokomosi diberikan oleh gambar berikut.



Gambar 2. Desain lokomosi robot

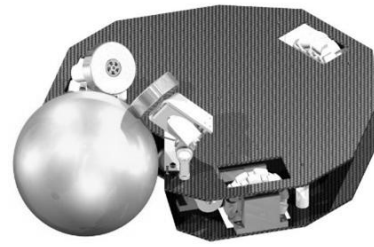
Sama seperti pengembangan robot versi sebelumnya, robot versi 2 ini juga menggunakan metode kontrol PID pada kecepatannya. PID digunakan karena metode ini mempunyai keunggulan sangat aplikatif yang paling sering digunakan dalam kendali industri maupun otomasi [2]. Nilai konstanta PID motor lokomosi yang diperoleh menggunakan MATLAB®. Nilai konstanta pada Tabel 1 didapatkan dengan waktu *sampling* 0.02 s.

Tabel 1. Nilai konstanta PID motor lokomosi

Motor	Kp	Ki	Kd
1	0.29743024	9.0211605	0
2	0.30734395	9.0330844	0
3	0.31027798	8.54558353	0

B. Sistem Penggiring

Sistem penggiring dirancang supaya robot dapat menggiring bola sehingga bola dapat berputar sesuai dengan arah gerak alami bola. Sistem penggiring ditopang dengan material alumunium dan hard nylon. Sistem penggiring terdiri dari dua motor PG36 dengan *encoder* 7ppr yang dihubungkan dengan roda berdiameter 50mm sebagai komponen aktif serta 2 *omniwheel* mini 50mm sebagai komponen pasif.



Gambar 3. Desain sistem penggiring robot

Pada versi 2 ini, kami menambahkan metode kontrol PID untuk mengendalikan kecepatan motor penggiring. Dengan menggunakan metode *tunning* PID seperti sebelumnya didapat nilai konstanta seperti ditunjukkan oleh tabel berikut.

Tabel 2. Nilai konstanta PID motor penggiring

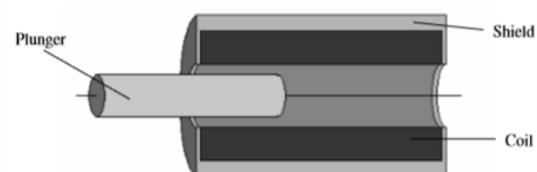
Motor	Kp	Ki	Kd
Kanan	0.2040256	7.9394200	0
Kiri	0.30374	7.4118	0

C. Sistem Penendang

Sistem penendang yang kami gunakan pada versi saat ini menggunakan solenoid yang merupakan pengembangan pada versi sebelumnya. Pada tabel 3 dijelaskan mengenai spesifikasi solenoid yang akan dibuat sebagai penghasil gaya pada sistem penendang.

Tabel 3. Spesifikasi solenoida

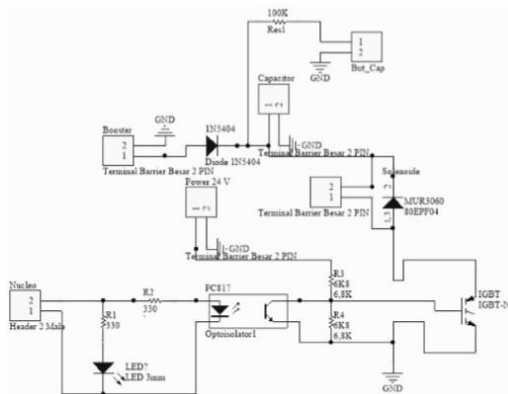
Coil	
Length	10.5 cm
Inner radius	1.25 cm
Outer radius	4.6 cm
Wire size	16 AWG
Turns	1800
Layers	24
Shield	
Position	Top, bottom, left, right
Width	0.4 mm
Space	0.5 mm
Plunger	
Length	16 cm
Radius	1.2 cm



Gambar 4. Skematik solenoid penendang

Shield pada sistem ini digunakan untuk melindungi komponen elektrik di sekitar penendang dari paparan *Electromagnetic Interference (EMI)*. Arus listrik yang mengalir pada kumparan dalam solenoid akan menimbulkan medan elektromagnetik, medan ini akan menarik plunger dan menghasilkan gaya yang cukup untuk melempar bola. Efek dari timbul gaya magnet ini juga menyebabkan terbentuknya EMI [3].

Modul kontroler pada sistem penenang bertugas dalam mengendalikan pengisian energi ke dalam kapasitor dan mengatur seberapa besar energi yang akan dikeluarkan ke solenoida. Desain modul kontroller ini dilakukan menggunakan Altium Deigner. Untuk mengatur banyaknya energi yang dilepaskan ke solenoid digunakan sinyal PWM. Skematik modul kontroler ini diberikan oleh gambar 5.



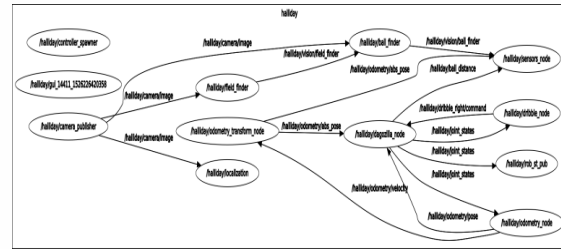
Gambar 5. Skematik modul kontrol penendang

D. Sistem Citra Robot

Pengembangan robot versi 2 menggunakan *single vision sensor* pada robot penyerang. Sensor vision ini diperoleh dari cermin katadioptrik yang dapat memberikan gambar 360°. Sedangkan gambar yang dipantulkan oleh cermin ini nantinya akan ditangkap oleh kamera webcam Logitech 922 pro kemudian diolah dalam *processor* pengolah berupa *personal computer (PC)* / laptop. Desain cermin yang digunakan berupa cermin parabolik yang dapat memberikan gambar 360° dan memiliki cakupan daerah yang cukup luas [4].

E. Arsitektur Software

Pada robot versi 2 ini, kami menggunakan *Robot Operating System (ROS)* untuk mengolah komputasi *high-level* [5]. Sedangkan untuk komputasi *low-level* dilakukan oleh mikrokontroller stm32 F446Re Nucleo. Model *publish-subscribe* kami gunakan dalam pengembangan robot ini. Kita dapat memvisualisasikan *node* dan *topic* menggunakan *rqt graph*. Gambar 6 menunjukkan visualisasi *rqt graph* yang digunakan.

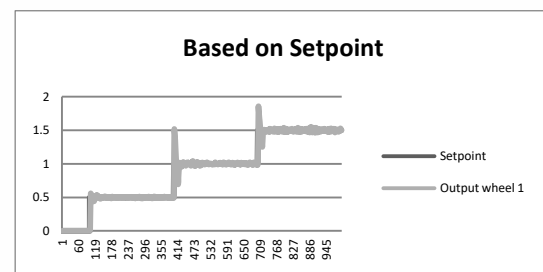


Gambar 6. *Rqt graph* memvisualisasikan hubungan antara *topic* dan *node* yang berjalan pada sebuah robot bernama Halliday

III. HASIL DAN PEMBAHASAN

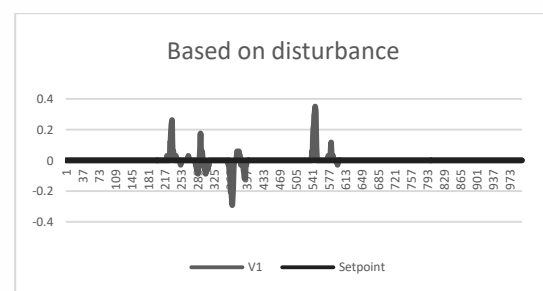
A. Sistem Lokomosi

Pengujian terhadap sistem lokomosi dibagi menjadi dua, yaitu pengujian implementasi PID dan pengujian gerakan dasar.



Gambar 7. *Setpoint dan output kecepatan motor DC*

Data PID merepresentasikan hubungan antara *setpoint* yang diberikan dan kecepatan *actual* dari setiap motor pada robot (Gambar 7). Grafik tersebut juga menunjukkan bahwa robot dapat menyesuaikan terhadap perubahan *setpoint* dengan cukup cepat.

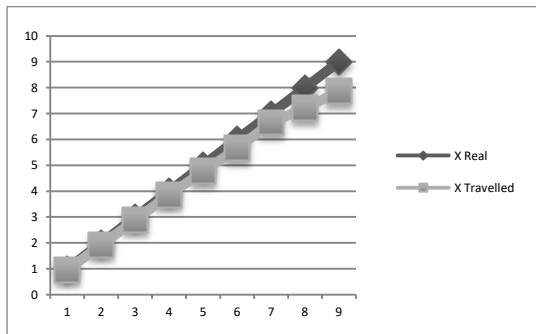


Gambar 8. Keluaran motor DC terhadap gangguan / disturbance

Pada Gambar 8, ketiga motor pada robot diberikan *setpoint* 0 m/s. Ketika mendapatkan gangguan, robot akan mempertahankan kecepatannya tetap pada 0 m/s dengan besar error maksimum 0.35062 m/s.

Dari pengujian gerakan dasar dihasilkan grafik seperti pada gambar 9. Dari grafik tersebut dapat dilihat bahwa hasil error yang diberikan oleh

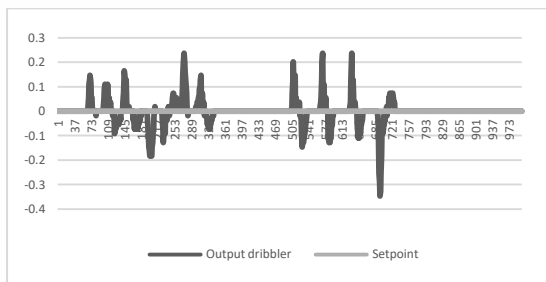
pembacaan odometri bersifat linear. Error ini terjadi karena faktor dinamik robot yang memang belum dimasukkan kedalam komputasi odometri robot.



Gambar 9. Posisi sumbu x pada robot (real vs odometri)

B. Sistem Penggiring

Pengujian sistem penggiring dilakukan dengan mengamati respon PID motor terhadap setpoint dan *disturbance* yang diberikan. Hasil maksimum overshoot yang diperoleh berada pada nilai 24,77% dengan error maksimum 0.34697 m/s (Gambar 10).



Gambar 10. Setpoint dan keluaran kecepatan motor DC terhadap gangguan

C. Sistem Penendang

Sudut elevasi maksimal yang dapat dihasilkan dari platform saat ini sebesar 26,391°. Pengujian yang dilakukan pada sistem penendang digunakan pendekatan persamaan gerak parabolik dan persamaan kinematika, dengan asumsi massa bola, gaya gesek, dan segala sesuatu yang berhubungan dengan dinamika robot-bola diabaikan. Hasil pengujian tersebut diberikan pada Tabel 4.

Tabel 4. Hubungan PWM dengan jarak lambung, waktu jatuh, dan tinggi maksimum bola lambung

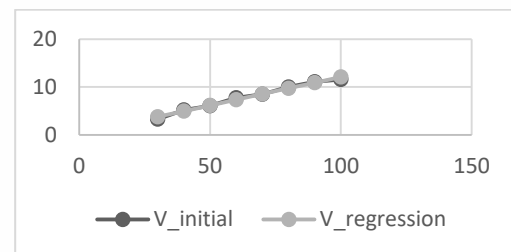
PWM	Xmax (m)	Traveled time (s)	H_max (m)
100	6.85	0.66	1.352293837
90	6.44	0.648	1.239936715
80	5.68	0.634	1.007618082
70	4.82	0.634	0.725593456
60	3.92	0.564	0.606445423

50	2.74	0.502	0.373999902
40	1.74	0.372	0.274656836
30	0.74	0.25	0.10999203

Dari hasil perhitungan didapatkan hubungan antara PWM dengan kecepatan awal bola adalah sebagai berikut.

Tabel 5. Hubungan PWM dengan kecepatan awal tendangan

PWM	V_initial (m/s)	V_regression (m/s)
100	11.58629287	12.10134824
90	11.09452537	10.91336714
80	10.00131127	9.72538603
70	8.487028227	8.537404923
60	7.758983517	7.349423816
50	6.093189877	6.161442709
40	5.221606913	4.973461602
30	3.304376899	3.785480495



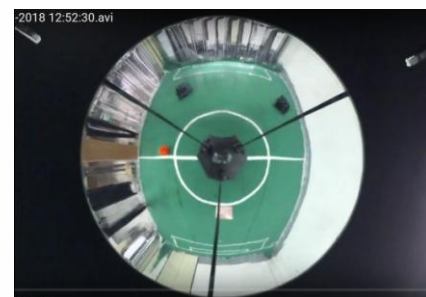
Gambar 10. Grafik hubungan antara kecepatan awal dengan PWM

Dari hasil regresi disimpulkan bahwa hubungan PWM dengan kecepatan awal adalah linear, sehingga kita dapat memperkirakan nilai kecepatan awal yang diinginkan dengan persamaan:

$$\text{Initial velocity} = 0.1188 \times \text{PWM} + 0.22$$

D. Sistem Citra Robot

Dari hasil pengolahan citra, robot dapat mendeteksi bola maskimal sejauh 3m dengan menggunakan cermin katadioptrik.



Gambar 11. Citra cermin katadioptrik v.1 yang diperoleh

E. Robot Operating System (ROS)

Tiga manfaat utama penggunaan ROS dalam pengembangan robot MSL adalah *data sharing*, abstraksi *hardware*, konkurensi program. Manfaat ini memberikan keuntungan bagi tim untuk dapat membangun *software* secara modular sehingga lebih mudah untuk dikembangkan lebih lanjut.

IV. KESIMPULAN

Paper ini berisikan tentang pengembangan robot Dagozilla untuk MSL generasi ke-dua. Robot ini dikembangkan pada periode 2017-2018. Pokok utama pengembangan robot ini adalah perombakan dan peningkatan fungsi *hardware* serta *software* sehingga menjadi lebih *robust* dan *maintainable*. Perombakan yang dimaksud berupa peningkatan dalam sistem lokomosi, penggiring, penendang, sistem citra, dan penggunaan ROS pada robot.

V. DAFTAR PUSTAKA

- [1] Robocup. *Middle Size Robot League Rules and Regulation*. Robocup, 2016
- [2] Ang, K. H., Chong, G., & Li, Y. (2005). PID control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*, 13(4), 559–576.
- [3] Goch, B. P. T. van. (2006). Optimizing a solenoid for a Robocup kicker. *Technology*, 7–32
- [4] Lopez, G., Ribeiro, F. “Catadioptric system optimisation for omnidirectional Robocup MSL robots”. In *RoboCup 2011: Robot Soccer World Cup XV*, 2012.
- [5] Garzón M. et al. (2017) Using ROS in Multi-robot Systems: Experiences and Lessons Learned from Real-World Field Tests. In: Koubaa A. (eds) *Robot Operating System (ROS)*. *Studies in Computational Intelligence*, vol 707. Springer, Cham

Development of Middle-Sized Mobile Robot FUKURO for Robot Soccer Competition

Igi Ardiyanto, Rudy Tri Hartanto, Fahri Ali Rahman, Yosef Adhitya Duta D, Okta Fajar Suryani, Yahya Abdurrahman, Resha Dwika Hefni Al-Fahsi, Fali F Fauzan, Ridwan Afwan K.F, Denanda Hendra, Tsanil Ma'arif, , Aleksander Patar J.P, Irziq Misbahul Arif, Dimas Arya Dewangga, Muhammad Fauzan A, Muhammad Fikri Al F, Annisa Soliha Hidayat

Team FUKURO, Gadjah Mada Robotic Team
Universitas Gadjah Mada
Yogyakarta, Indonesia

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 a 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 project consists of an undergraduate student from several fields contributing -to 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 that consists five base tiers. At the first base, this base is filled by the system of the robot drive, wherein the robot uses

three Omni wheels these are connected by the planetary gear motor with a torque of 20 kgf.cm. There is also dribble system on base 1 that **make a difference from the first generation of Fukuro**. The dribbler system function is to dribble the ball so that robot can keep the ball to stay attached to the robot in a state of motion. On the base two, there is kicker system. The kicker system is a combination of mechanic and an electric component that can give function to kick the ball. Most of the material used to make the robot framework is aluminum.

This robot uses a laptop as a hardware controller that is placed in the base three. For the visual component, Fukuro using a convex mirror placed on the camera with a proportional range so that the robot has 360° visions. Vision component is placed on base four and convex mirror place at base five. Dimension of this robot have length of 50cm x 50cm x 78cm. And the weight of this robot is 20kg. The mechanical design of Fukuro shown in Figure 1.

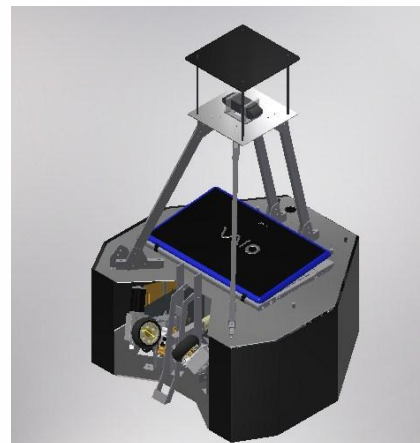


Figure 1 . Mechanical Design

A. Drive Mechanism

The first base such as Figure-2 consists of Planetary Gear 45 (PG-45) motors, omni wheel, lithium polymer (Li-Po) batteries, Rotary encoder, and dribble system. 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.



Figure 2. The lower base

B. The Kicker System

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 solenoid as the main actuator. **Different from previous year**, by utilizing the solenoid system and combine it with special kicking mechanism, we can produce a force that can be used to kick the ball.

C. Dribble system

For dribble system, we use motor planetary gear 36 (PG36) as main actuators and coupled with 6cm-diameter wheels. The support of the dribble contain damper spring to make the dribble system able to adjust the incoming ball. 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.

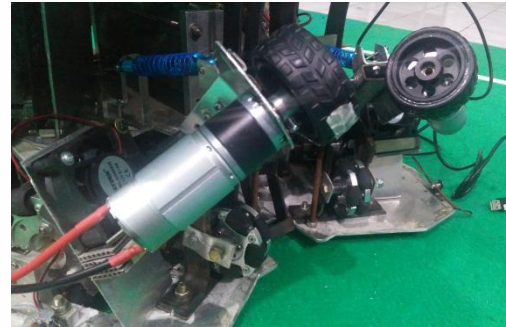


Figure 3. Dribble System

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 main different from previous year is on motor driver used.** This year we move from EMS to driver motor module with IC BTN 7970 as in Figure-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. MOSFET IRF540 has been used for regulating and supplying 12V to the motor PG-36. UBEC is used to regulating and supplying 5V to the rest of the circuit.

D. Motor Controller

Fukuro driver motor module with IC BTN 7970 is used to control the DC motors in our robots. The module is designed to operate in 2 driving direction with 3x H-Bridge Driver Motor, 2x Half Bridge Driver Motor, Maximum Voltages 45 V, Maximum Current 20A. The driver motor used to control both the wheel and dribbler motors.

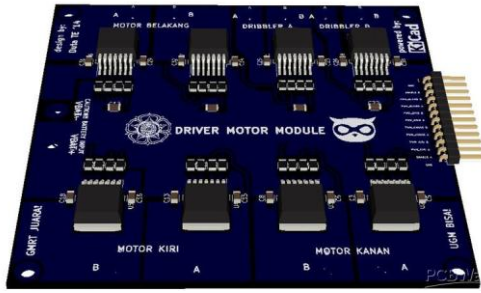


Figure-5. FUKURO Driver Motor Module.

E. Solenoid Kicker

The solenoid kicker consists of 1000 turn hand-wound solenoid mounted onto the base of the robot. It is powered by a 350V 5 mF capacitor, which is charged by a step-up converter at 350 V. Optocoupler with IGBT relays controlled by the STM32F446-Nucleo act as switches to control the charging and discharging of the capacitor. A kick is actuated by a 15ms pulse of current at 350 V from the capacitor to the solenoid.

IV. SOFTWARE ARCHITECTURE

This year we have implemented communication for our robot. **This is the main different from previous year in our software architecture.** The software system such as Figure-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.

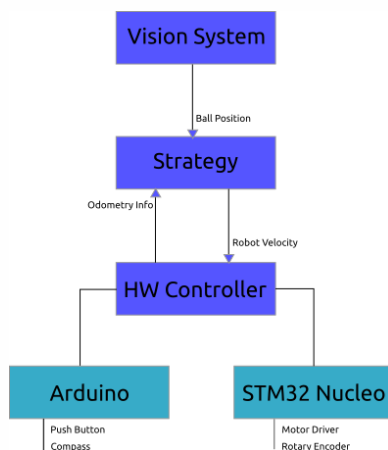


Figure-6. FUKURO Software Architecture Design.

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 shown in Figure-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.

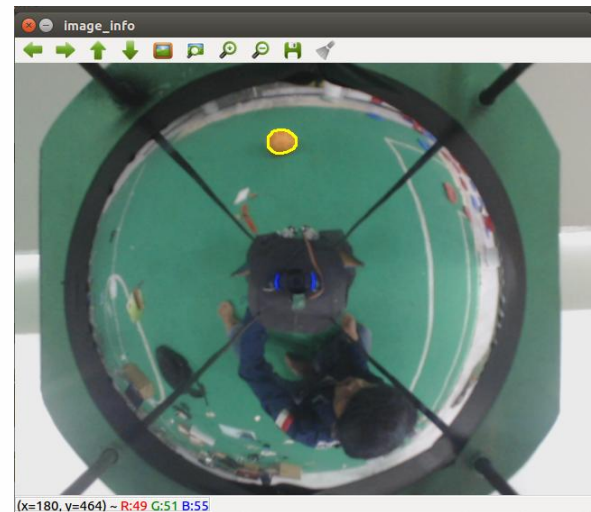


Figure-7. Ball detection result.

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 Figure-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.

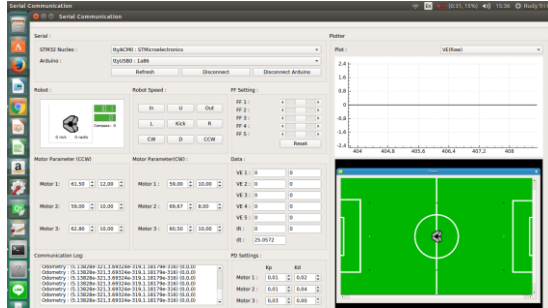


Figure-8. HW Controller Graphical User Interface.

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. Figure-9. illustrate the computation of v_x and v_y . The velocities are computed as follows:

$$v_x = v_{\text{linear}} \cos \theta \quad (1)$$

$$v_y = v_{\text{linear}} \sin \theta \quad (2)$$

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.

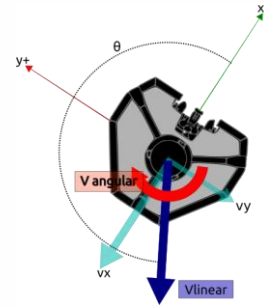


Figure-9. Velocity vector and decomposition.

F. FUKURO Communication

FUKURO Communication system is consist of Basestation and fukuro_communication_node. The Basestation connect information from RefereeBox to robot and allow information sharing among the robots. Every information such as robot position, ball position, obstacle position are shared to every robot realtime. In the FUKURO communication system these information are stored in FukuroDB, a database system that implement hashing algorithm developed by FUKURO team. The fukuro_communication_node handle the information that receive from Basestation and can be processed to determine the strategy of the robot.



Figure-10. Fukuro Communication GUI

V. CONCLUSION

Until now, our software architecture and hardware configuration work finely. In our software which based ROS Platform the major improvements are can implemented deep neural network and communication between robots and base station. While in our hardware system the major improvement are used double layer pcb and solenoid kicker. These improvement show that Fukuro always want to implement modern technology on it's robots for this year and future.

VI. REFERENCE

- [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.

Penentuan Posisi Robot Menggunakan *Odometry Omniwheel*

Iwan Kurnianto Wibowo, Mochamad Mobed Bachtiar, Bima Sena Bayu D, M. Labiyb Afakh, Renardi Adryantoro P, Muhammad Abdul Haq, Zanuar Tri R, Satria Rachmad S, Yuda Maulana, Gidion Tinto P, Yulian S.P, Agung P, Gani Nur F, Agung Tri P, Rizal Wahyu P, M. Zaenal A, M. Alan N, M. Jauharul M, Amre S, Fendi Nur W, M

Irfan Mas'udi, Alviansyah Arman Y

Politeknik Elektronika Negeri Surabaya

Jl. Raya ITS 60111 Sukolilo Surabaya

Email : {eone¹, mobed², bima³}@pens.ac.id, {renardi98⁵, muhabdulhaq⁶}@gmail.com

Abstrak -- Robot *soccer* beroda adalah salah satu robot yang berfungsi untuk melakukan permainan sepakbola dengan aturan-aturan tertentu seperti permainan sepakbola yang dilakukan manusia dengan sistem pergerakan menggunakan roda. Direncanakan pada tahun 2050, manusia akan bermain sepakbola dengan robot. Penelitian dalam bidang robot *soccer* beroda banyak dilakukan pada kecerdasan robot seperti bagaimana robot mendeteksi dan mencari bola, menggiring bola, menendang bola ke arah gawang lawan, menghindari lawan dan komunikasi robot dalam tim. Pada penelitian ini, difokuskan pada bagaimana robot berjalan. Robot *soccer* ini dibuat mirip seperti *X-Drive* dan *H-drive* di mana mereka semua memiliki tiga derajat kebebasan, yang berarti bahwa mereka dapat bergerak ke segala arah dalam bidang x-y, serta memutar terhadap pusatnya. Robot ini hanya menggunakan tiga motor dan tiga *omni-wheel*. Cara beroperasinya dengan cara yang sama dengan *X-drive*, karena arah gerakan robot akan berada pada arah yang sama dengan vektor gaya resultan net. Keuntungan dengan model seperti ini robot diharapkan dapat bergerak ke segala arah, robot hanya membutuhkan tiga roda omni, selain itu hanya membutuhkan tiga motor, dibandingkan dengan empat motor dari *X-Drive*. Pergerakan robot diatur dengan *odometry*. Presentase *error* dari tiga *omni-wheel* ini sebesar 5cm dengan percobaan sebanyak 5x untuk target tujuan yang sama.

Kata kunci: Robot Soccer Beroda, Penentuan Posisi, *Odometry*, *Omniwheel*

I. PENDAHULUAN

Sepakbola antara robot dengan tim juara dunia sepak bola pada tahun 2050 adalah cita-cita dari para pengembang robot *soccer*. Para peneliti robot *soccer* terus melakukan berbagai pengembangan riset. Ada dua kategori dalam robot *soccer* yaitu *humanoid soccer* dan *mobile soccer* (*soccer* beroda). Kategori *humanoid* mempunyai bentuk seperti manusia sedangkan *mobile soccer* mempunyai sistem penggerak berupa roda. Di Indonesia telah dimulai penelitian ini yang dipelopori oleh diselenggarakannya Kontes Robot Sepak Bola Indonesia (KRSBI) dimana dilombakan

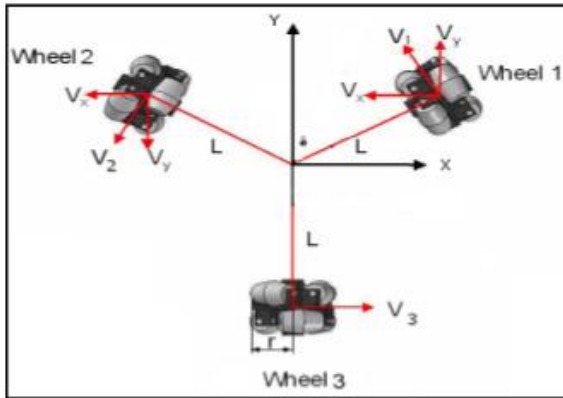
robot *soccer middle size* beroda (mulai tahun 2016) dan *humanoid soccer* (mulai tahun 2011). Penelitian dalam robot *soccer* berkonsentrasi pada kecerdasan robot untuk mengenali bola, penentuan posisi robot, penentuan arah gawang lawan, mengenali lawan, dan komunikasi robot. Semua fungsi tersebut berjalan secara otomatis berdasarkan kecerdasan di dalam robot [1][2]. Robot *soccer middle size* beroda juga menggunakan *computer vision* yang digunakan untuk mencari bola [1] dan untuk melakukan tembakan ke arah gawang digunakan *kicker* dengan menggunakan solenoid [2].

Dalam penelitian ini, difokuskan pada bagaimana robot berjalan di lapangan hanya dengan menggunakan 3 roda. Kelincahan robot sangat tergantung dari mekanik gerak robot itu sendiri. Dalam berbagai penelitian robot *soccer* ada yang mempunyai penggerak roda sebanyak 3 ada juga yang berjumlah 4. Roda-roda tersebut berupa roda omni. Robot *Soccer* dengan 4 roda omni lebih seimbang dalam bergerak ke segala arah. Sedangkan robot *soccer* yang menggunakan 3 roda lebih bebas dalam bergerak, dan lebih *simple* secara mekanik hanya membutuhkan 3 motor dan 3 roda. Diharapkan dengan penggunaan roda pada robot sejumlah 3, robot bisa bergerak lebih cepat dan fleksibel.

II. METODE

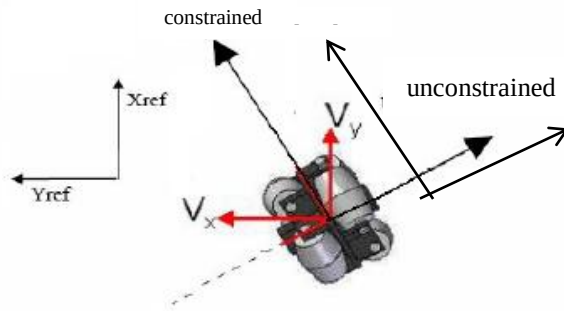
A. Kinematika Omni Wheels

Untuk memahami dasar dari *omni-directional robot* yang dikembangkan, kita dapat meneliti sistem kinematika dari robot tersebut [3]. Jika kita menginginkan pergerakan atau perpindahan robot terhadap posisi awalnya, kita harus mengetahui variabel-variabel yang berkaitan satu dengan yang lainnya yang dapat kita kontrol seperti posisi gerak dari robot dan kecepatan pada masing-masing roda. Perhitungan kinematik pada robot digunakan untuk menentukan perubahan posisi antar koordinat global dan posisi koordinat internal dari robot itu sendiri [4].



Gambar 1. Representasi kinematik dari sistem penggerak tiga omni-directional

Untuk arah gerak masing-masing roda seperti gambar 2 berikut.



Gambar 2. Kinematik dari sebuah penggerak Omni- Directional [3]

Keterangan :

X_B, Y_B = Sistem koordinat kartesian.

$V_i (1,2,3)$ = Arah putar roda omni.

L = Jarak pusat robot dengan titik center roda.

δ = Sudut roda omni terhadap sumbu Y = 30° .

Dari gambar 1 dan 2 persamaan kinematik dari sistem bisa didapatkan. Persamaan yang digunakan adalah.

$$V_y = V_1 \cos(\delta) - V_2 \cos(\delta)$$

$$V_x = V_3 - V_2 \sin(\delta) - V_1 \sin(\delta)$$

$$V_\theta = V_1/L + V_2/L + V_3/L$$

$$V_i (1,2,3) = w \cdot r$$

Dimana,

r = Radius roda omni (cm)

w = kecepatan angular roda (rad/sec)

Untuk mendapatkan kecepatan dari masing-masing roda omni adalah dengan cara mengalikan kecepatan dari motor dengan jari-jari dari roda omni. Dalam hal ini roda omni dirangkai secara simetris dengan perbedaan sudut antar roda sebesar

120° dan masing-masing roda memiliki sudut δ sebesar 30° . Sehingga dari persamaan (1) sampai persamaan (3) dapat ditulis dalam matriks

$$\begin{bmatrix} V_y \\ V_x \\ V_\theta \end{bmatrix} = \begin{bmatrix} \cos(\delta) & -\cos(\delta) & 0 \\ -\sin(\delta) & -\sin(\delta) & 1 \\ 1/L & 1/L & 1/L \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$$

B. Odometry

Odometry adalah penggunaan data dari pergerakan aktuator untuk memperkirakan perubahan posisi secara *real time* [4]. Odometry digunakan untuk memperkirakan posisi relatif terhadap posisi awal. Untuk itu digunakan perhitungan jumlah pulsa yang dihasilkan oleh sensor *rotary encoder* setiap satuan ukuran yang kemudian dikonversi menjadi satuan *milimeter*. Dengan persamaan sebagai berikut.

$$K \text{ roda} = 2\pi r$$

$$\text{pulsa per mm} = \frac{\text{resolusi encoder}}{K \text{ roda}}$$

Pada robot yang menggunakan roda omni directional dapat digunakan persamaan (1) sampai (3) untuk mendapatkan koordinat X dan koordinat Y. Dengan mengganti kecepatan setiap roda dengan jarak yang telah ditempuh masing-masing roda sehingga didapatkan persamaan.

$$V_y = S_1 \cos(\delta) - S_2 \cos(\delta)$$

$$V_x = S_3 - S_1 \sin(\delta) - S_2 \sin(\delta)$$

$$V_\theta = S_1/L + S_2/L + S_3/L$$

Dimana,

$S_i (1,2,3)$ = jarak tempuh dari masing-masing roda.

Persamaan diatas merupakan koordinat dari robot itu sendiri, sehingga jika diinginkan koordinat kartesian lapangan perlu ditransformasikan dengan menggunakan matriks transformasi dari rotasi sebagai berikut.

$$\begin{bmatrix} Y_{pos} \\ X_{pos} \\ \theta \end{bmatrix} = \begin{bmatrix} \cos(V_\theta) & -\sin(V_\theta) & 0 \\ \sin(V_\theta) & \cos(V_\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_y \\ V_x \\ V_\theta \end{bmatrix}$$

III. HASIL DAN DISKUSI

Dari model *kinematic odometry* terdapat nilai sudut peletakan *rotary* lantai yang tetap yaitu dengan sudut roda 1 sebesar 30° , roda 2 sebesar 150° dan roda 3 sebesar 270° , ketika robot bergerak maka nilai pulsa dari *rotary* lantai juga akan berubah, kombinasi dari perubahan nilai pulsa *rotary* dan sudut masing-masing *rotary* lantai akan dikalkulasi dalam satu fungsi yang sama dan akan menghasilkan letak posisi robot.



Gambar 3. Ilustrasi percobaan odometry

Tabel 1. Hasil percobaan odometry

TARGET	0,300			0,-300			-150,0			150,0		
PERCOBAAN	X	Y	θ	X	Y	θ	X	Y	θ	X	Y	θ
1	2	293	0	4	-295	1	-145	1	0	147	3	-1
2	-2	293	1	-1	-294	0	-146	0	0	146	0	-1
3	-5	292	1	-9	-295	-5	-144	-3	2	146	-4	-4
4	-3	293	2	2	-293	0	-143	0	2	146	-3	0
5	0	293	0	-2	-293	4	-145	0	1	146	1	1
POSISI RATA-RATA	-1.6	293	0.8	-1.2	-294	0	-145	-0.4	1	146	-0.6	-1
RATA-RATA ERROR DISTANCE	-7.185637048			-5.964054801			-5.393061031			-3.776072683		

Keterangan :

Terdapat 5 data percobaan untuk masing-masing posisi target, didapatkan rata-rata *error* posisi dan rata-rata *error distance* dengan satuan *centimeter*.

Selang waktu yang semakin lama, maka nilai *odometry* akan terdapat *error*, *error* ini bisa terjadi dari beberapa faktor, seperti pengaruh medan magnet dari *solenoid* saat dilakukan tendangan yang mengakibatkan nilai pulsa dari *rotary* lantai tidak terupdate dalam waktu tersebut, selain itu juga dapat terjadi karena faktor roda yang selip

IV. KESIMPULAN

Penentuan pergerakan dengan menggunakan *odometry* *rotary* lantai tidak sepenuhnya akurat karena terdapat beberapa faktor dan noise yang dapat mempengaruhi nilai *odometry*. Namun nilai *error odometry* dihasilkan masih dapat ditoleransi jika nilai *error* tidak lebih dari 30 cm dari posisi target.

V. UCAPAN TERIMA KASIH

Terimakasih atas kerjasama dengan laboratorium robot PENS.

VI. DAFTAR PUSTAKA

- [1] Iwan Kurnianto Wibowo, Mochammad Mobed Bactiar, Rizki Taufik Ulil Albab, Muhamad Khoirul Anwar, Zanuvar Tri Romadon dan Muhammad Jauharul Muqqorobin, 2017, "Desain Omnidirectional Vision pada Ersow"
- [2] Iwan Kurnianto Wibowo, Mochammad Mobed Bactiar, Rizki Taufik Ulil Albab, Muhamad Khoirul Anwar, Zanuvar Tri Romadon, Muhammad Jauharul, M. Sholahudin Al-Ayyubi, Agung Pambudi dan Rachmat Faizal Ajie., 2016, "Rancang Bangun Mekanik Penendang Pada Robot Soccer Beroda Menggunakan Solenoid"
- [3] Al-Amri, A. Salam and Iman Ahmed, 2010, *Control of Omni-Directional Mobile Robot Motion*, Al-Khawarizmi Engineering Journal, University of Baghdad, Vol.6, No.4, pp. 1-9.
- [4] Muhammad Labiyb Afakh, Fernando Ardilla, Iwan Kurnianto Wibowo, "Kontrol Pergerakan Omnidirectional Mobile Robot pada Robosoccer", TA PENS 2018.
- [5] Rojas, Raul, and Alexander Glove Forster, 2006, "Holonomic Control of a Robot with an Omni-Directional Drive", To appear in KI - Kunstliche Intelligent, BottcherIT Verlag.
- [6] Sanghani, Mayang and Sanket U. Vardhave, 2013, *Trajectory Following Robot*, UMIC Summer Project, Indian Institute Of Technology Bombay.
- [7] Ribeiro, F., 2000, *Robot Football Team from Minho University*. In *Robocup-99: Robot soccer World Cup III*, Berlin: Springer Verlag, pp. 731-734.
- [8] Borenstein, J. And L. Feng, 1996, "Gyrodometry: A New Method for Combining Data from Gyros and Odometry in Mobile Robots", Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, Apr. 22-28, pp. 423-428.
- [9] F. Ribeiro, I. Moutinho, P. Silva, C. Fraga, N. Pereira, "Three omni-directional wheels control on a Mobile robot". ResearchGate.

Penerapan Metode *Color Filtering* HSV untuk Pendeteksian Bola pada Robot KRSBI Beroda

Handriko, Irwan, Muhammad Iqbal Nugraha

Teknik Elektronika, Politeknik Manufaktur Negeri Bangka Belitung

Jalan Timah Raya Kawasan Industri Air Kantung, Sungailiat, Bangka, Indonesia

tenhanhan@gmail.com, irwan@ymail.com, iqbalbangka@gmail.com

Abstrak – KRSBI Beroda merupakan perlombaan robot yang bertemakan sepak bola. Didalam pertandingan robot sepak bola, pengolahan citra memegang peranan penting untuk mendeteksi bola sebagai objek utamanya. Bola yang digunakan pada KRSBI Beroda berwarna *orange* sehingga banyak sistem pendeteksian bola menggunakan metode warna seperti permodelan HSV (*hue, saturation, value*). Sistem yang digunakan didalam paper ini menggunakan permodelan warna HSV dan permodelan *hough circle transform* (HCT) untuk mendapatkan bentuk lingkaran pada bola. Proses pembacaan bola dimulai dari pengambilan objek gambar, kemudian diproses menggunakan permodelan HSV dan HCT menggunakan *tool* pada *openCV*. Hasil proses tersebut akan berupa koordinat yang dikirimkan ke aktuator untuk memposisikan robot. Hasil pengujian menunjukkan kemampuan robot dalam membaca bola dan koordinat bola dengan jarak 100cm hingga 500cm. Robot dapat melakukan pencarian bola dengan sudut baca hingga 120°. Sistem yang diterapkan pada robot mampu membaca bola dengan kecepatan hingga 1,6m/s.

Kata Kunci: *hough circle transform* (HCT), *hue, saturation, value* (HSV), *openCV*

I. PENDAHULUAN

Dalam robot sepak bola, kamera memegang peran penting dalam penempatan dan pergerakan robot karena kedua tugas tersebut bergantung pada hasil pembacaan gambar dalam pengambilan keputusan terutama pada robot penjaga gawang. Robot penjaga gawang membutuhkan respon pembacaan gambar yang cepat untuk menghalau arah datangnya bola menuju ke gawang. Salah satu metode yang digunakan untuk mendeteksi objek yaitu pengolahan citra digital (*digital image processing*).

Ada beberapa permodelan yang digunakan dalam pengolahan citra digital seperti permodelan warna yaitu permodelan RGB, CMY dan HSV [1]. Pada KRSBI Beroda, objek bola berwarna *orange* dengan lapangan berwarna hijau dengan garis putih, tiang berwarna putih dan robot pemain berwarna hitam sehingga penggunaan warna sebagai dasar pengolahan citra dapat digunakan dikarenakan objek yang dideteksi robot penjaga gawang hanya bola saja yang berwarna *orange*.

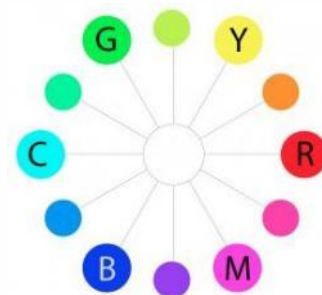
Dari permasalahan ini, penulis membuat sebuah sistem pendeteksian bola pada robot KRSBI Beroda *Polmanbabel*. Sistem ini dibangun dari

vision kamera pada robot. *Vision* pada kamera menggunakan *filter* warna HSV ditambah *filter* HCT dengan menggunakan *library OpenCV*. Dengan demikian, sistem diharapkan dapat mengurangi kesalahan deteksi sehingga proses pendeteksian warna dapat berjalan dengan akurat dan *real time*.

II. DASAR TEORI

A. Permodelan warna HSV

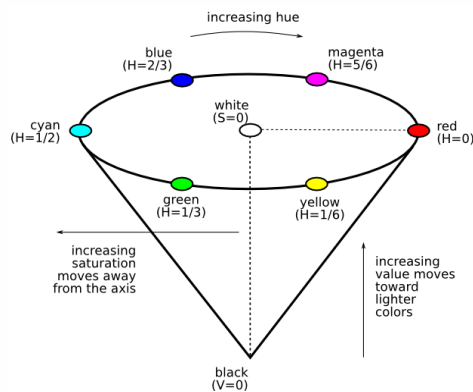
Model warna berbasis HSV (*hue, saturation, value*) merupakan model warna yang mengurangi ukuran informasi warna dan abu-abu pada gambar [1]. *Hue* adalah suatu ukuran panjang gelombang yang terdapat pada warna dominan yang diterima oleh penglihatan, *saturation* adalah ukuran banyaknya cahaya putih yang bercampur pada *hue* dan *value* menyatakan ukuran kecerahan (*brightness*) dari *hue* [2]. Permodelan HSV berfokus pada ruang warna tersier seperti pada gambar berikut.



Gambar 1. *Color space* pada permodelan HSV

Pada permodelan HSV, objek gambar diisolasi terhadap skala warna abu-abu dan dikelompokkan menjadi objek dengan ekstraksi warna. Penyaringan *hue* digunakan untuk menyegmentasikan warna yang ditentukan. Ketika nilai *hue* diatur, *mask* diterapkan ke gambar. Dalam gambar biner, ketika nilai piksel memenuhi kriteria yang ditentukan, seperti nilai *hue*, maka nilai tersebut akan ditransformasikan oleh *masking*. Fungsi diatur ke nol yang menunjukkan warna putih. Semua piksel dalam rentang warna yang ditunjuk ditandai sebagai latar depan yang ditampilkan dalam piksel putih dan lainnya

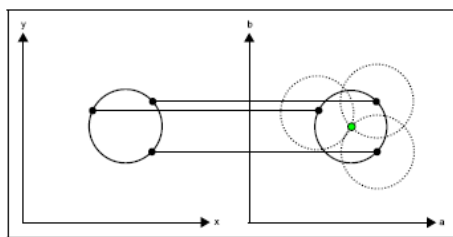
ditandai sebagai latar belakang ditampilkan dalam warna hitam [1].



Gambar 2. Fungsi set komponen HSV [3]

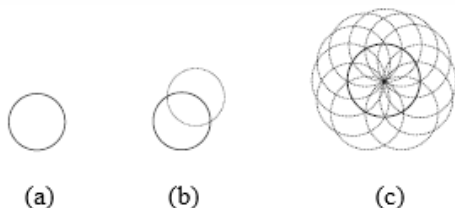
B. Filter Hough Circle Transform (HCT)

Hough circle transform merupakan suatu teknik untuk menentukan lokasi suatu bentuk dalam citra. Dalam implementasinya, HCT melakukan pemetaan terhadap titik-titik pada citra ke dalam parameter *space* berdasarkan fungsi yang mendefinisikan bentuk yang diinginkan, pada kasus ini lingkaran. Hal ini dapat dilihat pada gambar berikut.



Gambar 3. Ilustrasi *hough circle transform*

Gambar di atas merupakan contoh dari HCT untuk nilai radius tetap/konstan, dari jarak x, y ke parameter jarak. Untuk menemukan lingkaran pada sebuah gambar, maka perlu diketahui terlebih dahulu jari-jarinya. Contoh dari pendeteksian lingkaran melalui HCT dapat dilihat pada gambar dibawah ini



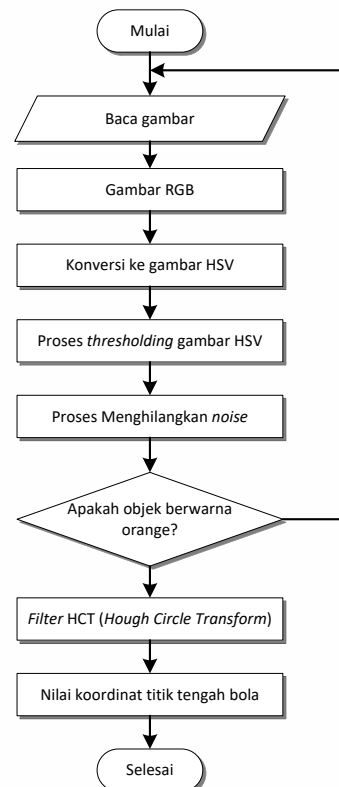
Gambar 4. Contoh dari pendeteksian sebuah lingkaran melalui penggunaan HCT

Pada gambar asli ada sebuah titik hitam yang dianggap sebagai titik radius (R) pada sebuah *background* yang terang pada gambar 4 (a). Untuk setiap piksel dengan titik hitam, sebuah pusat lingkaran di definisikan menjadi sebuah lingkaran dengan radius R dan pusatnya pada piksel tersebut pada gambar 4 (b). Frekuensi dengan piksel

gambar yang muncul di pusat lingkaran telah ditentukan hal ini dapat dilihat pada gambar 4 (c). Akhirnya, piksel dengan frekuensi yang paling tinggi dianggap sebagai pusat dari lingkaran dengan radius r [4].

III. METODOLOGI

Rancangan metode *color filtering* HSV dalam mendeteksi bola KRSBI Beroda berwarna *orange* dapat dilihat pada gambar berikut.

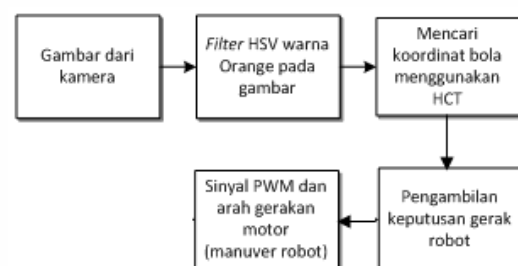


Gambar 5. Flow chart metode pendeteksian bola menggunakan *color filtering* HSV

Rancangan sistem yang dibuat berupa rancangan perangkat keras dan rancangan perangkat lunak. Rancangan perangkat keras yang dibuat ditunjukkan pada gambar 6 dan untuk perangkat lunak ditunjukkan pada gambar 7 berikut.



Gambar 6. Blok diagram *hardware* sistem



Gambar 7. Blok diagram *software* sistem

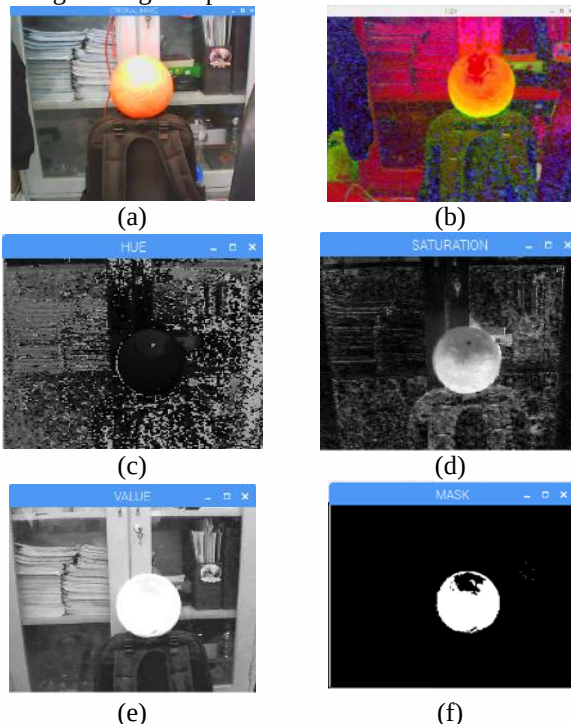
Pada metode diatas kamera mengambil gambar yang kemudian hasil gambar terbaca diproses menggunakan *filter* HSV dan HCT pada *library openCV*. Jika objek terdeteksi akan muncul data koordinat yang secara serial dikirimkan ke Arduino untuk menghasilkan sinyal PWM untuk menggerakkan motor.

IV. HASIL DAN PEMBAHASAN

Hasil yang didapat dari sistem ini adalah hasil dari beberapa pengujian yaitu pengujian kemampuan sistem dalam membedakan bola dan kemampuan robot dalam melakukan manuver terhadap pergerakan bola.

A. Pengujian Pembacaan Bola KRSBI Beroda dengan *Filter* HSV

Warna bola pada KRSBI Beroda yaitu berwarna *orange* cerah (*fluorescent*). Pada ruang warna HSV warna orange berada diantara merah dan kuning. Sehingga memiliki *range* sudut 0° hingga 60° . Dengan rentang nilai warna sebesar 255 maka ruang warna orange memiliki *range hue* sebesar 0 hingga 42,5. Berikut hasil pembacaan masing-masing komponen HSV.



Gambar 8. Hasil pembacaan gambar RGB (a), gambar HSV (b), *hue* (c), *saturation* (d), *value* (e) dan *masking* (f)

Tabel 1. Nilai set komponen HSV terhadap bola

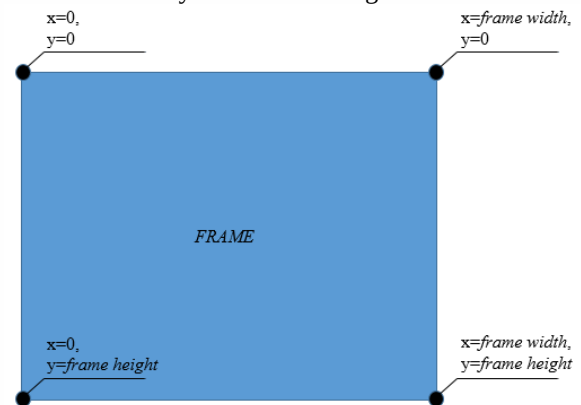
No	Keterangan	Nilai HSV
1	Hmin	0
2	Hmax	20
3	Smin	81
4	Smax	255
5	Vmin	225
6	Vmax	255

Pada gambar 8 nilai *hue* dan *saturation* diidentifikasi berdasarkan *color space* warna orange didalam permodelan HSV. Nilai *value* didapat dari pengaturan terhadap kecerahan ruangan sewaktu pengambilan gambar.

B. Pengujian Pembacaan Koordinat Bola KRSBI Beroda dengan *Filter* HCT

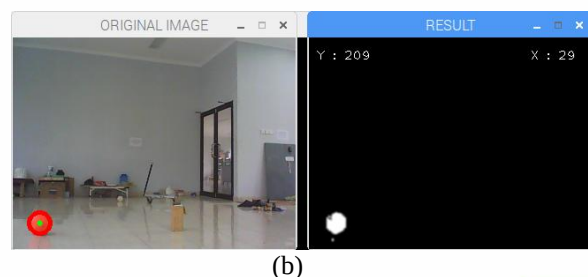
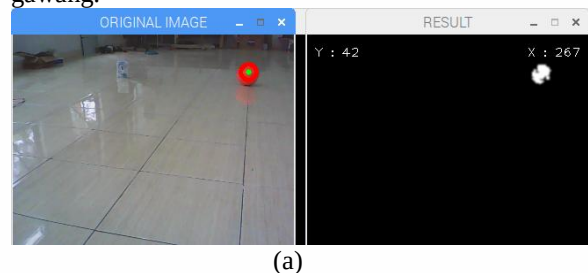
Koordinat bola didapatkan dengan menjadikan hasil *masking* dari proses HSV sebagai referensi. Koordinat bola didapat dengan menggunakan *tool hough circle transform* (HCT) pada *openCV*. Melalui *tool* ini hasil dari *masking* diolah untuk mendapatkan bentuk lingkaran dari objek bola. Bentuk lingkaran yang didapat kemudian diambil nilai koordinat bola pada *frame* kamera beserta radius dari objek bola.

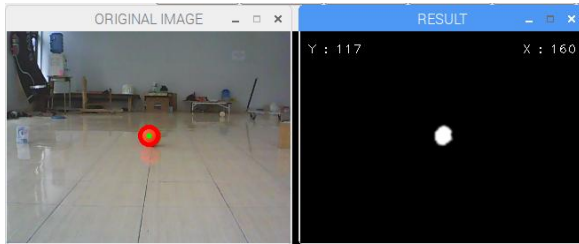
Koordinat x dan y direpresentasikan terhadap ukuran *frame* gambar yang ditangkap. Peta koordinat x dan y ditentukan sebagai berikut.



Gambar 9. Peta koordinat x dan y didalam *frame* gambar

Frame yang digunakan berukuran 320x240 pixel. Berikut hasil pembacaan bola dengan tambahan *filter* HCT di berbagai sisi didalam *frame* gambar pada jarak 3meter dari robot penjaga gawang.





(c)
Gambar 10. Pembacaan bola sisi kanan atas *frame* (a), sisi kiri bawah *frame* (b) dan sisi tengah *frame* (c)

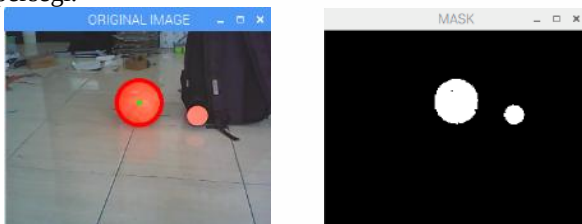
Tabel 2. Nilai koordinat bola terhadap jarak baca bola

No.	Jarak Bola	Koordinat x	Koordinat y
1	<100cm	0	0
3	100cm	160	205
4	150cm	162	151
5	200cm	165	124
6	250cm	172	105
7	300cm	172	94
9	400cm	173	80
11	500cm	175	71
12	>505cm	0	0

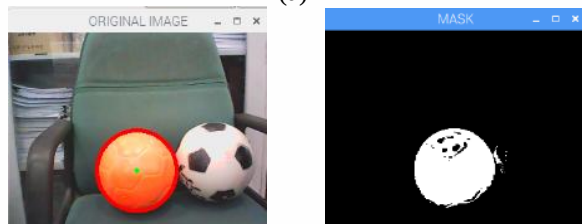
Pada pengujian kamera dimiringkan dengan sudut tertentu dan bersifat *fixed mount* sehingga hanya bisa membaca pada jarak seperti pada tabel diatas.

C. Pengujian Bola KRSBI Beroda Terhadap Objek Lain

Selain pengujian diatas dilakukan juga pengujian terhadap objek lain yaitu pengujian dengan objek berwarna sama dan dengan objek berdimensi dan bentuk sama. Pengujian terhadap objek berwarna sama dibagi 2 yaitu dengan bentuk sama dan bentuk berbeda. Dalam pengujian ini digunakan lingkaran lain dan objek berbentuk persegi.



(a)



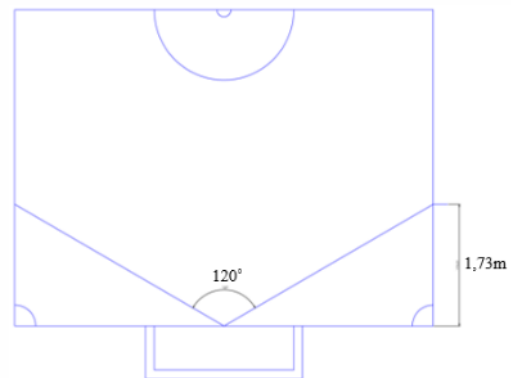
(b)

Gambar 11. Pembacaan bola terhadap lingkaran lain warna sama (a) dan bola lain warna berbeda

Pada gambar 11 (a) persegi tidak terdeteksi karena adanya *filter* HCT dan pada gambar (b) bola putih tidak terdeteksi karena adanya *filter* HSV.

D. Pengujian Penerapan *Filtering* HSV Pada Robot Penjaga Gawang

Pada pengujian ini bola digerakkan dan diposisikan diberbagai tempat didalam lapangan KRSBI Beroda. Robot penjaga gawang memposisikan diri sesuai dengan letak bola dan ukuran gawang. Pada penelitian ini sistem dituntut untuk dapat melakukan pembacaan bola dengan sudut baca hingga 120° . Sudut baca kamera hanya 58° sehingga robot bergerak kekiri dan kekanan hingga melakukan rotasi ditempat untuk melakukan pencarian bola. Untuk sudut baca bola 120° maka robot penjaga gawang harus dapat membaca bola pada area seperti pada gambar dibawah ini.



Gambar 12. Area baca bola robot penjaga gawang

Berikut hasil penempatan posisi robot penjaga gawang dalam menyesuaikan letak bola.



(a)

(b)



(c)

Gambar 13. *Positioning* robot penjaga gawang terhadap posisi bola kiri gawang (a), kanan gawang (b) dan tengah gawang (c)

Pengujian lain yang dilakukan yaitu pengujian kecepatan respon sistem terhadap kecepatan pergerakan bola. Didalam pengolahan citra objek gambar diproses dalam satuan pixel. Semakin dekat objek dengan bola maka pixel objek didalam

gambar akan semakin besar begitu pula sebaliknya. Berikut hasil kecepatan pemrosesan pengolahan gambar dengan berbagai jarak dapat dilihat pada tabel berikut.

Tabel 3. Kecepatan respon sistem dalam melakukan pengolahan citra terhadap jarak bola

No.	Jarak Bola	Rerata waktu pengolahan citra (ms)	Frame per second
1	100cm	53,4676	18,7 ~ 18
2	150cm	51,9896	19,2 ~ 19
3	200cm	51,4852	19,4 ~ 19
4	250cm	50,8734	19,6 ~ 19
5	300cm	49,9532	20,1 ~ 20
6	400cm	47,2987	21,1 ~ 21
7	500cm	45,8661	21,8 ~ 21
8	Tidak ada objek	44,3201	22,5 ~ 22

Tabel diatas adalah waktu pengolahan citra dari objek bola bukan waktu proses sistem. Waktu proses sistem yang dimaksud adalah waktu proses pengolahan citra dan waktu proses komunikasi ke robot. Komunikasi yang digunakan ialah USART dengan *baudrate* 9600 *bit per second* (bps) dengan mode pengiriman data 8 bit, 1 bit start, 1 bit stop. Terdapat 10 karakter yang dikirimkan artinya terdapat 100bit data. Dengan *baudrate* sebesar 9600 bps maka untuk mengirimkan 1 bit data membutuhkan waktu sebesar 104 mikro detik, maka waktu yang dibutuhkan untuk 10 karakter sebesar 100 bit x 104 mikrodetik/bit = 10,4ms.

Dengan menggunakan data diatas dilakukan pengujian kecepatan pembacaan bola terhadap *limit* yang ditentukan. Pengujian yang dilakukan menggunakan *stopwatch* dimana bola digerakkan dari kiri ke kanan atau sebaliknya sejauh 4m dengan waktu yang dihitung *manual* oleh *time keeper*. Pengujian ini tidak efektif dan tidak *valid* karena tidak ada alat ukur pembanding yang tepat untuk membaca kecepatan bergulirnya bola sehingga pengujian ini memiliki tingkat kesalahan yang besar. Berikut hasil pengujian pembacaan bola terhadap *limit* kecepatan pada jarak tertentu dapat dilihat pada tabel dibawah ini.

Tabel 4. Hasil Pengujian Kecepatan Pergerakan Bola Jarak Tempuh 400cm

Speed bola (m/s)	Distance (cm) Status pergerakan bola (G=good, NG= not good)						
	100	150	200	250	300	400	500
0,57	G	G	G	G	G	G	G
0,62	G	G	G	G	G	G	G
0,67	G	G	G	G	G	G	G
0,73	G	G	G	G	G	G	G
0,80	G	G	G	G	G	G	G
0,89	G	G	G	G	G	G	G
1,00	G	G	G	G	G	G	G
1,14	NG	NG	G	G	G	G	G

1,33	NG	NG	NG	G	G	G	G
1,60	NG	NG	NG	NG	NG	G	G
2,00	NG	NG	NG	NG	NG	NG	NG
2,67	NG	NG	NG	NG	NG	NG	NG

Berdasarkan hasil pengujian didapatkan hasil bahwa robot mampu membaca pergerakan bola dengan kecepatan maksimal 1,6m/s.

V. KESIMPULAN

Dari hasil penelitian dapat diambil kesimpulan bahwa sistem deteksi warna bola dengan model warna HSV cukup baik. Sistem mampu membedakan bola dengan obyek lain dengan bentuk dan warna tertentu. Proses pengolahan citra pada sistem ini memiliki kecepatan berkisar 18 hingga 22 *frame per second*. Keluaran dari sistem pengolahan citra berupa koordinat yang dikomunikasikan langsung ke robot. Dengan sistem ini robot mampu memposisikan diri sesuai dengan posisi bola. Robot mampu mendeteksi bola pada jarak 100cm hingga 500cm dan sudut baca bola hingga 120°. Robot mampu membaca bola yang bergerak dengan kecepatan hingga 1,6m/s.

VI. UCAPAN TERIMA KASIH

Ucapan terima kasih diucapkan kepada Direktur dan Pembantu Direktur Polmanbabel yang mendukung penulis dalam berkompetisi di KRSBI Beroda. Penulis juga mengucapkan terima kasih kepada para sponsor yang telah membantu dalam menyandang dana kegiatan ini. Tidak luput juga penulis haturkan terima kasih atas semangat peserta yakni para pengurus muda yang tergabung dalam tim KRSBI Beroda Polmanbabel 2018.

VII. DAFTAR PUSTAKA

- [1] Monika Deswal, Neetu Sharma, "A Fast HSV Image Color and Texture Detection and Image Conversion Algorithm", International Journal of Science and Research (IJSR), 2012.
- [2] Kusumanto R.D., Alan Novi Tomponu, Wahyu Setyo Pambudi, "Klasifikasi Warna Menggunakan Pengolahan Model Warna HSV", Jurnal Ilmiah Elite Elektro, Vol. 2, No. 2: 83-87, 2011.
- [3] John W. Shipman, Introduction to Color Theory, The hue-saturation-value color model. <http://infohost.nmt.edu/tcc/help/pubs/colortheory/web/hsv.html>
- [4] Bruno M.M. Riberio, Daniel A. Martins, Ant'onio J. R. Neves and Armando J. Pinho, "Arbitrary ball detection using the circular hough transform", Signal Processing Lab, DETI IEETA, University of Aveiro, 3810-193 Aveiro, Portugal, 2009.

Read TCP/IP Menggunakan *platform* Matlab pada Robot Sepak Bola Beroda

Niswatun Solehah, Astriana Siahaan, Widya Astuti, Rama Rifaldi Larengi,
Andhyka Vergiawan, Muh. Algifahri Ngareng
STMIK Adhi Guna, Jl. Undata No.3, Palu 94111

Email: niswatunsoleha740@gmail.com, butetsiahaan11@gmail.com, widyaastuti2201@gmail.com,
algolngareng@gmail.com, Andhykaoor@gmail.com

Abstrak -- Dalam kompetisi KRSBI Beroda, dituntut robot bisa bermain secara aktif dalam melakukan positioning. Dalam menunjang penempatan robot dibutuhkan sebuah kontroler dari luar lapangan untuk mengaktifkan robot seperti halnya *Referee Box*. *Referee Box* akan mengirim sebuah perintah ke robot yang nanti nya akan bergerak sesuai dengan instruksi perintah dari *referee box*. Tujuan dari penelitian ini adalah membaca data TCP/IP pada robot dalam hal ini menggunakan *platform* matlab sebagai wadah pengolah data karakter dari *basestation* agar bisa aktif ketika dikontrol oleh *referee box*. Metode yang digunakan dalam penelitian kali ini yaitu sistem komunikasi pada aturan Kontes Robot Indonesia Devisi KRSBI Beroda. Hasil yang diperoleh data TCP/IP Matlab Karakter *KickOff* memiliki nilai 75 dan *Start* memiliki nilai 115 dengan data yang diharuskan untuk dikonversi menjadi sebuah data karakter agar bisa menerima dengan sempurna data karakter yang ada pada *basestation*. Sehingga dapat disimpulkan bahwa *platform* matlab dapat difungsikan sebagai wadah penerima kontroler bagi robot.

Kata Kunci: TCP/IP, *Basestation*, Matlab

I. PENDAHULUAN

Kontes robot Indonesia merupakan sebuah acara yang bergengsi yang diikuti oleh seluruh mahasiswa yang berkecimpung dalam dunia robotika diseluruh Indonesia. Dalam acara tersebut ada 5 devisi yang akan bertanding dan salah satunya yaitu devisi KRSBI beroda. Pada kontes tersebut, digunakan *referee box* yang sudah disiapkan oleh panitia untuk mengatur jalannya pertandingan [1]. *Referee box* adalah aplikasi *game controller* yang digunakan juri untuk mengontrol jalannya pertandingan. *Referee box* game controller yang digunakan juri untuk mengontrol pertandingan, seperti pertandingan, akan dimulai kapan, *score* berapa, robot ingin diangkut, pelanggaran dan pinalti. *Basestation* menerima data dari *referee box* kemudian diteruskan ke robot.

Basestation merupakan penerima data dari *refree box* dan akan meneruskan data ke robot menggunakan *platform* matlab, *platform* matlab merupakan salah satu wadah yang dapat menerima Komunikasi TCP/IP juga dapat mengkonversi data float menjadi *char* atau karakter.

Maka dari itu, penulis menerapkan komunikasi antara *referee box* dan robot melalui perantara *basestation*, yang mana akan terhubung dengan SSID yang sama untuk melakukan komunikasi antara *referee box* dan robot.

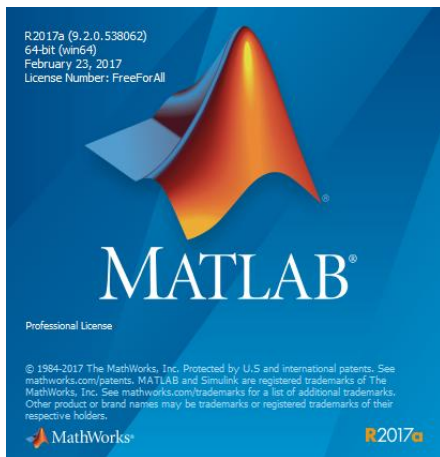
II. METODE

A. TCP/IP

TCP/IP (*Transmission Control Protocol/Internet Protocol*) merupakan standar komunikasi data yang digunakan dalam proses tukar-menukar data dari satu komputer ke komputer lain. TCP/IP merupakan jaringan terbuka yang bersifat independen terhadap mekanisme *transport* pada jaringan fisik yang digunakan, sehingga dapat digunakan di mana saja. Protokol ini menggunakan skema pengalamatan yang sederhana yang disebut sebagai alamat IP (*IP Address*) yang mengizinkan banyak komputer untuk dapat saling berhubungan satu sama lainnya di Internet. Protokol ini juga bersifat *routable* yang berarti protokol ini cocok untuk menghubungkan sistem-sistem berbeda untuk membentuk jaringan yang heterogen [2].

B. Matlab

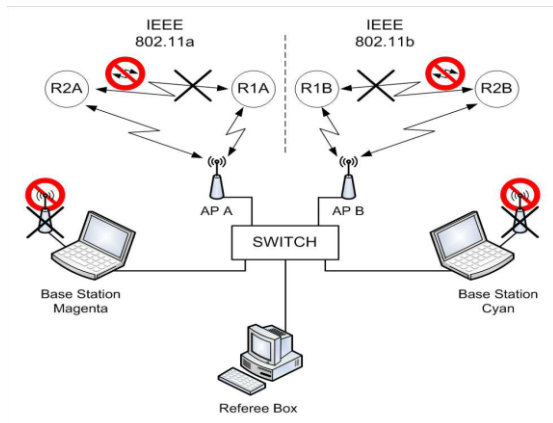
MATLAB (*Matrix Laboratory*) adalah sebuah program untuk analisis dan komputasi numerik dan merupakan suatu bahasa pemrograman matematika lanjutan yang dibentuk dengan dasar pemikiran menggunakan sifat dan bentuk matriks. Pada awalnya, program ini merupakan *interface* untuk koleksi rutin numerik dari proyek LINPACK dan EISPACK, dan dikembangkan menggunakan 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 [3]).



Gambar 1. Matlab picture

C. Komunikasi TCP/IP

Metode yang digunakan dalam menyusun sebuah sistem adalah mengacu pada peraturan Kontes Robot Indonesia (KRI) Divisi KRSBI Beroda.



Gambar 2. Komunikasi TCP/IP (Sumber : rules KRSBI beroda 2017) [4].

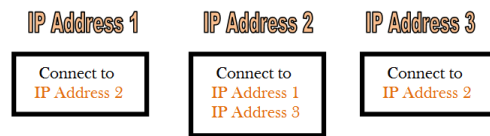
D. Blok Diagram



Gambar 3. Alur data

Dalam pengujian sistem yang dilakukan memiliki alur data dengan 3 (tiga) bagian yakni referee box menggunakan Aplikasi Processing I3 dengan penggunaan bahasa java yang tersedia didalamnya. kemudian selanjutnya adalah Basestation menggunakan aplikasi Python 2.7, terakhir adalah aplikasi Matlab versi 2017a.

E. Setting Komunikasi antar PC

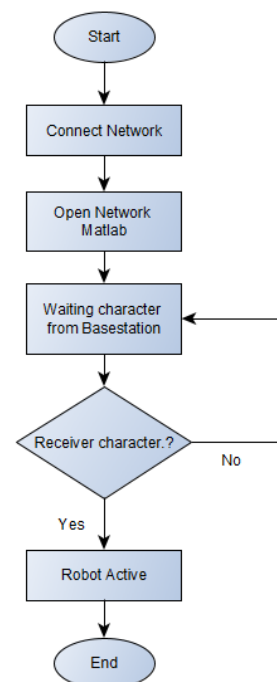


Gambar 4. Pengalamatan IP address

Terdapat beberapa PC (*Personal Computer*) yang saling berkomunikasi antara satu dan lainnya, gambar diatas menunjukkan *Basestation* yang merupakan IP Address 2 adalah sebuah sentral atau jembatan antara Referee Box (IP Address 1) dan Robot (IP Address 3) yang saling berkomunikasi.

F. Perilaku Matlab pada Proses Jaringan

Terlihat pada gambar 5 adalah proses mengolah data karakter pada Matlab. Beberapa langkah agar robot bisa aktif sesuai dengan perintah dari Referee Box.



Gambar 5. Flowchart menerima karakter platform matlab.

Awal PC pada matlab mengkonekkan jaringan ke PC basestation. Kemudian jaringan lokal pada PC mengharuskan matlab untuk membuka jaringan atau yang biasa di sebut dengan *fopen* pada matlab. setelah terbuka, proses *read* pada matlab akan berjalan untuk menunggu karakter dari basestation. setelah karakter diterima maka dengan waktu yang sama robot akan aktif dan bergerak sesuai dengan instruksi dari perintah tombol Referee Box.

G. Metode Pengujian

Dalam hal pengujian komunikasi data TCP/IP terdapat beberapa elemen penting yang mesti digunakan : 1. *Referee Box*, 2. *Basestation*, 3. *Matlab*, 4. *Robot*.

Tabel 1. Metode Pengujian

Kondisi referee box	Kondisi basestation	Kondisi Matlab	robot
Kick off	Kondisi karakter kick off	Perilaku data kick off	Perilaku robot
Start	Kondisi karakter star	Perilaku karakter star	Perilaku robot
Kondisi tidak ada	Perilaku karakter	Perilaku data	Perilaku robot

Keterangan :

1. *Referee Box* menggunakan beberapa tombol untuk diujikan yaitu *kickoff* dan *start*,
2. *Basestation* akan meneruskan karakter dari *referee box* untuk selanjutnya dilakukan pengiriman ke TCP/IP Matlab.
3. Pengiriman data matlab ke robot dilakukan ketika telah menerima karakter dari protokol *basestation*.
4. Perilaku robot akan aktif ketika karakter dari matlab telah tersampaikan.

III. HASIL DAN DISKUSI

A. Refreee Box



Gambar 6. Tampilan *referee box*

Terlihat pada *gambar 6* merupakan tampilan *Referee Box*. Untuk melihat pengiriman data pada tombol *Referee Box* pada penelitian kali ini adalah menggunakan *KickOff* dan *Start*. Dalam hal ini *Basestation* ataupun *Matlab* pada robot akan membaca karakter yang terdapat pada tombol *Referee Box* tersebut diatas.

```

17 public static final int appFrameRate = 15;
18
19 public static String[] Teamcmds= { "KickOff", "FreeKick", "GoalKick", "Throw In", "Corner", "P
20 public static String[] Commcmds= { "START", "STOP", "DropBall", "Park", "End Part", "RESET",
21
22 public static final String[] cTeamcmds= { "K", "F", "G", "T", "C", "P", "A", "O", "R", "Y" };
23 public static final String[] cTeamcmds= { "k", "f", "g", "t", "c", "p", "a", "o", "r", "y" };
24 public static final int CMDID_TEAM_KICKOFF = 0;
25 public static final int CMDID_TEAM_FREEKICK = 1;
26 public static final int CMDID_TEAM_GOALKICK = 2;
27 public static final int CMDID_TEAM_THROWIN = 3;
28 public static final int CMDID_TEAM_CORNER = 4;
29 public static final int CMDID_TEAM_PENALTY = 5;
30 public static final int CMDID_TEAM_GOAL = 6;
31 public static final int CMDID_TEAM_REPAIR_OUT = 7;
32 public static final int CMDID_TEAM_REDCARD = 8;
33 public static final int CMDID_TEAM_YELLOWCARD = 9;
34
35 public static final String[] cCommcmds= { "s", "S", "N", "L", "H", "Z", "e" };
36 public static final int CMDID_COMMON_START = 0;

```

Gambar 7. Struktur *listing* karakter *referee box*

B. Informasi Basestation

```

dataRobot1_temp = baseStation.myreceive()
dataRobot1 = dataRobot1_temp.decode('unicode
print(dataRobot1)

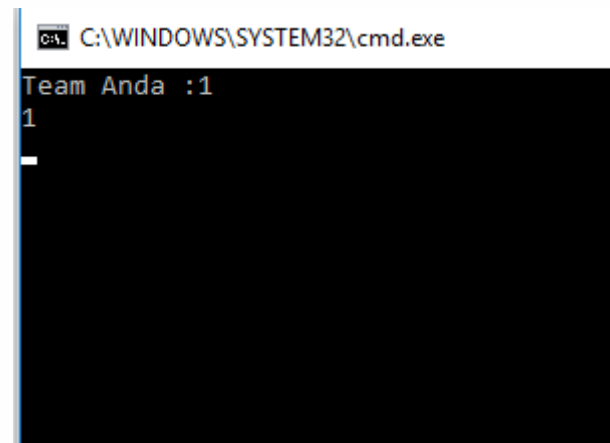
if (dataRobot1=='K' or dataRobot1=='k'):
    print("Kickoff")
    robot.mysend("K")

elif (dataRobot1=='s'):
    print("Start")
    robot.mysend("s")

```

Gambar 8. Kondisi karakter pada *basestation*

Terdapat *listing* program *Python* dalam hal ini adalah *basestation* yang terdapat pada *gambar 8* yakni kondisi *dataRobot1* untuk mengkodekan karakter *KickOff* dan *Start* yang terdapat pada *listing robot.mysend*.



Gambar 9. Ready kirim karakter

```
C:\WINDOWS\SYSTEM32\cmd.exe
Team Anda :1
1
----- Perintah Dikirim PORT 54321 -----
event
K
KickOff
0
***
Va
CK
0
0
----- Proses 54321 Selesai -----
----- Proses Base Station-----
K
Kickoff
----- Proses Base Station-----
```

Gambar 10. Kirim karakter *kickOff*

```
C:\WINDOWS\SYSTEM32\cmd.exe
***
Vd
CK
0
0
----- Proses 54321 Selesai -----
----- Proses Base Station-----
K
Kickoff
----- Proses Base Station-----

----- Perintah Dikirim PORT 54321 -----
event
s
START
-1
Vf
ST
0
0
----- Proses 54321 Selesai -----
----- Proses Base Station-----
s
Start
----- Proses Base Station-----
```

Gambar 11. Kirim karakter *start*

Informasi yang terlihat pada gambar 9 merupakan karakter yang ada pada *basestation* akan terkirim pada Matlab. pada gambar 10 karakter *KickOff* *basestation* telah terkirim ke informasi TCP/IP Robot dalam hal ini adalah Matlab. juga pada gambar 11 karakter *start* telah terkirim.

C. TCP/IP Matlab

1. Persamaan TCP/IP Matlab

Dalam mengkoneksikan TCP/IP pada matlab, beberapa persamaan dalam mengkoneksikan koneksi jaringan antar pc yakni *basestation* dan pc robot. `t = tcpip('0.0.0.0', 3000, 'NetworkRole', 'server');` merupakan persamaan mengalamatkan *ip address basestation* untuk selanjutnya saling

berkomunikasi. `fopen(t);` merupakan persamaan untuk membuka jalan masuk data dari *basestation*. `data = fread(t, t.ByteAvailable);` merupakan persamaan membaca data agar robot bisa aktif sesuai dengan instruksi dari *Referee Box*

Session 1: MATLAB Server

Accept a connection from any machine on port 30000.

```
t = tcpip('0.0.0.0', 30000, 'NetworkRole', 'server');
```

Open a connection. This will not return until a connection is received.

```
fopen(t);
```

Read the waveform and confirm it visually by plotting it.

```
data = fread(t, t.BytesAvailable);
```

Gambar 12. Persamaan koneksi TCP/IP (sumber *example matlab mathrworks*)[5]

2. Perilaku Data TCP/IP Matlab

Penerimaan data awal dari *basestation* ke TCP/IP Matlab adalah 0 (*no!*) dan ketika data karakter terbaca maka akan menghasilkan data lebih dari 0 (*no!*).

Terlihat pada gambar 13 yaitu hasil pembacaan data dari *basestation* yang sebelumnya telah ada penekanan pada tombol kontroler *referee box* yaitu pada gambar a *KickOff* dengan nilai data 75 dan pada gambar b *Start* dengan nilai data 115.

Command Window	Command Window
New to MATLAB? See resources for Getting Started .	New to MATLAB? See resources for Getting Started .
data =	data =
75	115
<code>f% KickOff</code>	<code>f% start</code>
(a)	(b)

Gambar 13. Command window print terima data TCP/IP. (a) *KickOff*, (b) *Start*

Terlihat pada gambar 14 yaitu hasil pembacaan data dari *Basestation* yang sebelumnya telah ada penekanan pada tombol kontroler *Referee Box* data `1x0 empty double row vector` merupakan sebuah data yang tidak menerima data penekanan tombol dari *Referee Box* sehingga data pada *Basestation* tidak mengalir ke robot dalam hal ini adalah *platform matlab*

```
Command Window
New to MATLAB? See resources for Getting Started.

Warning: Unsuccessful read: The specifie
data =
1x0 empty double row vector

Warning: Unsuccessful read: The specifie
data =
1x0 empty double row vector

fx |
```

Gambar 14. Command window print data TCP/IP tidak terbaca

D. Hasil TCP/IP Matlab

Pengaktifan robot agar bisa bergerak sesuai instruksi *Referee Box* yakni :

1. Koneksi TCP/IP

Terlihat pada gambar 15 adalah pengalaman *IP Address* dan penentuan *Port* dari *Basestation* dengan penentuan variabel *t*. *IP Address* dan *Port* yang dimasukkan sesuai dengan pengalaman dari *Basestation* itu sendiri.

```
13 % ===== BUKA JARINGAN SERVER =====
14 t = tcpip('192.168.2.99', 8081, 'NetworkRole', 'server');
15 fprintf('\nterbuka tcp/ip');
16
```

Gambar 15. Listing koneksi TCP/IP

```
Command Window
New to MATLAB? See resources for Getting Started.

>> tcp_ip_arduino_ImageProcessing

fx terbuka tcp/ip
```

Gambar 16. print hasil koneksi TCP/IP

2. Buka Pintu Jaringan

Terlihat pada gambar 17 merupakan *listing* program untuk membuka alamat IP dengan keterangan "ADA" pada fungsi *fprintf*. keterangan tersebut indikator bahwa robot *ready* menerima karakter dari *basestation*.

```
17 - fopen(t);
18 - fprintf('\nADA');
```

Gambar 17. Listing membuka pintu IP address

```
Command Window
New to MATLAB? See resources for Getting Started.

terbuka tcp/ip
ADA
masuk imageWarning: Unsuccessful read:
masuk imageWarning: Unsuccessful read:
fx masuk image
```

Gambar 18. print hasil pintu jaringan terbuka

3. Baca Data Karakter KickOff

Terlihat pada gambar 19 merupakan *listing* program menerima data karakter dari *Basestation* yakni karakter 'K', karakter yang dimaksud adalah perintah *KickOff* dari *Referee Box*. Perintah tersebut merupakan acuan awal robot memposisikan pada lingkungan lapangan yang telah ditentukan.

```
23 - data = fread(t,floor( t.BytesAvailable>=0),'char');
24 - if data == 'K'
25 -     fprintf('\nKickOff')
26 -     fwrite(s1,'1');
27 - end
28
```

Gambar 19. Listing baca karakter kickoff

```
Command Window
New to MATLAB? See resources for Getting Started.

masuk imageWarning: Unsuccessful
masuk imageWarning: Unsuccessful
masuk imageWarning: Unsuccessful
masuk imageWarning: Unsuccessful
masuk image
KickOff
fx masuk image
```

Gambar 20. Print hasil baca karakter kickoff

4. Baca Data Karakter Start

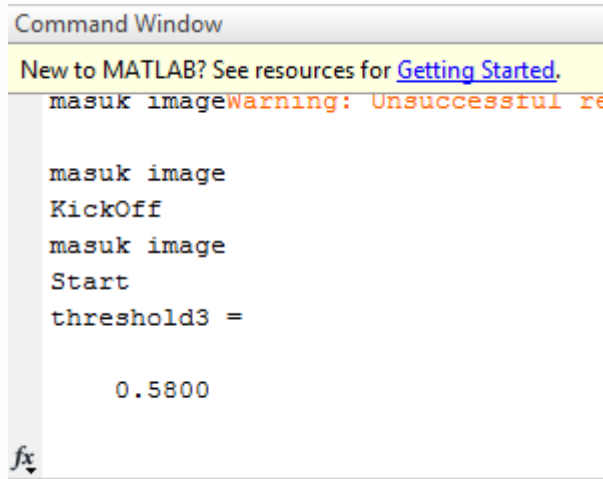
Terlihat pada gambar 21 merupakan *listing* program menerima data karakter dari *Basestation* yakni karakter 's', karakter yang dimaksud adalah perintah *Start* dari *Referee Box*. Perintah tersebut merupakan acuan robot aktif bergerak dalam melakukan permainan.

```

28 -
29 -     data = fread(t,floor( t.BytesAvailable>=0),'char');
30 -     if data == 's'
31 -         fprintf('\nStart')
32 -         fwrite(s1,'9');
33 -         fprintf('\nmasuk image');
34 -         while t.BytesAvailable == 0
35 -             trigger(vid);
36 -         % =====menangkap frames gambar=====
37 -         Image=getsnapshot(vid);
38 -         % =====deteksi warna bola=====
39 -

```

Gambar 21. Listing baca karakter start



Gambar 22. Print hasil baca karakter start

E. Hasil Pengujian

Tabel 2. Hasil pengujian

Kondisi Referee Box	Kondisi Basestation	Data Matlab	Robot
KickOff	Kirim Karakter "K"	75	Aktif
Start	Kirim Karakter "S"	115	Aktif
Kondisi Tidak Ada	Kondisi Tidak Ada	0	Tidak aktif

Dalam pengujian yang dilakukan terhadap data TCP/IP matlab menghasilkan sebuah data angka yang mana ketika eksekusi pada *referee box KickOff*, matlab akan menerima data 75 setelah dikirim karakter dari *basestation*. Selanjutnya, ketika tombol start di tekan maka tcp matlab akan menghasilkan angka 115 dan robot aktif ketika kedua tombol ditekan. Sebaliknya ketika tidak ada aktivitas tombol sama sekali maka data pun yang diterima TCP/IP matlab tidak ada berefek pada robot tidak aktif.

IV. KESIMPULAN

Dalam penelitian ini, peneliti berkesimpulan bahwa *platform* matlab dapat difungsikan sebagai wadah penerima kontroler bagi robot. dengan *basestation* sebagai *central* dalam menerima data karakter dari *referee box* sekaligus mengirim data karakter ke robot dalam hal ini adalah *platform* matlab dapat berkomunikasi sesuai dengan metode yang telah dibangun.

V. UCAPAN TERIMA KASIH

Dalam penulisan karya ilmiah ini, penulis mengucapkan terima kasih kepada team 10-D KRSBI Beroda STMIK Adhi Guna karena itu, dalam kesempatan ini penulis sangat mengharapkan adanya saran untuk menyempurnakan karya ilmiah ini.

VI. DAFTAR PUSTAKA

- [1] Fitri, "Penggunaan Bahasa Pemrograman Python Sebagai Pusat Kendali Pada Robot 10-D", STMIK Adhi Guna, Palu, 2017.
- [2] Santoso, Gatot, "Sistem Seluler WCDMA(Wideband Code Division Multiple)", Yogyakarta, 2006.
- [3] Anonymous, "Pengantar Untuk Pemrograman MATLAB", Institut Pertanian Bogor, Bogor, 2013.
- [4] Buku Panduan Kontes Robot Sepak Bola Indonesia Beroda (KRSBI Beroda) 2017, 2017.
- [5] Example Matlab Mathworks, "Cummunicate Using TCP/IP Server Sockets", 2017.

Implementasi Jaringan Saraf Tiruan untuk Meningkatkan Akurasi Sensor Kompas HMC5983 pada Robot KRSBI Beroda

Muamar Syahidan, Gerdy Pranaya Alfinal Hasni, Rif'al Ulum Zidni, Ibnu Kurnia Cahya, Cita Rahiim Tama, Shamsul Hadi

Universitas Brawijaya

Jalan Veteran, Malang, Jawa Timur 65145, Indonesia

Email: ¹syahidan.muamar1@gmail.com, ²gerdystar@gmail.com, ³ulumrifal@gmail.com, ⁴ibnukurniac11@gmail.com, ⁵crtama@ymail.com, ⁶kanghadishamsul@gmail.com

Abstrak --Di dalam Kontes Robot Sepakbola Indonesia Beroda (KRSBI Beroda), mahasiswa dituntut untuk dapat mengembangkan robot yang dapat bergerak secara *autonomous*, sehingga dibutuhkan suatu sensor yang dapat dipasang pada robot agar dapat mendeteksi kondisi lingkungan yang ada disekitarnya, salah satunya yaitu sensor kompas. Sensor kompas yang digunakan pada tim ENSPARTAN dari Universitas Brawijaya adalah HMC5983. HMC5983 memiliki sensitivitas yang tinggi, namun sayangnya keakurasian derajat putaran sensor terbilang cukup kecil pada beberapa kondisi. Oleh karena itu, tim ENSPARTAN mencoba mengimplementasikan Jaringan Saraf Tiruan untuk meningkatkan akurasi sensor kompas HMC5983 pada robot KRSBI Beroda. Dari hasil penelitian ini didapatkan bahwadengan menggunakan JST, akurasi sensor kompas HMC5983 menjadi lebih besar dengan akurasi rata-rata sebesar 98,43% dengan kesalahan terbesarnya yaitu 3 derajat.

Kata kunci : Jaringan Saraf Tiruan, Sensor Kompas, HMC5983, KRSBI Beroda, Magnetometer.

I. PENDAHULUAN

Kontes Robot Sepakbola Indonesia Beroda (KRSBI Beroda) diadakan untuk meningkatkan keilmuan dan kreatifitas mahasiswa di bidang robotika [1]. Di dalam kontes ini, mahasiswa dituntut untuk bisa mengembangkan robot yang dapat bergerak secara *autonomous*, baik dalam memposisikan robot didalam lapangan, mencari bola, mencari gawang, mengoper bola, menahan bola, menendang ke gawang lawan, dan sebagainya. Agar kemampuan tersebut dapat tercapai, dibutuhkan beberapasesor yang dapat dipasang pada robot agar dapat mendeteksi kondisi lingkungan yang ada disekitarnya, salah satunya yaitu sensor kompas.

Penggunaan sensor kompas pada robot dapat berguna untuk membedakan arah gawang tim lawan

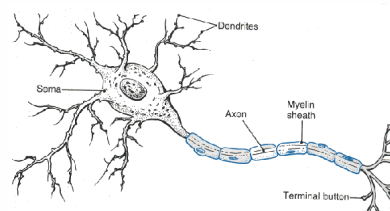
dan arah gawang tim sendiri. Selain itu, kita dapat membuat robot agar dapat mengetahui lokasi robot dilapangan secara akurat dengan menggunakan sensor kompas yang dikombinasikan dengan sensor lain, seperti sensor kamera atau sensor putaran roda robot. Masih banyak kegunaan sensor kompas yang dapat dimanfaatkan untuk meningkatkan kemampuan robot saat bertanding.

Sensor kompas yang digunakan pada tim ENSPARTAN dari Universitas Brawijaya adalah HMC5983. HMC5983 memiliki sensitivitas yang tinggi, namun sayangnya keakurasian derajat putaran sensor terbilang cukup kecil pada beberapa kondisi. Oleh karena itu, tim ENSPARTAN mencoba mengimplementasikan Jaringan Saraf Tiruan untuk meningkatkan akurasi sensor kompas HMC5983 pada robot KRSBI Beroda.

II. TINJAUAN PUSTAKA

A. Jaringan Saraf Tiruan(JST)

Otak manusia terdiri dari jutaan unit yang saling berhubungan, yang dikenal secara individual sebagai neuron biologis[2]. Pada Gambar 2.1, dapat dilihat bahwa setiap neuron terdiri dari sel yang terhubung dengan beberapa dendrit (*input*) dan satu akson (*output*). Akson terhubung dengan banyak neuron lainnya melalui titik koneksi yang disebut sinapsis. Sinapsis menghasilkan reaksi kimia sebagai respons terhadap *input*. Neuron biologis mengirim sinyal jika penjumlahan reaksi sinapsis cukup besar. Otak adalah jaringan kompleks neuron sensorik dan motorik yang mendukung manusia melakukan kemampuan untuk mengingat, berpikir, belajar, dan bernalar.



Gambar 2.1 Struktur neuron otak manusia

Jaringan Saraf Tiruan (JST) mencoba untuk meniru bagian biologis neuron[3]. McCulloch dan Pitts (1943) mengusulkan model sederhana dari sebuah neuron, dan Hebb (1949) menggambarkan teknik yang dikenal sebagai *Hebbianlearning*. Rosenblatt (1961), merancang satu lapisan neuron, yang disebut *Perceptron*, yang digunakan untuk pengenalan pola optik.

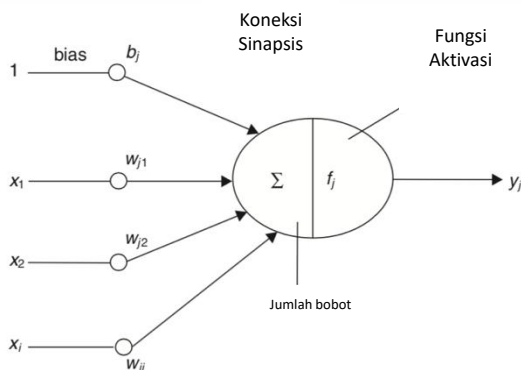
Jaringan Saraf Tiruan memiliki beberapa keuntungan untuk sistem kontrol cerdas:

- 1) JST belajar dari pengalaman daripada bergantung pada pilihan yang konstan.
- 2) JST memiliki kemampuan untuk menggeneralisasi dari data pelatihan yang diberikan ke data yang tidak terlihat.
- 3) JST memiliki proses yang cepat, dan dapat diimplementasikan secara real-time.

B. Operasi Neuron Buatan Tunggal

Model dasar dari neuron buatan tunggal terdiri dari penjumlahan bobot dan aktivasi (atau transfer) fungsi [3]. Gambar 2.2 menunjukkan sebuah neuron di lapisan ke- j , dimana :

- $x_1 \dots x_i$ adalah masukan
- $w_{j1} \dots w_{ji}$ adalah bobot
- b_j adalah bias
- f_j adalah fungsi aktivasi
- y_j adalah keluaran



Gambar 2.2 Model dasar dari neuron tiruan tunggal

Oleh karena itu jumlah bobot (s_j) adalah

$$s_j(t) = \sum_{i=1}^N w_{ji}x_i(t) + b_j \quad (2.1)$$

Persamaan (2.1) dapat ditulis dalam bentuk matrix dengan :

$$s_j(t) = \mathbf{W}_j \mathbf{x} + b_j \quad (2.2)$$

Fungsi aktivasi $f(s)$ (dimana s adalah jumlah bobot) dapat dibentuk dengan berbagai format, beberapa di antaranya bahwa bias b_j dalam Persamaan (2.1) dan (2.2) akan menggerakkan kurva sepanjang sumbu s , yaitu secara efektif pengaturan *threshold* di mana neuron mengirim sinyal. Jadi dalam kasus fungsi yang sulit dibatasi, jika $b_j = 0$, neuron akan

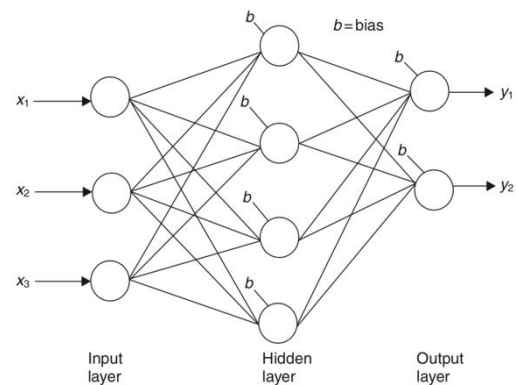
mengirim sinyal ketika $s_j(t)$ berubah dari negatif menjadi positif.

Fungsi aktivasi sigmoid sangat populer untuk aplikasi jaringan saraf karena dapat terdiferensiasi dan monotonik, keduanya merupakan syarat untuk algoritma *back-propagation*. Persamaan untuk fungsi sigmoid adalah :

$$f(s) = \frac{1}{1+e^{-s_j}} \quad (2.3)$$

JST adalah jaringan neuron tunggal yang disatukan oleh koneksi sinaptik. Gambar 2.3 menunjukkan jaringan syaraf *feedforward* tiga lapisan.

Jaringan feedforward yang ditunjukkan pada Gambar 2.3 terdiri dari tiga lapisan masukan (*input layer*) neuron, dua lapisan keluaran (*output layer*) neuron dan empat lapisan menengah neuron, yang disebut lapisan tersembunyi (*hidden layer*). Perhatikan bahwa semua neuron di lapisan tertentu sepenuhnya terhubung ke semua neuron di lapisan berikutnya. Ini umumnya disebut jaringan *multilayer* yang sepenuhnya terhubung, dan tidak ada batasan pada jumlah neuron di setiap lapisan, termasuk lapisan tersembunyi.



Gambar 2.3 Bentuk JST *feedforward* 3 lapisan

C. Sensor Kompas HMC5983

HMC5983 adalah sebuah sirkuit terpadu *magnetometer* tiga-sumbu[4]. HMC5983 mencakup sensor magnetoresistif seri-tinggi HMC118X serta ASIC yang mengandung amplifikator, *driver* otomatis *degaussing*, *offset cancellation*, dan 12-bit ADC yang memungkinkan akurasi kompas 1° hingga 2° . Bus serial I²C atau SPI memungkinkan untuk antarmuka yang mudah. Modul *multi-chip* yang bentuknya seperti pada Gambar 2.4 ini dirancang untuk mendeteksi medan magnet rendah untuk aplikasi seperti navigasi otomotif, personal navigasi, deteksi kendaraan, penunjuk, dan sebagainya.

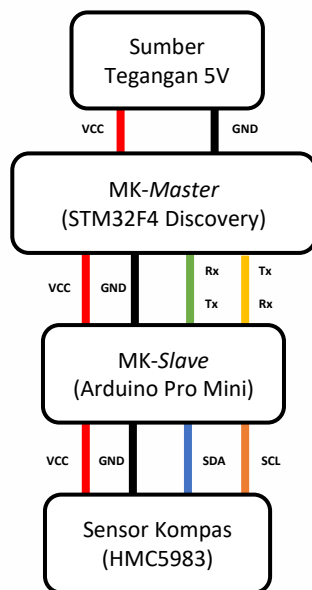


Gambar 2.4 Sensor kompas HMC5983 tampak depan dan belakang

III. PERANCANGAN SISTEM

A. Perancangan Perangkat Sistem

Perancangan perangkat elektrik pada penelitian ini dapat digambarkan pada blok diagram sebagai berikut.



Gambar 3.1 Diagram blok rangkaian elektrik

Pada perancangan perangkat elektrik, terdapat empat bagian utama yang dapat dilihat pada Gambar 3.1, yaitu sumber tegangan, mikrokontroler *master* (MK-Master), mikrokontroler *slave* (MK-Slave), dan sensor kompas HMC5983 itu sendiri. Sumber tegangan berfungsi untuk menyuplai semua rangkaian dengan tegangan 5 volt yang terhubung ke sumber tegangan, dimana sumber tegangan terhubung langsung dengan MK-Master. MK-Master pada penelitian ini menggunakan STM32F4 Discovery yang berfungsi sebagai tempat proses JST serta proses kalibrasi JST akan dilakukan. Sensor kompas yang disuplai dengan MK-Slave (yang dalam penelitian ini menggunakan Arduino Pro Mini), akan mengirim data mentah sensor (*raw data*) ke MK-Slave melalui komunikasi I²C untuk dibaca dan diolah dalam bentuk satuan derajat. Hasil dari proses pengolahan data sensor pada MK-Slave akan berkisar antara 1 derajat sampai dengan 360 derajat.

MK-Master menyuplai MK-Slave serta data yang telah diolah pada MK-Slave akan dikirim melalui komunikasi USART ke MK-Master untuk dilakukan proses JST. Fungsi adanya MK-Slave disini ialah agar proses pembacaan sensor dapat dilakukan terpisah dengan proses JST, sehingga pemrograman dapat lebih mudah.

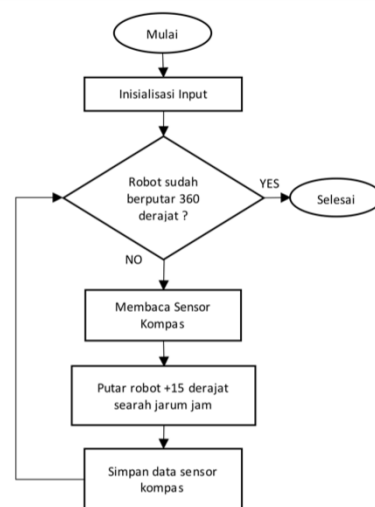
Adapun peletakan sensor kompas yang perlu dilakukan yaitu sensor kompas harus sejajar dengan tanah untuk mengurangi kesalahan akibat peletakan sensor yang miring. Selain itu, sensor juga lebih baik diletakkan pada bagian titik tengah robot serta sensor lebih baik diletakkan jauh dari sumber medan magnet yang dapat mengganggu performa sensor, contohnya rangkaian elektrik yang menghasilkan medan magnet, motor, aktuator, dan sebagainya.

B. Pembuatan Perangkat Lunak

Pada penelitian ini terdapat 3 bagian program, yaitu program kalibrasi, program pembacaan sensor kompas, dan program JST itu sendiri.

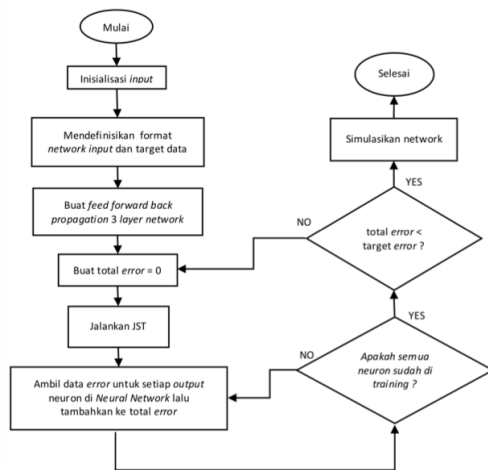
Sebelum dilakukan proses JST, robot perlu dikalibrasi terlebih dahulu untuk mengetahui kesalahan pengukuran dari sensor kompas, dimana kalibrasi hanya perlu dilakukan sekali saja. Setelah itu proses JST dapat dijalankan dengan mengambil acuan dari hasil kalibrasi sebelumnya. Untuk program pembacaan sensor kompas dilakukan pada MK-Slave, dimana hasil pembacaan dikirim langsung ke MK-Master melalui komunikasi USART.

Kalibrasi digunakan untuk menyimpan nilai derajat dari sensor kompas (*rawdata*) serta menyimpan nilai derajat sebenarnya untuk dijadikan target atau *setpoint*, yang selanjutnya akan digunakan pada proses JST. Diagram alur untuk kalibrasi JST dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram alur kalibrasi JST

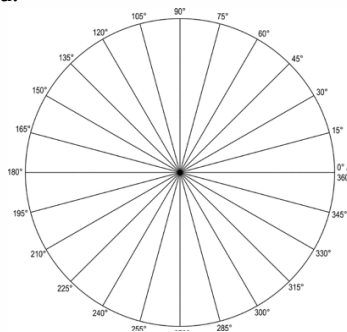
Pada alur proses JST yang dapat dilihat pada Gambar 3.3, inialisasi *input* yang akan dipakai terlebih dahulu, kemudian definisikan *input* dan target data yang telah diambil sebelumnya dari hasil kalibrasi. Setelah itu, buat *feedforwardbackpropagation 3 layernetwork* (yang terdiri dari *input layer*, *hidden layer*, *output layer*) dan mulailah perhitungan total *error* dari awal dengan cara menjadikan total *error*=0 terlebih dahulu lalu jalankan JST. Setelah proses JST dijalankan, bandingkan keluaran JST dengan target data, selisih antara keluaran JST dan target data adalah *error*. lalu jumlahkan semua *error* data yang diperoleh. Jika total *error* sudah lebih kecil dari yang diinginkan maka proses *training* selesai, akan tetapi jika *error* masih lebih besar dari yang diinginkan maka *training* akan dilakukan lagi. Setelah proses *training* selesai maka JST akan disimulasikan.



Gambar 3.3 Diagram alur proses JST

IV. HASIL DAN PEMBAHASAN

Pengujian dilakukan dengan meletakkan robot pada papan yang telah diberi penanda untuk mengukur sudut derajat putaran robot seperti pada Gambar 4.1. robot akan diposisikan di tengah lingkaran kemudian robot akan menghadap kesudut yang ingin diukur. Pengukuran akan dimulai dari sudut 0 derajat sampai dengan 360 derajat dengan penambahan sebesar 15 derajat dari pengukuran sebelumnya.

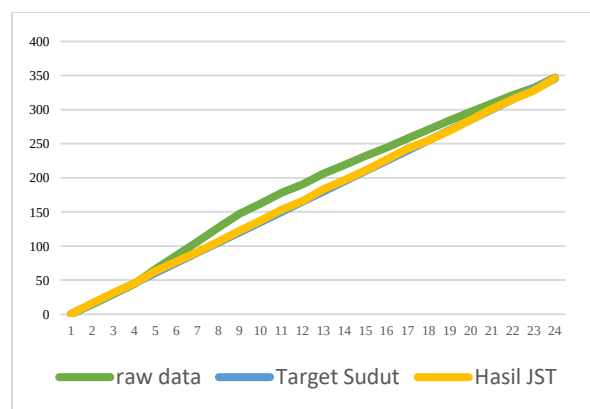


Gambar 4.1 Papan pengukuran sudut derajat putar robot

Setiap pengukuran tersebut akan dicatat hasil dari nilai sensor kompas yang berupa nilai mentah (*raw data*) serta akan dicatat pula nilai dari hasil proses JST. Hasil dari JST setiap pengukuran akan dibandingkan dengan sudut tujuan untuk mendapatkan persentase keakurasian pengukuran dengan JST.

Tabel 1 Hasil pengujian *raw data* dan pengujian JST

No	Sudut Tujuan (°)	Sudut raw (°)	Akurasi raw (%)	Sudut JST (°)	Akurasi JST (%)
1	0	0	100,00	0	100,00
2	15	14	93,33	16	93,33
3	30	29	96,67	31	96,67
4	45	44	97,78	45	100,00
5	60	66	90,00	63	95,00
6	75	86	85,33	77	97,33
7	90	106	82,22	91	98,89
8	105	127	79,05	106	99,05
9	120	147	77,50	122	98,33
10	135	162	80,00	137	98,52
11	150	178	81,33	153	98,00
12	165	190	84,85	166	99,39
13	180	206	85,56	183	98,33
14	195	219	87,69	197	98,97
15	210	232	89,52	211	99,52
16	225	244	91,56	227	99,11
17	240	258	92,50	243	98,75
18	255	271	93,73	255	100,00
19	270	284	94,81	269	99,63
20	285	297	95,79	285	100,00
21	300	309	97,00	301	99,67
22	315	321	98,10	315	100,00
23	330	332	99,39	328	99,39
24	345	347	99,42	346	99,71
Akurasi rata-rata		raw = 90,55 %		JST = 98,65 %	



Gambar 4.2 Grafik hasil pengujian sensor kompas

Hasil pengujian yang dapat dilihat pada Tabel 1 dan Gambar 4.2 bahwa akurasi rata-rata dari sensor kompas sebelum adanya JST (*raw data*) adalah sebesar 90,55%, sedangkan akurasi rata-rata sensor kompas setelah diberi JST adalah sebesar 98,65%. Akurasi paling kecil sebelum diberi JST adalah

sebesar 77,5% dengan kesalahan terbesarnya 28°, sedangkan akurasi paling kecil setelah diberi JST adalah sebesar 93,33% dengan kesalahan terbesarnya 3°.

Kedua hasil pengukuran tersebut dapat disimpulkan bahwa dengan penambahan JST sensor kompas dapat ditingkatkan keakurasiannya sebesar 98,43% dengan kesalahan terbesar yaitu 3°.

Peningkatan akurasi sensor kompas dapat digunakan untuk mengetahui sudut hadap robot yang selanjutnya digunakan untuk mengetahui perpindahan robot sehingga robot tersebut dapat mengetahui posisinya di lapangan secara akurat. Jika data sudut robot tidak akurat, maka posisi robot di lapangan menjadi tidak akurat yang berakibat robot akan salah mengarahkan bola ke gawang.

Adapun selama penelitian ini berlangsung, didapatkan kekurangan dan kelebihan dari penggunaan JST pada sensor kompas. Kekurangannya yaitu dikarenakan ada proses JST yang berlapis yang didalamnya terdapat banyak proses matematis, mengakibatkan proses *looping* pada program utama di mikrokontroler *master* menjadi lebih lambat. Akan tetapi, kelebihan dari penggunaan JST pada sensor kompas yaitu kesalahan sensor kompas dapat diperkecil serta proses kalibrasi tidak perlu dilakukan setiap saat.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan pada uraian yang telah dijabarkan sebelumnya serta berdasarkan hasil pengujian yang telah dilakukan, maka pada penelitian ini dapat ditarik beberapa kesimpulan sebagai berikut.

1. Dengan menggunakan JST, akurasi sensor kompas HMC5983 menjadi lebih besar dengan akurasi rata-rata sebesar 98,65% dengan kesalahan terbesarnya yaitu 3 derajat, dimana sebelum penambahan JST, kesalahan sensor kompas dapat mencapai 28 derajat.
2. Peningkatan akurasi sensor kompas dapat digunakan untuk mengetahui sudut hadap robot yang selanjutnya digunakan untuk mengetahui perpindahan robot sehingga robot tersebut dapat mengetahui posisinya di lapangan secara akurat.
3. Kekurangan dari penggunaan JST pada sensor kompas yaitu dikarenakan ada proses JST yang berlapis yang didalamnya terdapat banyak proses matematis, mengakibatkan proses *looping* pada program utama di mikrokontroler *master* menjadi lebih lambat. Akan tetapi, kelebihan dari penggunaan JST pada sensor kompas yaitu kesalahan sensor kompas dapat diperkecil serta proses kalibrasi tidak perlu dilakukan setiap saat.

B. Saran

Karena sensor kompas sangat sensitif terhadap medan magnet, maka peletakan sensor kompas lebih baik diletakkan pada bagian titik tengah robot serta sensor lebih baik diletakkan jauh dari sumber medan magnet yang dapat mengganggu performa sensor, contohnya rangkaian elektrik yang menghasilkan medan magnet, motor, aktuator, dan sebagainya. Selain itu, sensor kompas diletakkan sejajar dengan tanah untuk mengurangi kesalahan akibat peletakan sensor yang miring.

VI. REFERENSI

- [1] RISTEKDIKTI, "Buku Panduan Kontes Robot Sepakbola Indonesia Divisi Beroda (KRSBI Beroda)," Jakarta, 2017
- [2] F. Pakaja, A. Naba dan Purwanto, "Peramalan Penjualan Mobil Menggunakan Jaringan Syaraf Tiruan dan Certainty Factor," Universitas Brawijaya, Malang, 2012
- [3] R. S. Burns, "Advanced Control Engineering," Oxford, 2001
- [4] Honeywell, "3-Axis Digital Compass IC HMC5983," Honeywell International Inc., Plymouth, 2012

Sistem Kerja Robot Penjaga Gawang (*Keeper*) Divisi KRSBI Beroda Tim ROTA

Dedy Junaidy, Alvin Prayogo, Agus Halim
Universitas Tarumanagara
Jl. Letjen S. Parman No. 1, Jakarta Barat
Email: dedyjunaidy30@yahoo.co.id

Abstrak – Kontes Robot Sepakbola (KRSBI) Beroda merupakan salah satu cabang perlombaan yang diadakan dengan tujuan untuk meningkatkan keilmuan dan kreativitas mahasiswa dalam bidang robotika. Robot yang dibuat dalam bidang ini memiliki tujuan untuk dapat bermain bola layaknya pada pertandingan sepakbola pada umumnya. Hal yang menjadi keunikan pada robot ini yaitu sistem pergerakannya yang menggunakan roda. Pada divisi ini terdapat dua peran yang tersedia untuk robot yaitu sebagai penyerang dan penjaga gawang (*keeper*). Makalah ini memberikan penjelasan mengenai sistem kerja dari robot penjaga gawang dari tim ROTA. Mikrokontroler yang digunakan pada robot adalah Raspberry Pi dan Arduino Mega 2560. Sensor yang digunakan berupa kamera, sensor ultrasonik, *rotary encoder*, dan sensor *gyroscope*. Sistem kendali PID diterapkan pada robot untuk memperbaiki *error* selama pergerakan pengejaran bola. Proses yang dilakukan oleh robot ini yaitu mengambil data dari sensor yang ada untuk memposisikan dirinya pada gawang dan bergerak untuk menghadang bola yang ditendang ke arah gawang.

Kata kunci: KRSBI Beroda, sistem kerja, penjaga gawang, *keeper*.

I. PENDAHULUAN

Kontes Robot Sepakbola Indonesia (KRSBI) Beroda merupakan suatu ajang perlombaan dalam bidang robotika yang dapat meningkatkan keilmuan dan kreativitas mahasiswa untuk membangun sebuah robot pemain sepakbola dengan menggunakan penggerak berupa roda. Pada divisi ini, terdapat dua peran robot yang didesain: robot penyerang dan robot penjaga gawang (*keeper*).

Robot penjaga gawang adalah robot yang berperan sebagai penjaga gawang layaknya dalam permainan sepakbola pada umumnya. Robot ini memiliki peranan penting dalam pertahanan. Bila tim memiliki pertahanan yang baik, kekalahan dari suatu tim dapat dihindari. Oleh karena itu, pengembangan robot penjaga gawang merupakan salah satu hal yang penting dalam kesuksesan sebuah tim. Dengan banyaknya strategi yang dapat diterapkan untuk mencetak gol, hal ini menjadi tantangan yang cukup menarik untuk dilakukan.

Pada makalah ini, dijelaskan sistem kerja robot penjaga gawang yang digunakan oleh tim ROTA dengan tujuan pengembangan robot penjaga gawang yang lebih baik pada divisi KRSBI Beroda.

II. METODE

Kegiatan robot *keeper* pada ROTA berupa pergerakan robot untuk menghadang bola yang ditendang ke gawang. Untuk dapat melakukan hal tersebut, robot akan membaca posisi bola dan membandingkannya dengan posisi badan. Robot akan membaca letak posisi bola dengan kamera. Ketika bola ditemukan, robot akan berusaha memposisikan dirinya agar selurus dengan bola sehingga bola dapat ditangkap. Daerah pergerakan robot dibatasi pada gawang yang dijaga. Robot diposisikan sedemikian rupa sehingga memiliki jarak tertentu dari bagian belakang gawang dan batasan pergerakan secara horizontal berdasarkan tiang gawang.

Untuk dapat melaksanakan tugasnya sebagai *keeper*, robot KRSBI Beroda menggabungkan beberapa komponen utama, yaitu:

- Raspberry Pi sebagai mikrokontroler utama dan pemroses gambar dari kamera.
- Arduino Mega 2560 sebagai mikrokontroler subbagian yang mengatur pergerakan robot dalam mengejar bola.
- Kamera sebagai mata dari robot untuk mencari bola.
- Sensor ultrasonik untuk membaca jarak vertikal robot terhadap gawang.
- Rotary encoder* untuk membaca jarak horizontal robot terhadap gawang.
- Sensor *gyroscope* untuk membaca kemiringan badan robot pada tiap gerakan pengejaran bola.
- PID merupakan sebuah algoritma pengendali untuk memperbaiki posisi robot pada saat pergerakannya dalam mengejar bola.

Masing-masing komponen akan dijelaskan pada subbab berikutnya.

A. Raspberry Pi

Raspberry Pi adalah komputer berukuran kecil yang memiliki ukuran seperti kartu *ATM* yang dapat dihubungkan ke TV atau layar komputer dan *keyboard* (Gambar 1). Mikro ini memiliki 40 pin *input/output*, 4 *USB 2.0 port*, *HDMI* dan *DSI (Display Serial Interface)*, 3.5 mm *audio jack*.

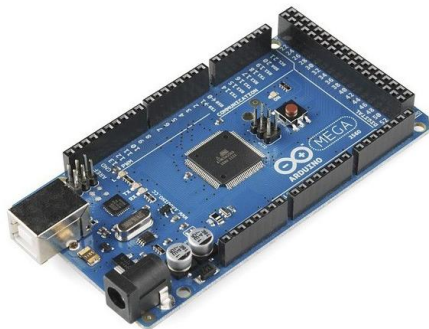


Gambar 1. Raspberry pi

Pada robot, mikro ini berperan sebagai kontroller utama, yang berfungsi untuk membaca dan mengolah data citra dari kamera dan sensor, dan mengirim perintah kepada Arduino Mega dengan komunikasi serial.

B. Arduino Mega

Arduino Mega 2560 adalah mikrokontroler yang menggunakan ATmega2560 (Gambar 2). Mikro ini memiliki 54 pin *input/ output* digital (15 pin di antaranya dapat digunakan untuk *output* PWM), 16 pin *input* analog, 4 pin UART(*Universal Asynchronous Receiver-Transmitter*), Kristal 16 MHz, koneksi USB, dan tombol *reset*(1).



Gambar 2. Arduino Mega 2560

Pada robot, mikro ini berfungsi untuk melakukan pembacaan sensor yang ada, memproses data yang didapatkan dan mengeluarkannya dalam bentuk pergerakan robot.

C. Kamera

Kamera digunakan untuk memperoleh data citra yang akan diolah dan diproses oleh Raspberry Pi (Gambar 3). Pada robot *keeper*, kamera berfungsi untuk mendeteksi posisi bola. Hal ini bertujuan agar robot dapat menjaga gawang dengan baik.



Gambar 3. Kamera Logitech 525

D. Sensor Ultrasonik

Sensor ultrasonik digunakan untuk mengukur jarak dengan menggunakan gelombang suara (Gambar 4). Sensor ini akan mengeluarkan gelombang suara dengan frekuensi tertentu dan menerimanya kembali setelah terpantul dengan suatu objek. HC-SR04 merupakan modul sensor ultrasonic yang digunakan pada robot. Sensor ini memiliki jangkauan dari 2cm hingga 400cm, dengan akurasi hingga 3mm. Modul terdiri dari pemancar dan penerima ultrasonik serta sirkuit pengendali(2).



Gambar 4. HC-SR04

HC-SR04 dipasang pada bagian belakang robot dan digunakan untuk menentukan jarak robot dengan bagian dalam dari gawang. Penentuan jarak ini bertujuan agar robot selalu berada pada posisi yang stabil saat menjaga gawang. Artinya, robot tidak akan terlalu maju sehingga keluar dari gawang dan terlalu mundur sehingga masuk ke dalam gawang.

E. Rotary Encoder

Rotary Encoder merupakan sensor yang mendekeksi posisi dan kecepatan dengan mengubah perubahan yang didapatkan dari putaran menjadi sinyal elektronik dan memprosesnya(3). Enkoder dapat dilihat pada Gambar 5.



Gambar 5. Rotary Enkoder

Enkoder digunakan agar robot mengetahui jaraknya dengan tiang gawang. Hal ini bertujuan agar robot tidak mengejar bola terlalu jauh dan menabrak tiang.

F. Sensor Gyroscope

Sensor *Gyroscope* merupakan sensor untuk mengukur pergerakan rotational(4). Salah satu modul sensor yang memiliki kemampuan ini adalah MPU-9150 (Gambar 6). Sensor ini merupakan kombinasi yang memiliki 3-axis *gyroscope*, 3-axis *accelerometer*, dan sensor AK8975 yang memiliki 3-

axis digital compass(5). Modul ini dapat mengukur *gyro* dalam tiga derajat kebebasan yaitu pada sumbu X, Y, dan Z. Dalam *gyroscope* arah putaran pada tiap sumbu dikenal dengan *pitch*, *roll*, dan *yaw*.



Gambar 6. MPU-9150

Sumbu yang diukur adalah sumbu Z atau *yaw*. Pada tiap pergerakannya dalam mengejar bola, robot akan terus mengakumulasi *error* dalam menghadap ke depan. *Error* tersebut akan menjadi masalah karena arah hadap robot terhadap bola akan terganggu. Modul ini dipakai agar robot dapat mengetahui kemiringan posisinya pada tiap gerakan yang dilakukan saat mengejar bola. Dengan mengetahui posisi ini, *error* yang terjadi dapat diperbaiki.

G. PID

Proportional-Integral-Derivative(PID)

merupakan algoritma kendali yang umumnya digunakan di industri dan telah diterima secara universal dan kendali industrial. Popularitas PID meningkat dikarenakan kemampuannya yang baik dalam berbagai kondisi dan kesederhanaannya, yang memungkinkan praktisi untuk menggunakannya. Konsep dari PID adalah pembacaan sensor, perhitungan *output* yang sesuai berdasarkan respon proporsional, integral, dan derivatif kemudian menggunakan nilai ketiga komponen tersebut untuk mengeluarkan *output* yang sesuai(6).

Penerapan algoritma ini pada robot memperbaiki posisi robot pada tiap pergerakannya sehingga robot tetap dapat berada pada jarak yang sesuai dari gawang dan meluruskan diri setiap adanya *error* yang timbul saat bergerak.

III. HASIL DAN DISKUSI

Berdasarkan komponen yang telah disebutkan, dilakukan percobaan untuk mendukung kerja yang diharapkan dari sebuah robot penjaga gawang. Percobaan yang dilakukan yaitu:

A. Deteksi Bola

Dengan digunakan kamera dan Raspberry Pi, pengolahan citra dilakukan dengan Open CV yang beroperasi pada bahasa pemrograman C++ dan Python. Proses pendeteksian bola diawali dengan penangkapan *image* dengan format matriks (Mat) RGB (*Red Green Blue*), lalu Mat RGB diubah

menjadi Mat HSV (*Hue Saturation Value*) (Gambar 7).



Gambar 7. Deteksi Bola

Selanjutnya, Mat HSV difilter dengan cara *threshold* dengan nilai *min* dan *max*. Jika nilai HSV yang di luar batas *min* dan *max* akan diubah menjadi warna hitam (*LOW/0*), dan yang di dalam batas menjadi warna putih (*HIGH/1*). Dengan mengetahui nilai batas untuk warna bola (*Orange*), Robot dapat mengenali bola berwarna *orange* yang berupa warna putih di matriks.

B. Pembacaan Jarak ke Bagian Belakang Gawang (Vertikal)

Dengan memanfaatkan HC-SR04, robot dapat mengetahui seberapa jauh posisinya dengan gawang bagian dalam. Data yang didapatkan dari HC-SR04 harus memiliki kepresisian yang tinggi, sehingga robot menjadi sensitif terhadap perubahan jarak vertikal. Rentang jarak yang diberikan akan membuat robot dapat bergerak secara konstan terhadap bagian belakang gawang.

Tabel 1. Hasil pembacaan dengan HC-SR04

Rentang Jarak yang Diiukur(cm)	Hasil yang Didapatkan
<=40	Pembacaan presisi
<=45	Pembacaan presisi
<=50	Pembacaan presisi
<=55	Pembacaan presisi
<=60	Pembacaan presisi
<=65	Pembacaan presisi
<=70	Terdapat pembacaan yang salah

Berdasarkan Tabel 1, disimpulkan bahwa jarak maksimum dimana pembacaan sensor masih cukup presisi adalah 65 cm ke bawah.

C. Pembacaan Jarak ke Tiang Gawang (Horizontal)

Posisi awal robot adalah di bagian tengah dari gawang. Percobaan dilakukan dengan memberikan Batasan pergerakan robot dalam mengejar bola tanpa menabrak tiang. Dengan memberikan batas yang pas, robot akan berhenti pada posisi maksimal untuk mempertahankan bola tanpa menabrak tiang. Tabel 2 menunjukkan Batasan yang diberikan kepada robot dan hasilnya.

Tabel 2. Batasan pada sumbu horizontal

Batasan yang diberikan(cm)	Hasil yang Didapatkan
70	Menabrak tiang
65	Menabrak tiang
60	Menabrak tiang
55	Tidak menabrak tiang
50	Tidak menabrak tiang namun bola dapat memasuki celah antara robot dan tiang gawang

Berdasarkan Tabel 2, disimpulkan bahwa batasan pergerakan maksimal robot dalam mengejar bola tanpa menabrak tiang dan dalam jangkauan penjagaan bola adalah 55 cm dan yang diaplikasikan ke robot adalah 58 cm.

IV. KESIMPULAN

Berdasarkan percobaan yang telah dilakukan, robot penjaga gawang untuk divisi KRSBI Beroda dapat dibentuk dan melakukan fungsinya dengan cukup baik. Dengan pengolahan citra yang dilakukan pada Raspberry Pi, robot dapat mengenali bola berdasarkan warnanya dan bergerak untuk mengejar bola serta memposisikan dirinya searah dengan bola. Robot memiliki jarak yang konstan terhadap bagian belakang gawang dengan memberikan Batasan sebesar 65 cm berdasarkan pengukuran sensor ultrasonik sehingga robot tidak akan keluar dari gawang maupun bergerak masuk ke gawang. Robot tidak menabrak tiang gawang dalam proses pengejaran bola dengan ditetapkannya jarak pergerakan secara horizontal dari

tengah gawang sebesar 58 cm berdasarkan pembacaan enkoder.

V. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak Universitas Tarumanagara yang memberikan dukungan dalam penelitian ini, para pembimbing yang memberikan banyak masukan dalam terwujudnya penelitian ini, dan kepada seluruh tim Robotik yang telah membantu selama proses penelitian yang telah dilakukan.

VI. DAFTAR PUSTAKA

- [1] Arduino. ARDUINO MEGA 2560 REV3 [Internet]. Available from: <https://store.arduino.cc/arduino-mega-2560-rev3>
- [2] Indoware. Ultrasonic Ranging Module HC - SR04. Datasheet [Internet]. 2013;1–4. Available from: <http://www.micropik.com/PDF/HCSR04.pdf>
- [3] Principles O. Technical Explanation for Rotary Encoders.
- [4] Further G. How a Gyro Works. 2017;1–5.
- [5] Incorporated I. MPU-9150 Specification Product 4.3 Revision. 2013;1(408):1–50. Available from: http://www.invensense.com/mems/gyro/documents/RM-MPU-9150A-00v4_2.pdf
- [6] Instruments N. PID Theory Explained. Available from: <http://www.ni.com/white-paper/3782/en/>
- [7] Arasada B, Suprianto B. Aplikasi Sensor Ultrasonik Untuk Deteksi Posisi Jarak Pada Ruang Menggunakan Arduino Uno. Tek Elektro. 2017;6(2):1–8.
- [8] Rochamnto. Implementasi Robot Three Omni-Directional Menggunakan Kontroler Pid Pada Robot Kontes Robot Abu Indonesia (Krai). 2014;
- [9] https://en.wikipedia.org/wiki/Raspberry_Pi

Perancangan Sistem Gerak pada Robot Kiper R2C-WARRIOR

Gidion Siwi Nugroho, Dani Septya Ardhian

Fakultas Teknik Elektronika dan Komputer

Universitas Kristen Satya Wacana

Jln. Diponegoro 52-60, Salatiga 50711 Jawa Tengah

Telp.: (0298)326-421212 mail: ftek@adm.uksw.edu, gidionsiwi@gmail.com

Abstrak -- Semakin hari teknologi semakin berkembang. Untuk membantu aktivitas manusia dibuatlah robot, alat mekanik yang didesain dapat melakukan tugas fisik tertentu. Di Indonesia, robot sudah mulai diperlombakan. Salah satunya adalah Kontes Robot Sepak Bola Beroda (KRSBI Beroda). Robot kiper R2C-WARRIOR menggunakan sistem pergerakan *differential drive* dengan empat buah roda. Dengan roda biasa sistem pergerakan tersebut tidak akan bisa bergerak kesegala arah, maka *mecanum wheel* dipasang agar dapat bergerak bebas kesegala arah. Untuk mengatur pergerakan robot, diberikan kecepatan masing-masing motor dengan menggunakan persamaan *invers kinematic*. Persamaan tersebut akan memberikan kecepatan masing-masing motor dengan input V_x dan V_y . Selanjutnya kecepatan motor dijadikan *set point* pada persamaan PID untuk mengontrol motor DC dengan *feedback* nilai *encoder* motor.

Dalam pengujian, robot didesain agar memiliki kecepatan 0.5 m/s kesemua arah tanpa mengubah orientasi robot. Lalu untuk pengukuran sudut dipasang busur derajat dan benang untuk mengukur arah pergerakan robot. Pada hasil percobaan menunjukkan bahwa robot memiliki ralat sebesar 1 derajat dikarenakan terjadinya slip antara roda dan lapangan.

Kata kunci : Kontes Robot Indonesia, Mobile Robot, Mecanum Wheel, Invers Kinematic, Kontrol PID

I. PENDAHULUAN

Kontes Robot Indonesia (KRI) adalah kontes robot nasional yang diselenggarakan tiap tahun. Ada lima divisi yang diperlombakan pada KRI yaitu (KRSBI - *Humanoid*), (KRSBI - Beroda), (KRPAL Berkaki), (KRAI), dan (KRSTI).[1] Setiap tahun selalu terjadi perubahan peraturan pada KRI. Pada KRI 2018, divisi KRSBI – Beroda merupakan divisi yang memiliki tantangan tersendiri bagi tim untuk membuat sistem baru, baik secara algoritma, citra, komunikasi, pergerakan robot, dan mekanik.[2] Pada bagian pergerakan robot tentunya sebagai divisi KRSBI, robot diharuskan untuk memiliki pergerakan yang cepat dan efisien.

Pada robot kiper, pergerakan robot sangatlah penting agar dapat menghalau bola yang ditendang robot lawan. Gerakan dasar robot kiper adalah kanan dan kiri, namun tidak menuntut kemungkinan kalau robot kiper diharuskan untuk

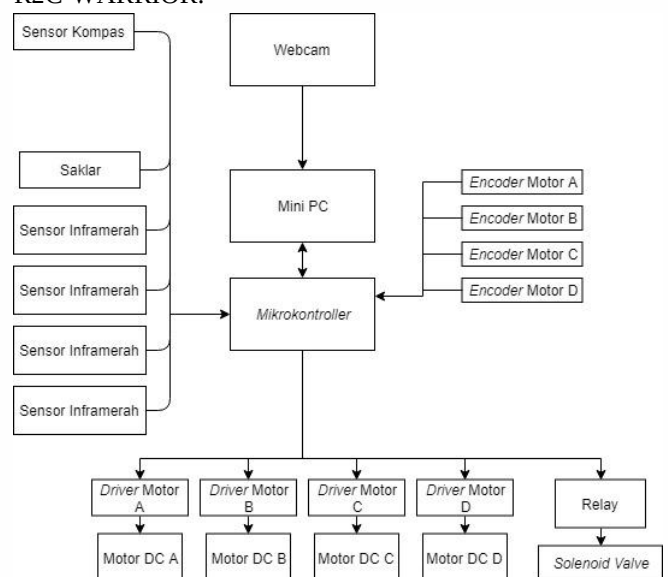
dapat bergerak dinamis ke semua arah yang biasa disebut sistem gerak *holonomic*.

Gerakan dasar kanan dan kiri akan mudah jika menggunakan sistem gerak *differential drive*. Agar gerak lebih dinamis kesegala arah menggunakan *mecanum wheel* adalah solusi yang tepat. Sehingga pemasangan *mecanum wheel* pada sistem gerak *differential drive* dapat memungkinkan robot bergerak dinamis kesegala arah, namun tetap mementingkan gerakan dasar robot kiper yaitu gerak ke kanan dan kiri.

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 pengoperasian sistem gerak *holonomic mecanum drive*. 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.

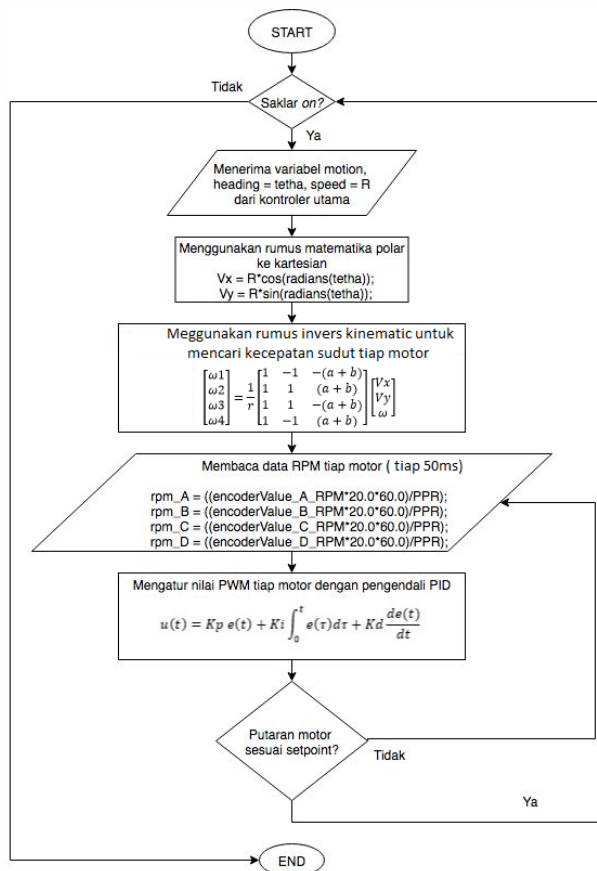
II. METODE

Diagram blok keseluruhan sistem robot kiper R2C-WARRIOR:



Gambar 1. Diagram blok sistem robot kiper

Cara kerja sistem gerak *holonomic mecanum drive* pada robot kiper R2C-WARRIOR ini dapat dijelaskan melalui diagram alir pada Gambar 1.



Gambar 2. Flowchart sistem gerak *holonomic mecanum drive*

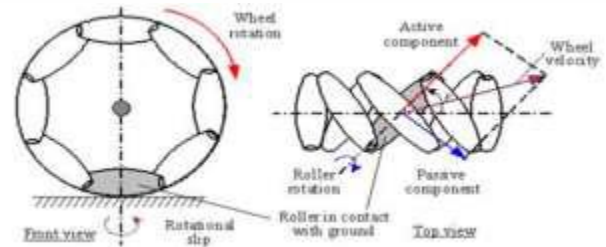
Dari diagram alir diatas, sistem gerak *holonomic mecanum* bergantung pada tiga *variable* yaitu *motion*, *heading*, dan *speed*. *Motion* adalah bentuk gerakan dari robot, *heading* adalah arah untuk menentukan *motion* (180 derajat hingga -180 derajat), sedangkan *speed* adalah *variable* untuk mengatur kecepatan *motion*. Dengan menggunakan rumus *invers kinematic* dari roda *mecanum* dan ditambah dengan menggunakan rumus matematika polar ke kartesian maka dari *variable* tersebut di dapatkan *setpoint* untuk tiap-tiap motor dengan satuan RPM (*Rotation Per Minute*). Setelah *setpoint* tiap – tiap motor diketahui langkah selanjutnya yaitu membaca RPM dari *encoder* pada masing – masing motor. Untuk memastikan roda berputar sesuai *setpoint* digunakan pengendalian PID dengan unpan balik berupa pembacaan *encoder* pada tiap-tiap motor.

III. Teori yang Digunakan

A. Roda Mecanum

Roda *mecanum* menggunakan desain roda yang dikelilingi oleh *roller* dengan sudut 45°. Gambar 3 menunjukkan bahwa sebuah roda

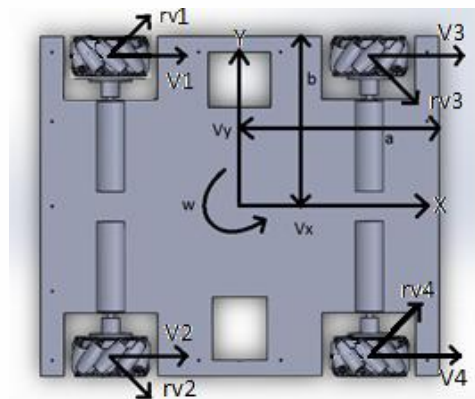
mecanum memiliki 3 DOF (*degree of freedom*). DOF pertama adalah putaran roda, DOF kedua adalah rotasi *roller*, sedangkan DOF ketiga adalah slip rotasi terhadap sumbu vertikal yang melewati titik kontak dengan permukaan.[3]



Gambar 3. Degree of freedom roda *mecanum*[6]

B. Invers Kinematic

Persamaan *kinematic* mendefinisikan beberapa bagian dari pergerakan robot, diantaranya kecepatan V_x , V_y dan ω menjadi persamaan kecepatan sudut masing-masing roda.



Gambar 4. Konfigurasi roda dan penamaan vektor

Konfigurasi parameter dan sistem kecepatan didefinisikan sebagai berikut:[4]

- X, Y adalah sumbu koordinat kartesius.
- V_x, V_y [m/s] adalah kecepatan linier robot.
- v_i adalah kecepatan linier tiap roda terhadap sumbu X .
- ω [rad/s] adalah kecepatan sudut robot.
- ω_i [rad/s] adalah kecepatan sudut pada motor i .
- r_{vi} adalah kecepatan pasif *roller* pada roda i .
- a adalah jarak roda depan dengan titik pusat robot.
- b adalah jarak roda samping dengan titik pusat robot.
- r adalah radius roda *mecanum*

Untuk menentukan kecepatan sudut masing-masing motor dalam satuan *radian per second*, digunakan persamaan *invers kinematic* seperti ditunjukkan pada Persamaan 1:[5]

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & -1 & -(a+b) \\ 1 & 1 & (a+b) \\ 1 & 1 & -(a+b) \\ 1 & -1 & (a+b) \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} \quad (1)$$

Dengan membuat tetap orientasi robot, maka didapatkan persamaan kecepatan sudut tiap-tiap motor sebagai berikut:

$$\omega_1 = \frac{V_x - V_y}{r} \quad (2)$$

$$\omega_2 = \frac{V_x + V_y}{r} \quad (3)$$

$$\omega_3 = \frac{V_x + V_y}{r} \quad (4)$$

$$\omega_4 = \frac{V_x - V_y}{r} \quad (5)$$

C. Pengendali PID

Pengendali PID (*Proportional–Integral–Derivative*) controller merupakan pengendali dengan mekanisme umpan balik yang biasanya dipakai untuk menentukan ketepatan suatu sistem. Sebuah pengendali PID secara *continue* menghitung nilai kesalahan berupa selisih antara *setpoint* yang diinginkan dan *variable* proses terukur. Pengendali PID mencoba untuk meminimalkan nilai kesalahan tiap satuan waktu dengan penyetelan *variable* kontrol seperti kecepatan menjadi nilai baru.[6]

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (6)$$

Dimana :

K_p : Gain proportional.

K_i : Gain integral.

K_d : Gain derivative.

e : Error = $Y_{sp} - Y_m$

Y_{sp} : Setpoint.

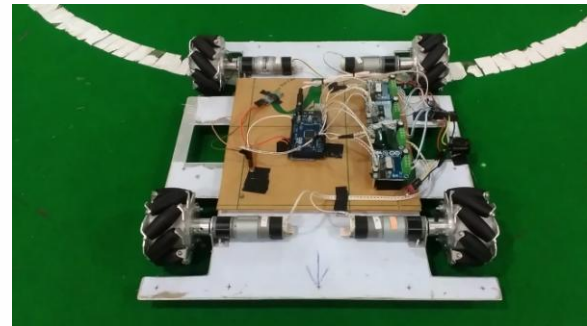
Y_m : Variabel proses.

T : Waktu.

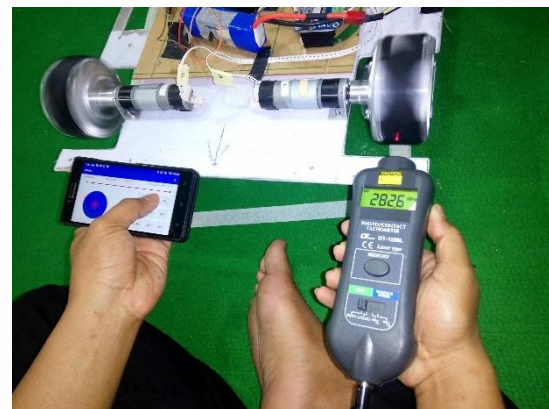
τ : Integrasi, nilainya diambil dari waktu nol sampai t .

IV. HASIL EKSPERIMEN

Berikut hasil mekanik dasar robot yang sudah dibuat:



Gambar 5. Hasil robot tampak depan



Gambar 6. Pengukuran kecepatan motor hasil perhitungan PID menggunakan alat tachometer

Pada Gambar 6 diuji kecepatan robot yang ditetapkan adalah 0.5 m/detik kesegala arah. Kecepatan robot yang dimasukkan adalah V_x dan V_y . Dengan menggunakan persamaan *Invers Kinematic*, didapatkan kecepatan masing-masing motor. Pada tabel 1 diperlihatkan hasil perhitungan *Invers Kinematic* pada tiap-tiap motor tanpa mengubah orientasi robot ($\omega = 0^\circ$) dan diasumsikan roller pada roda lincin dan roda tidak slip.

Tabel 1. Hasil perhitungan *invers kinematic* Robot

Sudut	V_x (m/s)	V_y (m/s)	Kecepatan Motor (rad/s)			
			ω_1	ω_2	ω_3	ω_4
0°	0.5	0	10	10	10	10
45°	1	1	0	40	40	0
90°	0	0.5	-10	10	10	-10
135°	1	1	-40	0	0	-40
180°	-0.5	0	-10	-10	-10	-10
225°	-1	-1	0	-40	-40	0
270°	0	-0.5	10	-10	-10	10
315°	1	-1	40	0	0	40

Dari tabel 1 didapatkan kecepatan masing-masing motor hasil dari perhitungan *invest*

kinematic. Nilai positif dan negatif menunjukkan arah putar dari roda *mecanum*.

Selanjutnya dilakukan pengujian arah gerak robot dengan reverensi sumbu X positif sebagai nilai 0 derajat. Pengujian dilakukan dengan memberikan nilai kecepatan pada tiap – tiap motor. Berikut adalah hasil pengukuran arah robot dengan media busur sebagai pengukur arah robot dari nilai reverensi.

Tabel 2. Hasil pengukuran sudut terhadap kecepatan motor

Sudut	Speed (m/s)	Hasil Pengukuran Busur	Ralat Busur
0°	0.5	1°	1°
45°	0.5	45°	1°
90°	0.5	91°	1°
135°	0.5	135°	1°
180°	0.5	181°	1°
225°	0.5	226°	1°
270°	0.5	271°	1°
315°	0.5	316°	1°

V. KESIMPULAN

Hasil pengujian arah gerak robot dengan nilai reverensi X positif sebagai nilai 0 derajat, dapat disimpulkan bahwa dengan memasang roda *mecanum* pada sistem gerak *differential drive* dapat menciptakan gerakan *holonomic*. Terdapat beberapa faktor seperti roda selip, kerasnya *roller* yang mengakibatkan gerakan robot melenceng dari sudut yang diharapkan.

VI. DAFTAR PUSTAKA

- [1] K. R. Indonesia, "Kontes Robot Indonesia Kri 2013," 2013.
- [2] Ristekdikti, "BUKU PANDUAN KONTES ROBOT SEPAKBOLA INDONESIA BERODA (KRSBI Beroda)," *Direktorat Kemahasiswaan Direktorat Jenderal Pembelajaran dan Kemahasiswaan Kementerian. Riset, Teknol. dan Pendidik. Tinggi Republik Indonesia* 2017.
- [3] P. F. Muir and C. P. Neuman, "Kinematic Modeling of Wheeled Mobile Robots," no. June, 1986.
- [4] Hendrayawan, V., Implementasi Invers Kinematics pada Sistem Pergerakan Mobile Robot Roda Mekanum, Tugas Akhir S-1, Universitas Brawijaya, Malang, 2014.
- [5] M. A. R. Arif Rahman Hakim, Sumardi, "KONTROL POSISI PADA SISTEM PERGERAKAN MOBILE ROBOT KINEMATIC," *Comput. Appl. (0975 – 8887)*, 2015.
- [6] F. Ribeiro, I. Moutinho, P. Silva, C. Fraga, and N. Pereira, "Three Omni-Directional Wheels Control on a Mobile Robot," *Control 2004*, no. September, 2004.

Sinkronisasi Pergerakan Robot Sepak Bola Beroda Menggunakan Metode PID dan *Odometry*

Achmad Yani¹, Koko Joni², Riza Alfita³

^{1,2,3}Universitas Trunojoyo Madura

Jl. Raya Telang, Kamal, Madura, Jawa Timur 69162

¹achmadyaniia3@gmail.com

Abstrak -- Penelitian ini bermaksud untuk sinkronisasi pergerakan robot sepak bola dengan menggunakan metode PID untuk mengatur kecepatan roda sehingga didapatkan respon kecepatan yang cepat mencapai set point serta setabil. Penentuan parameter kontroler PID menggunakan metode *hand tuning*. Dengan sistem umpan balik *odometry* pada robot sehingga diharapkan dapat menyingkronkan pergerakan robot sepak bola beroda. Penggunaan parameter kontroler PID yang tepat dapat membuat robot dapat bergerak sesuai dengan yang di harapkan. Selain itu sistem mekanik, elektronik, dan vision pada robot juga di perhatikan agar robot bisa bekerja dengan maksimal. Robot sepak bola beroda menggunakan motor dc untuk bergerak, solenoid untuk menendang bola, dan kamera untuk mendeteksi bola.

Kata kunci: Robot Sepak Bola Beroda, PID, *Odometry*.

I. PENDAHULUAN

Perkembangan teknologi robotika saat ini mengalami perkembangan yang sangat pesat. Perkembangan robotika sangat berpengaruh dan berguna terhadap bidang kedokteran, pendidikan, industri, pertanian, kesehatan, rumah tangga, dan alat bantu. Kontes Robot Indonesia (KRI) merupakan salah satu bukti perkembangan robotika di dibidang pendidikan yang diadakan oleh KEMENRISTEK DIKTI yang bertujuan untuk menumbuhkan, meningkatkan kreativitas, dan kepekaan mahasiswa perguruan tinggi dalam mengembangkan teknologi robotika. Kontes Robot Indonesia (KRI) terdiri dari 4 kategori yaitu Kontes Robot Abu Indonesia (KRAI), Kontes Robot Sepak Bola Indonesia (KRSBI), Kontes Robot Pemadam Api (KRPPI), Kontes Robot Seni Tari (KRSTI).

Robot Sepak Bola Beroda merupakan salah satu kategori dalam Kontes Robot Indonesia yakni KRSBI Beroda, pembuatan robot sepak bola beroda beracuan pada peraturan KRSBI beroda 2018. Terdiri dari peraturan pertandingan, dimensi lapangan pertandingan, jumlah robot, dan spesifikasi robot.

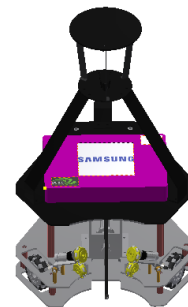
Sistem pergerakan robot sepak bola beroda bisa menggunakan media dari kamera untuk deteksi bola, lawan, serta gawang, juga bisa digunakan sensor garis, dan sensor jarak, untuk memperbaiki pergerakan robot agar bisa sinkron

antara roda maka diperlukan rotary encoder yang mana data dari pembacaan sensor akan dimasukkan perhitungan *odometry* sehingga dapat digunakan untuk acuan dalam mensinkronkan putaran roda kanan dan roda kiri, *rotary encoder* juga menghasilkan menghasilkan posisi (x,y). Agar pergerakan robot sepak bola responsif dan stabil maka digunakan metode PID sehingga robot sepak bola beroda dalam menyelesaikan misinya lebih cepat dan tepat.

II. TEORI PENUNJANG

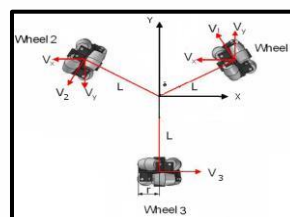
A. Sistem Pergerakan Robot *Three Omni-directional Drive*.

Merupakan suatu sistem pergerakan robot yang efisien dalam menentukan arah pergerakan dari robot. Bentuk dari Robot *Three Omni-directional Drive* adalah segitiga sama sisi dengan jumlah roda omni ada tiga buah yang terletak di setiap ujungnya membentuk sudut 60 derajat. Gambar 1 merupakan bentuk Robot *Three Omni-directional Drive*.



Gambar 1. Bentuk robot *three omni-directional drive*.

Untuk menentukan pergerakan dari robot ini menggunakan persamaan kinematik sebagai sistem kontrol pergerakan robot.



(a)



(b)

Gambar 2. (a) Desain mekanik roda (b)

Representasi kinematika dari sistem pergerakan robot

$$V_x = V_3 - V_1 \cos(\delta) - V_2 \cos(\delta) \quad \dots\dots\dots(2.1)$$

$$V_y = V_1 \sin(\delta) - V_2 \sin(\delta) \quad \dots\dots\dots(2.2)$$

$$V_\phi = V_1/L + V_2/L + V_3/L \quad \dots\dots\dots(2.3)$$

$$V_i(1,2,3) = w.r \quad \dots\dots\dots(2.4)$$

Dimana :

r : jari-jari roda omni (cm)

w : kecepatan angular roda (rad/detik).

Untuk mendapatkan kecepatan tiap masing-masing roda dengan cara mengalikan kecepatan sudut roda dengan jari-jari roda. Dalam desain Robot *Three Omni-directional Drive* roda tersusun secara simetris dengan tiap rodanya mempunyai perbedaan sudut sebesar 120° dan dengan $\delta = (60^\circ)$. Persamaan matrik Robot *Three Omni-directional Drive*. [1]

$$\begin{bmatrix} V_x \\ V_y \\ V_\phi \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} & 1 \\ \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 \\ 1/L & 1/L & 1/L \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \quad \dots\dots\dots(2.5)$$

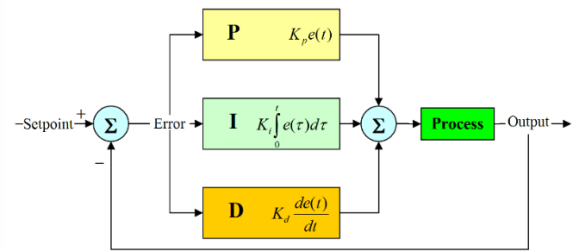
B. Kontrol PID (*Proporsional + Integral + Derivatif*).

Sistem kontrol PID terdiri dari tiga buah cara pengaturan yaitu kontrol P (*Proporsional*), I (*Integral*), dan D (*Derivative*) dengan masing-masing memiliki kelebihan dan kekurangan.

Gabungan aksi proporsional, integral, dan difensial mempunyai keunggulan dibandingkan dengan masing-masing dari tiga aksi kontrol tersebut. Masing-masing kontroler P, I, dan D berfungsi untuk mempercepat reaksi sistem, menghilangkan offset, dan mendapatkan energi ekstra ketika terjadi perubahan *load*. [2]

Dalam perancangan sistem kontrol PID yang perlu dilakukan adalah mengatur parameter P, I atau D agar tanggapan sinyal keluaran sistem terhadap masukan tertentu sebagaimana yang diinginkan.

PID mempunyai karakteristik *reset control* dan *rate control* yaitu meningkatkan respon dan stabilitas sistem serta mengeliminasi steady state error. Gambar 3 merupakan gambar diagram blok kontroler PID. Persamaan PID dapat dinyatakan pada persamaan berikut



Gambar 3. Diagram kontroler PID [3]

Tuning kontroler PID diperlukan dalam menentukan nilai K_i , K_p , dan K_d . Metode-Metode Tuning dilakukan berdasarkan pada:

- Model matematika plant/sistem
- Jika model tidak diketahui, dilakukan eksperimen terhadap sistem.

Persamaan PID dapat dinyatakan pada persamaan berikut.

$$mv(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad \dots\dots(2.6)$$

C. Odometry

Odometry adalah penggunaan data dari pergerakan aktuator untuk memperkirakan perubahan posisi dari waktu ke waktu. *Odometry* digunakan untuk memperkirakan posisi relatif terhadap posisi awal.

Untuk memperkirakan posisi relatif dari robot, digunakan perhitungan jumlah pulsa yang dihasilkan oleh sensor *rotary encoder* setiap satuan ukuran yang kemudian dikonvolusikan menjadi satuan millimeter. Rumus untuk mendapatkan jumlah pulsa setiap kali putaran roda yaitu: [4]

$$K_{\text{roda}} = 2 * \pi * r \quad \dots\dots\dots(2.7)$$

$$Pulsa_per_mm = resolusi_enc / K_{\text{roda}} \quad \dots\dots\dots(2.8)$$

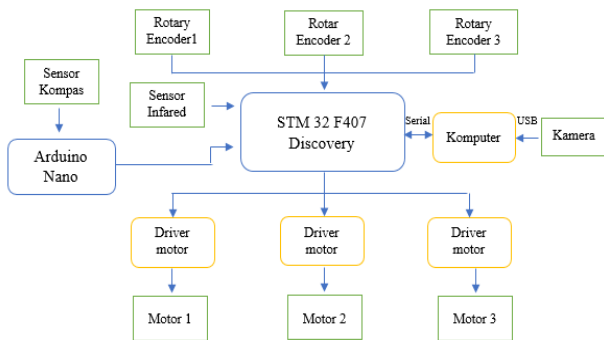
Dimana:

K_{roda} : Keliling Roda Omni

$Pulsa_per_mm$: konversi pulsa ke milimeter

D. Perancangan Sistem Keseluruhan

Pada tahapan ini dirancang cara kerja sistem kontrol robot menggunakan Sensor *rotary encoder*. Integrasi antara Sensor *rotary encoder*, Kontroler PID, dan mikrokontroler sehingga dapat menjalankan sistem sesuai dengan tujuan.



Gambar 4. Desain blok perancangan perangkat keras (*Hardware*) sistem secara keseluruhan.

E. Perancangan Desain Mekanik

Perancangan desain mekanik yang bagus dan presisi dapat pengaruh besar terhadap kinerja dan manuver dari robot tersebut. Gambar 5 merupakan mekanik Robot.



Gambar 5. Robot *three omni-directional drive*.

Spesifikasi mekanik robot terbuat dari aluminium dengan tebal 4 mm dan 5 mm, ketiga roda menggunakan roda *omni* berdiameter 10 cm, ukuran robot tinggi 80 cm dan lebar 50 cm.

F. Perancangan Sistem Kinematika Robot *Three Omni-directional Drive*.

Pada Robot *Three Omni-directional Drive* agar dapat bergerak ke segala arah dan sesuai dengan yang diinginkan maka perlu disinkronkan antara ketiga roda dengan cara mengetahui terlebih dahulu kecepatan motor (V_1 , V_2 , V_3). Nilai dari V_1 , V_2 , dan V_3 digunakan untuk perbandingan kecepatan tiap roda serta dikalikan dengan faktor pengali tertentu, dalam penelitian ini faktor pengali ditetapkan adalah 300 rpm. Dalam tabel 1 merupakan kecepatan tiga roda secara otomatis pada beberapa sudut yang telah ditentukan.

Tabel 1. Kecepatan pada tiap – tiap roda pada robot *three omni-directional drive*

No	Sudut Gerak Robot (°)	V1 (Rpm)	V2 (Rpm)	V3 (Rpm)
1	0	- 1500	3000	-1500
2	45	-2897,78	2121,32	776,46
3	90	-2598,08	0,00	2598,08
4	135	-776,46	-2121,32	2897,78

5	180	1500	-3000	1500
6	225	2897,78	-2121,32	-776,46
7	270	2598,08	-0,00	-2598,08
8	315	776,46	2121,32	-2897,78

Nilai rpm bernilai negatif putaran roda berlawanan searah jarum jam (CCW) sedangkan nilai rpm positif putaran roda searah jarum jam (CW).

G. Perancangan Sistem Tuning Kontroler PID Menggunakan Metode Kedua Ziegler-Nichols.

Dalam proses tuning PID menggunakan metode kedua Ziegler-Nichols mencari nilai K_p terlebih dahulu sampai sistem berosilasi berkesinambungan pada motor dengan nilai $K_i = 0$ dan $K_d = 0$, setelah itu baru menambah dengan nilai K_i dan juga nilai K_d secara berurutan. Dari nilai K_p saat terjadi osilasi berkesinambungan dapat dicari nilai K_{cr} dan P_{cr} . Dari nilai K_{cr} dan P_{cr} dapat digunakan untuk menghitung nilai K_i dan K_d dengan persamaan berikut.

$$K_p = 0,6 \times K_{cr}$$

$$T_i = 0,5 \times P_{cr}$$

$$T_d = 0,125 \times P_{cr}$$

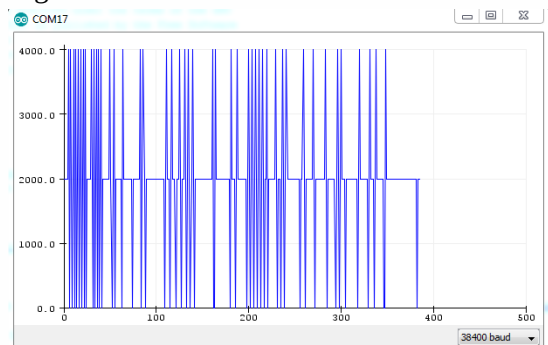
Untuk mengetahui nilai K_i dan K_d dengan menggunakan perhitungan berikut.

$$K_i = K_p / T_i$$

$$K_d = K_p \times T_d$$

H. Perencanaan Tuning PID Pada Roda

Hasil dari Pengujian repon roda dengan menggunakan kontroler proporsional dengan nilai $K_p = 70$, $K_i = 0$, $K_d = 0$. Dapat dilihat pada gambar 6.



Gambar 6. Grafik respon kecepatan roda 1.

III. HASIL DAN ANALISA

Pengujian dan analisa di maksud untuk mengetahui sejauh mana sistem yang di buat dapat bekerja sesuai dengan dengan perencanaan serta didapat analisa sistem secara keseluruhan untuk pengembangan selanjutnya . Pengujian dilakukan perblok sistem. Kemudian secara keseluruhan.

A. Pengujian Sensor Rotary Encoder

Pengujian ini bertujuan untuk melihat apakah sensor *rotary encoder* yang terpasang di motor planer gear (PG) 45 dapat menghasilkan pulsa. Pada pengujian ini, sensor *rotary encoder* diputar sebesar $\frac{1}{2}$ dan 1 putaran penuh. Pada masing-masing putaran dapat dilihat pada pulsa yang terbaca oleh mikrokontroler. Jumlah pulsa yang terbaca di ditampilkan pada LCD 40 X 2. Tabel 2 merupakan hasil pengujian yang diperoleh dari beberapa kali pengambilan data.

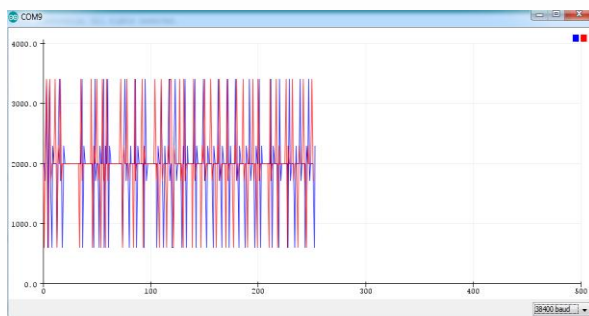
Tabel 2. Pengujian Sensor Rotary Encoder

No	Putaran	Rotary encoder 1	Rotary encoder 2	Rotary encoder 3
1	1/2	57	57	57
2	1	114	115	114

Berdasarkan tabel 2 dapat diperoleh hasil bahwa kesalahan pulsa yang terbaca oleh mikrokontroler adalah 1 pulsa. Pada pengujian, kesalahan pembacaan berupa jumlah pulsa kurang atau lebih besar dari nilai seharusnya. Salah satu kesalahan dari pembacaan nilai pulsa dari sensor *rotary encoder* karena terjadi kesalahan dalam memutar sensor *rotary encoder*.

B. Pengujian Hasil Tuning Kontroler PID

Pengujian ini bertujuan untuk melihat respon dari pergerakan robot dan putaran dari roda dengan menggunakan nilai parameter kontrol PID penerapan metode Osilasi Ziegler-Nichols. Grafik respon kecepatan roda hasil *tuning* dengan metode kedua Ziegler-Nichols dan *hand tuning* ditunjukkan pada gambar 7.



Gambar 7. Grafik respon kecepatan roda hasil *tuning* PID

Pada gambar 7 menunjukkan respon kecepatan roda setelah dilakukan *tuning* nilai PID. Garis biru mewakili roda 1 dan warna merah mewakili roda 2. Pada roda 1 nilai $K_p = 70$, $K_i = 10$, $K_d = 5$, sedangkan roda 2 nilai $K_p = 80$, $K_i = 10$, $K_d = 5$.

C. Pengujian Arah Gerak Robot

Pengujian ini bertujuan untuk mengetahui sinkronisasi pergerakan Robot *Three Omni-directional Drive* untuk bergerak pada sudut tertentu. Untuk pengujian robot dapat inputan pada

sistem (mikrokontroler STM32F407) berupa sudut tertentu dari inputan tersebut maka robot dapat bergerak sesuai dengan sudut kartesian. Hasil dari pengujian respon robot saat bergerak ke arah tertentu setelah beberapa kali melakukan percobaan dapat dilihat pada tabel 3.

Tabel 3. Hasil pengujian arah gerak robot

No	Sudut Inputan	Sudut Gerak Robot (°)
1	0	2
2		2
3		3
4	45	48
5		54
6		52
7	90	95
8		93
9		89
10	135	130
11		133
12		140
13	180	167
14		163
15		170
16	225	247
17		246
18		249
19	270	282
20		274
21		269
22	315	345
23		346
24		335

Berdasarkan tabel diatas dapat di peroleh rata-rata kesalahan sudut gerak robot sebesar $10,54167^0$.

D. Pengujian Pembacaan Posisi Robot

Pengujian ini dilakukan untuk mengetahui jarak perpindahan dari robot ke titik yang telah ditentukan. Prosedur percobaan untuk mencari nilai jarak perpindahan dalam senti meter (cm), dapat dilakukan dengan cara menggerakkan robot secara manual sejauh 1 meter. Tabel 4 merupakan ujicoba untuk mendapatkan nilai pulsa dalam lintasan sejauh 1 meter.

Tabel 4. Mencari jumlah nilai pulsa pada *rotary encoder* dalam 1 meter

NO	Panjang lintasan	Pulsa 1	Pulsa 2
1	1 Meter	309	297
2		311	308
3		312	302
4		313	306
5		307	301
6		309	296
7		314	296

8		308	298
9		312	301
10		310	303

Berdasarkan tabel 4 pulsa 1 merupakan jumlah pulsa untuk roda kiri depan yang terbaca dan pulsa 2 jumlah pulsa untuk roda kanan depan yang terbaca. setelah 10 kali melakukan sampling rata-rata dalam 1 meter jumlah pulsa pada *rotary encoder* adalah sebesar 305,65 pulsa. Jadi dalam 1cm perpindahan robot pulsa yang terbaca adalah 30,565 pulsa. Hasil pengujian dengan *tuning manual* pembacaan nilai jarak pada lintasan lurus sejauh 1 meter ditunjukkan pada tabel 5.

VI.

Tabel 5. Pengujian Untuk Mencari nilai jarak.

NO	Jarak sebenarnya	Jarak yang terbaca (mm)
1	1 Meter	96,5
2		99,2
3		99
4		98,5
5		98,5
6		99
7		98,5
8		97,5
9		98,5
1p		99,5

Bersarkan hasil pengujian diperoleh pembacaan nilai carak oleh robot pada lintasan lurus dengan jarak rata-rata yang terbaca adalah 98,47 cm.

IV. KESIMPULAN

Berdasarkan hasil pengujian yang di dapatkan, maka dapat disimpulkan bahwa untuk mensinkronkan pergerakan robot diperlukan metode persamaan kinematika untuk menentukan kecepatan masing-masing roda dan agar respon lebih cepat dapat diatasi dengan kontroler PID serta metode odometry untuk mengukur jarak.

Parameter yang digunakan untuk tuning kontroler PID menggunakan metode *hand tuning* pada roda 1 diperoleh nilai $K_p = 150$, $K_i = 1$, dan $K_d = 95$. Pada roda 2 diperoleh nilai $K_p = 150$, $K_i = 1$, dan $K_d = 100$. Pada roda 3 diperoleh nilai $K_p = 150$, $K_i = 1$, dan $K_d = 100$.

Rata-rata kesalahan sudut hadap robot three omni-directional adalah sebesar $10,54167^\circ$.

Jumlah pulsa rata-rata dari *rotary encoder* yang terbaca oleh mikrokontroler dalam 1 meter adalah 305,65 pulsa dan dalam 1 meter jarak sebenarnya rata-rata terbaca oleh mikrokontroler adalah 98,47 cm.

V. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada kedua orang tua yang selalu memberikan dukungan moril maupun materiil kepada penulis. Tak lupa penulis mengucapkan terimakasih kepada Bapak Koko Joni S.T., M.Eng. dan Bapak. Riza Alfita S.T.,MT selaku dosen pembimbing penulis dalam menyelesaikan penelitian ini serta Bapak Achmad Fiqhi Ibadillah, S.T., M.Sc selaku pembimbing tim robot KRSBI Beroda. Dan juga penulis ucapkan terima kasih kepada semua pihak yang telah membantu dalam mengerjakan penelitian ini.

VI. DAFTAR PUSTAKA

- [1] Al-Ammri, Salam dan Iman Ahmed.2010. *Control of Omni-Directional Mobile Robot Motion*. Baghdad: *Al-Khwarizmi Engineering journal*.
- [2] Raditya Artha Rochmanto. 2014.Implementasi Robot Three Omni- Directional Menggunakan Kontroler PID Pada Robot Kontes Robot Abu Indonesia (KRAI) [Skripsi] Malang (ID): Universitas Brawijaya.
- [3] <https://id.wikipedia.org/wiki/PID>,(diakses 12 Juni 2018).
- [4] Fernando Ardila, Bayu Sandi Marta, A.R.Anom Besari, 2011, “ *Path Tracking* Pada Mobile Robot Dengan Umpan Balik *Odometry*”, Politeknik Elektronika Surabaya, 26 Oktober 2011.

Implementasi Pergerakan *Omnidirectional* pada *Three-Omniwheeled Robot*

Simon Siregar, Muhammad Ikhsan Sani, M. Muchlis Kurnia, Dzikri Hasbiallyh
Universitas Telkom

Jln. Telekomunikasi Terusan Buah Batu, Bandung

Email korespondensi: simon.siregar@tass.telkomuniversity.ac.id

Abstrak – Robot dengan kemampuan bergerak ke segala arah menjadi sebuah kebutuhan pertandingan Kontes Robot Sepak Bola Beroda Indonesia. Salah satu metode yang dapat digunakan adalah dengan menggunakan *omni wheels* dengan konfigurasi tiga roda. Penelitian ini akan membangun dinamika dan kinematika konfigurasi tiga roda *omni wheels* yang dilengkapi dengan motor dan *driver motor* dengan karakteristik masing-masing roda, *driver motor* dan motor mendekati sama. Konfigurasi sudut antara sumbu roda satu dengan roda yang lain dibagi dengan sudut 120° . Kemudian konfigurasi sistem tersebut diterapkan pada sebuah robot dan diuji pergerakannya. Hasil pengujian menunjukkan bahwa pergerakan dari robot dengan menggunakan metode *omni wheels* dapat bergerak sesuai rancangan yang dibangun.

Kata kunci: *three-wheeled Robot*, *holonomic mobile robot*, kinematika dan dinamika.

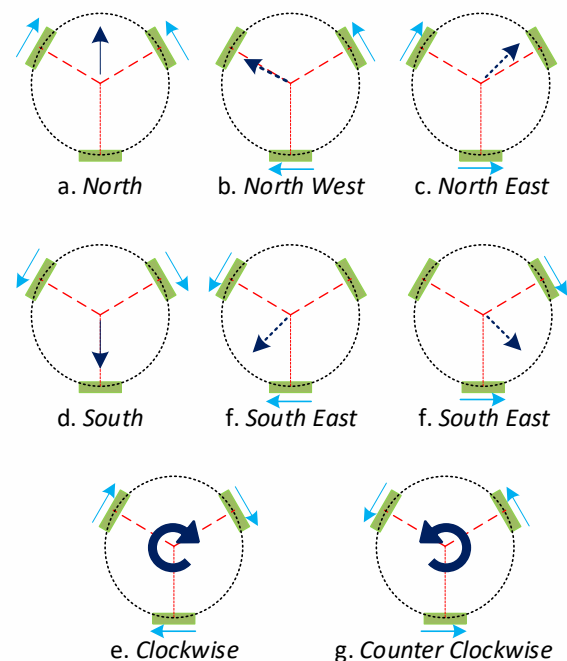
I. PENDAHULUAN

Kontes Robot Sepak Bola Beroda Indonesia (KRSBI) adalah salah satu kategori dalam Kontes Robot Indonesia yang diselenggarakan oleh Kementerian Riset Teknologi dan Pendidikan Tinggi (KEMENRISTEKDIKTI) yang mengacu pada RoboCup MSL 2016. Robot sepak bola beroda merupakan robot dengan penggerak menggunakan roda yang dirancang dengan kemampuan yang dapat berpindah-pindah tempat ke segala arah dengan tujuan untuk mengejar, menggiring dan memandang bola. Konstruksi dari robot beroda sendiri dapat dibagi menjadi beberapa tipe, yakni robot yang dapat bergerak ke segala arah (*omnidirectional* atau yang dikenal juga sebagai *holonomic*) dan robot yang tidak dapat bergerak ke segala arah. Robot *holonomic* adalah robot *mobile* yang memiliki kemampuan untuk dapat merubah arah pergerakan tanpa harus melakukan beberapa langkah rotasi dan dapat melakukan gerakan tersebut pada saat itu juga. (1)

Beberapa jenis pergerakan *omnidirectional* menggunakan roda dapat dibagi menjadi beberapa jenis yakni dengan menggunakan *cylindrical wheels*, *omni wheels* dan roda *mecanum wheels* (2). Agar dapat bergerak ke segala arah, jenis roda yang mungkin digunakan adalah *omni wheels* dan *mecanum wheels*. Dengan jenis yang berbeda, konfigurasi peletakan setiap roda menjadi berbeda pula. Selain konfigurasi, terdapat perbedaan juga dalam jumlah

roda yang digunakan. Secara garis besar, tipe mobilitas dari robot *mobile* dapat dibagi menjadi tiga: beroda, berkaki dan menggunakan bola. Sementara robot *mobile* sendiri dibagi menjadi dua: *omnidirectional* dan yang tidak *omnidirectional* (1). Pemilihan jenis roda sendiri berdasarkan hasil penelitian Soni (3), menunjukkan bahwa penggunaan roda dengan jenis *mecanum wheel* lebih rumit, sehingga pada penelitian ini digunakan jenis roda *omni wheels*.

Konfigurasi *Omnidirectional* dari robot *mobile* umumnya menggunakan konfigurasi simetris tiga roda maupun empat roda *omni wheels*. Beberapa bentuk konfigurasi peletakan roda menggunakan *omni wheels* dengan konfigurasi 4 roda dengan kondisi simetris maupun tidak simetris ditunjukkan pada penelitian Bemis(4).



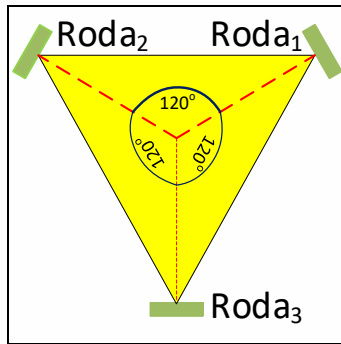
Gambar 1. Arah gerak dengan konfigurasi tiga *omni wheels* (2)

Beberapa desain yang menggunakan konfigurasi dengan tipe tiga roda *omni wheels* dan empat roda *omni wheels* telah dilakukan oleh Sani (5) dan Kantori (6). Dengan alasan kemampuan manuver yang baik dan kemudahan dalam sistem kendali seperti pada

penelitian Ribeiro(7), penelitian ini mengacu pada penelitian *Turtle Robot*(8), di mana robot *mobile* tersebut menggunakan roda dengan jenis *Omni wheels* dan dengan konfigurasi 3 roda dengan jarak sudut antar roda adalah 120° . Untuk dapat bergerak ke segala arah, terdapat beberapa metode dalam melakukan gerak beberapa arah seperti pada Gambar 1. Dengan konfigurasi ini, akan dibangun sebuah sistem pergerakan dengan menggunakan konfigurasi tiga roda *omni wheels* simetris dengan sudut peletakan masing-masing roda sebesar 120° .

II. ROBOT DENGAN TIGA OMNI WHEELS

Dalam membangun sistem pergerakan *omnidirectional* sistem, konfigurasi dari peletakan roda ditunjukkan pada Gambar 2. Konfigurasi tersebut membagi posisi tiga roda *omni wheels* dengan sudut masing-masing sumbu roda adalah 120° .



Gambar 2. Arah gerak dengan konfigurasi tiga *omni wheels* dengan sudut antara masing-masing sumbu roda sebesar 120°

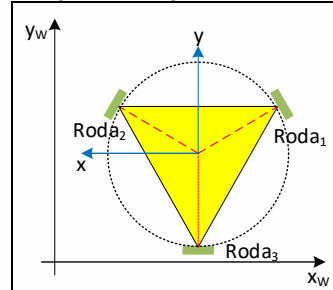
A. Kinematika dan Dinamika dari Robot Tiga Roda Omni Wheels

Kinematika pergerakan dari ketiga roda *omni wheels* tersebut digambarkan pada Gambar 3. Dengan menggunakan konfigurasi tersebut, robot jenis ini menghasilkan tiga derajat kebebasan (*Degree of Freedom*), dengan pergerakan arah linear sumbu x , pergerakan linear sumbu y dan pergerakan angular. Model kinematika sebuah robot *omnidirectional* digunakan untuk menggambarkan posisi dan kecepatan robot dan perubahan posisi robot dalam *frame World* dan *frame Robot* itu sendiri (9). Vektor posisi robot umumnya dinotasikan dengan x , y , dan ω pada *frame robot* dan notasi x_w , y_w , dan ω_w pada *frame world* seperti yang ditunjukkan pada Persamaan 1. Sementara untuk vektor kecepatan digambarkan dengan v_x , v_y , dan v_ω pada *frame robot* serta v_{x_w} , v_{y_w} , dan v_{ω_w} pada *frame world*. Agar posisi robot dapat ditransformasikan dari *frame robot* ke *frame world* ataupun sebaliknya, dibutuhkan matrik transformasi $R(\theta)$ dengan θ adalah perbedaan sudut yang dibentuk antara *frame world* dan *frame robot* seperti yang ditunjukkan pada Persamaan 2.

$$\begin{bmatrix} x_w \\ y_w \\ \omega_w \end{bmatrix} = R(\theta) \begin{bmatrix} x \\ y \\ \omega \end{bmatrix} \text{ dan } \begin{bmatrix} v_{x_w} \\ v_{y_w} \\ v_{\omega_w} \end{bmatrix} = R(\theta) \begin{bmatrix} v_x \\ v_y \\ v_\omega \end{bmatrix} \quad (1)$$

Dengan:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$



Gambar 3. Konfigurasi robot tiga roda *omni wheels* terhadap sumbu x dan y *frame Robot* dan sumbu x_w dan y_w *frame World*

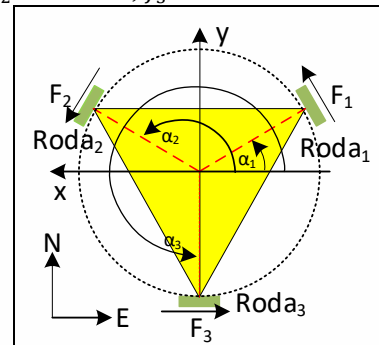
Dinamika sistem *omnidirectional*, seperti yang ditunjukkan pada Gambar 4, digunakan untuk menggambarkan pengaruh gaya yang diberikan roda terhadap pergerakan arah linear dan angular terhadap posisi robot. Jika masing-masing roda diberikan gaya $\vec{F}_1$, $\vec{F}_2$ dan $\vec{F}_3$ sudut α_1 , α_2 , dan α_3 merupakan sudut sumbu roda terhadap sumbu global untuk roda 1, 2 dan 3 secara berurutan, maka diperoleh resultan vektor gaya $\vec{F}_R$ dari masing-masing roda $\vec{F}_1$, $\vec{F}_2$ dan $\vec{F}_3$ pada Persamaan 3 dengan.

$$\vec{F}_R = \vec{F}_1 + \vec{F}_2 + \vec{F}_3 \quad (3)$$

Dengan asumsi massa yang digunakan adalah sama, maka akan diperoleh matriks berdasarkan metode yang digunakan pada Alaerds(8), diperoleh Persamaan 4.

$$\begin{pmatrix} a_x \\ a_y \\ \omega \end{pmatrix} = \begin{pmatrix} \sin(\alpha_1 + \frac{\pi}{2}) & \sin(\alpha_2 + \frac{\pi}{2}) & \sin(\alpha_3 + \frac{\pi}{2}) \\ \cos(\alpha_1 + \frac{\pi}{2}) & \cos(\alpha_2 + \frac{\pi}{2}) & \cos(\alpha_3 + \frac{\pi}{2}) \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} \quad (4)$$

Di mana, a_x adalah percepatan arah x , a_y adalah percepatan arah y dan ω adalah percepatan sudut yang dihasilkan pada gaya yang diberikan pada roda 1, f_1 , roda 2, f_2 dan roda 3, f_3 .



Gambar 4. Konfigurasi gaya gerak yang bekerja dari masing roda terhadap sumbu x dan y *frame robot*

B. Motor dan Driver Motor

Semua jenis motor yang digunakan dalam penerapan metode kontrol arah gerak linear dan angular pada penelitian ini memiliki karakteristik yang ditunjukkan pada Tabel 1.

Tabel 1. Karakteristik Motor PG45RS775

Operating Voltage	24	Volt (dc)
No load speed	468±10%	RPM
No load current	<1,5	Ampere
Rate Load Torque	15	Kgf.cm
Rated Current	<6,5	Ampere
Rated Load Speed	398±10%	RPM
Stall Torque	40	Kgf.cm
GearBox Maximum Torque	60	Kgf.cm
Rotation Direction	CCW/CW	

Dengan bentuk motor yang digunakan ditunjukkan pada Gambar 5. Jenis motor ini merupakan motor dengan *gear reduction speed* sehingga dapat menghasilkan torsi maksimal hingga 15 kg.



Gambar 5. Motor PG45RS775.

Untuk dapat mengendalikan motor sesuai dengan karakteristik yang dimiliki motor tersebut, digunakan driver motor berjenis BTS7960 (Gambar 6). Driver Motor ini merupakan driver motor bertipe H-Bridge yang memiliki karakteristik yang cukup aman dibandingkan terhadap kebutuhan arus rata-rata dari motor yang digunakan. Karakteristik dari driver motor BTS7960 ditunjukkan pada Tabel 2.

Tabel 2. Karakteristik H-Bridge Motor BTS7960

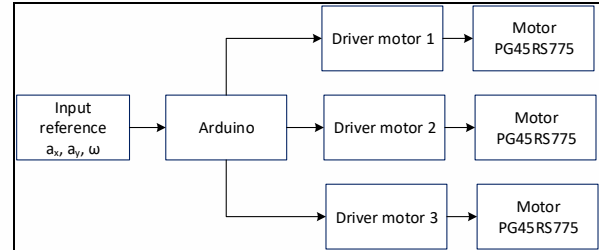
Supply Voltage	5.5-27	Volt (dc)
Logic Input Voltage	-0.3-5,3	Volt (dc)
Continuous Drain Current	40	Ampere



Gambar 6. Driver Motor H-Bridge BTS7960.

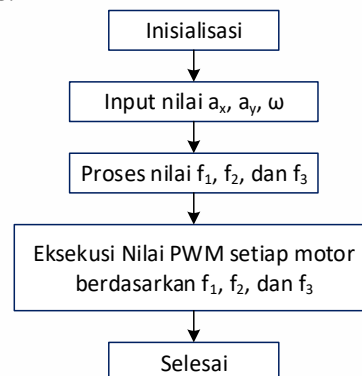
C. Sistem Penggerak yang Dibangun

Desain kinematika dan dinamika yang telah dibangun pada sub bab sebelumnya kemudian diterapkan dalam sebuah sistem kendali menggunakan mikrokontroler arduino, dengan bentuk diagram blok yang ditunjukkan pada Gambar 7.



Gambar 7. Blok diagram sistem penggerak *omni directional*

Dengan menggunakan Persamaan 2, dibentuk sebuah program menggunakan IDE Arduino yang dapat mengatur pergerakan berdasarkan nilai input yang diberikan. Flow chart dari sistematis kerja dari Mikrokontroler yang digunakan ditunjukkan pada Gambar 8.



Gambar 8. Flow chart sistem penggerak *omni directional*

Perubahan dari nilai Input a_x , a_y , dan ω menjadi nilai gaya f_1 , f_2 dan f_3 merupakan proses dengan menggunakan Persamaan 2. a_x merupakan percepatan robot pada sumbu x, sementara a_y merupakan percepatan robot pada sumbu y dan ω adalah percepatan angular robot. Nilai f_1 , f_2 dan f_3 , dapat berupa nilai positif dan negatif, kemudian dikonversi menjadi sebuah nilai yang dapat dieksekusi oleh arduino yakni sebuah nilai antara 0 hingga 255 (8bit).

III. PENGUJIAN DAN DISKUSI

A. Pengujian Menggunakan Matlab

Berdasarkan hasil simulasi menggunakan Matlab, dapat dihitung dengan memberi nilai parameter $\alpha_1 = 30^\circ$, $\alpha_2 = 150^\circ$, dan $\alpha_3 = 270^\circ$, maka diperoleh tabel pergerakan yang ditunjukkan pada Tabel 3.

Tabel 3. Hasil tabel simulasi arah pergerakan robot

a_x	a_y	ω	f_1	f_2	f_3	Arah Gerakan
1	0	0	0,58	-0,58	0	East
0	1	0	-0,33	-0,33	0,67	North
-1	0	0	-0,58	0,58	0	West
0	-1	0	0,33	0,33	-0,67	South
1	1	0	0,25	-0,91	0,67	North East
-1	-1	0	-0,25	0,91	-0,67	South West
1	-1	0	0,91	-0,25	-0,67	South East
-1	1	0	-0,91	0,25	0,67	North West
0	0	1	0,33	0,33	0,33	Berlawanan Arah Jarum Jam (CCW)
0	0	-1	-0,33	-0,33	-0,33	Searah Jarum Jam (CW)

B. Pengujian Sistem Gerakan pada Robot

Dengan menggunakan blok diagram pada Gambar 5 dan Flow Chart Sistem Penggerak *Omnidirectional* pada Gambar 6. Diperoleh hasil pergerakan robot pada Tabel 4.

Tabel 4. Hasil implementasi pergerakan arah robot

a_x	a_y	ω	Arah Gerakan	Kesesuaian gerakan
1	0	0	East	Sesuai
0	1	0	North	Sesuai
-1	0	0	West	Sesuai
0	-1	0	South	Sesuai
1	1	0	North East	Sesuai
-1	-1	0	South West	Sesuai
1	-1	0	South East	Sesuai
-1	1	0	North West	Sesuai
0	0	1	Berlawanan Arah Jarum Jam (CCW)	Sesuai
0	0	-1	Searah Jarum Jam (CW)	Sesuai

C. Analisis Sistem Gerakan pada Robot

Dari hasil percobaan dan hasil simulasi matlab, dihasilkan pergerakan robot yang sesuai dengan rencana penelitian. Beberapa pengembangan yang dapat dilakukan pada penelian ini adalah penerapan sistem kendali arah sesuai yang dapat disesuaikan dengan sistem navigasi yang dimiliki robot seperti sensor kompas, *Inertial Measurement Unit* (IMU) dan *encoder rpm sensor*(10) serta menggunakan kamera(11).

Terdapat beberapa pergerakan yang mengalami deviasi dari arah pergerakan seharusnya. Dengan adanya deviasi ini, dibutuhkan suatu sistem kendali yg dapat mengakomodasi galat arah yg dihasilkan. Terdapat beberapa kemungkinan sistem kendali yang dapat digunakan seperti sistem kendali Proportional, Integral dan Derivatif (PID)(1), *Kalman Filter*, dan *Neural Network*.

IV. KESIMPULAN

Dari hasil simulasi dan pengujian, tujuan penelitian yakni menghasilkan pergerakan omnidirectional dengan menggunakan konfigurasi 3 *omni wheels* dengan sudut antara masing-masing sumbu roda sebesar 120° dapat dicapai. Adapun mengurangi kondisi galat arah gerakan yang dihasilkan dapat menjadi salah satu rencana penelitian selanjutnya.

V. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Universitas Telkom yang membiayai penelitian ini melalui Dana Penelitian Internal Universitas Telkom dan Fasilitas yang diberikan pada Laboratorium Robotik *School of Applied Science* (SAS) Universitas Telkom.

VI. DAFTAR PUSTAKA

- [1] Moreno J, Clotet E, Lupiañez R, Tresanchez M, Martínez D, Pallejá T, et al. Design, implementation and validation of the three-wheel holonomic motion system of the assistant personal robot (APR). *Sensors* (Switzerland). 2016;16(10).
- [2] Parmar JJ, Savant C V. Selection of Wheels in Robotics. 2014;5(10):339–43.
- [3] Soni S, Mistry T, Hanath J. Experimental Analysis of Mecanum wheel and Omni wheel. *IJSET -International J Innov Sci Eng Technol* [Internet]. 2014;1(3):292–5.
- [4] Bemis S, Quinton B. Design and development of a novel omni-directional platform. *ProQuest Diss Theses* [Internet]. 2009;(October):209.
- [5] Sani MI, Damayanti I, Ismail, Saleh MI. Vision System on Wheeled Robot Soccer. In: *Indonesian Symposium on Robot Soccer Competition*. 2016.
- [6] Kantori MT, Sugianto Z, Nugraha MI, Febriansyah A. Mobile Robot Pemain Bola : Aplikasi Pada KRSBI 2017. 2017;
- [7] Ribeiro F, Moutinho I, Silva P, Fraga C, Pereira N. Three Omni-Directional Wheels Control on a Mobile Robot. *Control* 2004 [Internet]. 2004;
- [8] Alaerds RJG. Mechanical design of the next generation Tech United Turtle. 2010;
- [9] Oliveira HP, Sousa AJ, Moreira AP, Costa PJ. Dynamical Models for Omni-directional Robots with 3 and 4 Wheels. *Proc 5th Int Conf Informatics Control Autom Robot* [Internet]. 2008;(May 2008):189–96.
- [10] Garcia-Saura C. Self-calibration of a differential wheeled robot using only a gyroscope and a distance sensor. 2015;(September).
- [11] Jesus F, Ventura R. Simultaneous Localization and Mapping using Stereo and Mono Vision for Tracked Wheel Robots. 2012;7.

Sistem Pergerakan Robot pada Robot Bola Beroda Tim CHAKRAMAKARA UI

Aziz Fikri Hudaya, Dheavira Hadina Putri, Aldinisa Rahma Sabillah, dan Muniati Kurnia
Universitas Indonesia
Depok, Indonesia
Email : azizhudaya39@gmail.com

Abstrak - Robot dalam pertandingan KRSBI Beroda dituntut dapat melakukan manuver dan perpindahan yang baik dan efisien agar dapat menggiring bola ke gawang lawan. Supaya robot dapat melakukan perpindahan yang efisien, maka diperlukan roda yang dapat bergerak secara dinamis. Roda yang dipakai, yaitu roda *Omniwheel*. Roda ini memiliki roda-roda kecil yang terpasang tegak lurus dengan roda utama di ujung-ujung roda utama, sehingga pergerakan robot dapat lebih mulus saat menggiring bola. Dengan manuver dan perpindahan robot yang dinamis dan efisien, robot dapat bergerak lebih lincah dan dapat meraih kemenangan.

Kata kunci: *omniwheel*, rpm, kecepatan linier

I. PENDAHULUAN

Setiap tahunnya Direktorat Jenderal Pembelajaran dan Kemahasiswaan, Kementerian Riset, Teknologi dan Pendidikan Tinggi (DIKTI) selalu mengadakan Kontes Robot Indonesia (KRI). Dalam kontes tersebut terdapat suatu divisi yang bernama kontes robot sepak bola indonesia (KRSBI). Untuk divisi KRSBI, Universitas Indonesia diwakilkan oleh Tim CHAKRAMAKARA UI.

Tahun ini tim CHAKRAMAKARA UI memiliki 3 robot sepak bola, 2 robot dengan tipe menyerang dan 1 robot bertipe penjaga gawang. Dalam mengembangkan robot sepak bola ini masih banyak tantangan yang harus dihadapi, salah satunya bagaimana cara robot untuk bergerak secara dinamis dan efisien.

Oleh sebab itu fokus penelitian ini untuk membahas tentang cara pergerakan robot sepak bola yang akan digunakan dalam kontes. Roda yang penulis gunakan dalam kontes ini ialah 4 roda *omniwheel* dengan konfigurasi X. konfigurasi ini dapat membuat robot dapat bergerak ke segala arah tanpa adanya gerakan tambahan dan lebih sedikit gesekan yang terjadi. yang lebih sedikit ini akan menambah efisiensi pergerakan robot. Dan, pergerakan roda yang dinamis membuat robot dapat lebih mudah melakukan manuver saat robot dijalankan.

II. SISTEM PERGERAKAN ROBOT

A. *Omniwheel*

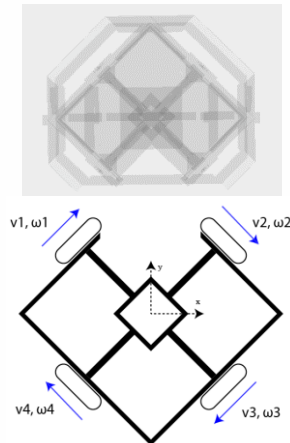
Omniwheel adalah roda yang mampu bergerak ke segala arah dalam 2 dimensi (X,Y). hal ini dikarenakan *Omniwheel* memiliki banyak roda-roda kecil yang tegak lurus dengan roda utama dan menempel pada ujung roda utama. Roda utama bergerak sesuai penggerak motor yang digunakan (maju atau mundur), sementara roda kecil berfungsi untuk menggerakkan roda ke kiri atau ke kanan, sehingga apabila roda di translasi ke arah kiri atau kanan gesekan yang terjadi antara roda dengan permukaan kecil sehingga mudah digerakkan. Robot yang menggunakan roda *omni* akan bergerak secara *holonomic*, yang mana robot dapat berpindah ke segala arah langsung tanpa membuat gerakan tambahan.



Gambar 1. Roda *omni* yang digunakan

B. Model Robot 4 *Omniwheel* Dengan Konfigurasi x

Model robot tim CHAKRAMAKARA UI merupakan konfigurasi dari kombinasi 4 roda *omniwheel* konfigurasi ini membuat setiap roda *omni* terpisah sebesar 90 derajat dan terhadap sumbu – sumbu tegak sejauh 45 derajat.



Gambar 2. Kerangka robot yang digunakan

Keterangan gambar 2:

- v_1, v_2, v_3, v_4 [m/s] merupakan kecepatan linier roda robot
- $\omega_1, \omega_2, \omega_3, \omega_4$ [rad/s] merupakan kecepatan angular roda robot

konfigurasi ini membuat seluruh roda dapat bergerak bersamaan untuk bertranslasi ke arah tertentu. Dengan mengombinasikan gerak setiap rodanya, robot dapat bertranslasi tanpa ada hambatan. Robot bergerak sesuai dengan gambar yang ada di tabel

Gambar	Keterangan
	Maju
	Mundur
	Ke kiri
	Ke kanan

Tabel 1. Gerak setiap arah pada robot

Keterangan

- = bergerak searah jarum jam
- ← = bergerak berlawanan arah jarum jam

Dari tabel tersebut, kita dapat menghitung kecepatan linier dari robot

$$V_{maju} = \sin(45^\circ) (v_1 - v_2 - v_3 + v_4)$$

...(1)

$$V_{mundur} = \sin(45^\circ) (-v_1 + v_2 + v_3 - v_4)$$

...(2)

$$V_{kiri} = \cos(45^\circ) (v_1 - v_2 - v_3 + v_4)$$

...(3)

$$V_{kanan} = \cos(45^\circ) (v_1 - v_2 - v_3 + v_4)$$

...(4)

Sehingga dari persamaan diatas kita mendapatkan persamaan rumus sebagai berikut:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \\ v_d \end{bmatrix} = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & \sqrt{2}/2 & -\sqrt{2}/2 \\ -\sqrt{2}/2 & \sqrt{2}/2 & \sqrt{2}/2 & -\sqrt{2}/2 \\ -\sqrt{2}/2 & -\sqrt{2}/2 & \sqrt{2}/2 & \sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 & -\sqrt{2}/2 & -\sqrt{2}/2 \end{bmatrix} \times \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \dots (5)$$

Dengan

- v_a = Kecepatan linier maju,
- v_b = Kecepatan linier mundur
- v_c = Kecepatan linier ke kiri
- v_d = Kecepatan linier ke kanan

Diketahui bahwa $\omega = v/r$, dengan r = jari – jari roda (m) dan $\omega = 2.\pi.f$, dengan f = frekuensi putaran roda (Hz) sehingga

$$v_a = (\sqrt{2}.\pi.r)(f_1 - f_2 + f_3 - f_4) \dots (6)$$

$$v_b = (\sqrt{2}.\pi.r)(-f_1 + f_2 + f_3 - f_4) \dots (7)$$

$$v_c = (\sqrt{2}.\pi.r)(-f_1 - f_2 + f_3 + f_4) \dots (8)$$

$$v_d = (\sqrt{2}.\pi.r)(f_1 + f_2 - f_3 - f_4) \dots (9)$$

III. ANALISIS DAN DISKUSI PERGERAKAN ROBOT

A. Analisis Waktu Tempuh Robot

Robot bola beroda CHAKRAMAKARA UI menggunakan *omniwheel* tipe PG36 dengan diameter 10 cm, maksimal frekuensi rotasi motor sebesar 600 rpm, jarak dari tepi lapangan ke tengah lapangan 4,5 m, dan jarak antar tepi lapangan 9 m, Kecepatan dan waktu yang dihasilkan dapat dilihat dari tabel berikut (asumsi frekuensi setiap roda sama):

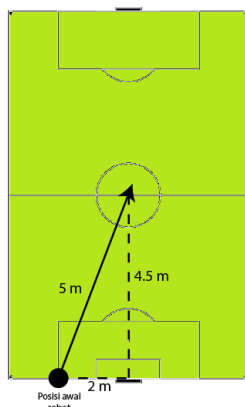
Tabel 2 Hubungan antara rpm dengan waktu tempuh

Rotation per Minute (rpm)	Kecepatan Linier (m/s)	Jarak (m)	Waktu (s)	Waktu sebenarnya (s)	Efisiensi (%)
100	1.4809	4,5	3.0385	3.6173	84%
100	1.4809	9	6.0771	7.5963	80%
200	2.9619	4,5	1.5192	1.9730	77%
200	2.9619	9	3.0385	4.0514	75%
300	4.4428	4,5	1.0128	1.3504	75%
300	4.4428	9	2.0257	2.7749	73%
400	5.9238	4,5	0.7596	1.0550	72%
400	5.9238	9	1.5192	2.1398	71%
500	7.4048	4,5	0.6077	0.8559	71%
500	7.4048	9	1.2154	1.7118	71%
600	8.8857	4,5	0.5064	0.7132	71%
600	8.8857	9	1.0128	1.4265	71%
Rata – rata efisiensi					74%

Dari tabel 2 dapat dilihat bahwa rata - rata efisiensi dari waktu yang diperlukan untuk menempuh jarak tertentu sebesar 74%. Dapat dilihat pula masih ada perbedaan antara waktu yang diperkirakan dengan waktu aslinya. Hal ini disebabkan karena adanya faktor luar yang mempengaruhi. Seperti adanya waktu robot untuk mencari objek bola dengan sensor, dan juga rpm yang dihasilkan dari motor tidak sepenuhnya bernilai sama dengan yang di programkan.

B. Analisis Gerak Robot Pada Saat *Kickoff*

Pada kontes KRSBI Beroda, terdapat aturan *kickoff*, keadaan ini membuat pihak lawan tidak bisa menggerakkan robotnya hingga tujuh detik setelah wasit meniup pluit pertandingan berjalan atau bola telah disentuh oleh pihak kita. Oleh karena itu dapat dibuat strategi untuk menciptakan peluang mencetak gol.

**Gambar 3.** Posisi awal saat *kickoff*

Asumsikan waktu robot untuk mencari bola selama 1 detik, robot harus sampai ke tengah lapangan dan telah mengambil bola 2 detik sebelum waktu *kickoff* berakhir. Sehingga kita mempunyai waktu bersih 4 detik untuk

menggerakkan robot serta semua roda bergerak dengan frekuensi yang sama, dapat dihitung

$$v = \frac{s}{t} = \frac{5 \text{ m}}{4 \text{ s}} = 1,25 \text{ m/s}$$

$$v = 4\sqrt{2} \cdot \pi \cdot r \cdot f \dots (6)$$

$$1,25 = 4\sqrt{2} \cdot \pi \cdot \frac{5}{100} \cdot f$$

$$f = \frac{5 \cdot 100}{4 \cdot 4\sqrt{2} \cdot 5 \cdot \pi}$$

$$f = 1.4067 \text{ Hz} = 84.404 \text{ rpm}$$

Sehingga rotasi per menit minimal yang dibutuhkan untuk bergerak ke tengah lapangan sebesar 84.404 rpm.

IV. KESIMPULAN

Berdasarkan hasil pengujian dan analisa, dapat diambil kesimpulan yaitu :

- Kecepatan linier maksimal yang dapat ditempuh oleh robot sebesar 8.88 m/s.
- Semakin besar rpm yang diberikan pada setiap roda dan semakin jauh jarak tempuhnya, semakin kecil pula efisiensi waktu tempuh robot.
- Efisiensi pada waktu tempuh robot berkisar dalam jangkauan 70% – 85%

Pergerakan pada robot ini masih akan dikembangkan lebih lanjut agar efisiensi waktu tempuh akan meningkat dan target waktu tempuh yang diinginkan dapat tercapai.

V. UCAPAN TERIMA KASIH

Penulis dan rekan kerja tim ini berterimakasih kepada Tuhan Yang Maha Esa, karena dengan kasih sayangnya kami dapat menyelesaikan paper ini. Kepada tim Chakramakara UI yang telah berjuang keras dan bekerjasama untuk membuat dan membuat penelitian tentang robot, terutama kepada kakak tingkat yang telah membimbing kami dari hal yang kami tidak ketahui. Dan juga kepada semua pihak yang telah mendukung kami dalam penelitian ini.

VI. DAFTAR PUSTAKA

- [1] Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia. "Buku Panduan Kontes Robot Sepakbola Indonesia Divisi Beroda (Krsbi Beroda)", Jakarta, 2018
- [2] Helder P. Oliveira, Armando J. Sousa, A. Paulo Moreira and Paulo J. Costa. "Dynamical Models For Omni-Directional Robots With 3 And 4 Wheels", Porto, Portugal, 2014
- [3] Rafiuddin Syam, Irham dan Widhi Erlangga. "Rancang Bangun Omni Wheels Robot Dengan Roda Penggerak Independent" Makassar, 2012

Mekanisme Penendang Menggunakan Solenoid pada Robot Sepak Bola Beroda

Joseph Elroy, Rizky Herlambang, Rizki Wira Pratama, dan Qurratu Aini Hasby

Universitas Indonesia

Depok, Indonesia

Email : josephelroy@gmail.com

Abstrak-Dalam permainan sepakbola, tujuan utamanya adalah mencetak gol sebanyak – banyaknya, begitu pula pada pertandingan KRSBI beroda ini. Penendang merupakan kunci utama dalam kemenangan, oleh karena itu dibutuhkan penendang kuat yang mampu menendang minimal dari tengah lapangan sampai ke gawang lawan. Solenoid digunakan untuk tujuan tersebut, arus yang mengalir pada kumparan solenoid menghasilkan medan magnet yang berfungsi untuk mendorong batang besi dan akhirnya terjadilah mekanisme penendang. Desain rangkaian yang digunakan pada driver solenoid dapat menghasilkan kecepatan maksimum penendang 10.58 km/h sedangkan kecepatan minimumnya 7.48 Km/h.

Kata kunci: Solenoid, Kapasitor, Relay

I. PENDAHULUAN

Kontes Robot Indonesia (KRI) adalah program kreatifitas yang dapat diikuti oleh tim mahasiswa dari Institusi atau Perguruan Tinggi Negeri dan Swasta yang terdaftar di Kementerian Riset, Teknologi dan Pendidikan Tinggi [1]. KRI ini memiliki lima cabang lomba yang diantaranya, Kontes Robot Sepak Bola Beroda (KRSBI Beroda). CHAKRAMAKARA merupakan tim dari Universitas Indonesia yang mengikuti Kontes Robot Sepak Bola Indonesia Beroda (KRSBI).

Dalam kontes tahun ini, CHAKRAMAKARA telah membuat robot sebanyak 3 buah yang terdiri dari 2 penyerang dan 1 penjaga gawang. Dalam melakukan pembuatan robot tim CHAKRAMAKARA melakukan penelitian. Penelitian pada divisi elektrik difokuskan pada sistem penendang robot dengan menggunakan solenoid. Dalam KRSBI beroda dimensi robot yang diperbolehkan adalah minimum 30x30 cm dan maksimum 52x52 cm. Dengan dimensi yang sesuai dengan aturan tersebut, dibutuhkan komponen – komponen yang tepat dan penempatannya yang terstruktur dan efisien agar dapat dimasukkan ke dalam robot.

Penendang merupakan salah satu komponen vital pada robot. Terdapat 3 pilihan *core* penendang, yaitu pneumatik, motor DC (*Direct Current Motor*) dan solenoid. Pada pneumatik, dibutuhkan tempat

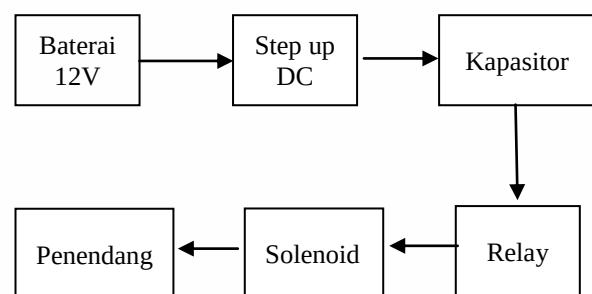
yang lebih besar untuk menempatkan botol – botol yang berisi gas. Selain itu, jumlah tembakan yang dihasilkan dari pneumatik terbatas oleh gas yang dapat ditampung botol. Motor DC bisa menjadi solusi dari keterbatasan dari ruang robot, namun masih terdapat kelemahannya. Untuk penendang menggunakan Motor DC, kekuatan yang dihasilkan semakin lama akan semakin lemah sehingga bola tidak terlempar jauh. Penggunaan Solenoid dirasa tepat karena tendangan yang dihasilkan dari robot bisa lebih konsisten dan lebih kuat serta untuk keterbatasan ruang dapat diatasi. Sistem penendang Solenoid ini memanfaatkan prinsip elektromagnetik.

II. SISTEM SOLENOID

A. Komponen Elektrik

Pada mekanisme penendang robot CHAKRAMAKARA digunakan sumber tegangan 12 V yang telah di distribusikan oleh *power distribution*, *relay*, *capacitor charging module*, kapasitor dan solenoid.

Secara sederhana dapat dilihat sebagai berikut:.



Tegangan 12 V dinaikan oleh capacitor charging module, tegangan yang dinaikan ini akan disimpan di kapasitor. Energi yang disimpan pada kapasitor dialirkan ke solenoid dengan *switch* relay yang membutuhkan *trigger*.

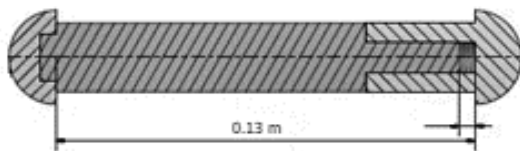
B. Desain Penendang Solenoid

Komponen penendang robot CHAKRAMAKARA menggunakan bahan besi dan nilon. Terlihat pada Gambar 1. Komponen nomor 3

terbuat dari besi yang akan merespon medan magnet yang dihasilkan solenoid. Komponen nomor 1 akan dikaitkan per sehingga ketika besi tersebut terdorong untuk menendang bola, per akan menarik kembali besi dan komponen nomor 2 yang akan kontak langsung dengan kaki penendang.



Gambar 1. Komponen pendorong



Gambar 2. Ukuran kompen pendorong



Gambar 3. Solenoid

Ukuran besi yang dibuat adalah sepanjang 25,5 cm. Besi ini harus berada didalam lilitan solenoid, ketika solenoid dialiri arus listrik, maka besi akan terdorong oleh medan magnet dan akan mendorong tuas yang akan menendang bola.

Lilitan solenoid yang digunakan sebanyak 1000 lilitan menggunakan kawat tembaga 1.5 mm, dengan panjang solenoid 13 cm dan diameter dalam 1.1 inch. Dengan menggunakan rumus :

$$L[mH] = \frac{0.015 \cdot N^2 \cdot \left(\frac{R1 + R2}{2}\right)^2}{6 \cdot \frac{R1 + R2}{2} + 9 \cdot l + 10(R2 - R1)} \quad (1)$$

kita dapat menghitung induktansi solenoid. Dimana N adalah jumlah lilitan, R1 adalah jarak jari – jari [m] dalam pipa dan R2 adalah jari jari luar pipa [m].

Hambatan dari solenoid juga dapat kita cari dengan persamaan[2]

$$R = \rho_{kawat} \cdot l_{kawat} \cdot \quad (2)$$

Dan

$$l_{kawat} = 2\pi \left(R1 + \frac{R2-R1}{2}\right) \cdot N \quad (3)$$

Untuk menghitung gaya pada solenoid, dapat menggunakan formula Lorentz yaitu

$$F = B \cdot l \cdot i \quad (4)$$

Dimana B adalah *flux density* [T], i adalah arus [A] dan l adalah panjang kawat [m]

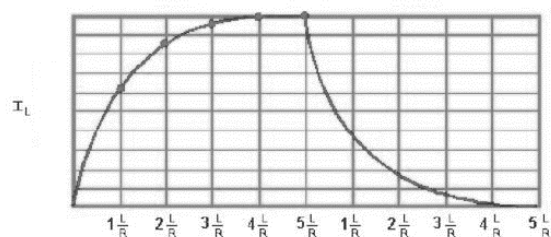
Pada kenyataannya, solenoid terdiri dari resistor dan induktor. Dimodelkan dengan rangkaian R-L. Persamaan arus yang terdapat pada L dapat ditulis

$$I = \frac{\varepsilon}{R} (1 - e^{-\left(\frac{R}{L}\right)t}) \quad (5)$$

Pada saat $t = R/L$, arus naik menjadi $(1-1/e)$. Jadi bisa disimpulkan R/L adalah ukuran seberapa cepat arus menuju nilai akhir. R/L disebut *time constant* dilambangkan dengan τ [3]

$$\tau = \frac{R}{L} \quad (6)$$

Reaksi waktu saat solenoid arus maksimum adalah ketika 5τ . Seperti grafik berikut[4].



Gambar 4. Grafik konstanta waktu

Hal ini menandakan, pendorong harus ditahan selama waktu reaksi ini. Setelah itu, solenoid bekerja dengan kekuatan penuh.

C. KEBUTUHAN ENERGI PENENDANG

Lapangan yang digunakan pada KRSBI Beroda memiliki ukuran $12 \times 8 \text{ m}^2$. Dengan asumsi robot bisa membuat goal pada gawang lawan dengan kecepatan bola 10 m/s dari tengah lapangan. Energi yang dibutuhkan oleh solenoid adalah

$$E_{bola} = \frac{1}{2} m_{bola} v_{bola}^2 + \frac{1}{2} J_{bola} \omega_{bola}^2 \quad (7)$$

$$m_{bola} = 0.41 \text{ kg}$$

$$V_{bola} = 8 \text{ m/s} = 28.8 \text{ km/h}$$

$$r_{bola} = 0.315 \text{ m}$$

$$J_{bola} = \frac{2}{3} m_{bola} r_{bola}^2 = 2.71 \cdot 10^{-2} \text{ kg} \cdot \text{m}^2 \quad (8)$$

$$\omega_{bola} = \frac{v_{bola}}{r_{bola}} \approx 25 \text{ rad/s} \quad (9)$$

$$E_{bola} = \frac{1}{2} \cdot 0.41 \cdot 8^2 + \frac{1}{2} \cdot 2.71 \cdot 10^{-2} \cdot 25^2$$

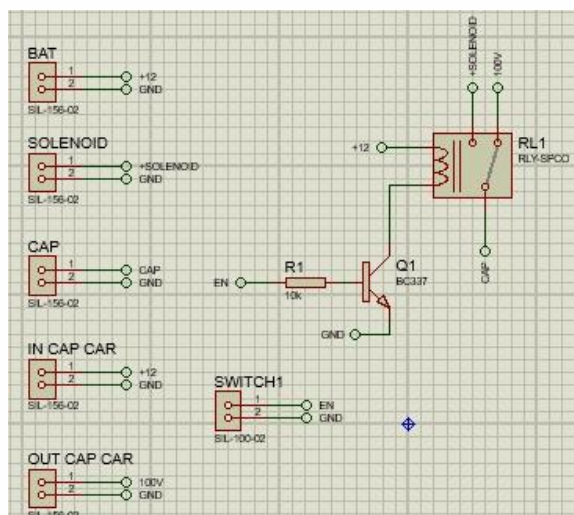
$$E_{bola} \approx 21.5 \text{ j}$$

Dengan konservasi energi didapat

$$E_{bola} = 21.5 \text{ j} = E_{Solenoid} \quad (10)$$

D. KEBUTUHAN ENERGI SOLENOID

Dalam rangka memenuhi energi yang dibutuhkan untuk menggerakkan bola sesuai target, robot membutuhkan driver solenoid sebagai berikut



Gambar 5. Skematik Driver Solenoid

Pada rangkaian tersebut digunakan Capacitor Charging module, relay, switch, dan kapasitor.

Capacitor Charging Module digunakan untuk menaikkan tegangan dari baterai 12 v hingga dapat

mencapai tegangan 450 v . Module ini dihubungkan secara langsung dengan lippo baterai 12 v dan output dari module ini dihubungkan ke capacitor dengan kapasitas hingga 100 v untuk disimpan sementara waktu sebelum dilepaskan.

Relay yang digunakan harus diberikan tegangan minimal 12 v untuk melakukan switching. Tegangan 12 v ini akan menghasilkan medan magnet pada koil yang terdapat di dalam relay sehingga dapat menarik tembaga dihadapan koil sehingga akan membuat jalur short dan tegangan dari kapasitor dapat mengalir ke solenoid.

Driver solenoid adalah rangkaian yang digunakan untuk menghubungkan komponen-komponen seperti *capacitor charging module*, relay, kapasitor, solenoid, dan *mikrocontroller*. Pada *driver solenoid* terdapat *switching* yang dapat dikontrol oleh *mikrocontroller* yaitu menggunakan transistor jenis bipolar junction transistor (BJT). Ketika basis pada transistor diberikan tegangan 5 v maka transistor akan dalam keadaan saturasi dan dapat menghubungkan jalur antara baterai dan relay sehingga relay dapat mengalirkan tegangan ke solenoid. Sebaliknya ketika basis pada transistor diberikan tegangan 0 v maka transistor dalam keadaan cut off dan jalur antara baterai dan relay dalam keadaan open circuit.

Kapasitor digunakan untuk menampung tegangan sementara sebelum dilepaskan ke solenoid. Kapasitor yang digunakan memiliki kapasitansi 4700 mfd dan tegangan maksimum 450 v .

$$E_{Kapasitor} = C \cdot V^2 = 4.7 \cdot 100^2$$

$$E_{Kapasitor} = 470000 \text{ j}$$

III. HASIL DAN DISKUSI

Dalam rangka menguji keberhasilan rangkaian, dilakukan simulasi. Simulasi yang dilakukan adalah memasang rangkaian solenoid pada robot, lalu robot diprogram untuk menendang bola. Pengamatan yaitu mengukur waktu tempuh bola bergerak sepanjang 1 meter .



Gambar 6. Simulasi solenoid

Tabel 1. Data pengamatan simulasi solenoid

Percobaan ke	Waktu (s)	Kecepatan (km/jam)
1	0,34	10,58
2	0,36	10,026
3	0,328	10,94
4	0,4	9
5	0,48	7,48

Dari data yang didapat diketahui bahwa bola dapat melesat dengan kecepatan maksimum 10,58. Kecepatan melemah dikarenakan baterai yang mulai berkurang tengahnya sehingga mengurangi tegangan kapasitor. Kecepatan minimumnya yaitu 7,48 km/jam. Percobaan ini membuktikan bahwa rangkaian belum dapat mencapai target 28,8 km/h.

IV. KESIMPULAN

Berdasarkan analisis dan hasil pengujian dapat diambil kesimpulan yaitu:

- Solenoid mampu menendang bola dengan kecepatan maksimum 10,58 km/h
- Solenoid belum mampu mencapai target 28,8 km/h
- Kekuatan solenoid melemah seiring dengan proses menendang yang dilakukan beberapa kali

Mekanisme penendang robot ini akan dikembangkan lebih lanjut. Konversi energi kapasitor ke solenoid belum berhasil dengan maksimal. Diperlukan peningkatan mutu solenoid dan mengurangi resistansi dan impedansi pada sistem.

V. UCAPAN TERIMA KASIH

Penulis berterima kasih kepada Tuhan Yang Maha Esa dan kepada tim CHAKRAMAKARA UI yang telah bekerjasama dan memberi dukungan dalam penelitian ini.

VI. DAFTAR PUSTAKA

- [1] Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia. "Tentang KRI". (Online). (<https://www.kontesrobotindonesia.org/>), 2017. Diakses pada tanggal 9 Juni 2018
- [2] Charles K. Alexander, Matthew N. O. Sadiku, "Fundamentals of Electric Circuits". McGraw-Hill, New York, 2013
- [3] R. C. Hibbeler, "Engineering Mechanics, Dynamics, 12th edition" Prentice Hall, Upper Saddle River 2010
- [4] Kasaei, S.H., Kasaei, S.M., Kasaei, S.A., Taheri, M. "Design and Development of Novel Solenoid-Based Kicking System". 9th ICME, Birjand, 2009

Sistem Deteksi Objek dan Posisi pada Robot Sepak Bola Beroda *Middle Size* pada Sistem Kamera *Omni Vision* dengan *Scan Lines*

Agung Raharjo, Achmad Fiqhi Ibadillah*, Koko Joni.*

Universitas Trunojoyo Madura
Bangkalan, Indonesia
Email: agungresistorc828@gmail.com

Abstrak -- Sistem *omni vision* adalah sistem yang banyak digunakan pada mobile robot soccer middle size. Dengan sistem ini cakupan area deteksi robot menjadi lebih luas sehingga bisa diterapkan untuk mapping. Mapping area lapang digunakan untuk mengetahui posisi robot dan posisi objek-objek disekitar robot. Posisi objek tersebut direpresentasikan dalam diagram kartesian dua dimensi dengan dua sumbu yang saling tegak lurus. Untuk mengenali objek berupa lapangan, garis, bola, gawang dan obstacle menggunakan metode *color segmentation*, *tresholding*, *image contours*, dan *LUT*. Konsep *LUT* digunakan untuk menghemat waktu proses pengolahan gambar dengan membagi warna kedalam beberapa klasifikasi.

Kata kunci: deteksi objek, posisi objek.

I. PENDAHULUAN

Teknologi robotika berkembang pesat dewasa ini baik di tingkat perangkat keras atau perangkat lunak dan salah satu pengaplikasiannya pada robot soccer. Robot Soccer adalah robot yang didesain agar dapat bermain sepak bola layaknya manusia. Robot Soccer dibagi menjadi dua jenis yaitu *humanoid robot soccer* dan *mobile robot soccer*. Pada liga mobile robot soccer, liga *mobile robot soccer middle size* adalah yang paling menantang karena karakteristik dalam pertandingan. Di liga ini, *mobile robot soccer* didesain dengan ukuran base maksimal 50 x 50 cm, tinggi 80 cm, berat maksimal 40 kg, dan bermain di area lapangan dengan ukuran 6 x 9 m. Aturan permainan dibuat mirip dengan aturan pada sepak bola manusia dengan perubahan untuk menyesuaikan dengan robot. Setiap robot dibuat *autonomus* dan memiliki sensor tersendiri. Setiap robot dapat berkomunikasi dengan robot yang lain dalam satu tim dengan komputer eksternal sebagai *base station* melalui jaringan nirkabel.

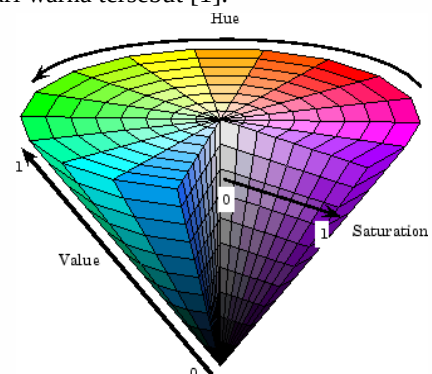
Salah satu elemen sensor yang berperan penting pada *mobile robot soccer* adalah kamera. Kamera pada *mobile robot soccer* difungsikan

seperti mata pada manusia. Kamera dibantu dengan cermin cembung atau kaca omni untuk memperluas area tangkap agar dapat menangkap gambar di sekitar robot secara 360 derajat. Dengan memanfaatkan kamera sebagai sensor, robot dapat mengenal objek disekitarnya seperti bola, gawang, lapangan, robot lawan dan robot kawan. Dengan objek-objek yang telah dikenali dan di *mapping* pada area lapangan didapatkan informasi berupa posisi objek di lapangan.

II. METODE.

A. Ruang Warna HSV

Pada sistem ruang warna HSV, Hue menyatakan warna sebenarnya, seperti merah, biru, kuning, dan lain sebagainya. Saturation menyatakan kemurnian atau kekuatan warna. Value menyatakan kecerahan dari warna, nilainya berkisar antara 0-100 % apabila nilainya 0 maka warnanya akan menjadi hitam., semakin besar nilai V maka semakin cerah dan muncul variasi-variasi baru dari warna tersebut [1].



Gambar 1. Model warna HSV

Untuk merubah nilai RGB menjadi nilai HSV dapat menggunakan teori travis, dapat dilihat pada persamaan

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B} \quad (1)$$
$$V = \max(r, g, b)$$

$$S = \begin{cases} 0, & V = 0 \\ 1 - \frac{\min(r,g,b)}{V}, & V > 0 \end{cases} \quad (2)$$

$$H = \begin{cases} 0, & V=0 \\ \frac{60 * (g-v)}{s * v}, & V=r \\ 60 * \left[2 + \frac{b-r}{s * v} \right], & V=g \\ 60 * \left[4 + \frac{r-g}{s * v} \right], & V=b \end{cases} \quad (3)$$

$$H = H + 360 \text{ jika } H < 0$$

HSV merupakan ruang warna yang cocok untuk mengidentifikasi warna-warna dasar, dimana warna dasar ini digunakan dalam penelitian sebagai warna identifikasi robot. Selain itu, HSV menoleransi terhadap perubahan intensitas cahaya. Inilah yang menjadi keunggulan HSV dibandingkan dengan ruang warna lainnya. Salah satu keunggulan dari ruang warna HSV dari pada RGB adalah dalam kemudahan menentukan rentang warna untuk proses segmentasi warna.

B. Color Filtering and Thresholding

Color filtering merupakan metode untuk menentukan piksel baru dengan menentukan batas atas serta batas bawah nilai dari sebuah piksel pada gambar, ketika dalam suatu gambar terdapat piksel yang memiliki nilai intensitas warna diantara batas atas dan bawah yang dikehendaki maka piksel tersebut akan diloloskan. Pada proses *thresholding*, piksel-piksel obyek dan *background* dikelompokkan menjadi dua mode yang dominan. Cara untuk mengekstrak obyek dari *background* adalah dengan memilih nilai *threshold* T yang memisahkan dua mode tersebut. Nilai T didapat dari proses *color filtering* menggunakan ruang warna HSV. Kemudian untuk sembarang titik (x,y) yang memenuhi $f(y,x) > T$ disebut titik obyek, selain itu disebut titik *background*. Hasil citra *thresholding* dapat didefinisikan sebagai persamaan 4.

$$g(y,x) = \begin{cases} 1, & f(y,x) > T \\ 0, & f(y,x) \leq T \end{cases} \quad (4)$$

Keterangan:

$g(y,x)$: Piksel Output

$f(y,x)$: Piksel Input

T : Variable Ambang untuk Threshold

Penentuan *range* warna yang tepat sangat mempengaruhi dari hasil *color filtering*[1].

C. LUT (Look Up Table)

LUT digunakan untuk klasifikasi warna yang cepat, kelas warna didefinisikan dengan penggunaan tabel look-up (LUT). LUT mewakili struktur data, dalam hal ini sebuah array, digunakan untuk mengganti perhitungan *realtime* dengan operasi pengindeksian array dasar. Pendekatan ini telah dipilih untuk menghemat waktu pemrosesan yang signifikan. Gambar dapat diperoleh dalam format HSV, RGB, YUV atau Bayer dan formatnya dikonversi menjadi gambar indeks (citra label) menggunakan LUT yang sesuai untuk masing-masing dari ketiga kemungkinan tersebut.

Warna terdiri dari 16.777.216 entri (2^{24} , 8 bit untuk R, 8 bit untuk G dan 8 bit untuk B) dengan masing-masing satu byte. Ukuran tabel sama untuk dua kemungkinan lainnya (YUV atau Bayer), namun makna masing-masing komponen berubah. Setiap bit dalam tabel entri mengungkapkan jika salah satu warna yang menarik (putih, hijau, biru, kuning, oranye, merah, langit biru, abu-abu - tidak berwarna) ada di dalam kelas yang sesuai atau tidak. Warna yang diberikan dapat diberikan ke beberapa kelas sekaligus. Untuk mengklasifikasikan sebuah pixel, pertama nilai warna pixel dibaca dan kemudian digunakan sebagai indeks ke dalam tabel. Nilai 8-bit kemudian dibaca dari tabel disebut "color mask" dari pixel. Hal ini dimungkinkan untuk melakukan subsampling gambar pada tahap ini dalam sistem dengan kemampuan pemrosesan terbatas untuk mengurangi waktu pemrosesan lebih lama lagi. Klasifikasi warna hanya diterapkan pada piksel yang valid jika ada masking[2].

Table 1. Contoh tabel LUT untuk klasifikasi warna

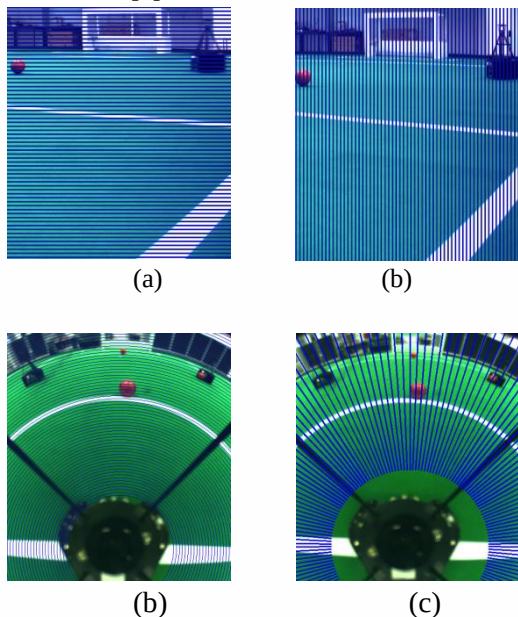
warna output	Range warna input (HSV)					
	H max	H min	S max	S min	V max	V min
Biru	197	118	59	15	177	105
Kuning	107	43	100	23	255	50
Hijau	180	112	100	41	255	52
Oranye	16	334	100	43	255	50
Putih	359	0	39	0	255	165
Hitam	359	0	74	0	41	0

Pada tabel 1 merupakan pemetaan warna inputan pada setiap pixel pada gambar dengan format warna HSV, menghasilkan warna output

yang terdiri dari warna biru, kuning, hijau, oranye, putih, dan hitam.

D. Scan Lines

Untuk mengekstrak informasi warna dari gambar dibuat tiga jenis garis pencarian, yang disebut scanlines: radial, linear (horizontal atau vertikal) dan melingkar. Mereka dibangun sekali, saat aplikasi dijalankan, dan disimpan dalam struktur untuk meningkatkan akses piksel-piksel ini dalam warna modul ekstraksi Pendekatan ini sangat penting untuk mengurangi waktu pemrosesan. Pada Gambar 2, empat tipe scanlines yang berbeda diilustrasikan[2].



Gambar 2. Contoh dari tipe *scanline* yang berbeda: a) *horisontal scanline*; b) *vertikal scanline*; c) *circular scanline*; d) *radial scanline*

III. HASIL DAN DISKUSI

A. Segmentasi Gambar

Segmentasi gambar digunakan untuk mengelompokkan beberapa intensitas warna pada gambar menjadi warna tertentu. Dalam percobaan beberapa intensitas warna pada gambar di kelompokkan menjadi empat warna, yaitu hijau, hitam, putih, dan oranye dan intensitas warna yang tidak sesuai di rubah menjadi warna gray.

Sebelum proses segmentasi gambar hasil tangkapan kamera harus dirubah kedalam format HSV dengan tujuan untuk mempermudah penentuan nilai *threshold* masing-masing warna. Dalam citra dengan format HSV ada enam nilai *thresholding* untuk setiap warna seperti yang ditunjukkan pada tabel 2 untuk citra hasil tangkapan kamera omni dan tabel 3 untuk citra hasil

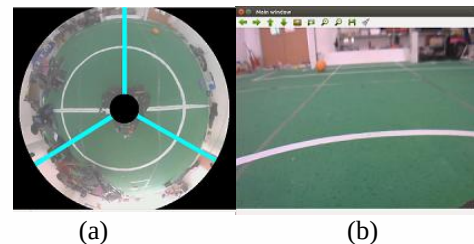
tangkapan kamera depan.

Tabel 2. Nilai *threshold* warna pada citra dari kamera omni

Warna output	Range warna input(HSV)					
	H max	H min	S max	S min	V max	V min
Putih	357	0	32	0	255	189
Hitam	359	0	33	0	114	0
Oranye	226	195	72	37	255	158
Hijau	169	53	60	4	248	72

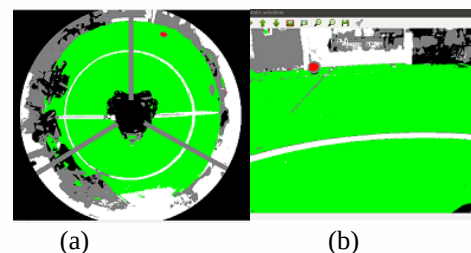
Tabel 3. Nilai *threshold* warna pada citra dari kamera depan

Warna output	Range warna input(HSV)					
	H max	H min	S max	S min	V max	V min
Putih	359	0	16	5	255	157
Hitam	359	0	23	0	115	0
Oranye	225	199	75	48	255	177
Hijau	169	53	60	0	248	68



Gambar 3. Citra asli yang di ambil kamera, omni (a) dan depan (b)

Sumber : Dokumentasi Pribadi



Gambar 4. Citra segmentasi dari citra asli, omni (a) dan depan (b)

Sumber : Dokumentasi Pribadi

B. Pembuatan Scanline

Scanline digunakan untuk mengetahui informasi warna setiap piksel yang dilintasi *scanline* pada gambar. dalam pengujian *scanline* yang digunakan adalah *radial scanline*, *circular scanline*, dan *horisontal scanline*. Dalam

pembuatan radial *scanline* S pertama tentukan n yang merupakan jumlah *scanline* yang digunakan. Selanjutnya tentukan nilai *delta* (Δ) dengan persamaan 5 berikut.

$$\Delta = \frac{2\pi}{n} \quad (5)$$

Nilai *delta* digunakan untuk menentukan jarak antara *scanline*. Selanjutnya tentukan nilai *teta* (θ) yang merupakan sudut *scanline* ke i (S_i) terhadap S_0 .

$$\theta_i = -\left(\frac{\pi}{2}\right) + (\Delta \cdot i) \quad (6)$$

Kemudian tentukan titik awal pembuatan *scanline* ke i dengan persamaan 8 dan 9 sebagai berikut.

$$Sstart(x)_i = center(x) + jd * \cos\theta_i \quad (8)$$

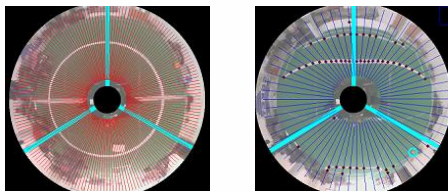
$$Sstart(y)_i = center(y) - jd * \sin\theta_i \quad (9)$$

Selanjutnya tentukan titik akhir pembuatan *scanline* ke i dengan persamaan 10 dan 11 sebagai berikut

$$Send(x)_i = center(x) + jl * \cos\theta_i \quad (10)$$

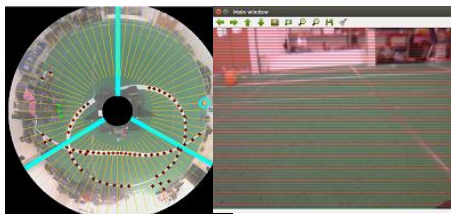
$$Send(y)_i = center(y) - jl * \sin\theta_i \quad (11)$$

$center(x)$ dan $center(y)$ adalah koordinat titik pusat dalam pembuatan *radial scanline*. jd adalah jarak antar titik pusat terhadap titik awal pembuatan *scanline* dan jl adalah jarak antar titik pusat terhadap titik akhir pembuatan *scanline*.



(a)

(b)



(c)

(d)

Gambar 5. Radial *scanline* untuk deteksi objek: (a) bola; (b) garis lapangan; (c) obstacle; (d) horisontal *scanline* bola

Sumber : Dokumentasi Pribadi

Pada kamera omni untuk mengekstra informasi warna bola digunakan radial *scanline* dengan jumlah 180 dan circular *scanline*. informasi warna garis lapangan dan warna obstacle diperoleh menggunakan radial *scanline* dengan jumlah 64 *scanline*. Kamera depan memperoleh informasi warna bola menggunakan horisontal *scanline* dengan jarak antar *scanline* 8 pixel seperti yang ada pada gambar 5.

Dalam memperoleh informasi, semakin banyak *scanline* yang digunakan untuk mendeteksi suatu objek akan meningkatkan sensitifitas tapi membuat semakin lama proses komputasinya.

C. Koordinat Objek

Ada tiga objek yang dideteksi oleh kamera omni robot, yaitu bola, obstacle dan garis lapangan. koordinat posisi objek di tentukan dengan acuan titik tengah robot sekaligus sebagai titik tengah frame gambar kamera omni sebagai titik nol. Frame gambar berukuran 640 x 480 pixel. Dalam frame gambar titik nol berada di bagian pojok kiri atas. Untuk merubah titik nol ke tengah frame digunakan fomula sebagai pada persamaan 12

$$Pusat(x, y) = \left(\frac{frame\ width}{2}, \frac{frame\ heigh}{2} \right) \quad (12)$$

$$pObjek' = Pusat - pObjek \quad (13)$$

Titik *Pusat* merupakan titik yang digunakan sebagai titik awal atau titik nol. *pObjek* merupakan posisi koordinat objek terhadap titik nol koordinat *frame*. *pObjek'* merupakan koordinat objek terhadap *Pusat* sebagai titik nol. Untuk menentukan jarak atau radius objek ($\square\square$) terhadap robot dapat menggunakan rumus pythagoras pada persamaan 14.

$$\square\square = \sqrt{\square^2 + \square^2} \quad (14)$$

x dan y adalah elemen dari $\square\square\square\square\square'$.

Untuk menentukan jarak sesungguhnya objek yang di deteksi dengan kamera omni dilakukan pengambilan sampel data perbandingan jarak sesungguhnya dalam (cm) terhadap data jarak pada frame gambar dalam *pixel*. Dari data tersebut di tentukan rumus fungsinya menggunakan algoritma interpolasi Lagrange. Tabel 4 merupakan data hasil pengambilan sampel.

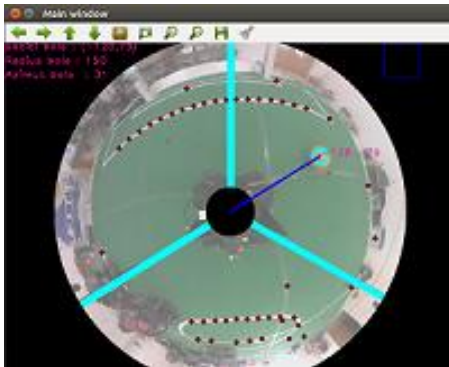
Tabel 4. Data perbandingan jarak kamera omni

Jarak (cm)	Jarak (pixel)
0	0
50	116
100	152
150	177

Rumus fungsi hasil interpolasi ditunjukkan dalam persamaan 15 dibawah ini. x merupakan jarak dalam *pixel* dan $\square(\square)$ adalah jarak dalam (cm).

$$\square(\square) = \frac{(1499.\square^3)}{71389764} + \frac{(48143.\square^2)}{71389764} + \frac{(836053.\square)}{11898294} \quad (15)$$

Implementasi dari persamaan 15 untuk menentukan jarak objek pada kamera omni dapat dilihat pada gambar 6.



Gambar 6. Objek bola terdeteksi pada jarak dan koordinat tertentu

Sumber : Dokumentasi Pribadi

IV. KESIMPULAN

Dari hasil pengujian yang telah dilakukan sistem vision robot dapat mengenali objek yang ada di lapangan khususnya objek bola, *obstacle*, dan garis lapangan dengan garis pencarian *scanline*. Robot juga dapat mengetahui posisi dan jarak objek dalam posisi sebenarnya.

V. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada kedua orang tua yang selalu memberikan dukungan moril maupun materiil kepada penulis. Tak lupa penulis ucapkan terimakasih kepada Bapak Achmad Fiqhi Ibadillah, S.T., M.Sc. dan Bapak Koko Joni, ST., M.Eng. selaku dosen pembimbing penulis dalam menyelesaikan penelitian ini. Dan juga penulis ucapkan terima kasih kepada semua pihak yang telah membantu dalam mengerjakan penelitian ini

VI. DAFTAR PUSTAKA

- [1] Zainal Abidin, "Rancang Bangun Robot Penghindar Halangan Berbasis Kamera Menggunakan Deteksi Kontur," Program Studi S1 Teknik Elektro, Universitas Trunojoyo Madura, Indonesia, 2017.
- [2] Antonio J.R. Neves, Alina Trifan, Bernardo Cunha, "UAVision: A Modular Time-Constrained Vision Library for Color-Coded Object Detection," IRIS Group, DETI / IEETA University of Aveiro, Portugal, 2014.
- [3] Bernardo Cunha, Jose Azavedo, Nuno Lau, Luis Almeida, "Obtaining the Invers Distance Map from a NON-SVP Hyperbolic Catadioptric Robotic Vision System," LSE-IEETA/DETI, University of Aveiro, Portugal, 2007.
- [4] Huimin Lu, Xun Li, Hui Zhang, Mei Hu and Zhiqiang Zheng, "Robust and Real-time Self-Localization Base on Omnidirectional Vision for Soccer Robots," College of Mechatronics Engineering and Automation, National University of Defense Technology, China, 2013.
- [5] Martin Lauer, Sascha Lange, dan Martin Riedmiller, "Calculating the Perfect Match: an Efficient and Accuration Approach for Robot Self-Localization," University of Osnabruck, Institute of Cognitive Science and Institute of Computer Science, Germany, 2005.

Teknik Pengambilan Warna untuk Deteksi Objek Berbasis OpenCV

Prayuda Ari Guntoro, Muhammad Wildan Nur Karim
Universitas Negeri Surabaya
Surabaya, Indonesia
yoedavegas18@gmail.com

Abstrak---Kontes Robot Indonesia selalu diadakan tiap tahun, semakin lama semakin canggih robot-robot yang diikuti pada ajang ini, saat ini terdapat divisi KRSBI dimana setiap robot diharuskan memiliki kemampuan untuk mengolah gambar untuk mendeteksi sebuah objek sehingga menuntut para mahasiswa, praktisi dan peneliti untuk turut serta mengembangkannya, robot ini juga disertai dengan sistem AI (Artificial Intelligence) atau kecerdasan buatan, sehingga dengan mampu mendeteksi objek maka akan berpengaruh juga dalam pengambilan keputusan gerakan robot.

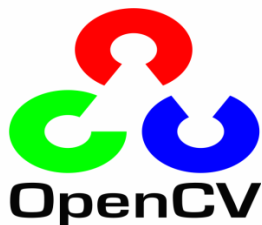
I. PENDAHULUAN

Saat ini banyak sekali robot-robot yang berkembang dengan sistem kecerdasan di dalamnya, sistem ini mampu membantu robot mengambil keputusan dengan melihat gejala-gejala yang terjadi disekitarnya, banyak digunakan sensor-sensor untuk melihat perubahan-perubahan disekitar atau saat ini juga sedang dikembangkan robot dengan pengolahan citra menggunakan kamera sehingga kamera ini memiliki fungsi seperti mata pada robot, robot mampu mengenali objek yang ada dihadapannya kemudian diolah sehingga robot mampu bergerak sesuai dengan program yang sudah dibuat.

II. TINJAUAN PUSTAKA

A. OPENCV

Salah satu yang terkenal untuk deteksi adalah OpenCV.



Gambar 1. Logo OpenCV

Saat ini library OpenCV sudah banyak digunakan pada robot-robot berbasis vision untuk pendeteksian objek menggunakan kamera. Walau begitu, pada library yang sama terkadang terdapat perbedaan cara dalam penggunaannya. Pada makalah ini adalah bagaimana proses pengambilan

citra pada objek kemudian prosesnya sehingga sampai dikirim ke mikrokontroller yang kemudian data tersebut dapat diolah kembali yang berfungsi sebagai penggerak motor.

Perlu diinstall pada miniPC beberapa library pendukung yang dibutuhkan seperti (numpy dll), untuk membantu pengolahan gambar sehingga outputnya berupa data nilai x dan y.

B. KOMPONEN



Gambar 2. Komponen

Untuk kamera dapat digunakan webcam/Omnivision. Dalam hal ini kami menggunakan kamera webcam Logitech C922 karena kualitas gambarnya sudah cukup baik dengan kualitas 1080 HD.



Gambar 3. Webcam logitech C922

Kemudian untuk pemrosesan gambar (Image Processing) kami gunakan Raspberry pi 3 Type B, untuk pengolahan hal-hal yang sederhana raspberry ini cukup murah dan mudah didapatkan di Marketplace atau toko-toko komputer dan ukurannya sebesar kartu ATM saja.



Gambar 4. Raspberry pi 3 model B

Output dari raspberry kemudian dikirimkan ke Arduino menggunakan USB to TTL secara serial. lalu program arduino akan merubah data-data tersebut kedalam putaran motor.



Gambar 5. USB to TTL dan arduino mega 2560

III. METODE

A. Pengambilan nilai pada warna.

Berikut ini adalah rumus sederhana untuk mencari nilai HSV melalui nilai RGB :

$$\begin{aligned} H &= \tan\left(\frac{3(G-B)}{(R-G)+(R-B)}\right) \\ S &= 1 - \frac{\min(R,G,B)}{V} \\ V &= \frac{R+G+B}{3} \end{aligned} \quad (1)$$

Keterangan :

H : Nilai Hue

S : Nilai Saturation

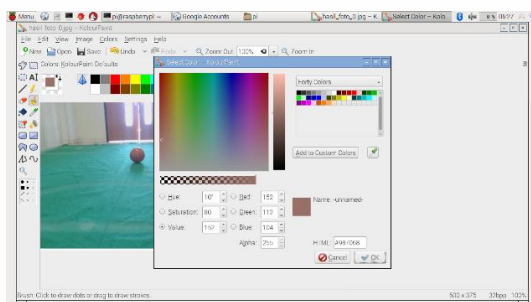
V : Nilai Value

R : Nilai dari warna merah (Red)

G : Nilai dari warna hijau (Green)

B : Nilai dari warna biru (Blue)

Dalam pengambilan sampel warna, bisa memanfaatkan aplikasi yang sudah ada pada raspberry, didalamnya kita bisa langsung mendapatkan nilai HSV untuk memilih warna yang akan kita pakai, sehingga tidak perlu terlalu lama menghitung.



Gambar 6. Mengambil nilai HSV

Kemudian kita ambil nilai Hue, Saturation, Value
 Hue : yaitu menyatakan warna sebenarnya
 Saturation : kadang disebut chroma, adalah kemurnian atau kekuatan warna.
 Value : kecerahan dari warna.

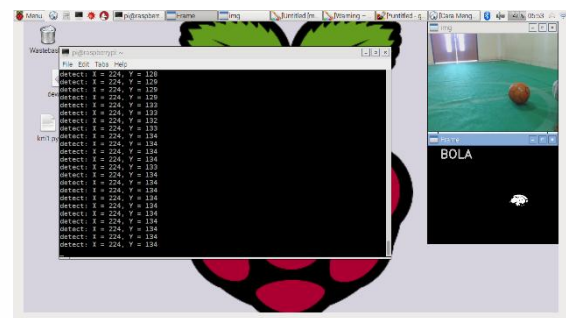
Pada saat pengambilan nilai, kita membutuhkan dua nilai dari warna paling gelap dan warna paling cerah sehingga pendeteksian objek akan terlihat bagus, perlu diingat bahwa warna itu sangat sensitif dan mudah terpengaruh oleh lingkungan sekitar terutama pencahayaan terhadap objek, jika pembiasan warna terlalu cerah bisa saja kamera tidak bisa mendeteksi adanya objek tersebut, jadi usahakan lingkungan mendukung untuk melakukan tracking object

Sebelum dimasukkan kedalam program utama, kita perlu menganalisis apakah warna yang kita ambil sudah bagus atau belum, hal tersebut dapat diketahui apakah pada frame hitam putih apakah objek terbentuk dengan sempurna.

Data tersebut kemudian akan dirunning oleh program dan diolah untuk menghasilkan nilai sehingga data tersebut dapat dikirimkan dan selanjutnya diolah oleh mikrokontroler.

IV. HASIL DISKUSI

Berikut adalah hasil running program, dapat kita lihat disini warna oranye pada bola terdeteksi dan pada frame hitam terdeteksi bola dan objek nya berwarna putih.



Gambar 7. Analisis warna

Berdasarkan (Gambar 7), kita dapat menganalisis bahwa bola terdeteksi pada koordinat dengan rata-rata X = 129, dan Y = 16. Sumbu X dan Y didapat berdasarkan jendela yang terbuka pada layar (gambar bola hitam putih) pada saat program dijalankan.

Lalu hasil informasi koordinat tersebut dikirim secara serial ke Arduino dengan menggunakan USB to TTL agar robot bisa menuju arah (koordinat) bola.

V. KESIMPULAN :

1. OpenCV adalah perpustakaan yang telah disediakan untuk pengolahan citra digital.
2. Untuk pendeteksian warna, perlu dicari nilai HSV pada sebuah objek.
3. Diperlukan nilai dari warna terendah sampai warna tercerah untuk mengatasi adanya perubahan deteksi warna akibat cahaya dilingkungan sekitar.
4. Untuk pengolahan data yang besar, tidak disarankan menggunakan Raspberry Pi karena keterbatasan memori sehingga akan kinerja akan menjadi lamban.

VI. REFERENSI

- [1] Dzikri, "Seleksi Warna RGB ke HSV". <http://dzikriugm.blogspot.com/2014/07/opencv-ubuntu-c-seleksi-warna-rgb-ke-hsv.html>.
- [2] Anonym, "Introduction to OpenCV Tracker". https://docs.opencv.org/3.1.0/d2/d0a/tutorial_introduction_to_tracker.html.
- [3] Rosebrock, Adrian. 2014. "Opencv and Python Colour Detection". <https://www.pyimagesearch.com/2014/08/04/opencv-python-color-detection/>.
- [4] Anonym, Choosing the correct upper and lower, HSV <https://stackoverflow.com/questions/10948589/choosing-the-correct-upper-and-lower-hsv-boundaries-for-color-detection-withcv>

Lokalisasi Robot Sepak Bola Terhadap Lapangan

Muhammad Rifqi Haryadi Pratomo, Muhammad Irsyad Thoyib, Dakhilullah Muhazzib,
Zaky Nuryasin, Johannes Calvin Tjahaja
Universitas Indonesia
Depok, Indonesia
Email : muhammad.rifqi78@ui.ac.id

Abstrak -- Lokalisasi atau *localization* merupakan salah satu metode yang digunakan pada robot untuk menentukan koordinat atau posisi secara presisi. Pada paper ini ditampilkan salah satu metode lokalisasi dengan menggunakan perhitungan odometri dengan menggunakan motor dc dan roda *omniwheel*.

Kata kunci: Lokalisasi, Odometri

I. PENDAHULUAN

Lokalisasi pada robot dibutuhkan untuk mendapatkan lokasi atau koordinat secara presisi dari robot untuk melakukan rencana pergerakan selanjutnya. Lokalisasi dilakukan dengan menggunakan encoder dengan 7ppr yang terhubung dengan motor pg36. Metode lokalisasi ini menggunakan odometri dengan menghitung perputaran roda untuk mendapatkan data koordinat x dan y serta *heading* pada robot.

Penelitian ini difokuskan pada bagaimana cara penerapan metode lokalisasi odometri untuk mendapatkan posisi yang akurat pada robot. Metode ini dapat mengurangi waktu yang diperlukan bagi robot untuk melakukan rencana pergerakan dikarenakan gerakan yang akan dilakukan menjadi lebih efektif dan akurat. Pada Kontes Robot Indonesia divisi KRSBI Beroda, metode ini dapat digunakan untuk memperoleh koordinat tiap robot serta koordinat gawang.

II. METODE

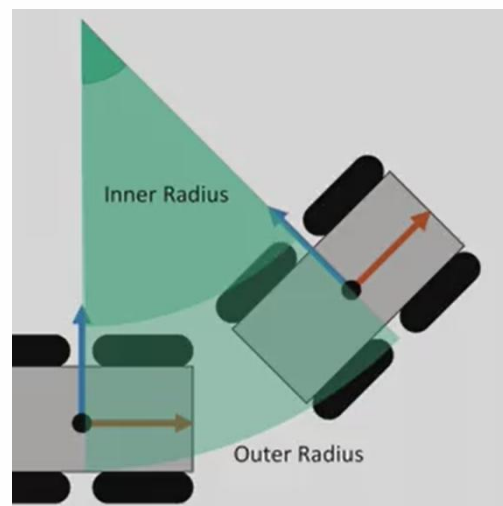
Metode yang digunakan dalam lokalisasi atau menghitung posisi robot adalah odometri. Odometri adalah proses perhitungan banyak perputaran yang dilakukan oleh roda-roda pada robot untuk menentukan posisi robot. Odometri dilakukan dengan menggunakan encoder dan kompas sebagai penghitung perputaran dan pada robot untuk mendapatkan data koordinat robot dan *heading*.

Kekurangan dari metode ini adalah rentan terhadap kesalahan akibat slip, dan desain sistem yang buruk (misalnya dimensi roda).



Gambar 1. Motor beserta encoder

Pada perhitungan odometri diperlukan sebuah motor encoder seperti pada gambar 1. Motor encoder akan menghasilkan *ticks* (e) yang dihasilkan dari perkalian tetha dengan radius dalam untuk e_i , dan perkalian antara tetha dengan radius luar untuk e_o .



Gambar 2. Menentukan radius dalam dan radius luar

Pada Gambar 2 diatas dapat ditentukan radius dalam yaitu antara roda bagian dalam dengan pusat perputaran dan radius luar antara roda bagian luar robot dengan pusat perputaran. Rumus untuk menentukan *ticks* (e) yang didapat pada radius dalam dan radius luar

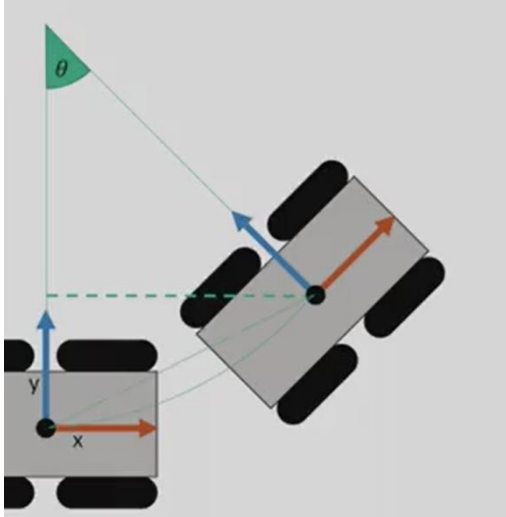
- $e_i = \theta r_i$
- $e_o = \theta r_o$
- $w = r_o - r_i$

w adalah jarak antara roda bagian luar dan roda bagian dalam.

Maka dapat ditentukan nilai derajat odometri.

Rumus:

$$\theta = \frac{e_o - e_i}{w}$$



Gambar 3. Tetha, koordinat x ,dan koordinat y

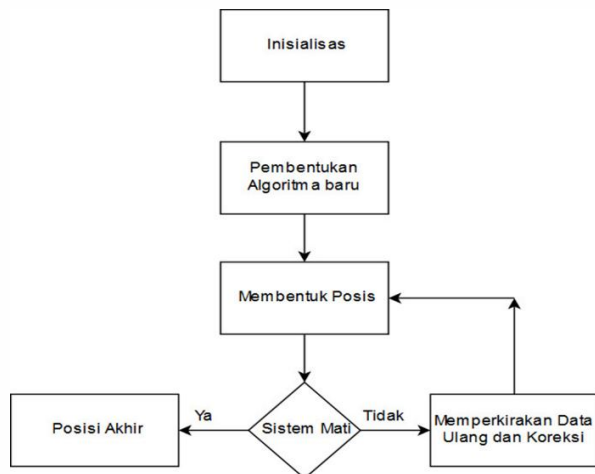
Pada proses menghitung koordinat lokasi robot menggunakan odometri, posisi robot dapat digambarkan seperti pada bidang 2 dimensi dengan adanya variabel x dan y yang membentuk 90 derajat.

Berikut perhitungan perubahan jarak pada lokasi koordinat x dan koordinat y yang dapat dilihat pada gambar 3.

$$y = \frac{e_o + e_i}{2} \cos \theta$$

$$x = \frac{e_o + e_i}{2} \sin \theta$$

Gambar 4. Perhitungan nilai x dan y



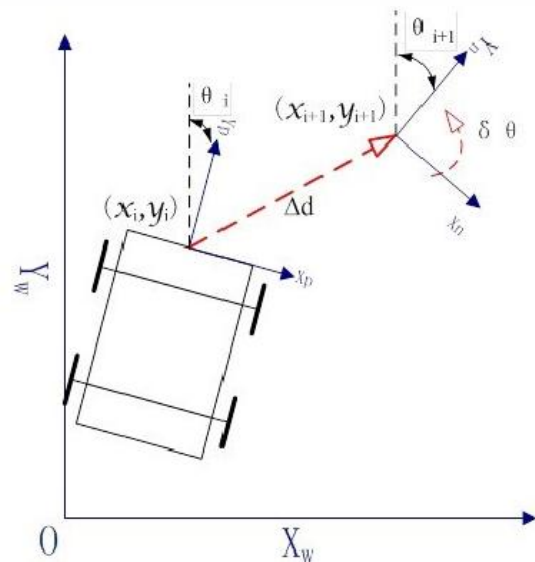
Gambar 5. Flowchart Algoritma Penggunaan Odometri

Dapat dilihat pada gambar diatas setelah dilakukan inisialisasi yaitu posisi awal robot dengan koordinat $x(0)=0$ dan $y(0)=0$, sistem akan membentuk algoritma perhitungan koordinat robot yang kemudian mencatat posisi awal robot. Robot akan berhenti memperbaharui posisi ketika dalam keadaan mati, namun ketika robot masih menyala maka koordinat robot akan terus diperbaharui bersama dengan pergerakannya.

Untuk menentukan posisi robot jika ditinjau melalui variable waktu (dalam bentuk diskrit) didapat rumus sebagai berikut,

$$d(i) = [d(i), \phi(i)]^T$$

i adalah variabel dalam bentuk bilangan bulat positif yang akan bertambah setiap kali data diperbaharui, $d(i)$ adalah variabel panjang di dalam bidang 2 dimensi, dan ϕ adalah sudut terhadap sumbu linear.



Gambar 6. Tinjauan terhadap variable integer i

Perhitungan Pembentukan Algoritma Baru :

D = Jarak Antara Roda sisi kanan dan sisi kiri
 $F1(i)$ = ticks(i) per revolusi dari roda kanan depan
 $F2(i)$ = ticks(i) per revolusi dari roda kiri depan
 $E1(i)$ = ticks(i) per revolusi dari roda kanan belakang
 $E2(i)$ = ticks(i) per revolusi dari roda kiri belakang

- $X_w = D \times 4$
- $X_R = \left(\frac{F1(i) + E1(i)}{2} - \frac{F1(i-1) + E1(i-1)}{2} \right)$
- $X_L = - \left(\frac{F2(i) + E2(i)}{2} - \frac{F2(i-1) + E2(i-1)}{2} \right)$
- $X_F = - \left(\frac{F1(i) + F2(i)}{2} - \frac{F1(i-1) + F2(i-1)}{2} \right)$
- $X_B = \left(\frac{E2(i) + E1(i)}{2} - \frac{E2(i-1) + E1(i-1)}{2} \right)$

6. $\phi = \left(\frac{X_R - X_L}{X_w}\right)$
7. $x(i) = \left(\left(\frac{X_R - X_L}{2}\right)\right)(\sin(\phi)) + \left(\left(\frac{X_F - X_B}{2}\right)\right)(-\cos(\phi))$
8. $y(i) = \left(\left(\frac{X_R - X_L}{2}\right)\right)\cos(\phi) + \left(\left(\frac{X_F - X_B}{2}\right)\right)\sin(\phi)$

III. HASIL DAN DISKUSI

Untuk mengetahui ketepatan dari lokalisasi odometri dilakukan 2 percobaan. Tujuan dari percobaan ini adalah untuk melihat seberapa besar kesalahan dari koordinat yang dihitung oleh robot terhadap koordinat aslinya. Pengambilan data dilakukan sebanyak 10 kali untuk tiap variasi dalam percobaan, yang kemudian dibuat nilai rata-rata.

Untuk percobaan pertama, robot bergerak dalam satu sumbu koordinat saja, yaitu sumbu x. Misalkan robot dianggap bergerak kanan dan kiri saja. Gerakan ini dilakukan berulang kali dari kanan ke kiri lalu ke kanan dan seterusnya, percobaan ini diharapkan dapat menunjukkan seberapa besar perbedaan nilai lokalisasi dengan jarak aslinya.

Berikut adalah tabel hasil percobaan.

Tabel 1. Hasil nilai *error* dari percobaan pertama

N	pos	Nilai kesalahan (cm)
1	1,9	0,95
5	6,6	3,3
10	21,1	10,55

N adalah banyak pengulangan gerakan yang dilakukan. Pos adalah nilai variabel perhitungan dalam program. Nilai kesalahan adalah perbedaan jarak hasil lokalisasi setelah gerakan dilakukan terhadap nilai aslinya dalam sentimeter.

Hasil dari percobaan ini menunjukkan perbedaan yang cukup besar dengan jarak asli jika gerakan dilakukan dalam jumlah yang banyak. Semakin banyak gerakan berbeda yang dilakukan, maka nilai kesalahan makin membesar. Pada pengulangan gerakan yang ke-10, mulai terlihat perbedaan yang cukup besar, yaitu 10,55 cm. Kesalahan perhitungan ini mungkin disebabkan oleh beberapa faktor yang mempengaruhi gerak robot selain perputaran roda, misalnya roda dan alas tidak bergesekan dengan baik. Hal ini menjadikan metode lokalisasi yang dipakai tidak cukup untuk menghitung koordinat robot secara tepat. Tetapi hasil data ini dapat digunakan robot untuk mengetahui lokasi robot dalam tiap area dari lapangan sepak bola, misalnya area penalti dan area *kick off*.

Untuk percobaan kedua, robot digerakan berputar membentuk persegi, dengan jarak 1 meter. Perbedaan besar daripada percobaan pertama adalah arah gerak robot yang berubah-ubah tiap satu gerakan. Tujuan kali ini adalah untuk melihat perbedaan perhitungan lokalisasi odometri untuk koordinat dan *heading* robot.

Tabel 2. Hasil nilai *error* dari percobaan kedua menggunakan *encoder*

N	Nilai kesalahan (cm)	Arah hadapan
1	5,85	5
5	11,0	12,2
10	31,9	29,5

Tabel 3. Hasil nilai *error* dari percobaan kedua menggunakan *encoder* dan kompas

N	Nilai kesalahan (cm)	Arah hadapan
1	1,55	1,6
5	7,95	1,7
10	11,75	1,1

Seperti sebelumnya, nilai kesalahan dari perhitungan enkoder semakin membesar jika gerakan berbeda terus dilakukan. Dengan tambahan kesalahan perhitungan *heading* robot, koordinat dari robot pun semakin menjauhi koordinat semestinya, untuk lebih jelasnya lihat tabel 2.

Terdapat perbedaan hasil jika menggunakan kompas, karena nilai kesalahan dari kompas tidak terus bertambah tiap ada perubahan gerakan. Hal ini karena data perhitungan kompas berbeda dengan data *encoder*. Nilai kesalahan dari kompas hanya berubah dengan selisih 2 derajat. Karena itu perhitungan koordinat tidak diperburuk oleh data arah hadap robot.

IV. KESIMPULAN

Lokalisasi dengan odometry merupakan salah satu metode yang dapat digunakan untuk mengetahui posisi robot dengan cukup akurat dengan menggunakan kompas eksternal sebagai data *heading* robot. Pemanfaatan dari diketahuinya posisi robot pada KRSBI Beroda ini adalah untuk memposisikan robot secara otomatis pada saat bola mati serta penentuan koordinat gawang. Namun beberapa faktor seperti *slip* pada roda akibat dari penggunaan *omniwheel*, dan terdapat halangan yang membuat robot tidak bergerak namun roda tetap berputar dapat mengacaukan hasil dari odometri.

V. UCAPAN TERIMA KASIH

Penulis berterima kasih kepada Tuhan Yang Maha Esa dan kepada tim Chakramakara UI yang telah bekerjasama dan memberi dukungan dalam penelitian ini

VI. DAFTAR PUSTAKA

- [1] Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia. "Tentang KRI". (Online). (<https://www.kontesrobotindonesia.org/>), 2017. Diakses pada tanggal 9 Juni 2018
- [2] Houshang Nasser and Farouk Azizi, "Mobile Robot Position Determination Using Data Integration of Odometry and Gyroscope" Purdue University Calumet, 2015

- [3] Ben-Ari, Mordechai and Francesco Mondada, "Elements of Robotics
- [4] <http://planning.cs.uiuc.edu/node576>. Diakses pada tanggal 12 Juni 2018
- [5] https://en.wikipedia.org/wiki/Rotary_encoder. Diakses pada tanggal 30 Juni 2018
- [6] <https://en.wikipedia.org/wiki/Odometry>. Diakses pada tanggal 30 Juni 2018

Optimasi Deteksi Bola pada Robot Sepak Bola Beroda Menggunakan *Region-based Segmentation*

Gilang PrilianArinata, Irfan Marzuqi, Hanif Ramadhan, Agus Khumaidi
Teknik Otomasi Politeknik Perkapalan Negeri Surabaya
Kampus ITS Sukolilo, Jl. Teknik Kimia, Keputih, Sukolilo, Keputih, Jawa Timur 60111
E-mail: aguskhumaidi@ppns.ac.id^[4]

Abstract - Setiap robot pemain sepak bola wajib memiliki kemampuan mendeteksi bola dengan akurat tanpa terganggu adanya *noise* dari luar lapangan. *Noise* berupa warna yang menyerupai bola yang berada diluar lapangan merupakan masalah utama. Dengan adanya *noise* tersebut, robot akan sering salah menentukan letak bola yang sesungguhnya. Untuk melakukan segmentasi (*Segmentation*) menggunakan parameter threshold Hue, Saturation, dan Value (HSV). Umumnya robot hanya menggunakan threshold warna tertentu tanpa dilakukan filter terhadap *noise* yang ada sehingga proses deteksi bola tidak begitu akurat. Penelitian ini bertujuan untuk membandingkan penggunaan *Region-Based Segmentation* (RBS) dan tidak, RBS merupakan metode untuk melakukan segmentasi bertingkat. Dari 10 kali percobaan tanpa menggunakan RBS, tingkat keberhasilan robot mendeteksi bola dengan benar adalah 30%, sedangkan dengan menggunakan metode RBS, tingkat keberhasilan robot mendeteksi bola adalah 100%. Oleh karena itu, penggunaan *Region-Based Segmentation* pada robot sepak bola sangat efektif untuk meningkatkan akurasi dan mengurangi *noise* yang berada di luar lapangan pertandingan.

Keyword—Segmentation, Region-Based Segmentation, Robot Pemain Sepak Bola, HSV.

I. PENDAHULUAN

Kemampuan utama dari robot pemain sepak bola adalah mendeteksi bola dan gawang. Untuk meningkatkan akurasi robot harus mampu membedakan antara *noise* dan objek yang dideteksi. Permasalahan yang sering terjadi adalah robot salah dalam mendeteksi objek yang diinginkan karena nilai threshold berbeda (perubahan intensitas cahaya) dan objek berada di luar lapangan dengan warna menyerupai objek (bola).

Salah satu solusi yang dapat dilakukan adalah mendeteksi objek yang hanya berada pada area lapangan saja (*Region-Base*). Dalam penelitian ini akan dibandingkan tingkat akurasi penggunaan RBS dan tidak untuk mendeteksi bola. Secara sederhana metode RBS adalah segmentasi objek yang dilakukan bertahap [1] [2]. Pengecekan di piksel sekitar bola, apabila terdapat warna lapangan, maka objek bola akan terdeteksi.

Tahapan pengolahan citra diawali dari konversi ruang warna RGB menjadi HSV [3], binerisasi citra berdasarkan threshold yang telah ditentukan [4] [5], hasil binerisasi diperbaiki dengan menggunakan morfologi citra berupa filling holes, area opening, dan erosi [6], luasan objek (moment) akan di cari titik tengahnya (centroid) sehingga didapatkan koordinat dari bola (sumbu x dan y), sudut bola, dan jarak bola.

Koordinat bola yang didapatkan akan menjadi input kinematik untuk pengaturan kecepatan putaran dan arah dari ketiga roda omni [7] [8].

II. METODOLOGI

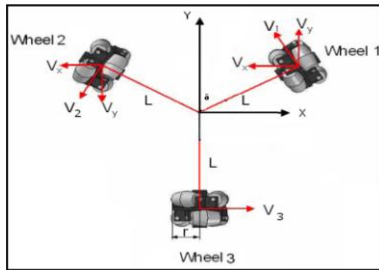
A. Perancangan Mekanik

Sistem penggerak robot dilengkapi dengan 3 roda omni yang masing-masing membentuk sudut 120° maka masing-masing roda memiliki sudut θ sebesar 60°. Diameter roda omni adalah 10 cm dengan tebal 3 cm. Badan robot terbuat dari aluminium dengan dimensi robot 52 x 50 x 70 cm. Gambar 1 adalah hasil rancangan mekanik yang telah dibuat.



Gambar 1. Desain Mekanik Robot

Untuk persamaan kinematic robot yang telah dibuat adalah sebagai berikut:



Gambar 2. Representasi Kinematika dari Sistem Pergerakan Three Omni-directional

Dimana

- X, Y : Sistem Koordinat Kartesian.
- V_x, V_y : Kecepatan roda ke sumbu X, Y
- $V_{i(1,2,3)}$: Vektor Putaran dari masing-masing roda omni.
- L : Jarak Pusat Robot ke titik pusat roda.
- r : Jari-jari dari roda omni.

Seperti pada Gambar 2 pergerakan dari roda omni memiliki 2 DOF pergerakan yaitu bergerak rotasi sesuai dengan putaran motor dan bergerak translasi karena pengaruh dari dorongan roda lainnya yang diasumsikan memiliki kecepatan V_x, V_y dan V_θ terhadap titik referensi koordinat kartesian. Hal ini dapat digunakan untuk menentukan kecepatan dari ketiga roda robot yang telah di rangkai menjadi satu seperti yang ditunjukkan pada Gambar 2, untuk mendapatkan kecepatan dari roda robot (V_i) dapat kita turunkan untuk mendapatkan kecepatan ketiga roda robot pada bidang koordinat kartesian robot. Untuk mendapatkan kecepatan dari masing-masing dari roda omni sama dengan kecepatan dari motor yang dikalikan dengan jari-jari dari roda omni yang dirangkai secara simetris dengan perbedaan sudut antar roda, seperti pada penelitian Raditya tahun 2014:

$$V_1 = -\frac{1}{3} \cos \Psi + \frac{1}{\sqrt{3}} \sin \Psi + \frac{L}{3} V_\phi \quad (2.1)$$

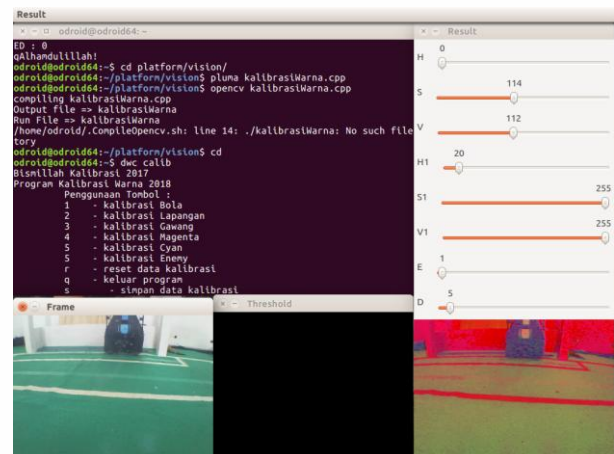
$$V_2 = -\frac{1}{3} \cos \Psi - \frac{1}{\sqrt{3}} \sin \Psi + \frac{L}{3} V_\phi \quad (2.2)$$

$$V_3 = \frac{2}{3} \cos \Psi + \frac{L}{3} V_\phi \quad (2.3)$$

B. Perancangan Software Kalibrasi

Sebelum melakukan pengolahan citra, perlu ditentukan jenis warna yang akan di deteksi dengan mengatur range nilai HSV. Proses kalibrasi warna menggunakan software yang berbeda untuk proses deteksi objek. Pemisahan software kalibrasi warna dengan program utama ini bertujuan untuk mempercepat proses pengolahan citra.

Langkah pertama untuk kalibrasi warna adalah proses pengambilan gambar dari kamera, dilanjutkan dengan mengkonversi ruang warna RGB ke HSV. Setelah proses konversi dari RGB ke HSV, dilakukan *thresholding* atau pemberian ambang batas pada ROI (Region of Interest) yaitu berdasarkan warna. Pemberian ambang batas ini dilakukan untuk menentukan nilai H max, S max, V max, H min, S min, V min, Dilasi (Morfologi), Erosi (Morfologi). Dilasi dan erosi ini bertujuan untuk mengurangi *noise* dengan menambah atau mengurangi ukuran segmentasi objek pada sekeliling objek.



Gambar 3. Software Kalibrasi Warna

C. Perancangan Software Deteksi Bola

- 1) Pengambilan citra (*Capture*): OpenCV menyediakan sebuah struktur data yaitu VideoCapture. Struktur data ini dapat dengan mudah memilih *index* kamera yang akan digunakan. Pada penelitian ini, ukuran frame yang digunakan adalah 544 x 288 piksel. Semakin besar ukuran frame, maka semakin jelas gambar yang didapatkan, namun kelemahannya adalah waktu proses yang relatif lama sehingga tidak cocok untuk robot yang membutuhkan data *realtime*.
- 2) Konversi RGB ke HSV: Proses ini dilakukan menggunakan fungsi `cvtColor()` untuk memperoleh citra HSV.
- 3) Thresholding nilai HSV : Setelah proses konversi dari RGB ke HSV, perlu dilakukan *thresholding* atau pemberian ambang batas pada ROI (*Region of Interest*) yaitu berdasarkan warna. Pemberian ambang batas ini dilakukan menggunakan trackbar

yang berisi nilai H max, S max, V max, H min, S min, V min, Dilasi (Morfologi), Erosi (Morfologi). Dilasi dan erosi ini bertujuan untuk mengurangi *noise* dengan menambah atau mengurangi ukuran segmentasi objek pada sekeliling objek.

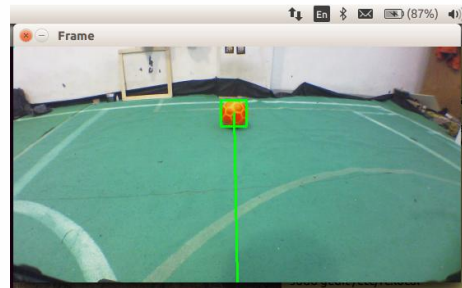
- 4) Memperoleh kontur: citra biner hasil thresholding diatas kemudian dilakukan pencarian kontur untuk setiap piksel yang bernilai 255. Proses pencarian kontur dari objek bola ini dilakukan untuk mempermudah deteksi bola berdasarkan bentuk.
- 5) Menggambar persegi pada citra: Proses penggambaran persegi ini bertujuan untuk mempermudah mendapatkan koordinat dari bola. Proses penggambaran persegi ini yaitu berdasarkan titik yang bersinggungan dengan kontur bola.
- 6) Penerapan Region-Based Segmentation: Proses ini dilakukan pengecekan piksel sekitar objek bola apakah masuk kedalam area pixel lapangan atau tidak, yaitu dengan mengecek area pixel bagian atas, kanan, bawah, kiri, pojok kanan atas, pojok kanan bawah, pojok kiri atas, pojok kiri bawah. Proses pengecekan piksel sekitar inilah yang disebut *Region-Based Segmentation*.

```

For K=0 to N
  Memberi kotak pada kontur
  Set minimum radius
  Counter = 0
  If titik Y kotak - 10 > 0
    Cek piksel lapangan atas kotak (Y kotak
    - 10, X kotak + Lebar kotak / 2)
    If piksel lapangan = 255
      Counter++
    End If
  End If
End For

If counter > 0
  ballFound = true
End If
    
```

Pseudocode diatas merupakan contoh pengecekan piksel bagian atas bola, pada penerapannya perlu dilakukan pengecekan piksel seluruh area sekitar bola agar lebih akurat.



Gambar 4. Software Deteksi Bola

- 7) Memperoleh nilai koordinat bola: Nilai koordinat bola berupa x dan y ini yang kemudian dikirim menuju mikro kontroler untuk menggerakkan aktuator.

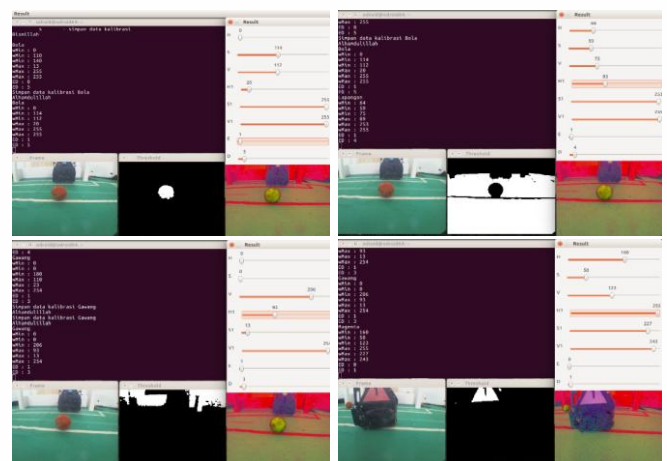


Gambar 5. Diagram Blok Vision Secara Umum

III. HASIL PENGUJIAN

A. Pengujian Software Kalibrasi Warna

Pengujian software kalibrasi warna ini bertujuan untuk mengetahui seberapa akurat hasil kalibrasi yang dilakukan dengan cara membandingkan nilai HSV dari software kalibrasi warna dengan tabel warna HSV. Pengujian dilakukan menggunakan warna dasar yang ada pada lapangan robot sepak bola yaitu warna oranye, hijau, putih, cyan, dan magenta.



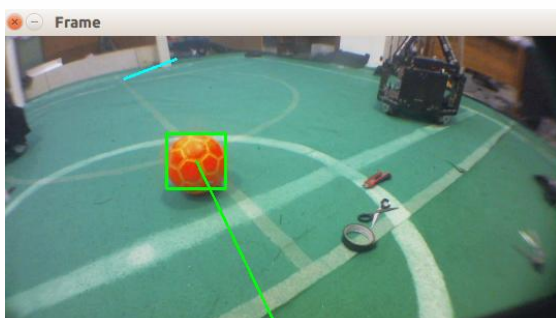
Gambar 6. Hasil Pengujian Kalibrasi Warna

Tabel 1. Hasil Pengujian Software Kalibrasi Warna

No	Tabel HSV			Software Kalibrasi			Ket
	H	S	V	H	S	V	
1	21.25	255	127.5	0-20	114-255	112-255	Oranye
2	84.9	255	63.75	64-93	59-255	75-255	Hijau
3	0	0	255	0-93	0-13	200-254	Putih
4	127.5	255	255	93-127	71-167	182-255	Cyan
5	211.5	255	255	160-255	50-227	123-243	Magenta

B. Pengujian Deteksi Bola dengan RBS

Dalam pengujian ini, dilakukan menggunakan metode RBS dan tanpa menggunakan RBS. Dalam deteksi bola ini menggunakan range 0-255 dengan nilai H : 0-20, S : 114-255, dan V : 112-255. Gambar 7 dan 8 adalah deteksi bola dengan menggunakan RBS.


Gambar 7. Deteksi Bola Di Luar Lapangan

Gambar 8. Deteksi Bola di Dalam Lapangan

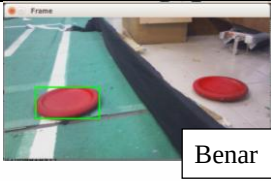
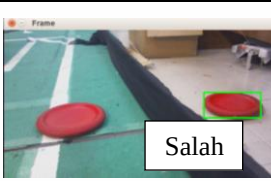
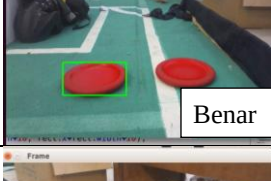
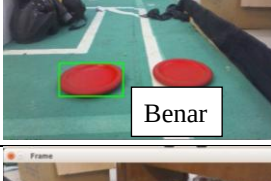
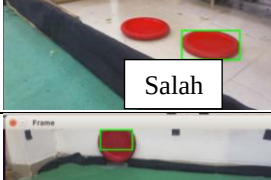
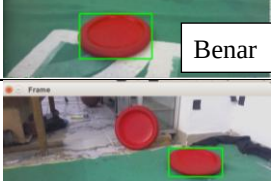
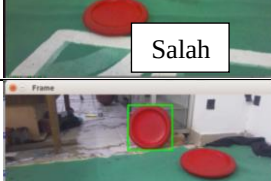
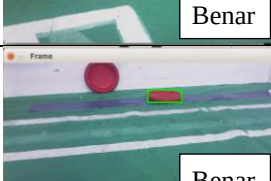
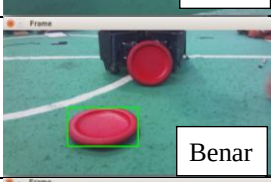
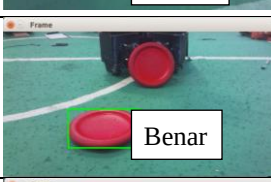
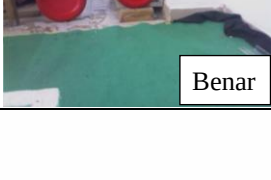
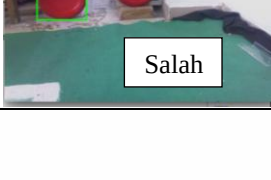
Pengujian selanjutnya yaitu dengan menggunakan thresholding tanpa RBS, dengan kondisi bola berada di luar lapangan.


Gambar 9. Deteksi Bola Tanpa RBS.

Penggunaan thresholding tanpa RBS pada Gambar 9 menyebabkan seluruh objek yang memiliki warna menyerupai bola masih tetap terdeteksi. Berbeda dengan menggunakan RBS, bola yang berada di luar lapangan tidak akan terdeteksi karena dianggap sebagai *noise*. Tabel 2 adalah detail hasil pengujian dengan menggunakan RBS dan tidak dengan asumsi objek yang harus di deteksi adalah objek yang berada di dalam lapangan (area hijau).

Dari Tabel 2 dapat diketahui bahwa dengan menggunakan metode RBS dapat membedakan 10 kondisi objek dengan benar (100%), sedangkan tanpa menggunakan RBS hanya mampu membedakan 3 kondisi objek dengan benar dari 10 kondisi (30%). Jadi penggunaan metode region-base sangat optimal untuk dapat membedakan antara objek yang dituju dengan *noise*.

Tabel 2. Hasil Pengujian Deteksi Objek Dengan Menggunakan RBS dan Tidak

No	Menggunakan Region-base	Tanpa Region-base
1	 Benar	 Salah
2	 Benar	 Salah
3	 Benar	 Benar
4	 Benar	 Salah
5	 Benar	 Salah
6	 Benar	 Salah
7	 Benar	 Salah
8	 Benar	 Benar
9	 Benar	 Benar
10	 Benar	 Salah

IV. KESIMPULAN DAN SARAN

Penggunaan metode RBS untuk mendeteksi bola pada robot sepak bola lebih efektif daripada tanpa menggunakan RBS dengan tingkat akurasi 100% sedangkan tanpa RBS adalah 30%. Dari pengujian yang dilakukan diluar lapangan hijau, penggunaan metode RBS dapat meminimalkan *noise* sehingga objek berwarna orange diluar lapangan tidak terdeteksi, sedangkan tanpa menggunakan RBS, semua warna orange yang berada di luar lapangan masih terdeteksi.

Namun kelemahan penggunaan RBS yaitu apabila bola berada di ujung lapangan, bola kadang tidak dapat terdeteksi karena bola dianggap berada di luar lapangan, namun untuk mengatasi hal ini, diperlukan pengecekan di seluruh area sekitar bola yang mengakibatkan proses deteksi bola cukup memakan alokasi CPU yang besar.

V. REFERENSI

- [1] Q. Salih, "Region based segmentation technique and algorithms for 3D image," in *Proceedings of the Sixth International Symposium on Signal Processing and its Applications (Cat.No.01EX467)*, Kuala Lumpur, Malaysia, Malaysia, 2001.
- [2] H. G. Kaganami, "Region-Based Segmentation versus Edge Detection," in *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Kyoto, Japan, 2009.
- [3] G. P. A. P. A. K. Irfan Marzuqi, "Segmentasi dan Estimasi Jarak Bola dengan Robot Menggunakan Stereo Vision," in *5th Indonesian Symposium on Robotic Systems and Control 2017*, Bandung, 2017.
- [4] Puneet, "Binarization Techniques used for Grey Scale Images," *International Journal of Computer Applications (0975 – 8887)*, vol. 71, pp. 8-11, 2013.
- [5] A. Mukherjee, "Enhancement of Image Resolution by Binarization," *International Journal of Computer Applications (0975 – 8887)*, vol. 10, pp. 15-19, 2010.
- [6] M. P. Beham, "Morphological image processing approach on the detection of tumor and cancer cells," in *2012 International Conference on Devices, Circuits and Systems (ICDCS)*, Coimbatore, India, 2012.
- [7] A. S. Al-Ammri, "Control of Omni-Directional Mobile Robot Motion," *Al-Khwarizmi Engineering Journal*, vol. 6, no. 4, pp. 1-6, 2010.
- [8] R. Artha, "Implementasi Robot Three Omni-Directional Menggunakan Kontroler Pid Pada Robot Kontes Robot Abu Indonesia (KRAI)," in *Publikasi Jurnal Skripsi*, Universitas Brawijaya, 2014.
- [9] P. Hidayatullah, *Pengolahan Citra Digital (Teori dan Aplikasinya)*, Bandung: Informatika, 2017.
- [10] Ristekdikti, *Buku Panduan Kontes Robot Sepak Bola Indonesia Beroda (KRSBI Beroda) 2017*, Jakarta: Ristekdikti, 2016.

- [11] T. Röfer, "Region-Based Segmentation with Ambiguous Color Classes and 2-D Motion Compensation," *RoboCup 2007: Robot Soccer World Cup*, vol. XI, pp. 369-376, 2007.
- [12] T. B. M. V. James Bruce, "Fast and Inexpensive Color Image Segmentation for Interactive Robots," Takamatsu, Japan, 2000.
- [13] F. Albregtsen, Region and Edge Based Segmentation, INF 4300 – Digital Image Analysis, 2010.
- [14] J. Zhang, "ROI Based Natural Image Retrieval Using Color and Texture Feature," in *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, Haikou, China, 2007.

Penggunaan Kamera Global untuk Menentukan Koordinat Robot pada Lapangan

Pandu Surya Tantra, Muhammad Azhar Ismail, Dhiaul Ma'ruf, Ivan Falahul Alam, Maulana Falithurrahman, Muhammad Revo Khairullah, Achmad, Dzulfikar Ahmad Samhan, Helga Dendy Ardianika, Kamal Arief, Deksaraka Danier, Dwiki Febrianto, Reynald Sapoeira, Ilham Dwiki Ramadhan, Aulia Rayimas Tinkar, Fatma Cahyani, Rudy Dikairono
Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111
Email: rudydikairono@gmail.com

Abstrak – Penentuan posisi robot adalah salah satu masalah utama dalam pembuatan robot sepak bola. Apabila robot tidak mengetahui posisinya pada lapangan, maka koordinasi antar robot dan tindakan selanjutnya akan sulit ditentukan. Salah satu cara untuk mengetahui posisi robot adalah dengan menggunakan metode *odometry*. Komponen utama *odometry* adalah *rotary encoder* dan kompas. *Rotary encoder* digunakan untuk mengetahui seberapa jauh robot telah berpindah dari posisi awal, sedangkan kompas digunakan sebagai penunjuk ke arah manakah robot berpindah. Namun metode ini mudah terpengaruh slip sehingga menghasilkan nilai *error* terintegral yang membuat nilai *error* posisi semakin besar seiring dengan semakin jauhnya robot berpindah. Untuk itu diperlukan suatu metode yang dapat menentukan posisi robot dengan benar tanpa terpengaruh *odometry*. Cara paling mudah untuk mewujudkan metode tersebut adalah dengan menggunakan kamera global sebagai sensor luar. Perlu diketahui bahwa metode ini hanya dipergunakan untuk riset selama latihan dan tidak dipergunakan dalam pertandingan sesungguhnya.

Kata kunci: *error, omnidirectional, artificial neural network, global positioning*

I. PENDAHULUAN

Robot sepak bola beroda difungsikan untuk dapat bermain sepak bola dengan alat penggerak berupa roda. Robot dirancang dan dibuat dengan memiliki roda sebagai penggerak dan diprogram untuk dapat melakukan strategi penyerangan dan bertahan selayaknya permainan sepakbola. Oleh karena itu tentunya robot harus bisa melakukan beberapa gerakan sepakbola seperti: mencari bola, mengejar bola, mencari gawang lawan, menjaga gawang, merebut bola dari lawan, mengumpan bola ke kawan dan mencetak goal. Untuk memenuhi fungsinya, robot harus mampu mengetahui posisinya pada saat itu. Kamera global digunakan untuk mengetahui posisi robot sebenarnya yang tidak terpengaruh error.

II. TEORI PENUNJANG

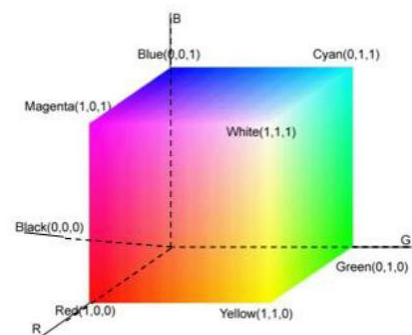
A. Pengolahan Citra Digital

Citra dapat didefinisikan sebagai fungsi $f(x,y)$, di mana x dan y adalah koordinat bidang Cartesius dan f adalah amplitudo pada setiap titik

koordinat (x,y) atau dapat diartikan sebagai intensitas atau tingkat keabuan pada titik koordinat tersebut. Apabila sebuah citra memiliki nilai f pada keseluruhan bidang (x,y) dengan range yang terbatas dan bernilai diskrit, maka citra tersebut merupakan citra digital. Pemrosesan citra digital dapat didefinisikan sebagai pemrosesan citra dengan menggunakan komputer digital. Citra digital tersusun atas elemen dengan jumlah terbatas dan masing-masing elemen tersebut menempati koordinat tertentu dan memiliki amplitudo/nilai. Elemen-elemen tersebut kemudian disebut dengan elemen citra atau *pixel*.

Citra dengan Ruang Warna RGB

Secara umum mata manusia dapat membedakan warna merah, hijau, dan biru (*Red, Green, Blue*) sehingga warna tertentu dapat dipandang sebagai kombinasi dari ketiga warna tersebut. Ruang warna RGB biasa digunakan pada sistem grafik komputer dan kamera digital. Ruang warna ini dimodelkan dalam bentuk kubus tiga dimensi dengan pusat warna merah, hijau, dan biru berada pada ujung sumbu R, G, dan B seperti pada gambar 1.

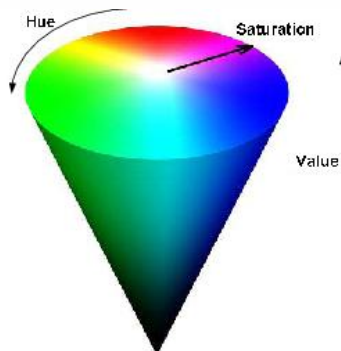


Gambar 1. Model ruang warna RGB

Citra dengan Ruang Warna HSV

HSV merupakan ruang warna yang merepresentasikan warna seperti yang dilihat oleh mata manusia. HSV kependekan dari *Hue*, *Saturation*, dan *Value* merupakan besaran warna yang biasa digunakan dalam pengolahan citra digital. Ruang warna HSV dimodelkan dengan kerucut terbalik, pusat warna hitam berada pada

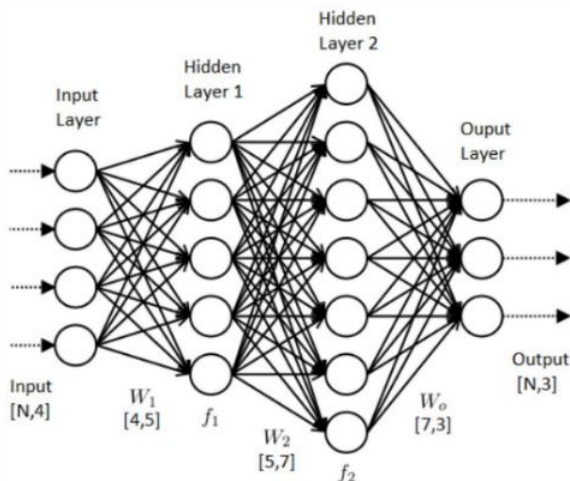
dasar kerucut dan pusat warna putih berada pada pusat tutup kerucut dikelilingi oleh campuran komposisi warna merah, hijau, dan biru seperti ditunjukkan pada gambar 2 [1].



Gambar 2. Model Ruang Warna HSV

B. Jaringan Syaraf Tiruan (Artificial Neural Network)

Machine learning merupakan istilah yang dipakai dalam bidang keteknikan komputer di mana komputer dapat melakukan suatu tugas tanpa benar-benar diprogram. Dengan menggunakan sampel data dan algoritma tertentu, komputer akan dibuat seolah-olah belajar dari sampel data yang diberikan. Kualitas dan jumlah sampel data yang diberikan sangat mempengaruhi tingkat akurasi *machine learning*. Semakin benar dan semakin banyak jumlah sampel data yang diberikan maka tingkat akurasi dari *machine learning* akan semakin tinggi, dan juga berlaku sebaliknya.



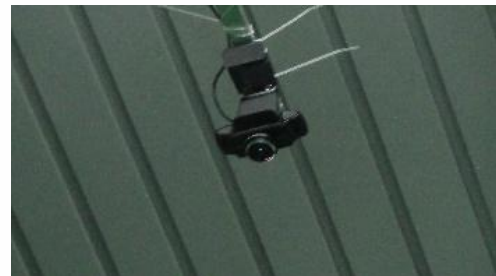
Gambar 3. Struktur umum jaringan syaraf tiruan

Salah satu metode *machine learning* yang populer dan paling banyak digunakan saat ini adalah jaringan syaraf tiruan (*artificial neural network*). Jaringan syaraf tiruan terdiri dari sejumlah *node* yang tersusun berlapis-lapis (*layers*) dan saling terhubung satu sama lain. Masing-masing *node* memiliki koefisien pembobotan yang disebut *weight*. Dan masing-masing *layer* memiliki nilai konstanta yang disebut *bias*. Struktur umum jaringan syaraf tiruan ditunjukkan pada gambar 3.

Nilai *weight* dan *bias* digunakan jaringan syaraf tiruan untuk menghitung keluaran dari masukan yang diberikan. Keluaran *layer* pertama digunakan sebagai masukan *layer* kedua, begitu juga keluaran *layer* kedua digunakan sebagai masukan *layer* ketiga, dan seterusnya. Sebelumnya, sebuah fungsi aktivasi diaplikasikan pada masing-masing *layer* untuk mengukur seberapa aktif *node* pada *layer* tersebut. Jaringan syaraf tiruan perlu dilatih agar nilai *weight* dan *bias* nya beradaptasi sehingga dapat memberikan nilai keluaran yang sesuai untuk masukan yang diberikan. Terdapat dua metode pelatihan jaringan syaraf tiruan, yaitu *supervised learning* dan *unsupervised learning* [2]. Metode pelatihan yang umum dilakukan adalah *supervised learning*

C. Kamera Global

Kamera Global yang digunakan adalah kamera Logitech C922 dengan lensa *Fish eye* yang terpasang diatas lapangan dengan tinggi kurang lebih 4.5 meter dari lapangan. Pada ketinggian tersebut, kamera mampu menangkap citra lapangan secara keseluruhan. Keuntungan ini lah yang kita gunakan untuk mengetahui posisi robot pada lapangan yang sebenarnya.



Gambar 4. Kamera global

III. METODE

A. Perencanaan Mekanik

Mekanik robot didesain menggunakan bahan dasar aluminium padat untuk bodinya dan akrilik untuk beberapa bagian lain semisal penyangga sensor dan alat elektronik lainnya. Desain dari robot ditunjukkan pada Gambar. Dibagian atas diberikan silinder tabung akrilik sebagai pelindung kamera dan kamera omnidireksional ditempatkan didalamnya. Kamera omnidireksional ini didesain sedemikian rupa sehingga kamera ini menghadap keatas, dan pada bagian atas kamera merupakan cermin cembung yang berfungsi agar jangkauan kamera lebih luas serta mengurangi titik buta terutama di daerah bagian dekat robot. Robot ini didesain mengerucut juga merupakan bagian dari strategi untuk membantu kamera omnidireksional menjalankan fungsinya dengan lebih baik dan

efektif. Dengan metode ini, performa maksimum dari kamera yang dimiliki robot dapat mencakup hampir keseluruhan bagian lapangan.



Gambar 5. Design 3D robot

B. Perencanaan Metode Pengambilan Data

Kelemahan terbesar dalam penggunaan kamera global dengan *Fish eye* seperti ini adalah tidak linear nya hasil pembacaan data yang didapat. Hal ini sangat berpotensi meningkatkan *error* pada data yang telah diambil. Ketidaklinearan ini disebabkan hasil pengambilan citra menggunakan *Fish eye* akan menyebabkan lapangan terlihat melengkung seperti pada gambar 6.



Gambar 6. Pengambilan citra lapangan

Sehingga data koordinat yang telah diambil menggunakan kamera global tidak dapat merepresentasikan koordinat robot yang sebenarnya. Untuk itulah diperlukan ANN untuk mendapatkan koordinat robot sebenarnya.

Pengambilan data dilakukan dengan cara membandingkan tiap-tiap koordinat robot pada titik uji dengan koordinat yang terbaca oleh kamera. Titik uji diambil setiap 50 cm pada lebar lapangan, dan 100 cm pada panjang lapangan. Sehingga didapat 117 titik uji.

C. Perencanaan Software Learning dan Software Pengolahan Data Kamera Global

Pengambilan data posisi pada kamera global dilakukan melalui beberapa tahap sebagai berikut :

1. Mengubah ruang warna citra lapangan menjadi ruang warna HSV
2. Memberi nilai ambang HSV untuk segmentasi lapangan
3. Memotong dan menormalisasi citra lapangan
4. Memberi nilai ambang HSV untuk segmentasi robot
5. Mendapatkan koordinat pixel robot pada lapangan

Jaringan syaraf tiruan yang digunakan menggunakan 117 data kalibrasi yang telah diambil saat pengambilan data. Dengan 2 input dan 2 output Proses pelatihan jaringan syaraf tiruan dilakukan dengan memanfaatkan GPU NVIDIA GeForce 840M.

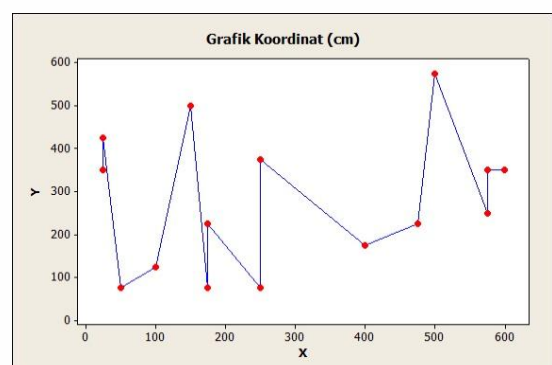
IV. HASIL DAN DISKUSI

A. Pengujian Posisi Global

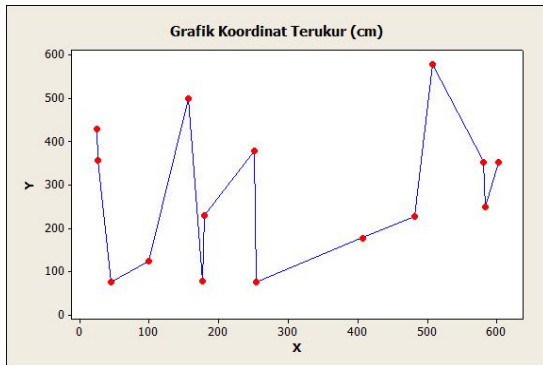
Pengujian dilakukan dengan memosisikan robot pada titik-titik di lapangan dan dibandingkan dengan keluaran ANN pada kamera global

Tabel 1. Hasil pengujian posisi global

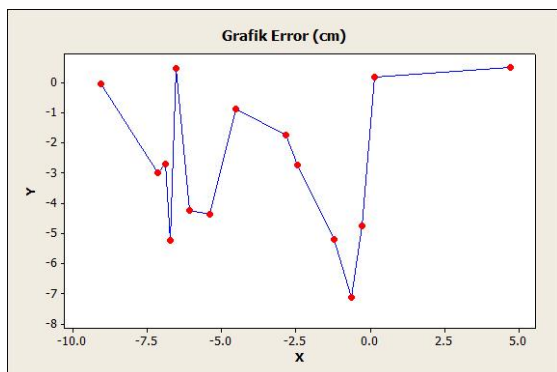
Koordinat (cm)		Koordinat Terukur (cm)		Error (cm)		Error Koordinat (cm)
X	y	x	y	x	y	
25	350	26	357	-0.62	-7.14	7.17
25	425	25	430	-0.28	-4.77	4.77
50	75	45	75	4.73	0.49	4.76
100	125	100	125	0.14	0.18	0.23
150	500	157	500	-6.52	0.47	6.54
175	75	177	78	-2.45	-2.73	3.67
175	225	180	229	-5.38	-4.38	6.94
250	75	254	76	-4.50	-0.87	4.58
250	375	251	380	-1.21	-5.21	5.35
400	175	407	178	-6.89	-2.70	7.40
475	225	482	228	-7.13	-2.99	7.73
500	575	507	580	-6.71	-5.24	8.51
575	250	584	250	-9.04	-0.04	9.04
575	350	581	354	-6.08	-4.24	7.41
600	350	603	352	-2.84	-1.75	3.33
RMSE (cm)				5.11	3.59	6.25



Gambar 7. Grafik perbandingan koordinat



Gambar 8. Grafik perbandingan koordinat terukur



Gambar 9. Grafik perbandingan error

Tabel diatas menggambarkan perbandingan posisi sebenarnya dengan posisi pada pixel yang diambil. Dengan menggunakan ANN kita bisa mendapatkan nilai perbandingan 2 dimensi x dan y pada tiap bagian lapangan, tidak hanya pada titik uji saja dengan tingkat keberhasilan yang tinggi.

V. KESIMPULAN

Metode penggunaan kamera global untuk mengetahui posisi robot secara real ini dapat diterapkan dengan baik untuk mengetahui error posisi yang terjadi akibat adanya *integral error* pada sensor dalam robot seperti *odometry* atau sensor-sensor lain. Sistem penentuan posisi global mampu memberikan data posisi dengan rata-rata *error* sebesar 6,25cm. Namun sistem ini sering kali gagal mendeteksi robot jika robot berada di pinggir lapangan. Hal ini dikarenakan kepala robot yang keluar dari jangkauan segmentasi lapangan.

VI. DAFTAR PUSTAKA

- [1] R. C. Gonzalez dan R. E. Woods, *Digital image processing*, 3. ed., internat. ed. Upper Saddle River, NJ: Pearson, 2010.
- [2] T. Munakata, *Fundamentals of the new artificial intelligence: neural, evolutionary, fuzzy and more*, 2. ed. London: Springer, 2008.

Implementasi Protokol TCP dan UDP dalam Sistem Komunikasi *Base Station* Robot Sepak Bola Beroda

Ata Lukman Fadlah¹, Achmad Fiqhi Ibadillah², Diana Rahmawati³

^{1,2,3}Universitas Trunojoyo Madura
Bangkalan, Indonesia

¹atalukman@gmail.com, ²fiqhi.achmad@gmail.com, ³diana_rahmawati@yahoo.com

Abstrak – Sistem komunikasi pada robot sepak bola berperan sebagai perantara pengiriman data antar robot selama pertandingan. Pada penelitian ini diimplementasikan sistem komunikasi didalam sebuah aplikasi *Base Station* robot sepak bola yang dibangun menggunakan perangkat lunak Qt-Creator berbasis bahasa pemrograman C++. Berdasarkan hasil pengujian yang dilakukan, *Base Station* berhasil menerima data dari *Referee Box* menggunakan protokol TCP (*Transmission Control Protocol*) sekaligus dapat meneruskan data yang diterima ke robot menggunakan protokol UDP (*User Datagram Protocol*). Selain itu *Base Station* dapat menerima data yang dikirimkan oleh robot. Dari 5 kali pengiriman data sampel didapatkan persentase data hilang/rusak sebesar 1,68%, dengan persentase terkecil 0% pada pengiriman 100 data, dan persentase terbesar yaitu 3% pada pengiriman 1000 data.

Kata kunci: *Base Station*; TCP; UDP; Qt-Creator; C++

I. PENDAHULUAN

Perkembangan dunia robotika global yang semakin pesat menuntut semakin berkembang pula robotika nasional, salah satunya robot sepak bola beroda yang setiap tahun diadakan oleh Kementerian Riset, Teknologi dan Pendidikan Tinggi dalam bentuk sebuah kontes atau pertandingan.. Kontes robot sepak bola beroda dipertandingkan antara 2 tim dimana masing-masing tim menggunakan 2 robot penyerang dan 1 robot kiper. Pada saat pertandingan, robot akan dikontrol dan diperintah berdasarkan instruksi wasit melalui perangkat lunak yang disebut *Referee Box*. Masing-masing robot diharuskan dapat menerjemahkan aksi berdasarkan intruksi yang diberikan, seperti *Free Kick*, *Goal Kick*, *Kick Off* dan sebagainya. Oleh karena itu diperlukan adanya sebuah sistem komunikasi untuk menjembatani antara *Referee Box* dengan robot sehingga robot dapat menerima dan menerjemahkan instruksi yang diberikan. Berdasarkan latar belakang ini dibuatlah aplikasi *Base Station* untuk mengakomodir tugas-tugas tersebut.

Aplikasi *Base Station* dibangun menggunakan

perangkat lunak Qt-Creator dengan bahasa pemrograman C++ dengan memanfaatkan protokol TCP (*Transmission Control Protocol*) untuk menerima data dari *Referee Box* dan protokol UDP (*User Datagram Protocol*) untuk meneruskan data yang diterima dari *Base Station* ke robot. Protokol TCP dioptimasi untuk kehandalan pengiriman serta menjamin pengiriman data, sehingga protokol ini cocok digunakan sebagai perantara antara *Base Station* dengan *Referee Box* dimana data yang dikirimkan oleh *Referee Box* hanya sekali pengiriman. Sedangkan protokol UDP dioptimasi untuk kecepatan pengiriman, sehingga protokol ini cocok digunakan sebagai protokol pengiriman antara *Base Station* dengan robot dimana pengiriman data dilakukan secara *realtime* dan terus menerus sehingga memerlukan mekanisme pengiriman yang cepat. Dengan dibuatnya aplikasi *Base Station* ini diharapkan dapat mengefektifkan fungsi *Referee Box* sebagai wasit serta meminimalisir campur tangan manusia secara langsung selama pertandingan.

II. DASAR TEORI

A. Referee Box

Referee Box merupakan sebuah perangkat lunak yang berfungsi untuk mengatur dan mengendalikan jalannya pertandingan. *Referee Box* mewakili peran dari wasit dalam sebuah pertandingan robot sepak bola dengan mengirimkan set instruksi berupa karakter-karakter yang akan diterjemahkan oleh robot. *Referee Box* dapat menerima koneksi dari dua buah *Base Station* sekaligus, dimana kedua *Base Station* tersebut mewakili 2 tim yang sedang bertanding. *Referee Box* menggunakan TCP (*Transmission Control Protocol*) sebagai protokol komunikasi dalam mengirimkan set perintah kepada *Base Station*, setelah instruksi diterima oleh *Base Station* kemudian diteruskan kepada robot untuk dieksekusi.



Gambar 1. GUI aplikasi *referee box* [1]

B. Base Station

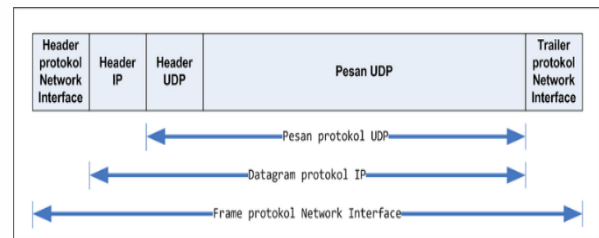
Base Station merupakan sebuah perangkat lunak yang dirancang sedemikian rupa untuk mengontrol, mengkoordinasi serta memonitoring kondisi robot selama bertanding. *Base Station* diharuskan dapat bekerja secara otomatis tanpa adanya interupsi selama pertandingan, oleh karena itu dibutuhkan koneksi yang handal serta kemampuan dalam pemrosesan data yang diterima secara cepat. Aplikasi *Base Station* terhubung dengan *Referee Box* melalui kabel *Ethernet* sedangkan terhubung dengan robot melalui perantara *wireless* (WLAN).

C. Transmission Control Protocol (TCP)

TCP (*Transmission Control Protocol*) adalah sebuah protokol yang berada di lapisan transport yang berorientasi sambungan (*connection-oriented*) dan bisa diandalkan (*reliable*). TCP dioptimalkan untuk kehandalan dalam hal komunikasi bukan kecepatan. TCP memberikan jaminan atas pesan-pesan yang disampaikan dari sebuah komputer ke komputer yang lain. TCP menggunakan mekanisme *handshake*.

D. User Datagram Protocol (UDP)

UDP (*User Datagram Protocol*) merupakan salah satu protokol lapisan transpor TCP/IP yang mendukung komunikasi yang tidak andal (*unreliable*) dan tanpa koneksi (*connectionless*). UDP dioptimasi untuk kecepatan pengiriman, berbeda dengan TCP, UDP tidak menjamin pengiriman data, aplikasi harus mengimplementasikan mekanisme *error recovery*-nya sendiri jika memerlukan mekanisme tersebut. Format segmen pengiriman UDP adalah sebagai berikut.



Gambar 2. Segmen UDP

Perbedaan antara segmen TCP dan segmen UDP yaitu pada ukuran header masing-masing protokol. Ukuran header TCP yaitu sebesar 20 byte dimana didalamnya terdapat banyak field yang harus dikirimkan oleh TCP untuk memastikan data yang diterima benar-benar sama dengan data yang dikirim. Sedangkan pada header UDP yang hanya memiliki besar 8 byte, hanya terdapat 4 field yang dikirimkan sehingga pengiriman menggunakan protokol ini tidak dapat diandalkan (*unreliable*) dan rentan *corrupt* atau *packet loss* pada saat pengiriman.

E. QT Creator

QT Creator merupakan cross-platform IDE (*Integrated Development Environment*) yang lengkap untuk pengembangan aplikasi dengan target berbagai platform desktop dan berbagai platform mobile. QT Creator merupakan IDE yang menyediakan tools untuk mendisain dan mengembangkan aplikasi menggunakan framework aplikasi Qt

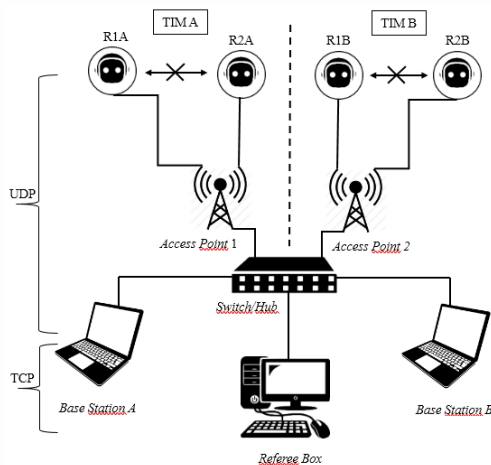


Gambar 3. Logo Qt Creator

III. DESAIN SISTEM

A. Skema Komunikasi

Skema komunikasi yang digunakan mengacu pada *rule* dan ketentuan yang termuat dalam Kontes Robot Sepak Bola Internasional, yaitu *Rococup Middle Size League* (MSL) dengan menyesuaikan kondisi di Kontes Robot Sepak Bola Indonesia (KRSBI). Skema komunikasi antara *Referee Box*, *Base Station*, dan juga robot selama bertanding dapat dilihat pada gambar 4 berikut.



Gambar 4. Skema Komunikasi

Sistem komunikasi antara *Referee Box* dengan *Base Station* dihubungkan dengan perantara kabel *Ethernet* (LAN) menggunakan protokol *TCP* (*Transmission Control protocol*), sedangkan antara *Base Station* ke masing-masing robot dihubungkan melalui perantara *wireless* (WLAN), dimana dalam penelitian ini penulis menggunakan *UDP* (*User Datagram Protocol*) sebagai protokol komunikasinya. Protokol *UDP* dipilih karena protokol ini dioptimasi untuk kecepatan pengiriman, sehingga protokol ini cocok digunakan sebagai perantara antara *Base Station* dengan robot dimana pengiriman data dilakukan secara *realtime* dan terus menerus sehingga memerlukan mekanisme pengiriman yang cepat. Robot selama betanding tidak diperkenankan melakukan komunikasi langsung antar robot, sehingga untuk dapat berkomunikasi robot diharuskan untuk melewati *Base Station* terlebih dahulu secara *wireless*. Standard *wireless* yang digunakan mengacu pada persyaratan yang ditetapkan oleh IEEE 802.11a/b/g/n.

B. Desain Aplikasi Base Station

Aplikasi *Base Station* dibuat menggunakan perangkat lunak Qt-Creator berbasis bahasa pemrograman C++ yang berjalan pada sistem operasi Linux Ubuntu 14.05. Aplikasi dibangun dengan menggabungkan beberapa *widget* kedalam sebuah GUI (*Graphical User Interface*) yang utuh.



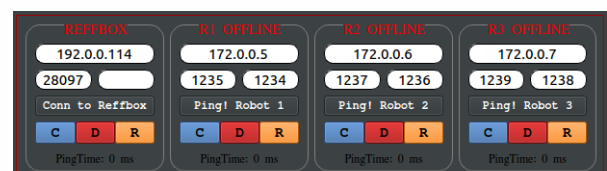
Gambar 5. GUI aplikasi base station

Seperti terlihat pada Gambar 5, dimana aplikasi *Base Station* merupakan gabungan dari beberapa *widget* yang dikemas menjadi sebuah aplikasi utuh guna menjalankan beberapa fungsi sekaligus. Diantara beberapa *widget* utama tersebut antara lain lapangan, koneksi, internal *Referee Box*, dan juga debug.



Gambar 6 Widget lapangan

Gambar 6 diatas merupakan tampilan dari *widget* lapangan. *Widget* ini berfungsi untuk menampilkan posisi robot yang sesungguhnya selama dilapangan. Hal ini dimaksudkan agar mempermudah dalam penyusunan strategi bermain serta memonitoring seluruh pergerakan robot.



Gambar 7. Widget koneksi

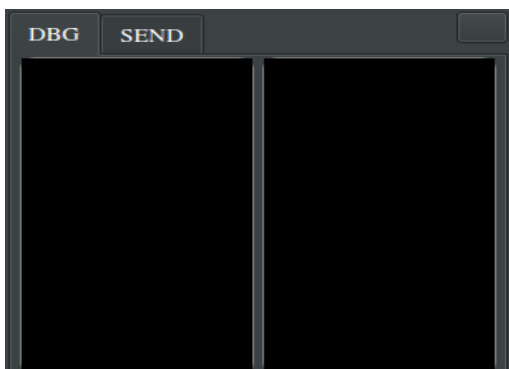
Widget ini berfungsi untuk mempermudah tim dalam koneksi antara *Base Station* ke *Referee Box* maupun dari *Base Station* ke masing-masing robot. Terdapat 4 buah *space* dimana masing-masing *space* mewakili 1 alamat IP *Referee Box* dan 3 *space* lain mewakili alamat IP 3 robot. *Widget* ini mengatur seluruh koneksi yang terjadi dalam *Base*

Station, dimana menggunakan protocol TCP (*Transmission Control Protocol*) untuk menghubungkan antara *Referee Box* dengan *Base Station*, sedangkan antara *Base Station* ke masing-masing robot menggunakan protocol UDP (*User Datagram Protocol*).



Gambar 8 Widget internal referee box

Widget ini berfungsi sebagai pengontrol jalannya pertandingan selayaknya *Referee Box official MSL*. Widget ini dibuat guna memudahkan tim dalam mengatur dan mengontrol permainan tanpa harus selalu terkoneksi dengan *Referee Box* secara terpisah. Widget ini mengirimkan set karakter yang sama dengan *Referee Box official* sehingga memudahkan robot dalam menerjemahkan instruksi yang diberikan.



Gambar 9. Widget debug

Gambar 9 yaitu widget debug. Widget ini berfungsi untuk merekam seluruh aktifitas *Base Station* serta lalu lintas data yang terjadi selama pertandingan. Perekaman ini dapat digunakan sebagai *debugging* dan bahan analisa jika sewaktu-waktu terjadi error maupun disaat sistem bekerja secara tidak normal.

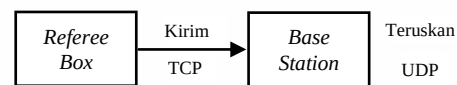
IV. HASIL DAN ANALISA

Pengujian dan analisa dimaksudkan untuk mengetahui sejauh mana aplikasi yang dibuat dapat bekerja sesuai fungsi dan tugasnya, serta didapatkan analisa sistem secara keseluruhan untuk pengembangan yang berkelanjutan. Pengujian dilakukan dengan cara mengirimkan beberapa data

sebagai sampel, baik dari *Referee Box* ke *Base Station* maupun dari robot ke *Base Station*.

A. Pengujian Pengiriman Data dari Referee Box

Dalam pengujian ini dilakukan pengiriman data dari *Referee Box* ke *Base Station* untuk mengetahui apakah data yang dikirim dapat diterima dengan baik. Pengujian dilakukan menggunakan 1 buah laptop yang sudah terinstall aplikasi *Referee Box official* dari *Robocup Middle Size League (MSL)*, 1 buah laptop yang sudah terprogram perangkat lunak *Base Station* serta 1 buah laptop sebagai robot. Alur pengiriman data dari *Referee Box* hingga sampai ke robot dapat dilihat pada gambar 10 berikut.



Gambar 10. Alur Pengiriman Data

Perangkat lunak *Base Station* terhubung ke *Referee Box* menggunakan alamat IP dan juga port yang disediakan oleh *Referee Box*, yaitu IP address laptop yang digunakan serta port default *Referee Box*, yaitu port 28097. Begitu juga dengan koneksi ke robot sama halnya dengan pengkoneksian sebelumnya namun menggunakan port yang berbeda. Koneksi antara *Referee Box* dengan *Base Station* menggunakan protokol TCP (*Transmission Control Protocol*) melalui perantara *Ethernet (LAN)* sedangkan antara *Base Station* dengan robot menggunakan protokol UDP (*User Datagram Protocol*) dengan perantara *wireless (WLAN)*. Proses *capture packet* dilakukan dengan bantuan perangkat lunak *Wireshark*.

Tabel 1 Uji coba data dari Referee Box

Data (char)	Sumber	Tujuan	Panjang	Protokol	Info	Keterangan
Kick Off Cyan ('K')	192.0.0.5	192.0.0.21	67 Bytes	TCP	[PSH, ACK]	Berhasil
	192.0.0.21	192.0.0.5	66 Bytes	TCP	[ACK]	
	192.0.0.21	192.0.0.10	50 Bytes	UDP	Len=1	
Start ('s')	192.0.0.5	192.0.0.21	67 Bytes	TCP	[PSH, ACK]	Berhasil
	192.0.0.21	192.0.0.5	66 Bytes	TCP	[ACK]	
	192.0.0.21	192.0.0.10	43 Bytes	UDP	Len=1	
Stop ('S')	192.0.0.5	192.0.0.21	67 Bytes	TCP	[PSH, ACK]	Berhasil
	192.0.0.21	192.0.0.5	66 Bytes	TCP	[ACK]	
	192.0.0.21	192.0.0.10	43 Bytes	UDP	Len=1	

Dari data pengujian pada Tabel 1 didapatkan bahwa dari semua perintah yang dikirimkan dari

Referee Box berhasil diterima oleh *Base Station*, mulai dari perintah *Kick Off Cyan (K)*, *Start (s)*, maupun *Stop (S)*. Dari hasil capture yang dilakukan oleh Wireshark, terdapat 3 paket yang tertangkap dalam sekali pengiriman, yaitu paket pertama berisi perintah dan ACK dari *Referee Box* (192.0.0.5) ke *Base Station* (192.0.0.21), dan paket kedua berisi ACK sebagai balasan dari ACK yang dikirimkan oleh *Referee Box*. Adanya ACK ini karena protokol yang digunakan adalah *Transmission Control Protocol* (TCP) sehingga memerlukan balasan dalam setiap pengiriman data. Sedangkan paket ketiga berisi paket perintah dari *Base Station* (192.0.0.21) ke robot (192.0.0.10). Selain dapat menerima, *Base Station* berhasil meneruskan data yang diterima dari *Referee box* ke robot untuk selanjutnya diterjemahkan sebagai intruksi yang harus dilakukan oleh robot.

B. Pengujian Pengiriman Data dari Robot.

Pengujian dilakukan dengan mengirimkan data dari robot secara kontinyu ke *Base station* dengan jumlah data yang bervariasi, mulai dari 100, 500, 1000, 3000 sampai 5000 data. Pengujian dilakukan dengan mengirimkan paket berupa data posisi bola, sudut bola terhadap robot serta deteksi keberadaan bola. Dimana data dikirim dengan format x_{Bola} , y_{Bola} , $azimut$, $deteksi_{Bola}$ dan dikirimkan secara *realtime* menggunakan protokol UDP (*User Datagram Protocol*) Unicast melalui perantara *wireless* (WLAN). Sampel data uji yang dikirimkan oleh robot ke *Base Station* dapat dilihat pada Tabel 2 berikut.

Tabel 2. Sampel data pengiriman

No	Data	Sumber	Tujuan	Protokol	Panjang	Info
1.	-76,-15,349,1	192.0.0.7	192.0.0.21	UDP	55	Len=13
2.	-78,-3,358,1	192.0.0.7	192.0.0.21	UDP	54	Len=12
3.	-78,1,0,1	192.0.0.7	192.0.0.21	UDP	51	Len=9
4.	-80,10,7,1	192.0.0.7	192.0.0.21	UDP	52	Len=10
5.	-80,15,10,1	192.0.0.7	192.0.0.21	UDP	53	Len=11

Data yang ditampilkan Tabel 2 merupakan 5 data sampel yang diambil secara acak selama pengiriman. Proses capture yang dilakukan menggunakan bantuan perangkat lunak pihak ketiga, yaitu Wireshark. Berikut deskripsi paket nomor 1 yang tertangkap oleh wireshark.

Tabel 3. Deskripsi paket nomor 1

Frame 110: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface 0
Ethernet II, Src: Azurewav_5f:68:d5 (74:c6:3b:5f:68:d5), Dst: Azurewav_81:6a:ae (6c:71:d9:81:6a:ae)
Internet Protocol Version 4, Src: 192.0.0.7, Dst: 192.0.0.21
User Datagram Protocol, Src Port: 43342, Dst Port: 1235
Data (13 bytes)
Data: 2d37362c2d31352c3334392c31
[Length: 13]

Dari data yang tertangkap dapat dilihat bahwa frame data yang dikirim berukuran 55 bytes, menggunakan protokol UDP dengan port sumber 43342 dan port tujuan 1235. Panjang data yaitu 13 bytes, dimana panjang data didapat dari karakter yang dikirimkan oleh robot, dalam hal ini yaitu -79,-15,349,1. Protokol UDP merupakan protokol yang dioptimasi untuk kecepatan pengiriman, namun UDP tidak menjamin pengiriman data, sehingga memungkinkan rusak/hilangnya data saat pengiriman. Berikut hasil pengujian hilang/rusaknya data pada sisi *Base Station*.

Tabel 4. Pengujian data hilang/rusak

Jumlah data	Data diterima	Data hilang/rusak	Persentase (%)
100	100	0	0 %
500	490	10	2 %
1.000	970	30	3 %
3.000	2.932	68	2,26 %
5.000	4942	58	1,16 %
Rata-rata			1,68 %

Pengujian dilakukan dalam 5 kali pengiriman, dari 100, 500 sampai 5000 data dengan pemantauan dilakukan pada sisi *Base Station*. Dari hasil pengujian yang didapatkan persentase rata-rata data hilang/rusak sebesar 1,68 %, dimana persentase data hilang/rusak terkecil yaitu 0 % pada pengiriman 100 data, dan persentase data hilang/rusak terbesar yaitu 3 % pada pengiriman 1000 data. Berikut deskripsi data yang ter-capture oleh wireshark pada jumlah data 100.

Tabel 5. Deskripsi data pengiriman 100

Measurement	Captured
Packets	100 Data
Time span (s)	4.188 s
Average pps	23.9 pps
Average packet size (B)	53.5 Bytes
Bytes	5362 Bytes
Average Bytes/s	1280 Bytes/s
Average bits/s	10 kbps
Packets	100 Data

Pada pengiriman 100 data, didapatkan rata-rata ukuran paket yaitu 53.5 bytes dengan waktu tempuh paket hingga sampai ke tujuan yaitu 4.188 s, sedangkan bandwidth aktual (*throughput*) yang didapat sebesar 10 kbps. Pada saat pengujian dilakukan pula proses pemarsingan data yang diterima oleh *Base Station*, dimana parsing dilakukan dengan membaca split karakter berupa tanda koma untuk memisahkan antara data satu dengan data lainnya. Hasil pemarsingan data selanjutnya ditampilkan dalam bentuk visual pada GUI aplikasi *Base Station*.

V. KESIMPULAN

Berdasarkan hasil pengujian yang dilakukan, aplikasi *Base Station* berhasil menerima data dari *Referee Box* menggunakan protokol TCP sekaligus dapat meneruskan data yang diterima ke robot menggunakan protokol UDP. Selain itu *Base Station* dapat menerima data yang dikirim oleh robot. Dari 5 kali pengiriman data sampel didapatkan persentase data hilang/rusak sebesar 1,68 %, dengan persentase terkecil 0 % pada pengiriman 100 data, dan persentase terbesar yaitu 3 % pada pengiriman 1000 data.

VI. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada kedua orang tua yang selalu memberikan dukungan moril maupun materiil kepada penulis. Tak lupa penulis mengucapkan terimakasih kepada Ibu Diana Rahmawati, S.T., M.T. dan Bapak Achmad Fiqhi Ibadillah, S.T., M.Sc. selaku dosen pembimbing penulis dalam menyelesaikan penelitian ini. Dan juga penulis mengucapkan terima kasih kepada semua pihak yang terlibat dalam menyelesaikan penelitian ini.

II. DAFTAR PUSTAKA

- [1] 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
- [2] 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.
- [3] William Stallings, "Data and Computer Communications," Pearson Education International, Upper Saddle River, New Jersey, NJ 07458, 2007

Desain dan Implementasi Komunikasi *Control* Robot Soccer Beroda Menggunakan User Datagram Protocol (UDP)

Syaiful Amri, Mahadir, Muhammad Rahmat, Ade Reza Septiadi, Nikita Supma Melani

¹ Politeknik Negeri Bengkalis, Staf Pengajar di Jurusan Teknik Elektro, Sungai Alam, Bengkalis

² Politeknik Negeri Bengkalis Prodi Teknik Elektronika, Sungai Alam, Bengkalis

³ Politeknik Negeri Bengkalis Prodi Teknik Mesin Produksi dan Perawatan, Sungai Alam, Bengkalis

⁴ Politeknik Negeri Bengkalis Prodi Teknik Listrik, Sungai Alam, Bengkalis

⁵ Politeknik Negeri Bengkalis, Prodi Jurusan Teknik Elektronika, Sungai Alam, Bengkalis

Email korespondensi: ¹syiafulamri@polbeng.ac.id, ²mahader.sob@gmail.com, ³muhammadrahmat120397@gmail.com, ⁴aderezaseptiadi@gmail.com, ⁵nikimelani26@gmail.com

Robot soccer beroda adalah robot yang dirancang untuk melakukan permainan sepakbola dengan aturan-aturan tertentu seperti permainan sepakbola dengan sistem pergerakan menggunakan roda. Pada umumnya robot soccer beroda dikendalikan secara otomatis menggunakan kamera dan mencari sebuah objek berupa bola yang pada dasarnya berwarna *orange*. Pada penelitian ini Robot soccer beroda dirancang agar bisa dikontrol dengan PC/laptop melalui aplikasi GUI (graphical user interface) yang didesain menggunakan NetBeans IDE (*integrated development environment*). Komunikasi yang digunakan pada sistem ini adalah UDP (user datagram protocol). UDP melakukan komunikasi secara sederhana dengan mekanisme yang sangat minimal. Aplikasi GUI controller akan dioperasikan pada PC/laptop dan mengkoneksikannya ke mini PC robot dengan sistem koneksi IP address. PC/laptop controller akan mengirim perintah data ke robot berupa data instruksi manuver robot dan nantinya data tersebut akan dieksekusi oleh mini PC dan diteruskan ke board Arduino melalui komunikasi serial port USB. Robot hanya bisa dikontrol dengan menggunakan jaringan *wifi*. Rata-rata jarak yang dapat dijangkau dalam pengontrolan ini adalah 125 meter.

Kata kunci : Robot soccer beroda, GUI (graphical user interface), UDP (user datagram protocol), NetBeans, komunikasi serial.

I. PENDAHULUAN

Robotika kini telah menjadi bagian dari kehidupan dan menjadi salah satu perkembangan teknologi suatu negara. Beberapa negara maju terus berupaya untuk terus mengembangkan penelitian di dunia robotika. Dari sekian banyak macam robot yang terus dikembangkan, salah satunya adalah robot pemain sepakbola (Robot Soccer). Sebuah robot yang beraksi di dalam lapangan sepakbola mini dengan peraturan yang lebih sederhana jika dibandingkan dengan sepakbola yang sebenarnya [3].

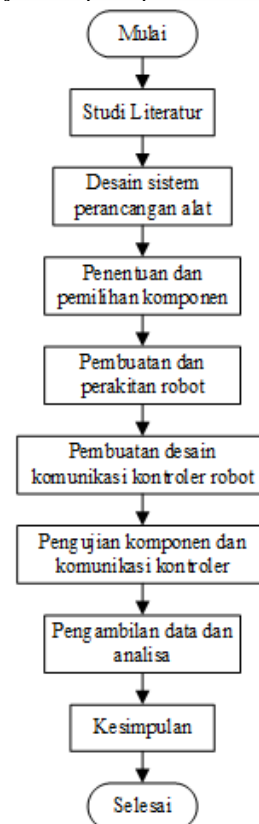
Para peneliti robot soccer ini terus melakukan berbagai riset. Pembagian divisi robot pun dibagi menjadi dua yaitu divisi robot soccer humanoid dan divisi robot soccer beroda. Divisi robot soccer humanoid berkonsentrasi dalam membangun sebuah

mekanik robot pemain sepakbola layaknya seperti manusia. Divisi robot soccer beroda berkonsentrasi pada Artificial Intelligent (AI) untuk mengenali pergerakan bola, lawan dan kawan [4].

II. METODE

A. Blok Diagram Penelitian

Secara sistematis langkah-langkah dalam penyelesaian penelitian ini dijadikan dalam bentuk diagram blok, yaitu seperti pada Gambar 2.



Gambar 2. Blok Diagram Penelitian

B. Studi Literatur

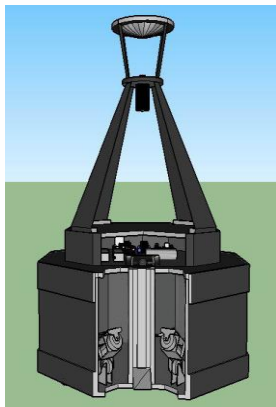
Studi literatur adalah mencari referensi teori yang relevan dengan kasus atau permasalahan yang ditemukan. Referensi tersebut berisikan tentang:

- Dasar teori tentang robot *soccer* beroda.
- Sistem gerak pada robot.
- Sistem pengontrolan melalui komputer.
- Dan sistem komunikasi yang digunakan untuk pengontrolan robot.

Studi literatur ini adalah kumpulan dari referensi-referensi yang terdahulu dengan perumusan masalah yang ada. Referensi ini dapat dicari dari buku, jurnal, artikel laporan penelitian dan situs-situs di *internet*. Tujuannya adalah untuk memperkuat permasalahan serta sebagai landasan teori dalam melakukan studi dan juga dalam penyelesaian penelitian ini.

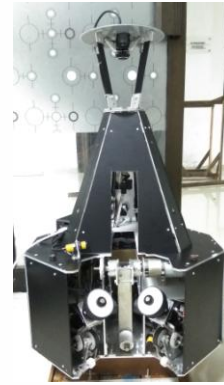
C. Perancangan Desain Mekanik

Untuk mempermudah dalam pembuatan penelitian ini, maka dibuat langkah-langkah dalam merealisasikan alat yang akan dibuat dan maka dengan itu mudah menemukan permasalahan pada pembuatan perancangan mekanik. Perancangan didesain dengan menggunakan aplikasi sketchup. Desain perancangan mekanik body/chasis robot dapat dilihat pada Gambar 3.



Gambar 3. Desain mekanik body/chasis robot

Adapun bahan yang digunakan untuk membuat mekanik *body* robot adalah terdiri dari plat *aluminium* dan *acrylic*. Penggunaan bahan untuk membuat mekanik *body* robot ini bertujuan agar robot tetap kuat pada saat ber-*manuver* ataupun berbenturan dengan benda keras. Pada lantai pertama digunakan plat *aluminium* dengan tebal 0,5 cm dan berat 350 gram. Lantai kedua digunakan *acrylic* dengan tebal 0,3 cm dan berat 130 gram. Lantai ketiga digunakan *acrylic* dengan tebal 0,2 cm dan berat 100 gram. Hasil mekanik *body/chasis* robot yang telah dibuat dapat dilihat pada Gambar 4.



Gambar 4. Hasil mekanik body/chasis robot

D. Perancangan Sistem Software

1. NetBeans IDE

Netbeans adalah sebuah aplikasi integrated development environment (IDE) yang berbasiskan Java dari Sun Microsystems yang berjalan di atas swing. Swing merupakan sebuah teknologi Java untuk pengembangan aplikasi dekstop yang dapat berjalan pada berbagai macam platform seperti windows, linux, Mac OS X dan Solaris. Sebuah IDE merupakan lingkup pemrograman yang diintegrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan Graphic User Interface (GUI), suatu kode editor atau text, suatu compiler dan suatu debugger [9]. Software NetBeans IDE ini adalah yang membuat form application controller robot.

2. Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (suite) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows ataupun aplikasi Web. Visual Studio mencakup compiler, SDK, integrated development environment (IDE), dan dokumentasi (umumnya berupa MSDN Library). [10]. Software Visual Studio ini digunakan untuk membuat console application receiver.

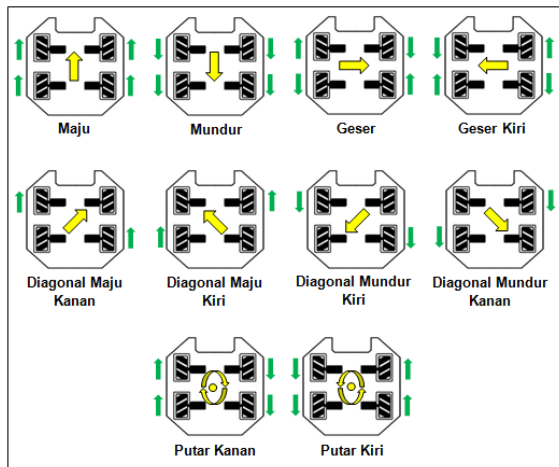
3. Arduino IDE

Arduino integrated development environment atau Arduino software (IDE) adalah sebuah software yang berisi editor teks untuk menulis kode, area pesan, konsol teks, toolbar dengan tombol untuk fungsi-fungsi umum dan serangkaian menu. menghubungkan ke Arduino dan Genuino hardware untuk meng-upload program dan berkomunikasi dengan mereka. Arduino IDE ini digunakan untuk memprogram board mikrokontroler Arduino.

E. Sistem Gerak Pada Robot Soccer Beroda

Alat gerak yang digunakan pada robot adalah menggunakan empat buah roda kenis *mecanum wheels*, tujuan penggunaan roda jenis ini adalah karena memiliki kelebihan yang tidak dimiliki oleh roda biasa. Dengan menggunakan roda jenis *mecanum wheels* ini tidak hanya dapat melakukan pergerakan maju dan mundur namun juga dapat melakukan pergerakan geser kiri, geser kanan serta pergerakan

diagonal kiri dan diagonal kanan [5]. Skema pergerakan robot soccer beroda dapat dilihat pada Gambar 5.



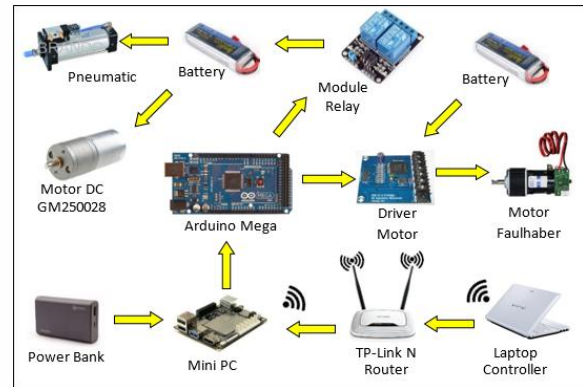
Gambar 5. Skema pergerakan yang dapat dilakukan oleh robot

F. UDP (User Datagram Protocol)

UDP singkatan dari user datagram protocol adalah salah satu protokol lapisan transpor TCP/IP yang mendukung komunikasi yang tidak andal (unreliable) yang merupakan bagian dari internet protocol. Dengan UDP, aplikasi komputer dapat mengirimkan pesan kepada komputer lain dalam jaringan lain tanpa melakukan komunikasi awal. UDP melakukan komunikasi secara sederhana dengan mekanisme yang sangat minimal. Ada proses checksum untuk menjaga integritas data [8].

G. Blok Diagram Sistem Hardware

Perancangan alat merupakan proses yang dilakukan terhadap alat, mulai dari perancangan kerja hingga hasil akhir alat sesuai dengan apa yang telah dirancang sebelumnya. Desain sistem pada robot soccer beroda ini terdiri dari perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*). Perancangan perangkat keras terdiri dari sistem mekanik dan komponen-komponen yang ada pada robot, sedangkan perancangan perangkat lunak terdiri dari desain sistem kontrol yang dibangun menggunakan *software NetBeans IDE*, *Microsoft Visual Studio* dan *Arduino IDE*. Blok diagram perancangan komponen sistem hardware robot dapat dilihat pada Gambar 6.



Gambar 6. Blok diagram perancangan hardware

Fungsi komponen-komponen pada Gambar 5 adalah sebagai berikut:

- *Arduino MEGA 2560* berfungsi sebagai mikrokontroler yang nantinya akan mengontrol keempat buah *driver motor* dan *module relay*.
- *Mini PC* berfungsi sebagai *receiver* data yang akan dikirim oleh *PC/laptop controller* dan meneruskan data yang diterima ke *board Arduino*.
- *TP-Link N Router* Berfungsi sebagai perantara penghubung antara *PC/laptop controller* dan *mini PC robot*.
- *Laptop controller* digunakan sebagai pengendali robot dan laptop juga yang akan mengirim data instruksi ke robot.
- *Driver motor* berfungsi sebagai mengontrol arah perputaran dan kecepatan *motor DC faulhaber*.
- *Motor faulhaber* berfungsi sebagai aktuator utama yang menggerakkan roda *mechanum wheels*.
- *Module relay* digunakan sebagai switch yang bekerja untuk memutuskan dan menghubungkan tegangan dari *battery* ke *valve pneumatic*.
- *Pneumatic* digunakan sebagai aktuator penendang objek yaitu berupa bola.
- *Battery* digunakan untuk men-supply *valve pneumatic*, *board Arduino* dan keempat *driver motor* yang digunakan.
- *Motor DC GM250028* digunakan sebagai menggiring bola.
- *Power bank* digunakan untuk men-supply *mini PC*.

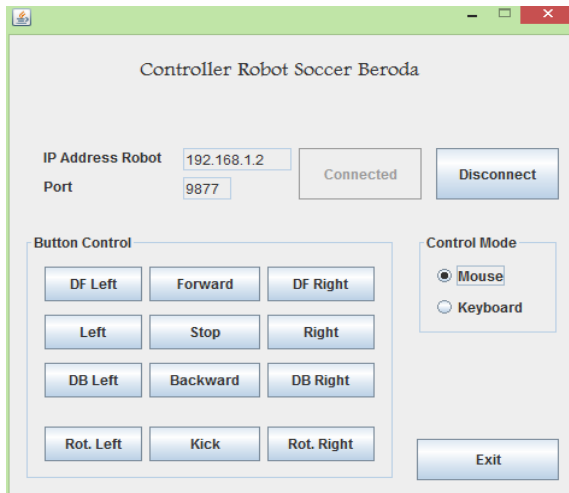
H. Tampilan Sistem Software

1. Form Application Controller

Form application controller bisa juga disebut dengan *GUI*. *GUI* adalah singkatan dari graphical user interface yang memungkinkan pengguna untuk berinteraksi dengan perangkat keras komputer serta memudahkan dalam mengoperasikan sebuah sistem operasi (user friendly).

Tampilan *GUI controller* robot soccer beroda ini didesain dengan sangat *simple* agar bisa dan mudah dalam penggunaannya. Maka dibuatlah pengontrolan dengan menggunakan dua sistem *mode*, yaitu *mode controll keyboard* dan *mode controll mouse*. *Mode*

controll keyboard adalah dimana *mode* pengontrolan dilakukan dengan menekan tombol *keyboard* pada PC/laptop *controller*. Sedangkan *mode contrtoll mouse* adalah dimana *mode* pengontrolan dengan mengklik *button controller* menggunakan *mouse*. Tampilan *GUI controller* robot soccer beroda dapat dilihat pada Gambar 7.

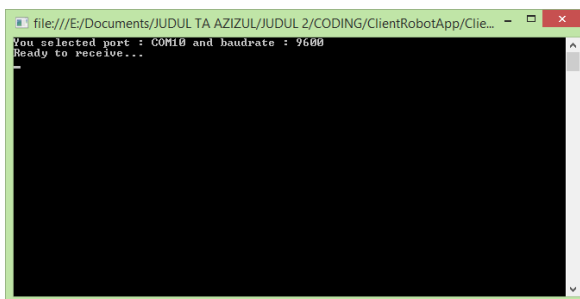


Gambar 7. Tampilan GUI kontroler

2. Console Application Receiver

Console application adalah program komputer yang dirancang dan digunakan antarmuka melalui komputer seperti *ISO*, *Unik*, *DOS* dll. *Console application* hanya menggunakan *keyboard* dan tampilan layar sebagai aplikasi, penggunaan *mouse*/perangkat penunjuk lainnya.

Console application ini dirancang sebagai program *receiver* dan dijalankan di *mini PC* robot. Program *receiver* ini akan terus *standby* menerima data masuk dan ketika program *receiver* ini telah menerima data yang dikirim oleh PC/laptop *controller* data tersebut akan diteruskan ke *board Arduino MEGA* dengan melalui komunikasi *serial USB* dan program *Arduino* akan menjalankan instruksi pergerakan yang telah dikirim dari PC/laptop *controller* tadi. Tampilan *console application* program *receiver* dapat dilihat pada Gambar 8.



Gambar 8. Tampilan console application program receiver

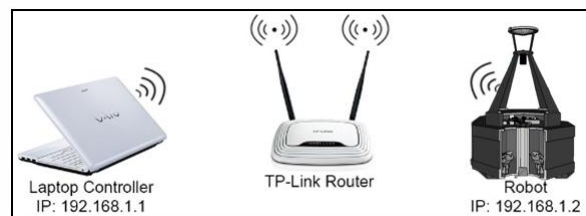
I. Cara Kerja Sistem

Secara garis besar robot hanya bisa dikendalikan oleh PC/laptop *controller* yang sudah saling terhubung dalam satu jaringan. Selain itu juga antara PC/laptop *controller* dan *mini PC* robot harus dalam alamat *IP address* yang sekelas. Robot hanya akan bergerak sesuai dengan data instruksi yang dikirim dari PC/laptop *controller*. Keterangan pergerakan robot dapat dilihat pada Tabel 1.

Tabel 1. Keterangan instruksi pergerakan robot

Tombol GUI Controller	Tombol Keyboard	Instruksi Robot	Simbol Pergerakan robot
Forward	W	Bergerak maju	↑
Backward	X	Bergerak mundur	↓
Left	A	Bergerak ke kiri	←
Right	D	Bergerak ke kanan	→
DF Right	E	Bergerak diagonal kanan depan	↗
DF Left	Q	Bergerak diagonal kiri depan	↖
DB Right	C	Bergerak diagonal kanan belakang	↘
DB Left	Z	Bergerak diagonal kiri belakang	↙
Rot. Left	J	Bergerak rotasi kiri	↻
Rot. Right	L	Bergerak rotasi kanan	↻
Kick	K	Menendang	-
Stop	S	Berhenti	•

Pengontrolan pada robot bisa dikendalikan dari jarak jauh, namun jarak pengontrolan masih terbatas dan akan mempengaruhi pada saat robot mengeksekusi data yang diterima. Skema sistem koneksi antara robot dan PC/laptop *controller* dapat dilihat pada Gambar 9.



Gambar 9. Skema sistem koneksi antara robot dan PC/laptop controller

III. HASIL DAN DISKUSI

Hasil perancangan ini adalah hasil yang didapat dari tahapan-tahapan perancangan alat. Baik itu pada sistem *software* maupun sistem *hardware*. Dari hasil pembuatan mekanik robot maka hal pertama yang harus diperhatikan adalah memastikan bahwa masing-masing komponen robot sudah terpasang atau terkoneksi dengan baik, seperti *board Arduino MEGA 2560*, *mini PC lattepanda*, *motor DC faulhaber*, *driver motor EMS 30A H-Bridge*, *module relay*, *motor dribble* serta *valve pneumatic*.

A. Pengujian Sistem Hardware

1. Pengujian Motor dan Driver Motor

Pengujian pertama adalah hanya dengan menggunakan satu buah motor *faulhaber* dan satu buah driver motor. Pengujian ini dilakukan dengan memberi input logika HIGH dan LOW atau 1 dan 0.

Hasil pengujian arah putaran motor dapat dilihat pada Tabel 2.

Tabel 2. Pengujian arah putaran motor

MOUT1	MOUT2	Kondisi motor
1	0	Motor berputar searah jarum jam
0	1	Motor berputar berlawanan arah jarum jam
0	0	Motor tidak berputar
1	1	Motor tidak berputar

Setelah melakukan pengujian arah putaran motor selanjutnya pengujian dengan merubah nilai PWM (*pulse width modulation*) dari rentangan nilai 0-255. Hasil pengujian kecepatan mutaran motor ini dapat dilihat pada Tabel 3.

Tabel 3. Pengujian kecepatan putaran motor.

Pin PWM (Pulse Width Modulation)	Kondisi motor
255	Motor berputar sangat laju
192	Motor berputar laju
128	Motor berputar sedang
64	Motor berputar lambat
0	Motor tidak berputar

Setelah pengujian dengan satu buah motor dan satu buah driver motor, maka selanjutnya pengujian dengan empat buah motor dan empat buah driver motor. Hasil dari pengujian ini dapat dilihat pada Tabel 4.

Tabel 4. Pengujian arah putaran empat buah motor pada *chasis* robot.

Motor 1	Motor 2	Motor 3	Motor 4	Aksi pergerakan robot
1	0	0	1	Robot bergerak maju
0	1	1	0	Robot bergerak mundur
0	1	0	1	Robot bergerak ke kanan
1	0	1	0	Robot bergerak ke kiri
0	0	0	1	Robot bergerak diagonal maju kanan
1	0	0	0	Robot bergerak diagonal maju kiri
0	0	1	0	Robot bergerak diagonal mundur kiri
0	1	0	0	Robot bergerak diagonal mundur kanan
0	1	0	1	Robot berputar ke kanan
1	0	1	0	Robot berputar ke kiri
0	0	0	0	Robot berhenti

2. Pengujian Module Relay

Module relay digunakan untuk menghubungkan dan memutuskan tegangan *battery* pada *valve solenoid* dan *motor dribble*. Hasil pengujian ini dapat dilihat pada Tabel 5 dan Tabel 6.

Tabel 5. Pengujian dengan *valve solenoid*

Input Logic	Output
1	Valve solenoid on
0	Valve solenoid off

Tabel 6. Pengujian dengan *motor dribble*

Input Logic	Output
1	Motor dribble on
0	Motor dribble off

Logika HIGH atau 1 adalah untuk menghidupkan *valve solenoid* dan *motor dribble*. Sedangkan logika LOW atau 0 adalah untuk mematikan *valve solenoid* dan *motor dribble*.

3. Pengujian Desain Kontroler

Ketika robot pertama kali dihidupkan PC/laptop *controller* bertugas untuk mengkoneksikan *IP address* robot melalui *button connect* yang terdapat pada *text box GUI controller*. Apabila PC/laptop *controller* dan robot sudah saling terhubung maka robot siap untuk dikontrol. Pengujian desain *controller* ini dilakukan pada komputer dan juga laptop dengan tujuan untuk merealisasikan sistem *controller* ini agar bisa digunakan pada komputer dan juga laptop. Hasil pengujian mode kontrol menggunakan *mouse* dan menggunakan *keyboard* dapat dilihat pada Tabel 7 dan Tabel 8.

Tabel 7. Pengujian mode kontrol menggunakan *mouse*.

Button control	Pergerakan robot	PC (Komputer)		Laptop	
		Berhasil	Gagal	Berhasil	Gagal
Forward	Robot bergerak maju	✓	-	✓	-
Backward	Robot bergerak mundur	✓	-	✓	-
Left	Robot bergerak ke kiri	✓	-	✓	-
Right	Robot bergerak ke kanan	✓	-	✓	-
DF Right	Robot bergerak diagonal kanan depan	✓	-	✓	-
DF Left	Robot bergerak diagonal kiri depan	✓	-	✓	-
DB Right	Robot bergerak diagonal kanan belakang	✓	-	✓	-
DB Left	Robot bergerak diagonal kiri belakang	✓	-	✓	-
Rot. Left	Robot bergerak rotasi kiri	✓	-	✓	-
Rot. Right	Robot bergerak rotasi kanan	✓	-	✓	-
Kick	Robot menendang	✓	-	✓	-
Stop	Robot berhenti	✓	-	✓	-

Tabel 8. Pengujian mode kontrol menggunakan keyboard.

Keyboard control	Pergerakan robot	PC komputer		Laptop	
		Berhasil	Gagal	Berhasil	Gagal
W	Robot bergerak maju	✓	-	✓	-
X	Robot bergerak mundur	✓	-	✓	-
A	Robot bergerak ke kiri	✓	-	✓	-
D	Robot bergerak ke kanan	✓	-	✓	-
E	Robot bergerak diagonal kanan depan	✓	-	✓	-
Q	Robot bergerak diagonal kiri depan	✓	-	✓	-
C	Robot bergerak diagonal kanan belakang	✓	-	✓	-
Z	Robot bergerak diagonal kiri belakang	✓	-	✓	-
L	Robot bergerak rotasi kiri	✓	-	✓	-
J	Robot bergerak rotasi kanan	✓	-	✓	-
K	Robot menendang	✓	-	✓	-
S	Robot berhenti	✓	-	✓	-

Dari hasil pengujian yang telah dilakukan bahwa robot telah bisa dikoneksikan dengan menggunakan komputer dan laptop. Pengujian dengan menggunakan *button control* pada *GUI controller* dan tombol *keyboard* ini sesuai dengan instruksi pergerakan robot.

4. Pengujian Jarak Kontrol

Setelah melakukan pengujian pada masing-masing komponen dan sistem, kemudian dilakukan pengujian secara keseluruhan yaitu mulai dari awal sampai akhir. Pengujian sistem secara keseluruhan ini adalah tahap akhir dalam proses pengujian pada robot.

Pengujian ini dilakukan dengan cara bertahap dengan jarak pengontrolan pertama adalah 5 meter. Setelah pengujian pertama selesai, jarak pengontrolan akan terus ditambah 5 meter dan sehingga nantinya jarak pengontrolan mencapai jarak 100 meter. Pengujian ini juga dilakukan dengan meletakkan sebuah penghalang antara robot dan PC/laptop *controller*, yaitu berupa kaca dan pengujian tanpa penghalang. Hasil pengujian ini dapat dilihat pada Tabel 9 dan Tabel 10.

Tabel 9. Pengujian jarak kontrol tanpa ada halangan

Jarak pengontrolan	Pengujian 1		Pengujian 2		Pengujian 3	
	Berhasil	Gagal	Berhasil	Gagal	Berhasil	Gagal
5 Meter	✓	-	✓	-	✓	-
10 Meter	✓	-	✓	-	✓	-
15 Meter	✓	-	✓	-	✓	-
20 Meter	✓	-	✓	-	✓	-
25 Meter	✓	-	✓	-	✓	-
30 Meter	✓	-	✓	-	✓	-
35 Meter	✓	-	✓	-	✓	-
40 Meter	✓	-	✓	-	✓	-
45 Meter	✓	-	✓	-	✓	-
50 Meter	✓	-	✓	-	✓	-
55 Meter	✓	-	✓	-	✓	-
60 Meter	✓	-	✓	-	✓	-
65 Meter	✓	-	✓	-	✓	-
70 Meter	✓	-	✓	-	✓	-
75 Meter	✓	-	✓	-	✓	-

80 Meter	✓	-	✓	-	✓	-
85 Meter	✓	-	✓	-	✓	-
90 Meter	✓	-	✓	-	✓	-
95 Meter	✓	-	✓	-	✓	-
100 Meter	✓	-	✓	-	✓	-

Tabel 10. Pengujian jarak kontrol dengan ada halangan

Jarak pengontrolan	Pengujian 1		Pengujian 2		Pengujian 3	
	Berhasil	Gagal	Berhasil	Gagal	Berhasil	Gagal
5 Meter	✓	-	✓	-	✓	-
10 Meter	✓	-	✓	-	✓	-
15 Meter	✓	-	✓	-	✓	-
20 Meter	✓	-	✓	-	✓	-
25 Meter	✓	-	✓	-	✓	-
30 Meter	✓	-	✓	-	✓	-
35 Meter	✓	-	✓	-	✓	-
40 Meter	✓	-	✓	-	✓	-
45 Meter	✓	-	✓	-	✓	-
50 Meter	✓	-	✓	-	✓	-
55 Meter	✓	-	✓	-	✓	-
60 Meter	✓	-	✓	-	✓	-
65 Meter	✓	-	✓	-	✓	-
70 Meter	✓	-	✓	-	✓	-
75 Meter	✓	-	✓	-	✓	-
80 Meter	✓	-	✓	-	✓	-
85 Meter	✓	-	✓	-	✓	-
90 Meter	✓	-	✓	-	✓	-
95 Meter	✓	-	✓	-	✓	-
100 Meter	✓	-	✓	-	✓	-

Pengujian jarak pengontrolan robot dilakukan dengan menggunakan tiga buah laptop yang berbeda *type* dan dengan jarak pengontrolan yang sama. Pengujian pertama menggunakan laptop *type SONY VAIO*, pengujian kedua menggunakan laptop *notebook Acer Aspire V5-132* dan pengujian ketiga menggunakan laptop *ASUS*. Pengujian ini telah dilakukan berulang kali dan waktu eksekusi yang diterima oleh robot tidak jauh berbeda meskipun jarak terus ditambah dengan jarak 5 meter.

Berdasarkan spesifikasi, jarak yang dapat dijangkau oleh *TP-Link Wireless N Router TL-WR841N* adalah 10-100 meter. Setelah melakukan pengujian jarak pengontrolan dengan jarak hingga 100 meter, penulis juga menguji jarak pengontrolan ini dengan jarak hingga 130 meter. Pengujian ini dilakukan dengan tujuan ingin mengetahui batas maksimal yang dapat dijangkau oleh *TP-Link Wireless N Router TL-WR841N*. Hasil pengujian jarak kontrol di atas 100 meter dapat dilihat pada Tabel 11.

Tabel 11. Pengujian jarak kontrol diatas 100 meter

Jarak pengontrolan	Pengujian 1		Pengujian 2		Pengujian 3	
	Berhasil	Gagal	Berhasil	Gagal	Berhasil	Gagal
105 Meter	✓	-	✓	-	✓	-
110 Meter	✓	-	✓	-	✓	-
115 Meter	✓	-	✓	-	✓	-
120 Meter	✓	-	✓	-	✓	-
121 Meter	✓	-	✓	-	✓	-
122 Meter	✓	-	✓	-	✓	-
123 Meter	✓	-	✓	-	✓	-

124 Meter	✓	-	-	✓	✓	-
125 Meter	✓	-	-	✓	✓	-
126 Meter	-	✓	-	✓	-	✓
127 Meter	-	✓	-	✓	-	✓
128 Meter	-	✓	-	✓	-	✓
129 Meter	-	✓	-	✓	-	✓
130 Meter	-	✓	-	✓	-	✓

Pada pengujian jarak kontrol dengan jarak di atas atau lebih dari 100 meter dilakukan dengan cara menambah jarak per 5 meter hingga mencapai jarak 130 meter. Setelah dilakukan beberapa kali percobaan pada pengujian kedua yaitu pengujian kontrol dengan menggunakan laptop *notebook Acer Aspire V5-132*, bahwa ketika pengiriman data instruksi ke robot pada jarak 125 meter robot sudah tidak bisa merespon data instruksi yang dikirim oleh laptop *controller* sedangkan untuk pengujian pertama yang menggunakan laptop *SONY VAIO* dan pengujian ketiga yang menggunakan laptop *ASUS* ketika pengiriman data instruksi pada jarak 125 meter robot masih bisa merespon dengan data instruksi yang dikirim. Jadi, penulis berinisiatif untuk mengurangi rentangan jarak kontrol dengan cara menambah jarak per 1 meter saja dari 120 meter hingga mencapai jarak 130 meter. Cara ini bertujuan agar mengetahui jarak pasti yang masih dapat dijangkau oleh TP-Link N Router per-type laptop itu berapa meter.

Terlihat pada Tabel 11 pengujian pertama dengan menggunakan laptop *SONY VAIO* dan pengujian ketiga dengan menggunakan laptop *ASUS* robot sudah tidak bisa merespon data instruksi yang dikirim oleh laptop *controller* pada saat jarak 126 meter. Sedangkan pada saat pengujian kedua, robot tidak bisa merespon data instruksi yang dikirim pada saat jarak 124 Meter. Jadi dapat disimpulkan bahwa jarak kontrol juga akan berpengaruh pada *type* laptop. Maksimal jarak yang dapat dijangkau oleh *TP-Link Wireless N Router* adalah 125 meter hanya berbeda atau lebih 25 meter saja dari spesifikasi yang telah ditentukan. Namun pengiriman paket data tersebut menjadi lambat dieksekusi oleh robot karena telah melewati jarak maksimal jangkauan *TP-Link Wireless N Router*. Jarak pengontrolan ini dilakukan robot berada di gedung elektro dan laptop *controller* berada di gedung C Politeknik Negeri Bengkalis. Ketika pengujian pada jarak 125 meter ke atas rata-rata robot sudah tidak bisa dikontrol lagi, karena paket data yang dikirim dari laptop *controller* ke robot sudah tidak bisa dieksekusi oleh robot lagi.

IV. KESIMPULAN

Setelah melalui beberapa proses dalam pengujian pada alat dan analisa sistem yang telah dilakukan, maka dapat diperoleh beberapa kesimpulan sebagai berikut:

1. Jarak maksimal yang dapat dijangkau oleh *TP-Link Wireless N Router TL-WR841N* rata-rata adalah 125 meter. Jika lebih dari 125 meter, maka paket data yang dikirim oleh PC/laptop

controller tidak akan sampai atau robot sudah tidak bisa dikontrol lagi.

2. Pengontrolan jarak juga akan berpengaruh dengan *type* PC/laptop.
3. Semakin jauh jarak pengontrolan kemungkinan besar robot akan lambat menerima data yang dikirim oleh PC/laptop *controller*.
4. Robot masih tetap bisa dikontrol meskipun ada halangan diantara robot dan PC/laptop *controller* seperti halangan kaca dan dinding.
5. Robot hanya bisa dikontrol melalui komunikasi *wireless* jaringan *wifi*, serta robot dan PC/laptop *controller* haruslah dalam alamat *IP address* yang sekelas.
6. Gangguan pada jaringan terkadang membuat robot akan mengalami kegagalan dalam mengeksekusi data masuk. Kegagalan ini biasanya terjadi pada saat pengontrolan dengan batas jarak yang maksimal, namun juga akan terjadi pada saat pengontrolan jarak dekat sekali pun.

V. DAFTAR PUSTAKA

- [1] Giyanto, Dr. Ir. Aulia Siti Aisyah, MT., Suwito, ST.MT. 2008. "Perancangan *server* Menggunakan *Visual Basic 6.0* untuk Mendukung Sistem *Monitoring* dan Pengendalian *Autopilot* pada Kapal MCST-1". Institut Teknologi Bandung (ITB). Jurusan Teknik Elektro. Bandung.
- [2] Muhammad Rizki Setiawan, M. Aziz Muslim, Goegoes Dwi Nusantara. 2012. "Kontrol Kecepatan Motor DC dengan Metode PID Menggunakan *Visual Basic 6.0* dan Mikrokontroler ATmega 16". Universitas Brawijaya (UB). Jurusan Teknik Listrik dan Mekatronika. Malang.
- [3] Muhammad Nurdin, Mohammad Adnan, Muhammad Luthfi Musra. 2016. "Penerapan Metode Disturbance Observer Pada Sistem Kontrol Pergerakan Motor DC Robot Soccer". Politeknik Negeri Ujung Pandang (PNUP). Jurusan Teknik Elektro. Makassar.
- [4] Iwan Kurnianto Wibowo, Mochamad Mobed. 2016. "Rancang Bangun Mekanik Penendang Pada *Robot Soccer* Beroda Menggunakan *Solenoid*". Politeknik Elektronika Negeri Surabaya (PENS). Jurusan Teknik Komputer. Surabaya.
- [5] Sahrul Nizam. 2016. "Desain Algoritma Pada *Robot Soccer* Beroda". Politeknik Negeri Bengkalis. Jurusan Teknik Elektronika. Bengkalis.
- [6] Finnata Bayu Samudra, Fernando Ardila, Setiawardhana. 2011. "Pengendalian robot soccer dalam mengambil bola dengan metode evolutionary artificial potential field". Politeknik Elektronika Negeri Surabaya (PENS). Jurusan Teknik Komputer. Surabaya.
- [7] Mannes, Poel. 2003. "Motion Planning In A Robot Soccer System", A Master's Thesis in Computer Science. Language, Knowledge and Interaction Group Department of Compute Science University of Twente The Netherlands.
- [8] https://id.wikipedia.org/wiki/User_Datagram_Protocol. "Pengertian UDP User Datagram Protocol". (Diakses pada tanggal 12 juni 2017). Wikipedia. 2017.
- [9] http://fahrun_it.blogspot.co.id/2013/05/pengertian-netbeans.html. "Pengertian NetBeans". (Diakses pada tanggal 12 juni 2017). Fahrun. 2013.
- [10] https://id.wikipedia.org/wiki/Microsoft_Visual_Studio. "Pengertian Microsoft Visual Studio". (Diakses pada tanggal 25 Maret 2017). Wikipedia. 2012.

Sistem Kendali Motor DC Menggunakan PID Dan Komunikasi I2C Pada *Omni Direction* Untuk Robot Soccer Ten-De (10-D)

Tria Rahmadani^{a 1}, Fitri^b, Jabal Rahman^c, Rama Rifaldi Larengi^d, Agung Kurniawan^e, Moh.Zabir^f, Fanrani, S^g
^{abcdegf} STMIK Adhi Guna, Jl.Undata No 3, Palu

¹Email :rhanyongk39@gmail.com, fitrilad01@gmail.com, Jabalrahman007@gmail.com, wargreymon4648@gmail.com, agungkurniawan2098@gmail.com, zabir0797@gmail.com, fanranigokill@gmail.com

Abstrak – Fokus yang ingin dicapai pada penelitian kali ini adalah bagaimana robot soccer dapat melakukan pergerakan atau manuver dilapangan dengan menanamkan algoritma PID sebagai basis kontrol motor DC dan sistem koordinasi serta komunikasi antar motor melalui komunikasi serial Integrated-Integrated Circuit (I2C). Dalam mencapai tujuan tersebut digunakan konsep master slave kontroler pada robot soccer. Dimana setiap motor DC akan terintegrasi dengan sebuah Board Arduino nano sebagai slave dan board arduino Mega sebagai master. Hasil yang telah diperoleh yakni setiap kali master mengirimkan karakter dan atau nilai maka setiap itu pula kontrol slave menerima dengan baik. Sementara nilai kontrol PID yang diterapkan yakni $P=1$, $I=4$, $D=0.001$. Kesimpulan yang diperoleh dalam penelitian ini adalah sistem komunikasi data master slave dengan menerapkan komunikasi serial I2C dapat berjalan dan berkomunikasi dengan baik dan kemudian algoritma kontrol PID dapat mengontrol motor DC melalui slave kontroler.

Kata kunci: *Serial I2C, PID, Master-Slave*

I. PENDAHULUAN

Kontes Robot Indonesia (KRI) merupakan salah satu kompetisi di bidang robotika yang dimana setiap tahunnya di adakan oleh Direktorat Jendral Pendidikan Tinggi (DIKTI). Pada kompetisi ini terbagi dari beberapa devisi salah satunya ialah (Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda. Dimana robot ini akan dituntun seperti pada umumnya manusia bermain bola. Untuk itu, diperlukan kendali dan komunikasi data pada motor DC yang menjadi salah satu dari bagian robot yang terpenting untuk menggerakkan robot ataupun bermanuver. Sistem pengendali kecepatan motor DC 24 Volt dengan menggunakan PWM sebelumnya pernah dibuat. Berupa simulasi sederhana dan pengamatan pulsa modulasi pada satu buah motor DC 24 Volt [1].

Pada tahun sebelumnya robot tim Ten-De ini menggunakan omni wheel dengan jumlah motor sebanyak 4 motor DC. Sistem kerja 4 roda pada robot dapat membantu kecepatan robot dan kuat menahan beban serta tahan terhadap benturan robot lawan. Namun algoritma kontrol pada motor DC masih belum baik karena belum menerapkan algoritma PID sebagai basis kontrol motor DC. Selain itu sistem kontrol

antara low level dan high level belum terpisah sehingga memperlambat kinerja pemrosesan pada robot. Dimana dari 4 buah motor DC yang digunakan hanya terdapat sebuah kontroler dan sekaligus kontroler tersebut terintegrasi dengan berbagai sensor yang digunakan pada robot.

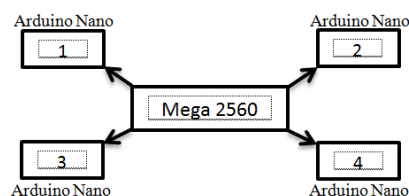
Dalam memetakan arah gerak sistem motor DC berdasarkan kombinasi tingkat kecepatan yang diberikan pada masing-masing motor DC. Mendesain bentuk mekanik, merakit rangkaian elektronika dan pembuatan program harus saling terkoordinasi dan melalui sistem komunikasi [2].

Untuk itu, dalam penelitian ini fokus peneliti pada pengembangan robot soccer tim Ten-De yang akan mengkaji lebih dalam tentang kontrol motor DC dengan menggunakan algoritma PID dan sistem koordinasi serta komunikasi antar motor DC dengan model master slave kontroler melalui serial I2C guna mendukung pergerakan yang baik ataupun manuver pada robot.

II. METODE

Dalam penelitian ini tim robot Ten-De menerapkan beberapa perubahan dari robot tahun sebelumnya dengan menerapkan sistem kontrol yang terkomunikasi dengan model Master-Slave.

A. Sistem Kontrol Master-Slave Robot

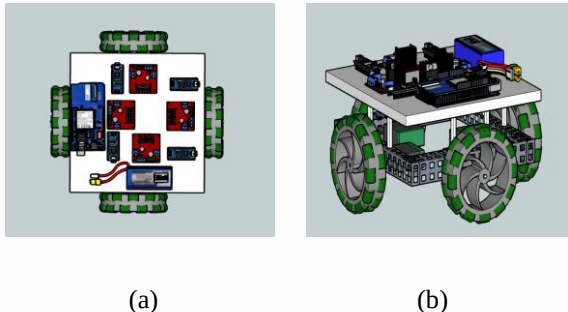


Gambar 1. Blok Diagram Kontrol Master-Slave

Pada gambar blok diagram kontrol di atas bahwa perancangan yang dilakukan menggunakan sistem master-slave dalam melakukan komunikasi data. Dengan memanfaatkan *board arduino Mega2560* sebagai master dan *board arduino nano* sebagai slave kontrol.

B. Desain Chasis Robot Menggunakan Omni Wheel

Gambar dibawah ini adalah satu satu bentuk casis robot tim Ten-De yang menggunakan 4 roda omni *directional*.

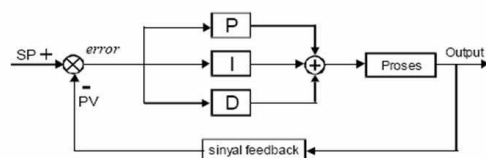


Gambar 2. Bentuk 3D Robot, (a) Tampak atas dan (b) Tampak Samping

Omni directional yang di maksud dengan gambar di atas bertujuan untuk mempermudah gerakan robot ke segala arah dan mampu menahan beban robot yang berat.

C. Sistem Kontrol PID Motor DC

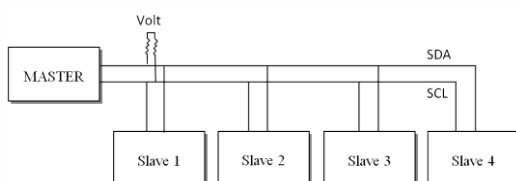
Sistem kontrol PID merupakan sistem kontrol loop tertutup yang cukup sederhana dan kompatibel dengan sistem kontrol lainnya sehingga dapat dikombinasikan dengan sistem kontrol lain seperti *Fuzzy control*, *Adaptif control* dan *Robust control* [3]. Pada penelitian ini robot tim Ten-De menanamkan algoritma PID pada setiap kontroler slave (arduino nano) untuk mengontrol setiap motor DC.



Gambar 3. Kontrol PID pada Motor DC

D. Komunikasi Serial I2C antar arduino

Beberapa pilihan komunikasi serial dalam mikrokontroler yakni *HSU*, *SPI*, *I2C*. Dalam penelitian kali ini peneliti menggunakan komunikasi serial I2C sebagai proses dalam komunikasi antar kontroler Master-Slave.



Gambar 4. Sistem Bus Komunikasi I2C

Gambar diatas dapat menjelaskan standar komunikasi I2C mempunyai dua bagian saluran untuk mengirim data maupun menerima. Saluran I2C sendiri terdiri dari saluran SCL (serial clock) dan SDA (serial data) yang membawa informasi data antara I2C dengan pengontrolnya. Piranti yang dihubungkan dengan sistem I2C bus dapat dioperasikan sebagai Master dan slave.

Karakter I2C adalah :

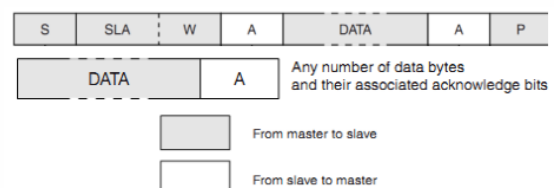
1. Data dikirim serial secara per-bit
2. Menggunakan dua penghantar koneksi dengan ground bersama. Dua penghantar tersebut adalah SCL (Serial Clock Line) untuk menghantarkan sinyal clock dan SDA (Serial Data) untuk mentransaksikan data.
3. Jumlah slave maximal 127. Slave dialamatkan melalui 7-bit –alamat.
4. Setiap transaksi data terjadi antara pengirim (Transmitter) dan penerima (Receiver).

Aturan komunikasi :

1. Device atau komponen yang mengirim data disebut transmitter, sedangkan device yang menerimanya disebut receiver
2. Device yang mengendalikan operasi transfer data disebut master, sedangkan device lainnya yang dikendalikan oleh master disebut slave
3. Master harus menghasilkan serial clock melalui pin SCL, mengendalikan akses ke BUS serial, dan menghasilkan sinyal kendali START dan STOP.

Mode pengoprasian transfer data yaitu:

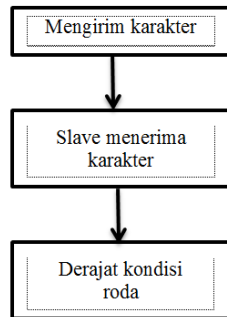
1. Transfer data dari master (*transmitter*) ke slave (*receiver*). Byte pertama yang dikirimkan oleh master adalah slave, setelah itu master mengirimkan sejumlah byte data. Slave atau receiver mengirimkan sinyal *acknowledge* setiap kali menerima 1- byte data. Pada tiap byte, bit pertama yang dikirim adalah MSB.



Gambar 5. Transfer data dari master ke slave

E. Model kerja program

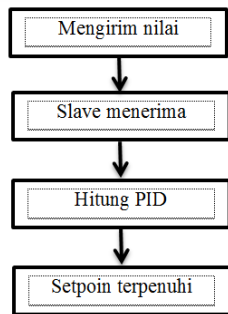
Adapun gambaran dari model kerja program terbagi menjadi dua bagian sebagai berikut:



Gambar 6. Komunikasi I2C mengirim karakter

Dari gambar 3 komunikasi I2C mengirim karakter di atas dapat dijelaskan sebagai berikut:

1. Master mengirim karakter ke slave sebagai penentu arah sudut robot yang ingin dituju.
2. Lalu slave menerima karakter dari master untuk langsung dikonversi menjadi program untuk menentukan arah pergerakan roda.
3. Kemudian karakter yang ada akan langsung dieksekusi menjadi sebuah pergerakan sesuai penentuan derajat.



Gambar 7. Komunikasi I2C mengirim nilai

Dari gambar 3 komunikasi I2C mengirim nilai di atas dapat dijelaskan sebagai berikut:

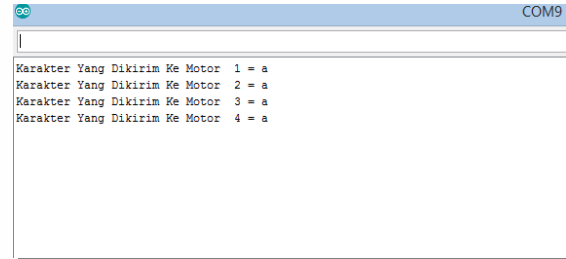
1. Master akan mengirim nilai ataupun *setpoint* ke slave
2. Slave menerima karakter nilai berupa *setpoint* untuk segera diproses dengan algoritma kontrol yang ada yaitu PID.
3. Setelah slave menerima karakter nilai dari master maka dengan waktu yang sama algoritma kontrol PID akan bekerja pula
4. Tujuan akhir yang dianggap berhasil ketika *setpoint* telah terpenuhi.

III. HASIL DAN DISKUSI

Berdasarkan hasil percobaan dan pengujian yang dilakukan didapatkan hasil sebagai berikut :

A. Hasil Pengujian Komunikasi I2C

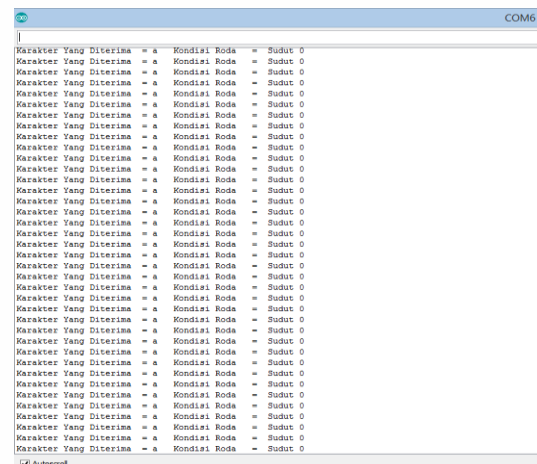
1. Master Mengirim Karakter (Transmitter)



Gambar 8. Master Mengirim Karakter

Pada gambar 8 diatas merupakan *capture* dari pengujian pertama pengiriman karakter “a” yang dilakukan oleh master melalui komunikasi I2C ke masing-masing slave controller.

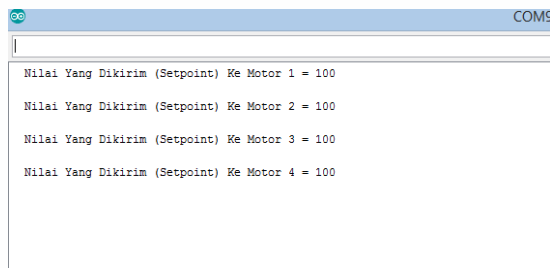
2. Slave Menerima Karakter Perlakuan Sudut (Receiver)



Gambar 9. Slave Menerima Karakter

Untuk gambar 9 diatas merupakan hasil pengujian penerimaan karakter pada slave, bahwa slave telah menerima karakter “a” dari master dan karakter tersebut langsung dikonversi menjadi arah sudut yang ingin dituju.

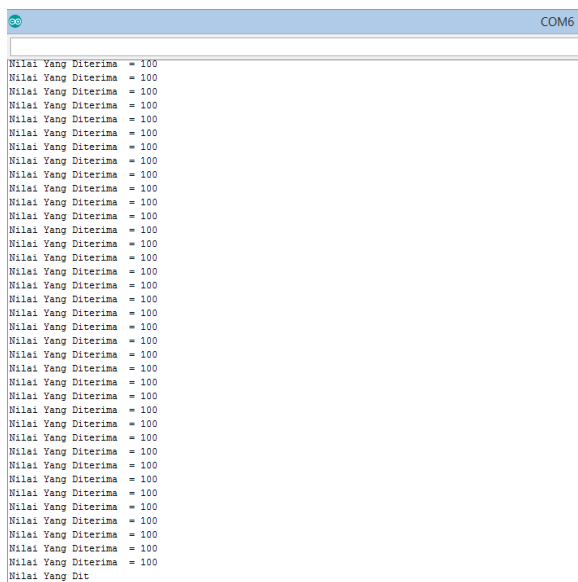
3. Master Mengirim Nilai Setpoint



Gambar 10. Master Mengirim Setpoint

Pada pengujian kedua master melakukan pengiriman nilai setpoint ke masing-masing slave kontroler yang merupakan setpoint dari kecepatan motor DC pada masing-masing roda yang ditunjukkan pada gambar 10.

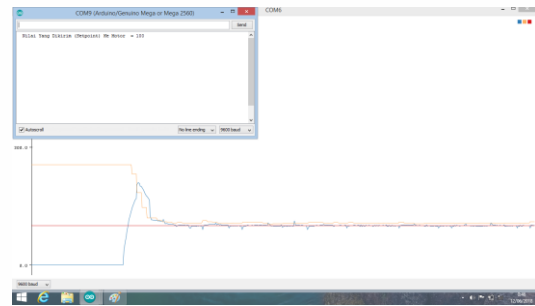
4. Slave Menerima Nilai untuk mengontrol kecepatan motor menggunakan kontrol (PID)



Gambar 11. Slave Menerima Karakter

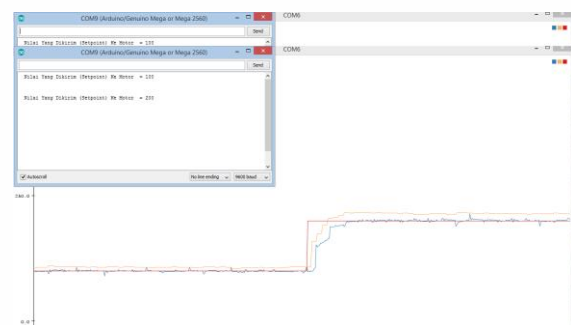
Pada gambar 11 diatas merupakan *capture* nilai yang telah diterima oleh slave dari master, nilai tersebut nantinya akan dijadikan sebagai setpoint kecepatan robot pada slave kontroler dimana telah ditanamkan algoritma kontrol PID didalamnya.

B. Hasil Pengujian Kontrol PID pada Slave Kontroller



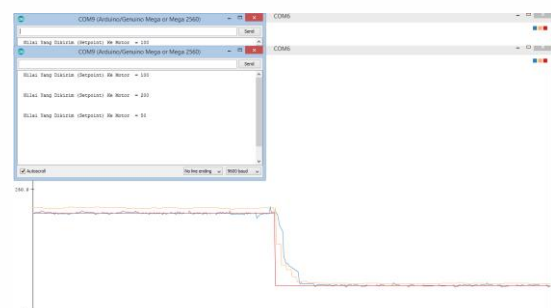
Gambar 12. Hasil Pengujian Kontrol

Gambar diatas merupakan hasil pengujian kontrol PID menggunakan Serial Ploter, garis berwarna merah merupakan Setpoint, garis kuning biru merupakan kecepatan dari hasil kontrol PID, dan kuning merupakan Kecepatan dari keadaan yang sebenarnya, pada pengujian kontrol PID pertama setpoint yang diberikan adalah 100 dalam satuan RPM dan secara otomatis kecepatan sebuah motor berubah mengikuti setpoint yang diberikan.



Gambar 13. Hasil Pengujian kedua Setpoint

Pada gambar pengujian kedua setpoint yang diberikan adalah 200 dalam satuan RPM, dan kecepatan motor secara otomatis berubah mengikuti setpoint yang diberikan menggunakan kontrol PID

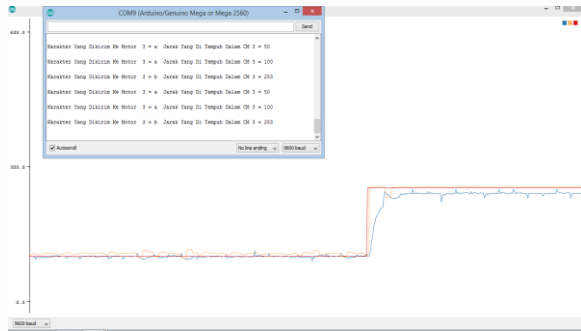


Gambar 14. Hasil Pengujian ketiga Setpoint

Pada gambar diatas merupakan hasil pengujian ketiga dimana perubahan dari nilai tinggi menjadi nilai rendah akan tetapi dengan menggunakan kontrol PID

kecepatan motor secara otomatis berubah mengikuti setpoint yang diberikan.

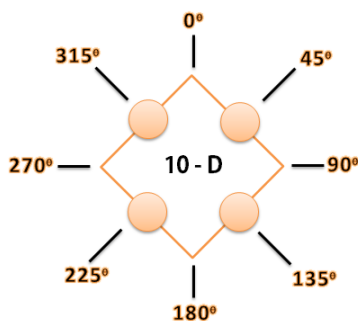
Kontrol PID akan melakukan pengontrolan kecepatan pada masing-masing motor secara otomatis berdasarkan setpoint yang telah diberikan, setpoint yang diberikan merupakan sebuah tujuan dari kecepatan berputar yang ingin dituju oleh motor DC dalam satuan Rotasi Per Menit (RPM).



Gambar 15. Aksi Kontrol PID dalam bentuk tampilan plotter pada slave kontroller

Gambar 15 diatas merupakan bentuk tampilan serial ploter pada slave kontroler untuk pengujian kontrol PID. Kecepatan motor berubah secara otomatis berdasarkan setpoint yang diberikan oleh master dengan masing-masing nilai $P=1$, $I=4$, $D=0.001$

C. Hasil Pengujian Sudut Pergerakan Robot



Gambar 16. Sudut Arah Pergerakan Robot

Pada gambar 16 merupakan sudut arah pergerakan robot. Sudut arah pergerakan tersebut merupakan aksi yang terjadi dari data karakter yang diperintahkan oleh kontroler master. Sehingga pola gerak motor DC akan pada omni direction tersinkronisasi dan akan terkontrol melalui slave kontroller dengan algoritma PID yang sudah ditanamkan pada setiap kontroler slave motor DC.

Tabel 1. Hasil Pengujian Arah Gerak Robot

Karakter Yang Diterima	Kondisi Robot (Sudut)	Kondisi Roda 1	Kondisi Roda 2	Kondisi Roda 3	Kondisi Roda 4
a	0°	Maju	Maju	Maju	Maju
b	180°	Mundur	Mundur	Mundur	Mundur
c	270°	Mundur	Maju	Mundur	Maju
d	90°	Maju	Mundur	Maju	Mundur
e	315°	Mati	Maju	Mati	Maju
f	45°	Maju	Mati	Maju	Mati
g	225°	Mundur	Mati	Mundur	Mati
h	135°	Mati	Mundur	Mati	Mundur

IV. KESIMPULAN

Kesimpulan yang diperoleh dalam penelitian ini adalah sistem komunikasi data master slave dengan menerapkan komunikasi serial I2C dapat berjalan dan berkomunikasi dengan baik dan kemudian algoritma kontrol PID dapat mengontrol motor DC melalui slave kontroler. Dengan peran omni directional bekerja dengan baik bermanuver sesuai dengan karakter yang dikirim melalui komunikasi serial I2C.

V. UCAPAN TERIMA KASIH

Terima kasih kepada Allah SWT yang telah melancarkan urusan kami untuk menyusun makalah ini. Kepada orang tua kami tercinta sudah mensupport serta memberika doa, tidak lupa pula pembimbing yang selalu memberikan masukan dan motivasi yang membangun kami dalam penulisan makalah ini dan teman – teman seperjuangan Adhi Guna Robotech Club.

VI. DAFTAR PUSTAKA

- [1] Ari Rahayuningtya" Pembuatan Sistem Pengendali 4 Motor DC Penggerak 4 Roda Secara Independent Berbasis Mikrokontroler AT89C2051"Subang Jawa Barat, 2009
- [2] Taufiq Yudi Sulistiyono" Komparasi Sistem Komunikasi Serial Multipoint Pada Robot Management Sampah Menggunakan I2c Dan SPI" Malang, 2014
- [3] Muhammad Irfan" Desain Dan Implementasi Kendali Kecepatan Motor Pada Robot Dengan Empat Roda Omni Menggunakan Metode Logika Fuzzy" Universitas Telkom Bandung, 2016
- [4] Muhammad Ruswandi Djalal" Optimisasi Kontrol Pid Untuk Motor Dc Magnet Permanen Menggunakan Particle Swarm Optimization" Makassar, 2017

ROBOSOC UM 2018: Deskripsi Tim Robot KRSBI Beroda Universitas Negeri Malang

M. Syaifuddin*, Dedy Tri Prasetyo, Ahmad Burhanuddin Fahmi, Ahmad Yunus Afghoni, Dedy Prasetyo, Nurfakhmi Ilham Firdaus, Yogi Dwi Mahandi, Mahfud Jiono, Muladi
Universitas Negeri Malang
Jl. Semarang 5 Malang 65145, Telp. (0341) 551312
Email: syaifuddin.ardha@gmail.com

Abstrak -- ROBOSOC UM yang berartikan Robot Soccer UM, merupakan Tim KRSBI Beroda Universitas Negeri Malang yang berlaga pada Regional IV. Pada paper ini, dijabarkan beberapa perancangan sistem robot Tim ROBOSOC UM 2018 yang dibuat untuk memenuhi standard pertandingan robot sepak bola beroda yang mengacu pada MSL Robocup. Besar harapan perancangan ini dapat memberikan manfaat bagi perkembangan robotika di Indonesia khususnya Divisi KRSBI Beroda.

Kata kunci: KRSBI Beroda, MSL Robocup, ROBOSOC UM

I. PENDAHULUAN

ROBOSOC UM yang berartikan *Robot Soccer UM*, merupakan Tim KRSBI Beroda Universitas Negeri Malang. KRSBI (Kontes Robot Sepak Bola Indonesia) Beroda [1] merupakan kontes bergengsi di Indonesia yang dimulai sejak tahun 2017. Kontes ini dibagi menjadi 4 regional di mana Tim ROBOSOC UM berada pada regional IV. Kontes ini mengadopsi pertandingan RoboCup pada liga ukuran menengah (*Middle Size League*) [2] yang memiliki ukuran panjang maksimal 50 cm, lebar maksimal 50 cm dan tinggi maksimal 80 cm dengan bobot maksimal 40 kg pada tiap robotnya.

II. PERANCANGAN ROBOT SEPAK BOLA BERODA

Bagian ini secara rinci mendeskripsikan mekanik robot, elektrik robot, penglihatan robot dan komunikasi robot.

A. Mekanik Robot

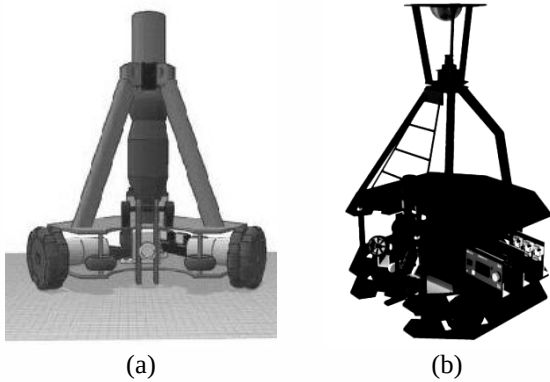
Mekanik robot merupakan sekumpulan rancangan komponen mekanik atau “raga” dari sebuah robot yang dalam hal ini menjadi medium terjadinya interaksi antara robot dengan lingkungannya. Perancangan mekanik merupakan sebuah penentu seberapa banyak robot dapat melakukan tindakan terhadap lingkungannya. Semakin banyak tindakan yang diharapkan, semakin kompleks pula perancangan mekanik robot yang dibutuhkan.

Pada rancangan mekanik robot kami pada tahun 2017 (Gambar 1.a), digunakan plat almunium 3 mm dengan tipe 6061 yang memiliki kekokohan lumayan tinggi sebagai basis robot dalam menyokong 3 buah motor penggerak yang digunakan. Selain itu juga

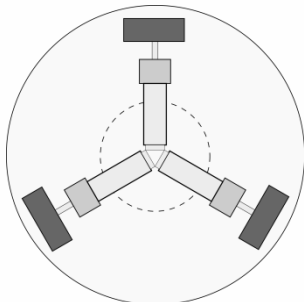
digunakan almunium lonjor berbentuk persegi dengan luas permukaan sebesar 4 cm² dengan ketebaan 1 mm dalam menyokong komponen pendukung seperti kamera, mini PC, pneumatic, dan komponen kecil lainnya. Sehingga didapatkan rancangan robot sepak bola yang ringan, kuat dan tahan benturan.

Pada perkembangan selanjutnya (Gambar 1.b), perancangan mekanik ROBOSOC UM mengalami perubahan secara besar jika dibandingkan dari perancangan mekanik tahun sebelumnya. Pada basis robot, kami menggunakan plat almunium dengan tebal 6 mm berbeda dengan tebal plat tahun sebelumnya. Pertimbangan penggunaan plat dengan tebal 6 mm yaitu kekokohan plat pada luas permukaan yang lebih besar. Plat dengan luas permukaan yang lebih besar, cenderung lebih mudah mengalami bengkok dan tidak stabil dalam menyokong komponen yang menempel di atasnya. Akan tetapi penggunaan plat almunium dengan ketebalan dua kali dari sebelumnya, berdampak pada berat robot yang naik dua kali lipat dan semakin menambah beban terima motor penggerak robot. Oleh karenanya, kami menggunakan penggerak robot yang dilengkapi dengan fitur *planetary gear-box*. Fitur ini memungkinkan penambahan torsi yang lebih optimal daripada sistem *gear-box* lainnya.

Robot ini menggunakan tiga buah motor penggerak yang pada tiap ujung penggeraknya menggunakan roda omni dengan diameter 100 mm. Baik pada perancangan tahun sebelumnya maupun tahun ini, keduanya menggunakan roda dan motor penggerak yang sama. Motor penggerak yang digunakan masing-masing mampu menahan beban sebesar 25 kgf.cm dengan kecepatan putar sebesar 500 rpm. Tiap motor penggerak dan rodanya (Gambar 2) diletakkan secara bersandingan membentuk sebuah segitiga dengan sudut yang terbentuk antar motornya sebesar 120°. Penggunaan dan peletakan motor ini diharapkan agar pergerakan robot dapat dengan bebas dan lancar bernavigasi sesuai dengan yang diinginkan.



Gambar 1. Rancang mekanik robot KRSBI Beroda (a) tahun 2017 dan (b) tahun 2018 ROBOSOC UM



Gambar 2. Rancang peletakan motor penggerak dan roda robot

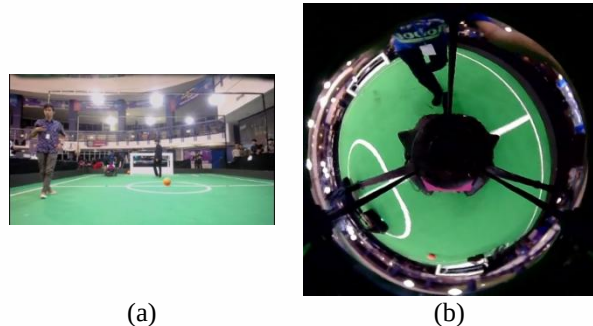
B. Sistem Elektrik Robot

Sistem elektrik robot merupakan sistem yang menjadikan mekanik robot dapat bekerja sesuai dengan yang diinginkan. Pada sistem ini, setiap robot menggunakan perangkat elektrik di antaranya, arduino uno sebagai pusat controller robot, solenoid sebagai komponen penendang robot, dan beberapa sensor seperti sensor jarak untuk mengukur adanya bola atau tidak pada sistem *dribbling*-nya.

C. Sistem Pengelihan Robot

Sistem pengelihan pada robot merupakan sebuah perangkat elektronika yang dibekali dengan sensor CMOS yang berfungsi menerima pantulan cahaya dari sebuah objek yang memiliki warna dan diubah menjadi sebuah matriks dalam satuan piksel yang lebih lanjut disebut kamera. Pada robot kami tahun 2017, hanya menggunakan 1 buah kamera dengan pabrikan Logitech tipe C525. Kamera ini cukup bagus untuk mendeteksi bola pada lapangan sepak bola. Akan tetapi *angle of view* dari kamera yang kurang besar, sehingga menjadikan pandangan yang dapat dijangkau hanya berkisar 60 derajat dari posisi kamera diletakkan. Oleh karenanya, pada rancang robot tahun 2018, kami menggunakan kamera dengan pabrikan Logitech dengan tipe C930e yang memiliki *angle of view* yang lebih besar yaitu 90 derajat. Hal ini memungkinkan robot dapat melihat posisi bola dengan lebih leluasa daripada sebelumnya dan menambah jangkauan pandangan robot dalam skala sudut.

Lebih lanjut, pada kompetisi robot sepak bola beroda regional IV pada tahun 2018, kami menggunakan dua buah kamera, yaitu kamera Logitech C930e sebagai kamera depan dan *low-cost omnidirectional camera* sebagai kamera yang dapat menjangkau 360° pada lingkungan disekitar robot. Kamera omni ini mampu bekerja hingga 120 fps dengan resolusi yang cukup tinggi. Penggunaan kamera omni ini diharapkan dapat membantu robot dalam bernavigasi mencari bola lebih baik lagi terutama dalam pencarian dan penentuan posisi bola. Gambar 3 menunjukkan hasil pengambilan gambar dari sistem pengelihan robot kami berikut derajat pengelihatannya pada lapangan pertandingan.



Gambar 3. Sistem pengelihan robot KRSBI Beroda tahun 2018 ROBOSOC UM: (a) kamera depan dan (b) kamera 360°

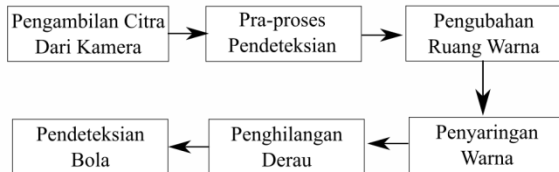
Pada pendeteksian bola oranye yang digunakan pada perlombaan, kami menggunakan library computer vision yaitu OpenCV [3]. Library ini memungkinkan penggunaannya dapat lebih mudah memberikan perintah kepada robot untuk mencari fitur dari bola oranye tersebut. Beberapa fitur yang disediakan di antaranya pengubahan ruang warna dari RGB menjadi HSV [4], RGB menjadi *grayscale*, begitu pula dengan fitur *blur* yang dikhususkan untuk penghalusan permukaan bola yang cenderung belum tampak bulat atau belum ternormalisasi dengan baik. Lebih lanjut, adanya fitur perhitungan *moment* pada library ini, mempermudah bagi robot untuk mencari titik tengah dari objek dan menentukan letak dimana bola berada.

D. Sistem Komunikasi Robot

Komunikasi antara robot merupakan koordinasi yang dilakukan oleh kesemua robot dengan *base-station* serta *base-station* dengan *referee box*. Pada perancangan ini, kami menggunakan *Robot Operating System (ROS)* [5] yang menyediakan fitur untuk mempermudah komunikasi dan manajemen antar komponen pendukungnya. Komunikasi yang terbentuk sebagaimana seperti peraturan yang telah ditentukan pada kontes robot sepak bola beroda 2018.

III. PENDETEKSIAN BOLA

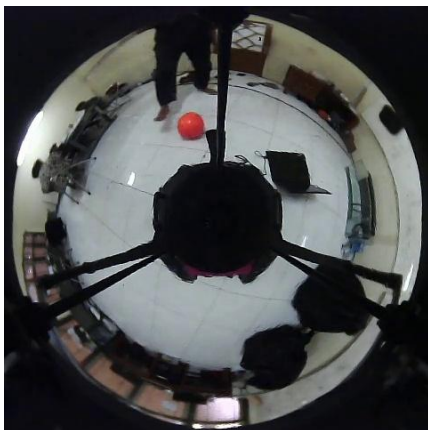
Pada bab ini, dijelaskan secara rinci proses pendeteksian bola. Bola yang digunakan merupakan bola futsal berwarna oranye dengan ukuran 4 dengan keliling 63-66 cm dengan berat sekitar 400 gram. Gambar 4 menunjukkan metode pendeteksian bola yang digunakan.



Gambar 4. Metode Pendeteksian Bola

A. Pengambilan Citra Dari Kamera

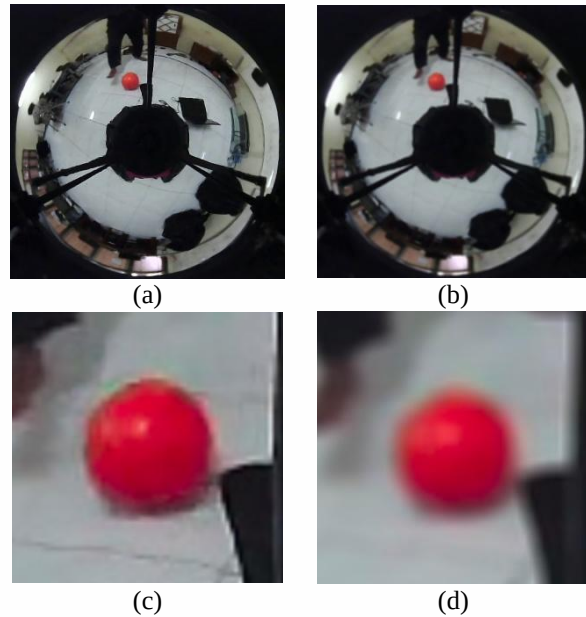
Pengambilan citra dari kamera merupakan tahapan di mana robot mengambil citra dari kamera omni. Pengambilan citra kamera tersebut diambil dalam ruang warna RGB. Ruang warna ini merupakan ruang warna yang umum digunakan untuk pengambilan citra digital menggunakan kamera. Gambar 5 menunjukkan hasil pengambilan citra menggunakan kamera omni.



Gambar 5. Pengambilan citra menggunakan kamera omni

B. Pra-proses Pendeteksian

Pada tahapan ini, dilakukan pemburaman citra yang didapatkan dari kamera. Pemburaman ini dimaksudkan untuk menormalisasi tiap piksel citra yang masih memiliki perbedaan warna RGB yang mencolok. Dengan kata lain, proses pemburaman ini dilakukan untuk menghilangkan derau warna akibat kualitas sensor kamera yang kurang baik (Gambar 6 (c)). Gambar 6 menunjukkan proses pemburaman yang dilakukan terhadap citra yang didapatkan. Gambar 6 (a) merupakan citra sebelum dan Gambar 6 (b) merupakan hasil dari pemburaman citra awal 6 (a). Apabila citra 6 (a) dan 6 (b) diperbesar, maka keadaan piksel pada objek bola sebelum dan sesudah dilakukan pemburaman ditunjukkan pada Gambar 6 (c) dan 6 (d).



Gambar 6. Proses pemburaman citra: (a) sebelum dan (b) sesudah dilakukan pemburaman, serta keadaan piksel (c) sebelum dan (d) sesudah dilakukan pemburaman pada citra (a) dan (b) ketika dilakukan perbesaran

C. Pengubahan Ruang Warna

Pengubahan ruang warna dari RGB (*Red, Green, Blue*) menjadi HSV (*Hue, Saturation, Value*) dilakukan untuk mempermudah penyaringan warna pada tahap selanjutnya. Pada tahap ini, tiap kanal penyusun warna RGB dikonversikan menggunakan persamaan 1 sampai 3, di mana *value* merupakan nilai tertinggi dari ketiga kanal penyusun RGB. *Saturation* merupakan kemurnian dari sebuah warna, semakin besar nilainya, maka semakin pekat pula sebuah warna terlihat. Begitu sebaliknya, warna biru muda memiliki kecenderungan warna terhadap warna putih atau *chromatic*. Dengan kata lain, kemurnian warna biru pada warna biru muda memiliki nilai yang kecil atau kemurnian rendah. Berbeda dengan warna biru sendiri yang memiliki kemurnian warna yang pekat atau tinggi.

Selain itu, kanal *Hue* merupakan representasi dari nilai warna yang dapat ditangkap oleh mata. Warna primer yang tampak, merah, hijau, dan biru, diletakkan pada sebuah sudut warna yang masing-masing warnanya memiliki jangkauan sudut berjumlah 120 derajat. Pada sudut warna ini, percampuran warna dari ketiga warna primer dapat dengan mudah ditentukan nilainya. Dimulai dari sudut 0 yang merupakan warna merah, sudut 120 adalah hijau, dan sudut 240 adalah biru.

$$V = \max(R, G, B) \quad (1)$$

$$S = \begin{cases} \frac{V - \min(R, G, B)}{V} & \text{jika } V \neq 0 \\ 0 & \text{lainnya} \end{cases} \quad (2)$$

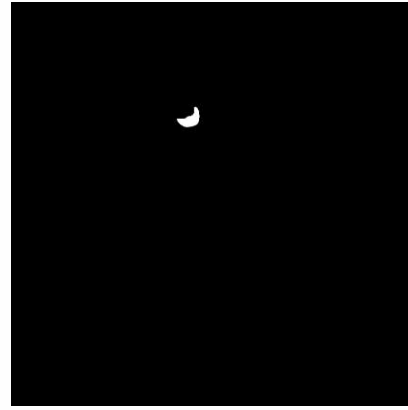
$$H = \begin{cases} \frac{60(G-B)}{V-\min(R,G,B)}, & \text{jika } V = R \\ \frac{120+60(B-R)}{V-\min(R,G,B)}, & \text{jika } V = G \\ \frac{240+60(R-G)}{V-\min(R,G,B)}, & \text{jika } V = B \end{cases} \quad (3)$$

D. Penyarangan Warna

Tahap ini merupakan tahap di mana penyarangan warna terhadap setiap kanal warna HSV. Penyarangan ini dilakukan untuk mencari letak warna objek bola dengan warna oranye. Warna oranye sendiri berada pada pencampuran warna merah dan kuning, sedangkan warna kuning merupakan pencampuran warna merah dengan hijau. Lebih lanjut, warna merah memiliki sudut 0, sedangkan warna hijau memiliki sudut 120. Pada percampuran keduanya ada warna kuning yang berada pada sudut 60. Pada pencampuran kuning dan merah, ada warna oranye. Oleh karenanya, sudut warna oranye berada pada sudut 30. Untuk memperoleh hasil yang lebih baik, dilakukan penyarangan antara sudut 30 yang menyerupai *band-pass filter*. Diberikan nilai toleransi 10 nilai sudut. Sehingga warna oranye pada penyarangan yang dilakukan, berada pada batas sudut bawah 20 dan batas sudut atas 40. Hasil penyarangan warna tersebut, berupa citra dengan dua jenis warna, yaitu hitam dan putih atau sering disebut dengan citra biner. Hitam berarti warna yang dihilangkan, warna putih merupakan warna yang tersaring atau diloloskan yang selanjutnya disebut objek bola.

E. Penghilangan Derau

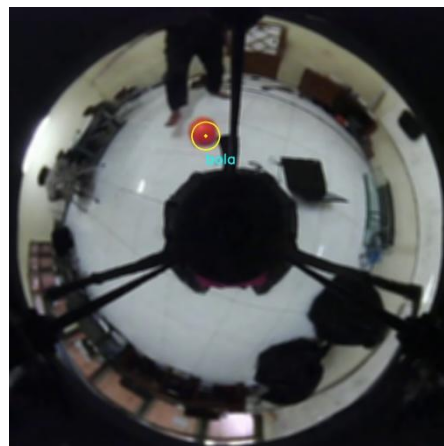
Pada tahap ini, citra hasil penyarangan tidaklah langsung memperoleh hasil yang bagus persis seperti objek yang diinginkan dan bersih. Oleh karenanya, perlu adanya penghilangan derau yang ditujukan untuk menghilangkan objek yang dianggap serupa namun bukan objek bola yang diinginkan. Hal ini muncul dikarenakan pada citra yang disaring, masih ada objek lain dengan warna yang menyerupai warna bola yaitu oranye atau piksel dari objek tersebut masih memiliki warna yang menyerupai. Oleh karenanya, dilakukan proses permukaan objek yang disebut proses bukaan. Proses ini terdiri dari proses erosi kemudian dilakukan proses dilasi. Proses erosi bertujuan untuk menghilangkan objek putih yang berupa titik-titik yang menjadi derau dan harus dibuang, sedangkan proses dilasi menebalkan objek yang tersisa dari operasi erosi sebelumnya agar objek tersebut tidak menjadi kecil dan menghilang karena dianggap sebagai derau oleh operasi erosi. Gambar 7 menunjukkan hasil deteksi objek bola setelah dilakukan penyarangan warna dan penghilangan derau.



Gambar 7. Hasil deteksi bola setelah dilakukan penyarangan warna dan penghilangan derau

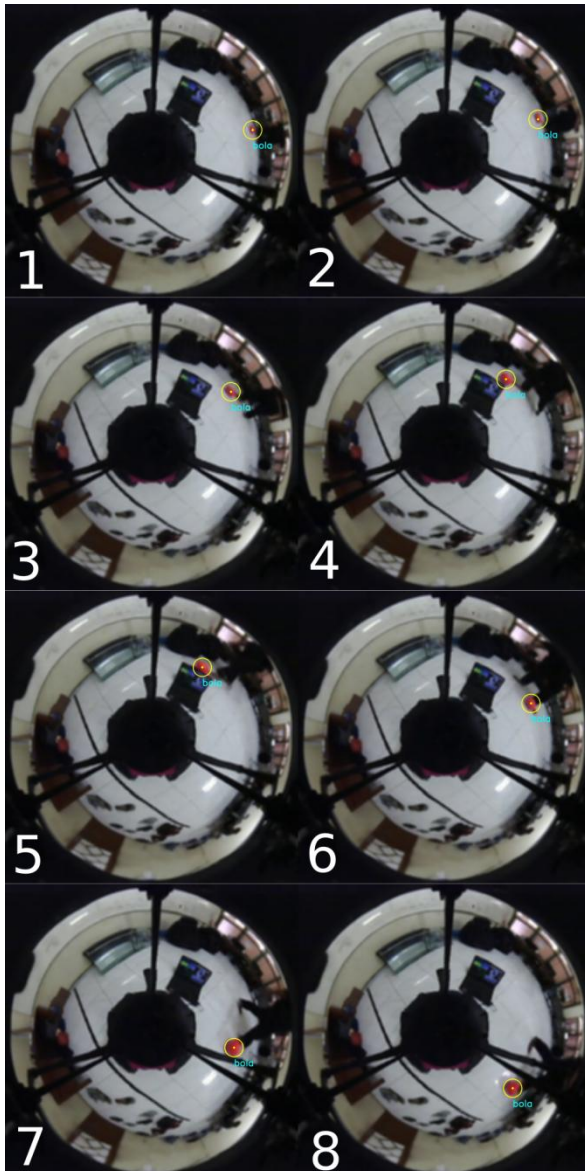
F. Pendeteksian Bola

Setelah objek bola dapat tersaring dengan baik antara objek bola dan latar belakangnya, barulah dilakukan pendeteksian bola. Pendeteksian bola yang dimaksudkan adalah pengasumsian objek yang terdeteksi sebagai bola oleh sistem. Objek putih hasil dari tahap sebelumnya, akan dihitung luasannya objeknya dan diketahui titik tengahnya. Perhitungan ini menggunakan metode moment berbasis contour atau luasan objek yang menghasilkan titik berat objek terdeteksi pada koordinat citra. Titik tengah tersebut selanjutnya digunakan sebagai acuan dimana posisi bola berada terhadap robot dan posisinya ditunjukkan dengan label “bola” yang berada disebelah objeknya yang mengikuti letak bola berada. Gambar 8 menunjukkan hasil pendeteksian bola dan posisinya.



Gambar 8. Pendeteksian bola

Pada percobaan yang telah dilakukan, bola diputar-putar mengitari robot selagi melakukan proses pendeteksian. Indikator sistem dapat mendeteksi bola ditandai dengan adanya label berupa garis lingkaran dengan label “bola” mengikuti kemana arah bola bergerak. Pada Gambar 8 menunjukkan percobaan pendeteksian bola oranye pada robot menggunakan metode yang telah diusulkan.



Gambar 4. Pendeteksian bola oranye pada robot menggunakan metode usulan

IV. RISET SELANJUTNYA

Kedepannya, Tim ROBOSOC UM akan terus mengembangkan robot kami terutama pada *low-cost omnidirectional camera* agar dapat lebih baik lagi dan dapat menjangkau keseluruhan lapangan pada dimensi yang sebenarnya di pertandingan MSL Robocup. Selain itu, solenoid pada sistem penendang robot kami, agar dapat lebih dioptimalkan kembali baik dari konsumsi daya dan jarak serta kecepatan tembakan bola.

V. KESIMPULAN

Dari metode pendeteksian bola yang diusulkan, bola dapat dideteksi dengan baik mengikuti kemana arah bola bergerak. Besar harapan kami dapat memberikan manfaat bagi perkembangan robotika di Indonesia khususnya Divisi KRSBI Beroda. Maju Terus Robotik Indonesia.

VI. UCAPAN TERIMA KASIH

Tim ROBOSOC UM mengucapkan terimakasih kepada Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Malang yang telah mendukung riset tim kami mulai dari pendanaan, tempat riset dan semangat untuk terus maju.

VII. DAFTAR PUSTAKA

- [1] "BUKU PANDUAN KONTES ROBOT SEPAKBOLA INDONESIA DIVISI BERODA (KRSBI Beroda) 2018," 2018. [Online]. Available: <http://kontesrobotindonesia.id/data/2018/PanduanKRSBIBeroda2018.pdf>.
- [2] M. Asada *et al.*, "MSL Technical Committee 1997–2016," p. 101.
- [3] G. Bradski, "The OpenCV Library," *Dr Dobbs J. Software Tools*, 2000.
- [4] R. Gonzalez and P. Wintz, *Digital image processing*. Addison-Wesley Publishing Co., Inc., Reading, MA, 1977.
- [5] M. Quigley *et al.*, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, 2009, vol. 3, p. 5.

Rancang Bangun Mekanik Ekspansi Robot Goalkeeper Sepak Bola Beroda Menggunakan Pneumatik

Muhammad Rafii Naufal¹⁾, Rila Inda Belga²⁾, Ovi Tri Hartatik³⁾, Suprayoga Erdin Wicaksono⁴⁾, Ahmad Wafi Nurmukti Wibowo⁵⁾, Alwi Widi Pradana⁶⁾, Sigit Yatmono, M.T.⁷⁾

Yogyakarta State University (Universitas Negeri Yogyakarta)
Kampus Karangmalang, Jl. Colombo No 1 Yogyakarta

⁽¹⁾muhammad.rafi@student.uny.ac.id

Abstract—*The newest division in Indonesian Robot Contest, Indonesian Wheeled Soccer Robot Contest, an interested division to followed, because they have many newest technologies and innovations that included in the robot, the wheeled soccer robots. One of the newest innovation is the expansion system in the goalkeeper robot. This innovation helps the goalkeeper robot to block the ball to avoid a goal made for the opponent team. The purpose of this paper is to discuss how expansion system works in the goalkeeper robot, so the robot can move the expansion part with the pneumatic system. Distance of the ball that can be detected by goalkeeper robot from range 5cm to 40cm, it means at a distance over that (41cm or above) expansion system of the robot can't be activated.*

Keywords—*goalkeeper robot; soccer; wheeled; expansion system; electro pneumatic.*

I. PENDAHULUAN

Pertandingan antara robot dengan tim juara dunia dalam permainan sepakbola yang diperkirakan dilaksanakan pada tahun 2050 adalah cita-cita dari para pendiri *Robocup*. Para peneliti robot sepak bola ini terus melakukan berbagai riset. Pembagian divisi robot pun dibagi menjadi dua, yakni divisi berkaki atau *humanoid* yang mempunyai bentuk seperti manusia dan divisi *middle size league* yang mempunyai penggerak berupa motor. Divisi *humanoid* berkonsentrasi dalam membangun sebuah mekanik robot pemain sepakbola layaknya manusia. Divisi *middle size league* berkonsentrasi pada AI (*Artificial Intelligence*) untuk mengenali pergerakan bola, lawan, dan teman.

Wibowo (2016) mengungkapkan, ada banyak kontes robot di Indonesia, salah satunya adalah Kontes Robot Sepak Bola Indonesia (KRSBI), dimana perlombaan kontes robot sepak bola terbagi dalam dua divisi, yakni divisi berkaki atau *humanoid* dan robot beroda. Robot sepak bola memiliki berbagai kemampuan dalam menjalankan fungsinya seperti mencari bola, mencari gawang, menghindari musuh, mengumpan bola, hingga menendang bola. Semua fungsi tersebut berjalan secara otomatis berdasarkan proses kendali yang dilakukan robot [1].

Dalam penelitian ini, difokuskan pada bagaimana robot sepak bola beroda, khususnya robot *goalkeeper* menangkis bola ke arah luar gawang. Cara konvensional, robot menangkis bola hanya dengan komponen tambahan berupa tiang besi tanpa penggerak aktuator. Pada saat bola ditendang robot lawan ke arah robot *goalkeeper*, robot *goalkeeper* mengarah ke bola yang ditendang, ditangkis, dan langsung ditendang ke arah depan gawang, agar tidak terjadi sebuah gol. Tangkisan ini terkadang membahayakan, karena bisa jadi ketika ditangkis, bola malah terpental ke gawang sendiri, sehingga tercipta sebuah gol untuk tim lawan.

Untuk itu, diusulkanlah sebuah desain mekanik penangkis bola agar tidak membahayakan gawang sendiri menggunakan pneumatik. Desain ini berupa ekspansi menggunakan besi diameter 5 mm dengan panjang 10 cm. Besi tersebut bisa bergerak bebas ke atas dan bawah dari bagian atas robot *goalkeeper*. Besi tersebut diletakkan segaris lurus dengan tiang tambahan robot *goalkeeper*. Ketika besi tersebut dialiri angin melalui pneumatik, maka secara otomatis besi tersebut keluar dari dalam tiang tambahan dan tepat berada di atas robot *goalkeeper*. Proses pengeluaran besi ke atas bagian robot *goalkeeper* ini, bisa dimanfaatkan untuk menjadikan pendorong untuk menangkis bola. Teknik ini mengadopsi mekanik pada sebuah sistem manufaktur fleksibel di industri yang menggunakan pneumatik.

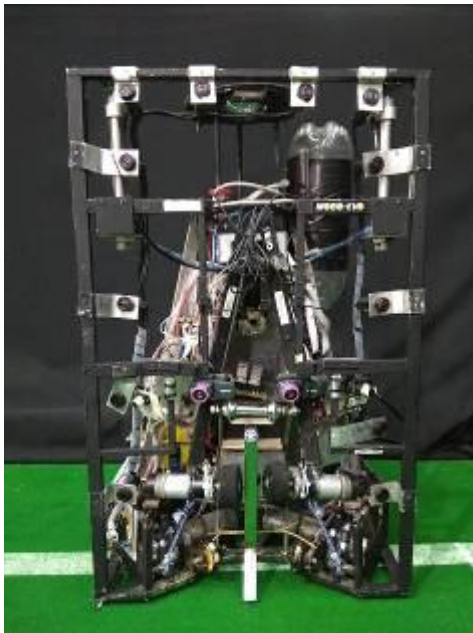
Ketika robot akan menangkis bola, maka tiang harus dialiri dengan tegangan listrik dan angin melalui elektro pneumatik sehingga sistem ekspansi bisa keluar menangkis bola yang dimanfaatkan untuk dorongan pada tangkisan. Ketika proses menangkis bola sudah selesai, aliran listrik dan angin pada tiang tambahan ini mati secara otomatis, sehingga posisi tiang besi kembali seperti semula. Pada saat proses keluarnya tiang besi oleh pneumatik ini, terjadilah gaya dorongan yang sangat kuat. Dorongan ini bisa membuat sesuatu yang ditabraknya, seperti bola bisa terlempar jauh, sehingga gawang aman dari tendangan lawan. Bila bola diletakkan tepat di depan sistem ekspansi ini, maka tiang besi ini akan

terdorong dengan kuat keluar dari tiang tambahan yang ada di robot *goalkeeper*.

II. DASAR TEORI

A. Robot Goalkeeper Sepak Bola Beroda

Dalam Aturan Main KRSBI (2018) dijelaskan, bahwa KRSBI Beroda adalah divisi terbaru pada KRSBI yang merupakan pengembangan dari KRPAI Beroda. Robot beroda ini dapat bermain sepak bola dan bergerak secara autonomus mencari bola dan gawang sendiri tanpa bantuan kendali manusia. Aturan kompetisi KRSBI Beroda mengacu pada *RoboCup Soccer Middle Size League (MSL)*. Tidak hanya dapat mencari bola dan gawang lawan saja, Robot beroda ini juga dapat menghadang bola untuk masuk kedalam gawangnya sendiri. Hal itu dapat dilakukan oleh Robot *goalkeeper* dengan menggunakan ekspansi ke kanan, kiri dan atas. Ukuran panjang dan lebar robot *goalkeeper* ini 30 hingga 52 cm dan tinggi antara 40 hingga 80 cm. Robot *goalkeeper* ini boleh bertambah panjang ke kiri, ke kanan dan ke atas hingga lebar maksimum 60 cm dan tinggi maksimum menjadi 90 cm. Perubahan hanya boleh terjadi sesaat saja, paling lama 1 detik, yaitu pada saat bola mendekat [2].



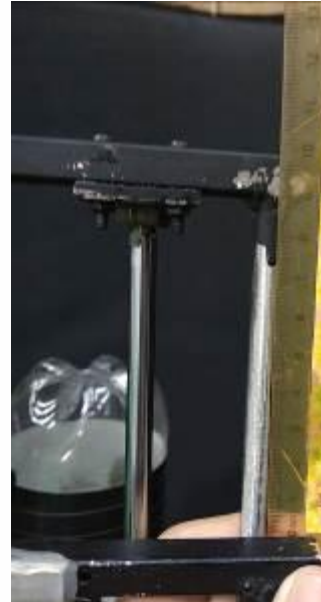
Gambar 1. Robot Goalkeeper KRSBI Beroda Tim MOBO-EVO UNY 2018 pada Posisi Default

Sumber : Dokumentasi Pribadi

B. Ekspansi

Ekspansi adalah suatu upaya untuk bertambah dalam segi ukuran, banyaknya, maupun kepentingannya atau dengan kata lain membuat suatu hal meningkat atau bertambah besar. Pada robot *goalkeeper* ini, sistem ekspansi digunakan ke arah kanan, kiri dan atas untuk menghadang bola

yang mendekat. Untuk ke arah atas, ekspansi hanya diperbolehkan menambah 10 cm saja, dikarenakan ukuran robot penyerang maksimum 80 cm, sedangkan robot *goalkeeper* diperbolehkan hingga 90 cm.



Gambar 2. Panjang Ekspansi pada Robot Goalkeeper Setinggi 10 cm

Sumber : Dokumentasi Pribadi

C. Pneumatik

Kurniawan (2008: 42) menjelaskan, sistem pneumatik adalah sistem yang menggunakan tenaga yang disimpan dalam bentuk udara yang dimampatkan serta dimanfaatkan untuk menghasilkan kerja. Komponen-komponen yang digunakan dalam sistem pneumatik adalah (1) kompresor, (2) *reservoir*, (3) *air service unit*, (4) katup, yang mencakup: katup pengatur arah, katup pengatur laju aliran, dan katup pengatur tekanan, (5) aktuator, baik gerakan linier maupun gerakan rotasi, dan pada akhirnya digunakan (6) sensor, untuk pendeteksian pada proses [3].



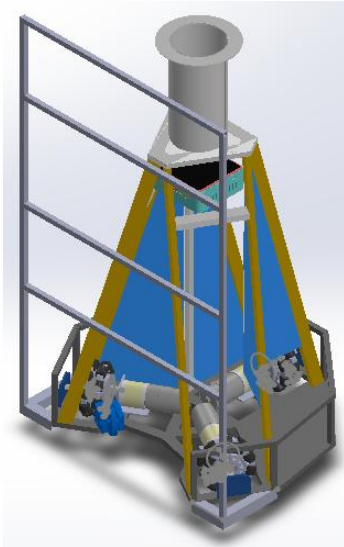
Gambar 3. Komponen utama pada sistem pneumatik

Sumber: <https://library.automationdirect.com/why-use-pneumatics/>

III. METODE

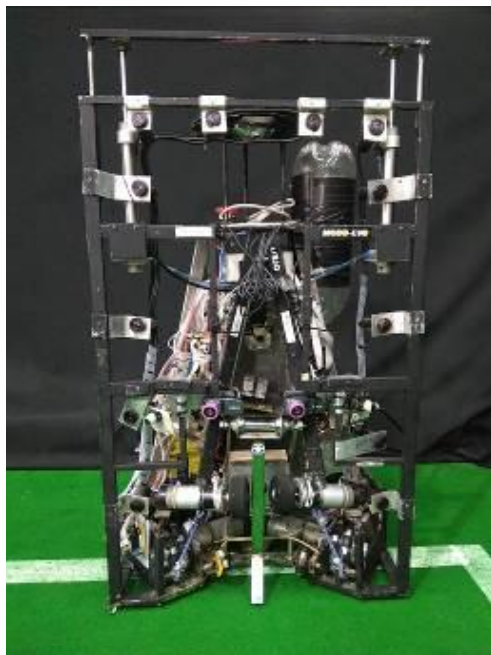
A. Desain Robot Goalkeeper

Desain robot *goalkeeper* sepak bola beroda ditunjukkan pada gambar 4. Penangkis berada pada bagian atas depan robot *goalkeeper*. Penangkis mempunyai satu mekanisme tangkisan yaitu tangkisan ke atas. Tetapi, sistem ekspansi ini dapat bergeser ke atas dan ke bawah. Untuk menghasilkan suatu tangkisan ke atas, maka robot menggunakan mekanik ekspansi bergeser ke atas. Untuk arah bawah berfungsi untuk kembali ke posisi *default* ekspansi robot *goalkeeper*.



Gambar 4. Desain awal robot *goalkeeper* posisi *default*

Sumber : Dokumentasi Pribadi

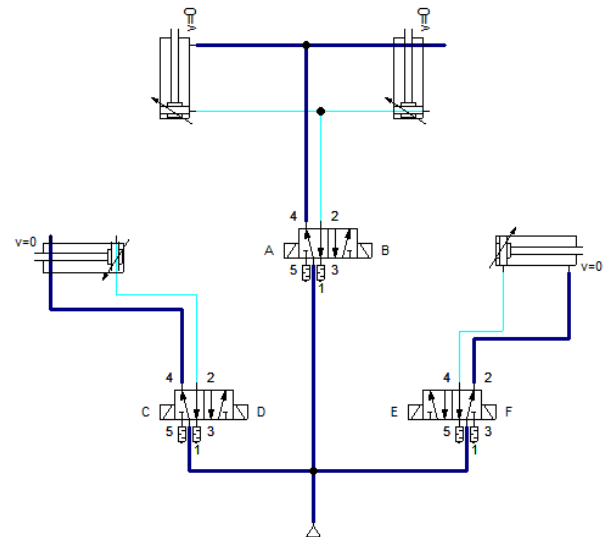


Gambar 5. Robot *Goalkeeper* KRSBI Beroda Tim MOBO-EVO UNY 2018 pada Posisi Ekspansi Atas Aktif

Sumber : Dokumentasi Pribadi

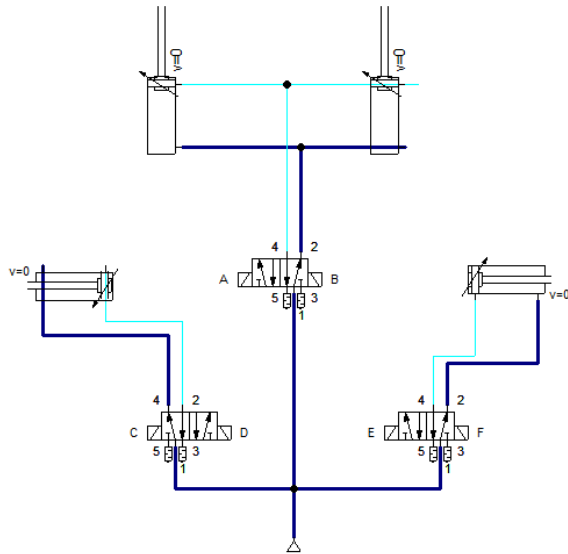
B. Desain Wiring Pneumatic

Pengisian angin dilakukan dengan mengatur regulator sesuai tekanan yang dibutuhkan *valve* sebesar 4 bar, sehingga tabung yang terdapat pada robot *goalkeeper* dalam keadaan berisi tekanan. Sensor *Proximity* sebelum menerima data, sistem ekspansinya belum aktif atau berada pada posisi *default*. Ekspansi akan aktif jika sensor *proximity* mengirim data yang diterjemahkan, kemudian dikirim pada 3/2 way *electric valve*, lalu katup 1 mengirim tekanan sehingga katup B akan terbuka. Tekanan dari katup 2 akan dialirkan ke silinder, sehingga silinder bergerak maju. Silinder bekerja selama 1 detik. Setelah 1 detik, silinder akan kembali pada kondisi *default*. Silinder lalu mundur, sehingga katup A akan terbuka dan tekanan akan terhisap. Penjelasan ini menjelaskan skema yang ada di Gambar 6 dan 7.



Gambar 6. Wiring Pneumatik Robot *Goalkeeper* Tim MOBO-EVO UNY 2018 pada Posisi Ekspansi Belum Aktif

Sumber : Dokumentasi Pribadi

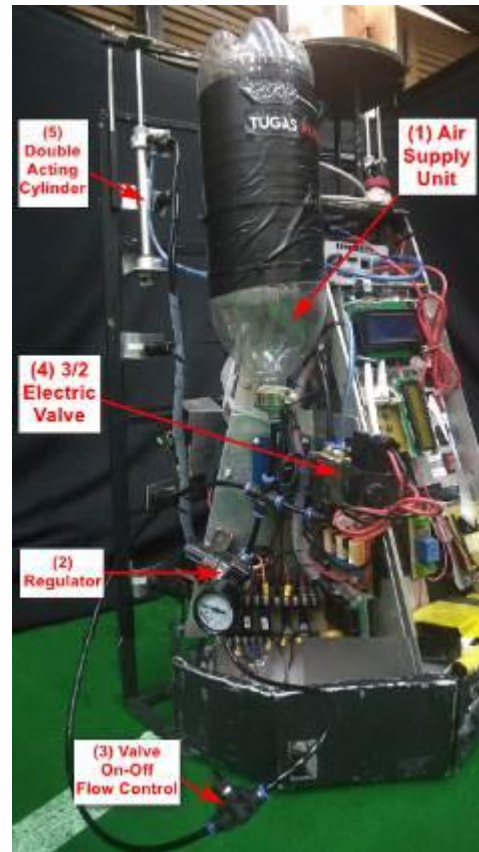


Gambar 7. Wiring Pneumatik Robot Goalkeeper Tim MOBO-EVO UNY 2018 pada Posisi Ekspansi Atas Aktif

Sumber : Dokumentasi Pribadi

C. Perakitan

Tahap perancangan (*design*) telah dilaksanakan, selanjutnya tahap perakitan. Dalam tahap ini, melaksanakan apa yang telah dirancang. Komponen utama pneumatik yang digunakan yakni: (1) *Air Supply Unit* sebagai penyedia stok udara yang digunakan dalam sistem pneumatik berupa dua buah botol, (2) *Regulator* sebagai pembatas tekanan angin yang masuk, tekanan angin dapat diketahui melalui indikator yang berupa barometer, (3) *Valve On-Off Flow Control* untuk menambah atau mengurangi laju aliran angin dengan cara membuka atau menutup katup, (4) *3/2 Electric Valve* sebagai pengendali aliran udara yang masuk dari *air supply unit* ke silinder, terdiri dari 3 buah, dan (5) *Double Acting Cylinder*, sebagai aktuator penggerak sistem ekspansi.



Gambar 8. Robot Goalkeeper Tim MOBO-EVO UNY 2018 yang sudah dirakit sistem ekspansi-nya dengan pneumatik

Sumber : Dokumentasi Pribadi

Berdasarkan Gambar 8, alur cara kerja dari sensor *proximity* hingga nyala-nya sistem ekspansi sebagai berikut:

1. Bola terdeteksi oleh sensor *proximity*,
2. Data berupa ON dikirim ke 3/2 way *electric valve*,
3. Aliran udara dari *air supply unit* mengalir ke 3/2 way *electric valve*,
4. Valve mengalirkan udara ke *double acting cylinder*,
5. Sistem ekspansi aktif hanya selama 1 detik, lalu kembali ke posisi semula lagi.

IV. HASIL DAN DISKUSI

Proses pengujian dilakukan untuk mengetahui jarak seberapa jauh bola dapat dideteksi oleh sensor *proximity* sehingga dapat mengaktifkan sistem ekspansi yang sudah terpasang pada *robot goalkeeper*.

Jarak Bola Terhadap Sensor Proximity	Status Sistem Ekspansi
5 cm	Aktif
10 cm	Aktif
15 cm	Aktif
20 cm	Aktif
25 cm	Aktif
30 cm	Aktif
35 cm	Aktif
40 cm	Aktif
45 cm	Tidak Aktif
50 cm	Tidak Aktif

Tabel 1. Uji coba jarak yang dapat mengaktifkan ekspansi

Pada pengujian jarak deteksi, robot *goalkeeper* dapat mengaktifkan sistem ekspansi dengan *range* jarak mulai dari 5 centimeter hingga 40 centimeter saja. Jarak 40 centimeter ke atas atau lebih, sistem ekspansi pada robot *goalkeeper* tidak dapat aktif.

V. KESIMPULAN

Sesuai dengan hasil pengujian, dapat ditarik kesimpulan bahwa jarak yang dapat dideteksi oleh robot *goalkeeper* berkisar dari 5 centimeter hingga 40 centimeter saja, sehingga pada jarak di atas itu (41 centimeter atau lebih), sistem ekspansi pada robot *goalkeeper* tidak dapat aktif.

VI. DAFTAR PUSTAKA

- [1] I. K. Wibowo, dkk. “Rancang Bangun Mekanik Penandang Pada Robot Soccer Beroda Menggunakan Solenoid”. Surabaya : Politeknik Elektronika Negeri Surabaya - The 4th Indonesian Symposium on Robot Soccer Competition 2016, June 1th, 2016.
- [2] Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia. “Aturan Main KRSBI 2018”. Halaman 12. (PDF). ([http://kontesrobotindonesia.id/data/2018/PanduanKRSBI Beroda2018.pdf](http://kontesrobotindonesia.id/data/2018/PanduanKRSBI%20Beroda2018.pdf)), 2018. Diakses pada tanggal 1 Juni 2018.
- [3] R. Kurniawan. “Rekayasa Rancang Bangun Sistem Pemindahan Material Otomatis Dengan Sistem Elektro-Pneumatik”. Yogyakarta : Universitas Atma Jaya - Jurnal Ilmiah Teknik Mesin CAKRAM Vol. 2 No. 1, Juni 2008 (42 –47).

Tracking Warna Menggunakan Ruang Warna HSV pada Robot Scada Rt

Muhammad Sadam*, Gunawan Wijaksono, Sahar Patra, Dodi Amir, Moh. Zulfikar H. Yasin, Aria Nurul Haq, Firmansyah, Ardi Amir
Univesitas Tadulako Palu

Jalan Soekarno Hatta Km 9 Kampus Bumi Tadulako Tondo Palu-Sulawesi Tengah
ukm.robotech.tadulako@gmail.com

Abstrak -- Pada Kontes Robot Sepak Bola Indonesia Beroda 2018, terdapat banyak masalah pada robot terutama pada pembacaan suatu objek. Agar robot mampu bertanding secara maksimal di ajang KRSBI BERODA 2018 ini dibutuhkan kemampuan-kemampuan dasar yang dapat dilakukan robot seperti membaca bola. Ini menjadi tantangan tersendiri bagi robot-robot yang akan bertanding, sehingga pada Robot SCADA RT menggunakan OpenCV untuk mendeteksi objek. Adapun prosesnya seperti pengambilan warna melalui camera kemudian dikonversi dari RGB ke HSV, dilakukan color filtering dan thresholding sehingga dapat membaca bola dengan baik.

Kata kunci: OpenCV, RGB, HSV, color filtering, thresholding

I. PENDAHULUAN

Perkembangan teknologi di bidang robotika, khususnya pada divisi KRSBI BERODA, terbilang cukup pesat. Para peserta dituntut untuk melakukan peningkatan kemampuan robot sehingga dapat menyesuaikan dengan robot yang berskala ROBOCUP MSL. Dalam bermain sepak bola, robot harus dapat membaca bola yang ada di lapangan, sehingga ini yang menjadi masalah tersendiri pada pertandingan.

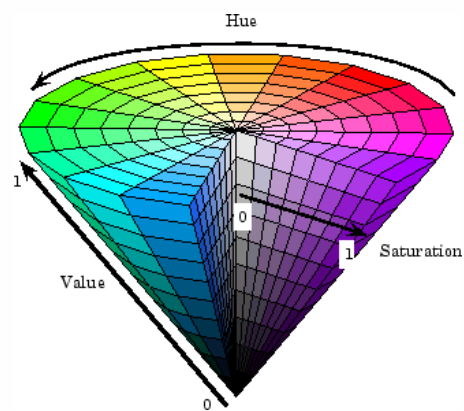
Kamera berfungsi sebagai pengambil citra dari lingkungannya dan untuk mengetahui letak posisi bola. Seperti halnya mata pada manusia yg memiliki batas pandang, diperlukan bantuan khusus untuk menambah luas arah pandang dengan menggerakkan ke atas ataupun ke bawah. Gerakan ini mengikuti jarak bola terhadap robot. Selain gerakan tersebut, ditambahkan pula gerakan manuver robot khusus seperti gerakan memutar kiri atau putar kanan sehingga robot dapat mencari letak bola dengan sempurna.

Dasar pengambilan citra warna oleh kamera yaitu dengan proses HSV. HSV mendefinisikan warna dalam terminologi Hue, Saturation dan Value. Keuntungan HSV adalah terdapat warna-warna yang sama dengan yang ditangkap oleh indra manusia. Sedangkan warna yang dibentuk model lain seperti RGB merupakan hasil campuran dari warna-warna primer[5].

Melalui model gambar 1, HSV memiliki 3 karakteristik pokok, yaitu Hue, Saturation dan Value. Apa yang dimaksud ketiganya?

a. Hue: menyatakan warna sebenarnya, seperti merah, violet, dan kuning dan digunakan menentukan kemerahan, kehijauan, dsb.

- b. Saturation: kadang disebut *chroma* yaitu kemurnian atau kekuatan warna.
c. Value: kecerahan dari warna. Nilainya berkisar antara 0-100%. Apabila nilainya 0, warnanya akan menjadi hitam. Semakin besar nilai maka semakin cerah dan muncul variasi-variasi baru dari warna tersebut.



Gambar 1. Model warna HSV

Dengan menggunakan sebuah laptop untuk basis kontrol robot terhadap kamera, pengolahan ruang warna oleh kamera dapat diproses secara penuh untuk mendeteksi sebuah objek.

II. ALAT DAN BAHAN

Adapun material yang digunakan tim SCADA RT yakni terbagi menjadi 2 yaitu *Hardware* dan *Software*

A. Hardware

1. Sensor

SCADA RT menggunakan kamera *webcam* sebagai sensor. Adapun kamera *webcam* yang dipakai yakni Logitech C525. *Webcam* adalah sebuah periferal berupa kamera sebagai pengambil citra/gambar dan mikrofon (*optional*) sebagai pengambil suara/audio yang dikendalikan oleh sebuah komputer atau oleh jaringan komputer. Gambar yang diambil oleh *webcam* ditampilkan ke layar monitor. Karena dikendalikan oleh komputer, ada *interface* atau *port* yang digunakan untuk menghubungkan *webcam* dengan komputer atau jaringan. Ada beberapa orang mengartikan *webcam* sebagai *web pages + camera*,

karena dengan menggunakan *webcam* untuk mengambil gambar video secara aktual bisa langsung diunggah bila komputer yang mengendalikan terkoneksi internet [4].



Gambar 2. Kamera Logitech C525

2. Interface

SCADA RT menggunakan laptop sebagai pemroses gambar pada *webcam* Logitech C525. Adapun laptop yang digunakan yaitu ASUS A455L. Laptop secara signifikan lebih lambat daripada komputer *desktop*. Naamun kemajuan teknologi manufaktur telah memungkinkan laptop melakukan hampir sama dengan komputer *desktop*. Bahkan, laptop *high-end* sering melakukan lebih baik daripada komputer *desktop* yang mempunyai spesifikasi rendah. 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, yang memungkinkan pengguna untuk mengakses internet tanpa memerlukan kabel[5].

Tabel 1. Spesifikasi laptop

Prosesor	Intel Core i3-4030U 1,9 GHz
RAM	DDR3L 2GB + DDR3L 4GB
Layar	14 Inch
Port	VGA, HDMI, 2x USB 2.0, 1x USB 3.0, RJ-45, SD card reader, 1x audio jack
Konektivitas	Bluetooth V4.0, WiFi, Ethernet



Gambar 3. Laptop ASUS A455L

B. Software

Adapun *software* yang digunakan untuk proses gambarnya yaitu *OpenCV* yang dipakai dalam mengimplementasikan *HSV* pada robot.



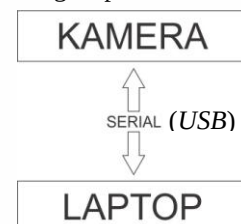
Gambar 4. OpenCV

OpenCV (*Open Computer Vision*) adalah program sumber terbuka berbasis C++ yang saat ini banyak digunakan sebagai program *computer vision*. Salah satu penerapannya yaitu pada robotika. Interaksi antara manusia dan robot (*human robot interaction*) dapat dihadirkan dengan *OpenCV*. Misalnya, wajah manusia dideteksi oleh *webcam*, lalu diproses oleh komputer dan kemudian oleh robot untuk aksi tertentu, misalnya mengikuti/mengenal wajah orang tersebut. Kesemuanya itu membutuhkan *OpenCV* sebagai program utama antara *webcam* dan pengolahannya, yaitu komputer[6]. Dengan *vision* tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali bentuk suatu objek. Beberapa pengimplementasian dari *computer vision* adalah *face recognition*, *face detection*, *face/object tracking*, *road Tracking*, dll. *OpenCV* adalah *library Open Source* untuk *computer vision* untuk C/C++, *OpenCV* didesain untuk aplikasi *real-time*, memiliki fungsi-fungsi akuisisi yang baik untuk *image/video* [1]. Adapun *OpenCV* yang digunakan yaitu *OpenCV* versi 3.4.0 yang terpasang di *python*.

II. METODE PENELITIAN

A. Komunikasi Kamera dan Laptop

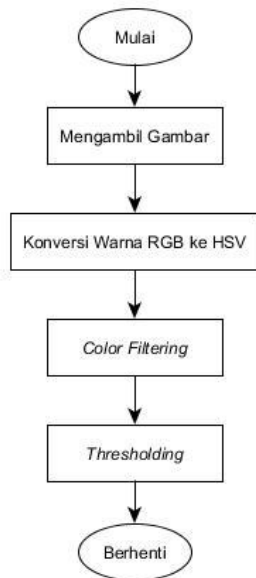
Koneksi dari masing-masing komponen yang digunakan adalah *serial* (Gambar 5). Komunikasi *serial* dapat lebih menghemat penggunaan penghantar dibanding komunikasi paralel. Karena, *serial* mengirim satu *bit* data dan selanjutnya diikuti oleh *bit-bit* data yang lain pada jalur yang sama. Namun kecepatan komunikasi *serial* tidak secepat komunikasi paralel. Secara umum penerapan komunikasi *serial* dapat memenuhi berbagai aplikasi mikrokontroler.



Gambar 5. Blok diagram komunikasi kamera dan laptop

B. Proses Pembacaan Gambar

Untuk membaca sebuah objek seperti bola, diperlukan pendekatan dengan warna objek yakni *HSV* yang memiliki karakteristik yang hampir sama dengan warna *RGB*.



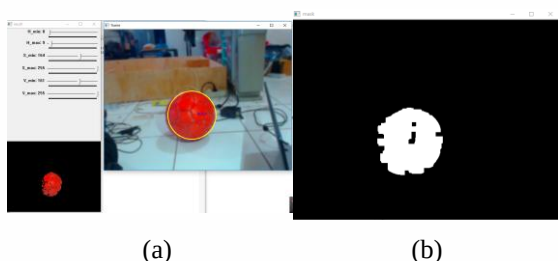
Gambar 6. Flowchart proses pembacaan gambar

Penjelasan tiap bagian Gambar 6 sebagai berikut:

1. *Mengambil gambar*
Tahapan pertama yaitu pengambilan gambar melalui kamera dan mengirim data tersebut ke laptop.
2. *Konversi ruang warna RGB ke HSV*
Mula - mula gambar yang diambil melalui Kamera ruang warnanya *RGB* perlu dikonversi ke ruang warna *HSV*.
3. *Color filtering*
Proses ini dilakukan untuk melakukan pengambilan piksel pada suatu *image* yang memiliki nilai intensitas warna tertentu.
4. *Thresholding*
Mengubah citra berderajat keabuan menjadi citra biner atau hitam putih sehingga dapat diketahui daerah mana yang termasuk obyek dan background dari citra secara jelas[2].

III. HASIL

Berikut adalah hasil visual dari proses tersebut.



Gambar 7. Konversi warna (a) dan *thresholding* (b)

Hasil citra warna yang ditangkap oleh kamera dipengaruhi oleh cahaya yang ada di lingkungannya.

Berikut data hasil deteksi berdasarkan intensitas cahaya tertentu:

Tabel 2. Hasil pengujian deteksi bola pada cahaya redup

(80 lux s/d 100 lux)

No	H		S		V		Jarak	Keterangan
	Min	Max	Min	Max	Min	Max		
1	0	9	74	255	84	255	1 m	Terdeteksi
2							3 m	Terdeteksi
3							5 m	Terdeteksi
4							7 m	Terdeteksi
5							9 m	Terdeteksi
6							12 m	Tidak Terdeteksi

Tabel 3. Hasil pengujian deteksi bola pada cahaya sedang

(500 lux s /d 700 lux)

No	H		S		V		Jarak	Keterangan
	Min	Max	Min	Max	Min	Max		
1	0	9	74	255	84	255	1 m	Terdeteksi
2							3 m	Terdeteksi
3							5 m	Terdeteksi
4							7 m	Terdeteksi
5							9 m	Tidak Terdeteksi
6							12 m	Tidak Terdeteksi

Tabel 4. Hasil pengujian deteksi bola pada cahaya terang

(1000 lux s/d 1500 lux)

No	H		S		V		Jarak	Keterangan
	Min	Max	Min	Max	Min	Max		
1	0	9	74	255	84	255	1 m	Terdeteksi
2							3 m	Terdeteksi
3							5 m	Terdeteksi
4							7 m	Terdeteksi
5							9 m	Tidak Terdeteksi
6							12 m	Tidak Terdeteksi

Dengan nilai *HSV* di atas, dilakukan tiga pengujian pencahayaan dengan intensitas yang berbeda. Pada intensitas cahaya redup (80 lux s/d 100 lux), jarak maksimal untuk mendeteksi bola adalah 9 meter. Ketika pengujian cahaya sedang (500 lux s/d 700 lux) dan cahaya terang (1000 lux s/d 1500 lux) hanya mencapai 7 meter. Hal ini menunjukkan cahaya pada lingkungan mempengaruhi nilai dari *Value*.

IV. KESIMPULAN

Penerapan konsep warna *HSV* sebagai *tracking* warna bola dapat bekerja cukup baik. Namun pada jarak tertentu yang cukup jauh, objek yang dideteksi akan perlahan menghilang. Ini dikarenakan objek tersebut akan semakin mengecil dan kamera sulit untuk melakukan identifikasi. Selain dari pengaruh jarak, konsep *HSV* terpengaruh oleh berkas cahaya di sekitarnya. Semakin kurang pencahayaan, warna yang terlihat semakin hitam atau gelap.

V. DAFTAR PUSTAKA

- [1] Priawadi,Ozi.<https://www.priawadi.com/2012/09/opencv.html>. OpenCV.
- [2] Laksono Budi, Prianggodo, “Perancangan Object Tracking Robot Berbasis Image Processing Menggunakan Raspberry PI,” S1 Teknik Elektro Fakultas Teknik, Universitas Muhammadiyah Surakarta, 2016.
- [3] Prosiding ISRSC 2017
- [4] <http://purmaiyasadeopy.blogspot.com/2016/11/pengertian-web-camera-kamera-web-secara.html>, Pengertian Web Camera (Kamera Web) Secara Spesifik”
- [5] <http://www.charisfauzan.net/2015/01/ruang-warna-hue-saturation-value-hsv.html>, Abdi Charis Fauzan, “Ruang Warna Hue Saturation Value (HSV) serta Proses Konversinya”.
- [6] Budiharto, Widodo et al. 2012, ”Robot Vision – Teknik membangun robot cerdas masa depan”.

Strategi Pergerakan Robot Sepak Bola Beroda Menggunakan Komunikasi TCP/IP

Rasidin, Graitto Usodo, Azzam Rifki Fardiandanu, Samsudin, Britantyo Wicaksono
Program Studi Elektronika Industri, Jurusan Teknik Elektro, Politeknik Negeri Jakarta
Jl. Prof. DR. G. A. Siwabessy, Kampus UI, Depok 16425, Telp/Fax: 021 7863531
E-mail: rasidinrasidin44@gmail.com,

Abstrak -- Permasalahan komunikasi antar robot pada Kontes Robot Sepak Bola Indonesia Beroda sangat kompleks, sehingga penting untuk diteliti. Khususnya yang dipengaruhi oleh *mode wifi* yang digunakan. Informasi robot mendapatkan bola dapat mengantisipasi bertabrakan dengan robot lain dalam satu tim. Penelitian tentang strategi pergerakan menggunakan komunikasi TCP/IP belum banyak dilaksanakan. Oleh karena itu, penting merancang strategi pergerakan robot dengan memanfaatkan komunikasi protokol TCP/IP termonitor pada *base station*. Interkoneksi perangkat (Raspberry Pi dan NodeMCU) dengan *base station*, sebagai komunikasi *Machine to Machine* (M2M) dan jaringan *internet* sebagai media pertukaran data. Pertukaran data berupa karakter (*string*). Pertukaran karakter antar robot harus melalui *base station*. Perancangan *base station* dengan software *Microsoft Visual Studio 2017 community edition* menggunakan bahasa pemrograman C#. Pengujian dilakukan pada dua robot dan *base station* yang terkoneksi dalam satu jaringan *internet*. Hasil dari pengujian didapat strategi pergerakan robot yaitu ketika robot mendapatkan bola, secara otomatis mengirim karakter "a" ke robot lainnya melalui *base station* untuk menginstruksikan bergerak mundur ke depan gawang. Kecepatan keseluruhan transmisi data ± 84 ms dan tertampil pada aplikasi *Wireshark*.

Kata kunci: *Base station*, Pemrograman C#, TCP/IP

I. PENDAHULUAN

Penggunaan strategi komunikasi penting untuk diteliti. Robot mentransmisikan data karakter (*string*) ke robot lain melalui *base station*. Permasalahannya sangat kompleks, hingga saat ini penelitiannya berkembang dan berkelanjutan. Pentransmisi data dipengaruhi oleh *mode wifi* yang digunakan. Kecepatan transmisi data dapat diukur dengan aplikasi *Wireshark*. Variabel-variabel pengukuran seperti; waktu kirim/terima data dalam satuan detik. Keseluruhan variabel berkaitan dengan kecepatan komunikasi antara robot dengan *base station*. Transmisi data *string* untuk komunikasi antar robot, mengantisipasi robot bertabrakan dalam satu tim. Saat ini, penggunaan komunikasi dengan *base station* belum banyak direalisasikan. Simulasi pemodelan transmisi data telah banyak diimplementasikan [1].

Kesesuaian data karakter dengan instruksi sesuai peraturan *RoboCup* merupakan celah penelitian. Klarifikasi kecepatan transmisi data menggunakan aplikasi *Wireshark* belum banyak dilaksanakan. Variabel waktu kirim/terima data ditampilkan di

Personal Computer (PC) yang terinstall aplikasi *Wireshark*. Oleh karena itu, penting mendata besaran hasil kecepatan kirim/terima data termonitor di PC. Transmisi data menggunakan mikrokontroler Raspberry Pi, telah diteliti dan diimplementasikan [2]. Interkoneksi modul mikrokontroler ke jaringan *router* memungkinkan data dapat ditransmisikan ke modul NodeMCU melalui *base station*. Mikrokontroler, *base station* dan modul NodeMCU, terkoneksi dalam satu jaringan *internet* [3].

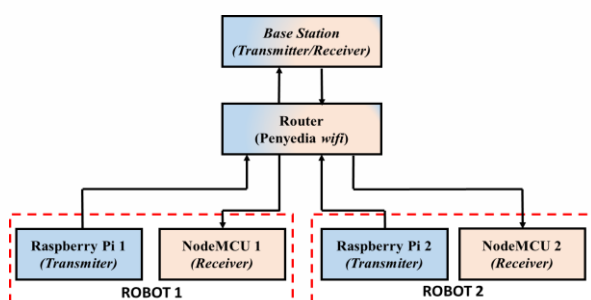
Tujuan penelitian untuk merancang strategi pergerakan robot dengan memanfaatkan komunikasi TCP/IP. Kecepatan keseluruhan transmisi ± 84 ms, menggunakan jaringan *mode A*, IEEE802.11a dengan frekuensi 5,8 GHz tertampil pada aplikasi *Wireshark*.

II. METODE

Penelitian dilaksanakan dengan beberapa metode agar target sistem terealisasi sesuai rencana, yaitu:

A. Perancangan sistem

Ketelitian dan perencanaan yang matang penting pada proses perancangan model strategi komunikasi termonitor *base station*. Perancangan alat mengacu pada blok diagram sistem (Gambar 1).



Gambar 1. Blok Diagram Sistem

Keterangan **Gambar 1:**

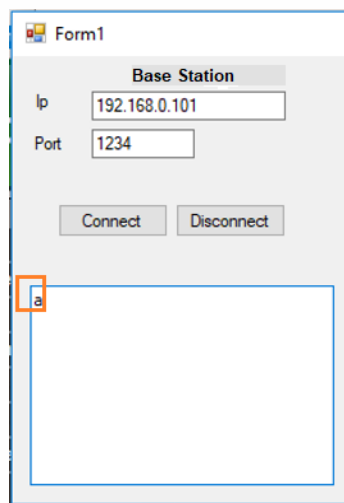
- Raspberry Pi 1 : Raspberry Pi 1 sebagai pen-transmitter data ke NodeMCU 2 melalui *base station*
- NodeMCU 1 : NodeMCU 1 sebagai penerima data dari Raspberry 2 melalui *base station*
- Raspberry Pi 2 : Raspberry Pi 2 sebagai pen-transmitter data ke NodeMCU 1 melalui *base station*

- NodeMCU 2 : NodeMCU 2 sebagai penerima data dari Raspberry 1 melalui *base station*
- Router : Router sebagai penyedia *wifi*. Raspberry Pi, *base station* dan NodeMCU, terkoneksi dalam satu jaringan *internet*
- Base Station : *Base station* sebagai media penghubung antara robot 1 dan robot 2 (sesuai *rule KRSBI*)

Proses transmisi data tertampil pada *base station* dirancang menggunakan *software Ms. Visual Studio 2017 community edition* dengan bahasa pemrograman C#. Data *string* “a” ditransmisikan dari Raspberry Pi 1 ke NodeMCU 2 ketika robot 1 mendapat bola dan jika robot 2 yang mendapat bola maka data dikirim dari Raspberry Pi 2 ke NodeMCU 1. Seluruh pertukaran data dilakukan melalui *base station* memanfaatkan komunikasi TCP/IP.

B. Tampilan Terima Data pada Base Station

Base station (Gambar 2) digunakan sebagai media penghubung antar robot dan pemonitor hasil transmisi data *string* dalam bentuk karakter. Karakter “a” yang tampil pada *base station* menjadi parameter dalam keberhasilan pentransmisian data.



Gambar 2. Base station Penampil data *string*

Keterangan Gambar 2:

- IP : IP sebagai pengalamatan NodeMCU
- Port : Port sebagai jalur koneksi NodeMCU
- Connect : *Connect* menunjukkan bahwa *base station* terkoneksi dengan robot
- Disconnect : *Disconnect* menunjukkan bahwa *base station* tidak terkoneksi dengan robot
- Text Box : *Text Box* sebagai penampil hasil data *string* yang telah ditransmisikan

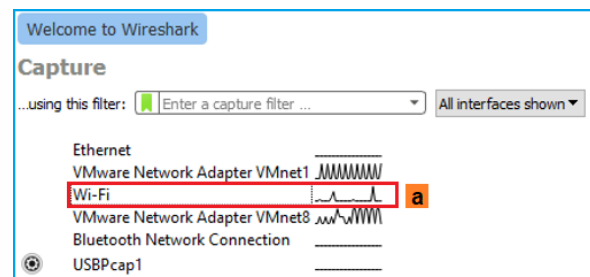
C. Realisasi Sistem

Realisasi sistem dengan beberapa tahapan yaitu:

- (1) Menginstalasi Alat/Sistem
Peletakan *hardware*, men-*setting* nomor *port* NodeMCU pada kedua robot.
- (2) Membuat Program
Pembuatan program *base station* sebagai media penghubung komunikasi antar robot dengan *software Microsoft Visual Studio 2017* menggunakan bahasa pemrograman C#.
- (3) Menampilkan Data *String*
Data *string* yang ditransmisikan dari Raspberry Pi akan diterima oleh modul NodeMCU dan ditampilkan pada *base station* sebagai monitor keberhasilan.
- (4) Menguji Kecepatan Transmisi Data
Transmisi data dari Raspberry Pi ke *base station* ditampilkan kecepatannya melalui aplikasi *wireShark* yang terinstall pada PC.

D. Tampilan Halaman Utama Wireshark

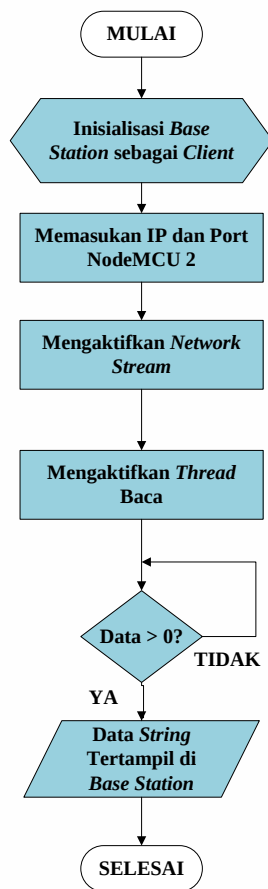
Penggunaan aplikasi *Wireshark* diperlukan untuk mengukur kecepatan transmisi data dari Raspberry Pi ke NodeMCU melalui *base station*. Halaman utama aplikasi *Wireshark* (Gambar 3) terdapat beberapa pilihan jaringan yang ingin dimonitor seperti; *Ethernet*, *VMware Network Adapter VMnet1*, *Wifi*, *Bluetooth Network Connection*, dan *USBcap1*.



Gambar 3. Tampilan Halaman Utama Wireshark

Pemilihan jaringan *wifi* (Gambar 3.a) dilakukan karena komunikasi yang diizinkan sesuai peraturan kontes KRSBI Beroda 2018 adalah komunikasi TCP/IP yang menggunakan jaringan *wifi*. Mode *wifi* yang digunakan adalah *mode A*.

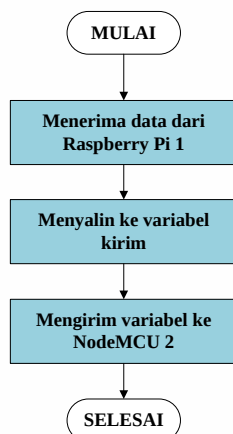
E. Flowchart Terima Data dari Raspberry Pi 1



Gambar 4. Flowchart terima data dari Raspberry Pi 1

Diagram alir terima data (Gambar 4) menjadi acuan untuk merealisasikan cara kerja sistem terima data dari Raspberry Pi 1. *Base station* diinisialisasikan sebagai *client*. IP (*Internet Protocol*) dan *Port* NodeMCU 2 dimasukkan nomor IP dan *port*-nya sesuai dengan yang telah di *setting* sebelumnya.

F. Flowchart Kirim Data ke NodeMCU 2



Gambar 5. Flowchart kirim data ke NodeMCU 2 Setelah *base station* menerima data dari Raspberry Pi

1, data *string* tersebut ditransmisikan menggunakan komunikasi TCP/IP ke NodeMCU 2. Karakter “a” menginformasikan bahwa robot telah mendapatkan bola dan menginstruksikan robot lain dalam satu tim untuk mundur menuju ke arah gawang.

III. HASIL DAN DISKUSI

Pengujian untuk mengetahui robot dan sistem dapat berfungsi sesuai yang diharapkan. Pemeriksaan terhadap kerja modul di tiap subsistem penting, interkoneksi keseluruhan *hardware* dan *software* dapat mengalami *error*.

A. Langkah Kerja

- (1) Aktifkan *router* sebagai penyedia jaringan *wifi*.
- (2) Koneksikan laptop dan kedua robot dengan jaringan *wifi* yang ada pada *router*.
- (3) Buka aplikasi *base station* pada Laptop.
- (4) Masukkan nomor IP dan Port robot 1 pada *base station*, kemudian klik tombol *connect*.
- (5) Tunggu hingga muncul tampilan notifikasi robot telah terkoneksi.
- (6) Lakukan pengujian pengiriman data dari Raspberry Pi 1 (robot 1) ke *base station*.
- (7) Ulangi langkah ke (4) sampai (6) untuk Raspberry Pi 2 (robot 2).
- (8) Operasikan kedua robot, kemudian berikan bola pada salah satu robot.
- (9) Perhatikan apakah robot yang tidak menerima bola kembali ke depan gawang.
- (10) Jika tidak sesuai, maka ulangi pengujian dari langkah (1).

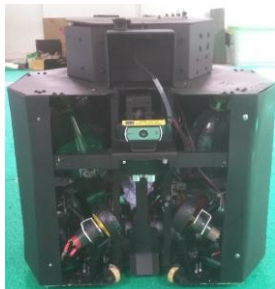
B. Daftar Alat dan Bahan

Tabel 1. Daftar Alat dan Bahan

No	Jenis	Fungsi
1	Power Supply	Sumber tegangan Modul Raspberry Pi dan NodeMCU
2	Modul Raspberry Pi	Pengirim data <i>string</i> ke modul NodeMCU melalui <i>base station</i>
3	Modul NodeMCU	Penerima data <i>string</i> dari Raspberry Pi melalui <i>base station</i>
4	Router	Penyedia jaringan <i>wifi</i> . untuk Raspberry Pi, <i>base station</i> dan NodeMCU, terkoneksi dalam satu jaringan <i>internet</i>
5	Laptop	Penampil data <i>string</i> pada <i>Base station</i> sebagai pemonitor komunikasi robot
6	Bola	Kebutuhan bahan penunjang untuk pengujian komunikasi

Gambar 6 (tampak depan) dan Gambar 7 (tampak atas) mendeskripsikan robot sepak bola beroda. Pada percobaan peneliti sebelumnya menggunakan

bluetooth sebagai media komunikasi data secara *nirkabel*. Pengujian dilakukan dengan dua robot yang dijalankan bersamaan. Keberhasilan diukur dengan nyala LED (*Light Emmiting Diode*) ketika salah satu robot telah mendapat bola dan LED mati setelah posisi bola jauh atau bola telah ditendang [4]. TCP/IP (*Transmission Control Protocol/Internet Protocol*) merupakan standar komunikasi data yang digunakan oleh komunitas *internet* dalam proses tukar-menukar data dari satu komputer ke komputer lain didalam suatu jaringan [5]. Dua robot dan *base station* pada laptop dikoneksikan dalam satu jaringan *internet* menggunakan *router*. Penggunaan jaringan *router* memanfaatkan protokol komunikasi TCP/IP [6] dalam melakukan pertukaran (kirim/terima) data.



Gambar 6. Robot Sepak Bola (Tampak Depan)



Gambar 7. Robot Sepak Bola (Tampak Atas)

Sistem komunikasi diuji dengan kedua robot. Variabel yang diukur adalah kecepatan kirim/terima data dalam satuan mili detik (ms). Pada percobaan peneliti sebelumnya menggunakan komunikasi *point to point*, sehingga pengirimannya dilakukan secara bergantian antara modul *slave 1* dan *slave 2* yang terdapat pada robot. *Oscilloscope* sebagai pengukur waktu proses perpindahan komunikasi dalam bentuk sinyal gelombang kotak [7]. Pengujian dilakukan untuk mengetahui lamanya proses pertukaran data. Hasil kecepatan kirim atau terima data ditampilkan di aplikasi *Wireshark* (Gambar 8). *Wireshark* merupakan salah satu aplikasi *open source* yang digunakan sebagai alat analisis sebuah jaringan, baik *ethernet* maupun *non-ethernet*. *Wireshark* sebagai *network analyzer* mampu melakukan pengamatan berupa IP address, protokol jaringan, jumlah *frame*, *filtering* dan *sniffing* [8]. Penggunaan aplikasi *Wireshark* sebelumnya telah diteliti sebagai *tools* pengukuran

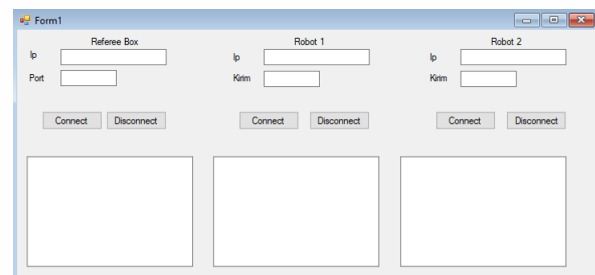
delay pengiriman sebuah paket data [9]. *Tools* ini seringkali digunakan untuk menemukan masalah-masalah pada jaringan, pengembangan perangkat lunak (*software*) dan protokol komunikasi.

No.	Time	Source	Destination	Protocol	Length	Info
163	15.972031	192.168.0.102	255.255.255.255	GVC	50	> DISCOVERY_CMD
164	16.300670	192.168.0.101	192.168.0.100	TCP	60	1234 → 4583 [PSH, ACK]
165	16.342657	192.168.0.100	192.168.0.101	TCP	54	4583 → 1234 [ACK] Seq=1
166	17.444361	192.168.0.100	255.255.255.255	GVC	50	> DISCOVERY_CMD
167	17.618696	192.168.0.1	255.255.255.255	UDP	215	39809 → 7437 Len=173
168	18.020686	192.168.0.102	255.255.255.255	GVC	50	> DISCOVERY_CMD

Gambar 8. Tampilan kecepatan transmisi data

C# adalah bahasa berorientasi objek yang memiliki kode visual menyerupai bahasa pemrograman C++ dan Java [10]. Bahasa pemrograman C# merupakan salah satu bahasa pemrograman tingkat menengah yang sederhana, aman, dan mudah dalam membuat program. Sebuah *file C#* menggunakan *file* berekstensi (.cs).

Cara mentransmisikan data oleh modul *bluetooth* menggunakan antarmuka C# telah dipelajari sebelumnya [11]. Proses transmisi data antar robot tertampil di *base station* dengan TCP/IP menggunakan antarmuka C# (Gambar 9).



Gambar 9. Tampilan *base station* menggunakan antarmuka C#

C. Data Hasil Pengujian

Time	Source	Destination	Protocol
163 15.972031	192.168.0.102	255.255.255.255	GVC
164 16.300670	192.168.0.101	192.168.0.100	TCP
165 16.342657	192.168.0.100	192.168.0.101	TCP
166 17.444361	192.168.0.100	255.255.255.255	GVC
167 17.618696	192.168.0.1	255.255.255.255	UDP
168 18.020686	192.168.0.102	255.255.255.255	GVC

Gambar 10. Data kecepatan transmisi data

Tabel 2. Data hasil pengukuran kecepatan kirim/terima data

No	Waktu Kirim (Detik Ke-)	Waktu Terima (Detik Ke-)	Selisih Waktu (Detik)
1	145.926313	145.966725	0.040412
2	5.700043	5.741562	0.041519
3	12.142015	12.182681	0.040666
4	4.502639	4.542896	0.040257
5	4.721452	4.765955	0.044503
6	5.887818	5.928608	0.04079
7	4.259899	4.301525	0.041626
8	6.307603	6.347965	0.040362
9	6.417965	6.458545	0.04058
10	2.98345	3.024565	0.041115
11	4.828154	4.868339	0.040185
12	4.051362	4.092184	0.040822
13	9.66451	9.704519	0.040009
14	5.408292	5.44835	0.040058
15	5.379143	5.419257	0.040114
16	3.269531	3.310747	0.041216
17	4.852486	4.893492	0.041006
18	5.437264	5.477735	0.040471
19	4.545096	4.585389	0.040293
20	6.886772	6.927313	0.040541
Rata-rata			0.04082725

D. Analisis dan Pembahasan

Tampilan *visual* dan variabel-variabel hasil pengukuran kecepatan transmisi data dari Raspberry Pi, NodeMCU, dan *base station*. Aplikasi *Wireshark* dapat mensimulasikan kecepatan transmisi data dari Raspberry Pi ke *base station* dan sebaliknya. Desain *Wireshark* dipilih untuk memonitor jaringan *wifi* yang terkoneksi pada robot (Raspberry Pi dan NodeMCU) dan *base station*. Jaringan komunikasi yang terdeteksi adalah protokol TCP/IP sesuai (Gambar 10.a). Data hasil pengukuran (Gambar 10.b) menunjukkan bahwa data dikirimkan Raspberry Pi 1 pada detik ke 16,300670 dan (Gambar 10.c) menunjukkan bahwa data diterima *base station* pada detik ke 16,342657. Selisih dari waktu kirim dan waktu terima merupakan kecepatan transmisi data yaitu 0,041987 detik atau 42 mili detik. Setelah itu, waktu kirim/terima data dari *base station* ke NodeMCU 2 adalah sama yaitu 42 mili detik, sehingga waktu transmisi data keseluruhan adalah 84 mili detik. Informasi dari Raspberry Pi 1 (Robot 1) tersampaikan ke NodeMCU 2 (Robot 2) mengantisipasi bertabrakan dengan robot satu tim.

Uji coba data mendokumentasikan keseluruhan data yang tertampil pada aplikasi *Wireshark*. Data yang tertampil adalah waktu kirim (detik ke-), dan waktu terima (detik ke-). Selisih waktu (detik) kirim atau terima data menjadi besarnya kecepatan

transmisi. Kecepatan transmisi dalam pertukaran data antara kedua robot dengan *base station* dipengaruhi oleh *mode wifi* yang digunakan.

IV. KESIMPULAN

Berdasarkan pengujian yang telah dilakukan, dapat disimpulkan bahwa ketika robot mendapatkan bola, secara otomatis mengirim karakter “a” ke robot lainnya melalui *base station* untuk menginstruksikan bergerak mundur menuju depan gawang. Pengiriman data *string* cepat menggunakan Raspberry Pi dan penerimaan datanya stabil menggunakan NodeMCU. Kecepatan keseluruhan transmisi data ± 84 ms dan tertampil pada aplikasi *Wireshark*.

V. UCAPAN TERIMA KASIH

Terimakasih kami sampaikan kepada Pembantu Direktur Bidang Kemahasiswaan Politeknik Negeri Jakarta, Ketua Jurusan Teknik Elektro PNJ, Ketua Jurusan Teknik Mesin PNJ, Ketua Jurusan Teknik Informatika PNJ, Ketua Tim Robotika PNJ, dan para dosen pembimbing di Tim Robotika PNJ serta kakak-kakak alumni Tim Verocia PNJ yang telah memberikan dukungan dan arahan selama ini terhadap Tim Robotika PNJ.

VI. DAFTAR PUSTAKA

- [1] Hermawan, G. Implementasi Algoritma Particle Swarm Optimization Untuk Penentuan Posisi Strategis Agent Pada Simulasi Robot Sepak Bola Dua Dimensi. *Jurnal KOMPUTA*. 2012 October; 5 (2): 63-70.
- [2] Purwanti, B. S. R. Wicaksono, B. Listiani, A. Hardian, B. Sistem Monitor Dan Denda Pelanggaran Batas Kecepatan Kendaraan Umum Termonitor Ke Website. *SenasPro*. 2017.
- [3] Adipranata, R. Implementasi Protokol TCP/IP Untuk Pengendalian Komputer Jarak Jauh. *Jurnal Informatika*. 2002 May; 3 (1): 34-41.
- [4] Santoso, D. Susilo, D. Chandra, B. W. Optimalisasi dan Perancangan Algoritma Pergerakan dan Komunikasi Pada Robot Penyerang Hummanoid Soccer. *Techne Jurnal Ilmiah Elektroteknika*. 2015 October; 14 (2): 113-125.
- [5] Masinambow, V. Najooan, M. E. I. Lumenta, A. S. M. Pengendali Saklar Listrik Melalui Ponsel Pintar Android. *E-Jurnal Teknik Elektro dan Komputer*. 2014; 2301-8402: 1-9.
- [6] Hidayati, N. Suwardi. Analisis Kinerja TCP/IP Untuk Jaringan Nirkabel Bergerak 3G Di Surabaya. *Jurnal Teknik ITS*. 2016; 5 (2): 941-946
- [7] Wibisono, H. Purwanto, Y. Prasetyanto, W. A. Rancang Bangun Sistem Komunikasi Data Game Controller Menggunakan Bluetooth Pada Robot Humanoid Soccer. *ISRSC*. 2015.
- [8] Azhar, R. Hariyadi, I. P. Simulator GNS3 dan Wireshark Sebagai Model Virtual Pembelajaran Praktikum Jaringan Komputer. *Jurnal Kependidikan*. 2015; 16 (3): 285-293.
- [9] Wulandari, R. Analisis QoS (*Quality of Service*) Pada Jaringan Internet. *JUTISI*. 2016 August 2; 2 (2): 162-172.
- [10] Wibowo, A. G. Wicaksono, A. P. Rancang Bangun Aplikasi Untuk Menentukan Jalur Terpendek Rumah sakit Dengan Metode Algoritma Dijkstra. *JUITA*. 2012 May; 2 (1): 21-35.
- [11] Engin, S. Aksoz, A. Dursun, M. The Design of Humanoid Robot Using C# Interface on Bluetooth Communication. *Journal of Automation Engineering*. 2016 June; 4(3):229-232.

Metode Pendeteksi Bola dan Gawang pada Robot Sepak Bola Beroda

Hidayat Putra¹, Rahmat Fanshuri², Salmi Surva Radar³.

^{1,3}Program Studi Elektronika, Jurusan Teknik Elektro, Politeknik Negeri Padang

²Program Studi Telekomunikasi, Jurusan Teknik Elektro, Politeknik Negeri Padang

Kampus Limau Manis, Jl. Kampus Limau Manis, Limau Manis, Pauh, Kota Padang, Sumatera Barat 25164

¹pm.hidayat88@gmail.com

Abstrak – KRSBI beroda merupakan kontes robot sepakbola beroda yang diadakan secara teratur setiap tahun oleh Direktorat Jenderal Pendidikan Tinggi. Dalam robot sepak bola beroda hal yang tidak bisa dilepaskan darinya yaitu gawang dan bola. Oleh karena itu, makalah ini akan membahas tentang salah satu metode yang digunakan untuk mendeteksi bola dan gawang. Pada saat pertandingan dimulai robot secara otomatis akan menjalankan program pengolahan citra dengan menggunakan *threshold* untuk warna bola. Ketika robot mendapatkan bola yang dideteksi dengan menggunakan sensor jarak yaitu *sharp gp*, mikrokontroler akan mengirim perintah ke pc bahwa bola telah didapat sehingga pc akan mengganti *threshold* warna yang digunakan menjadi *threshold* warna gawang. Dengan demikian robot dapat mendeteksi gawang setelah robot mendapatkan bola, sehingga pengolahan citranya tidak membebani kerja pc dikarenakan pengolahan warna hanya terjadi satu warna saja.

Kata kunci: robot sepak bola, deteksi bola, deteksi gawang

I. PENDAHULUAN

Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda merupakan salah satu kompetisi untuk mahasiswa yang diadakan oleh Ristekdikti. KRSBI Beroda diadakan untuk meningkatkan keilmuan dan kreativitas mahasiswa di bidang robotika dan kecerdasan buatan. Dalam kontes ini, tim yang terdiri dari 5 buah robot *full autonomous* beroda bertanding melawan tim lain. Selama pertandingan berlangsung, tidak diperbolehkan adanya campur tangan manusia [1].

Sebagai robot pemain sepak bola, robot harus dapat mendeteksi bola dan gawang agar dapat memenangkan pertandingan. Jika tidak dapat mengenal bola dan gawang, robot tidak dapat mencapai tujuannya di dalam pertandingan.

Untuk mendeteksi bola digunakan pengolahan citra yang didapat dari pembacaan kamera. Fungsi kamera pada robot yakni untuk mengetahui posisi bola maupun gawang. Dalam pengolahan citra bola dan gawang dibaca tidaklah bersamaan, melainkan secara bergantian sesuai tujuan robot itu sendiri yakni memasukkan bola ke dalam gawang. Di sini *threshold* warna pengolahan citra yang digunakan yaitu warna bola. Jika bola telah dikenali oleh robot, maka tujuan

robot selanjutnya yaitu mencetak gol ke gawang sehingga pengolahan citra yang diperlukan yaitu *threshold* warna gawang. Oleh karena itu dalam makalah kali ini akan dibahas mengenai metode mengidentifikasi bola dan gawang secara bergantian.

II. METODE

A. Sistem yang Digunakan

Robot yang digunakan pada penelitian ini yaitu robot KRSBI beroda milik tim Datuak Midun (Gambar 1). *Software* dan *hardware* yang digunakan sebagai berikut:



Gambar 1. Robot Datuak Midun PNP

1. Python 2.7

Python adalah bahasa pemrograman yang bersifat *open source*. Bahasa pemrograman ini dioptimalisasi untuk *quality software*, *productivity developer*, *portability program*, dan *component integration*. Python telah digunakan untuk mengembangkan berbagai macam perangkat lunak, seperti *internet scripting*, *systems programming*, *user interfaces*, *product customization*, *numeric programming*. Python saat ini telah menduduki posisi

4 atau 5 bahasa pemrograman paling sering digunakan diseluruh dunia.

2. Laptop ASUS e203nha

Laptop yang digunakan yaitu hxsj 720 pixel. Spesifikasi komputer yang digunakan untuk mengolah citra adalah sebagai berikut:

- a. Processor: Intel Celeron N3350
 - b. RAM : 2GB DDR3
 - c. HDD: 500GB
 - d. OS: Ubuntu 16.04
3. Arduino MEGA 2560

Arduino mega 2560 adalah piranti mikrokontroler menggunakan ATmega2560. Modul ini memiliki 54 digital *input* atau *output*, di mana 14 pin digunakan untuk PWM output dan 16 pin digunakan sebagai analog input, 4 pin untuk UART, 16 MHz osilator kristal, koneksi USB, power jack ICSP header, dan tombol reset. Modul ini memiliki segala yang dibutuhkan untuk memprogram mikrokontroler seperti kabel USB dan catu daya melalui adaptor atau baterai.

Semua ini diberikan untuk mendukung pemakaian mikrokontroler Arduino, hanya terhubung ke komputer dengan kabel USB atau listrik dengan adaptor dari AC ke DC atau baterai untuk memulai pemakaian. Arduino Mega kompatibel dengan shield yang dirancang untuk Arduino Duemilanove, Decimila maupun UNO.



Gambar 2. Arduino MEGA 2560

4. Kompas hmc58831

Sensor kompas GY-271 adalah sebuah modul sensor kompas yang siap pakai, modul kompas ini didesain untuk dapat mendeteksi medan magnet yang rendah dengan menggunakan *interface* digital dan memberikan informasi yang tepat. Sensor ini mengubah setiap medan magnet kedalam tegangan yang berbeda-beda pada 3 *axis*. Nilai tegangan ini akan berubah menjadi nilai digital, yang dapat digunakan untuk mengkalkulasikan medan magnet dari setiap arah yang berbeda [5].



Gambar 3. Sensor kompas

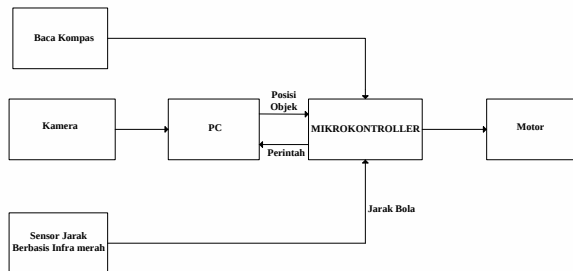
B. Robot Vision

Perancangan pengolahan citra pada mobile robot tidak bisa dilepaskan dari perangkat sensor dan kecerdasan buatan [2]. Oleh karena itu sistem vision merupakan salah satu bagian utama dalam mobile robot. Hal utama dari robot vision adalah kemampuan untuk merasakan setiap perubahan-perubahan pada lingkungannya. Rangsangan tersebut kemudian diolah oleh sistem robot tersebut sehingga dapat merespon kondisi atau keadaan yang terjadi. Sistem vision pada *mobile* robot umumnya tidak hanya satu melainkan bisa lebih. Sehingga diperlukan penelitian lanjutan mengenai stereo vision supaya robot dapat menangkap objek secara tiga dimensi. Tujuan utama robot vision adalah proses pengenalan objek yang ditangkap oleh kamera. Proses pengenalan objek pada *mobile* robot adalah proses pemetaan lingkungan, sehingga robot dapat mengenali lintasan yang akan dilalui dan dapat menghindari penghalang-penghalang yang ada dihadapannya. Proses pengenalan citra pada *mobile* robot adalah sebagai berikut [3]:

- a. *Penangkapan Citra* merupakan tahapan yang pertama dan paling penting di dalam robot vision. Apabila pada saat pengambilan citra terdapat kesalahan maka dapat mengakibatkan kegagalan dari seluruh proses yang ada. Pengambilan citra yang dilakukan oleh kamera merupakan peranan utama untuk pengambilan citra objek dengan kecukupan kontras dan fokus kamera tersebut.
- b. *Pemroses Awal* dilakukan setelah citra berhasil didapatkan. Pada tahapan ini noise yang didapatkan pada citra akan dihilangkan.
- c. *Penonjolan Citra*. Tahapan ini yakni citra diproses untuk memaksimalkan kontras dan mendapatkan output yang optimal untuk tahap berikutnya.
- d. *Pendeteksian Tepi dan Deteksi Warna*. Tahapan ini merupakan elemen yang paling penting dalam robot vision dan pengolahan citra. Bagian penting dalam tahapan ini adalah pengolahan sinyal yang masuk kedalam sistem dan teknik kecerdasan buatan untuk pengolahan citra tersebut.
- e. *Analisis Tepi*. Tahapan ini berupa proses analisis terhadap objek dan warna yang telah didapatkan untuk memperoleh keputusan sebagai bahan instruksi pada proses selanjutnya

C. Rancangan Metode

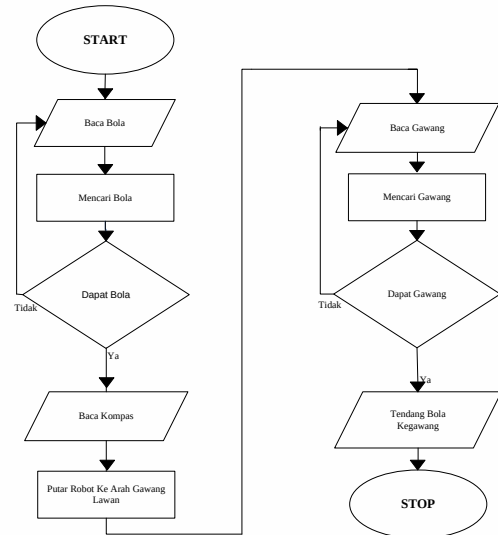
Rancangan metode yang digunakan untuk pendeteksian bola dan gawang dapat dilihat dari blok diagram dan flowchart pada Gambar 4.



Gambar 4. Blok diagram sistem

Gambar kamera sebagai input untuk menentukan posisi baik itu bola maupun gawang. Untuk mendapatkan posisi maka robot vision diperlukan yang mana dalam robot vision melakukan pengolahan citra yang dilakukan oleh PC.

Setelah tahap pengolahan citra maka didapatkan posisi objek yang diinginkan. Posisi tersebut akan dikirimkan ke arduino yang mana akan menjadi controller utama dalam robot krsbi beroda. Pengiriman data dari pc ke arduino maupun sebaliknya melalui komunikasi serial. Data yang diterima arduino dari komunikasi serial akan menjadi input untuk melakukan tindakan yang akan dilakukan robot. Dengan input posisi yang didapat dari pengolahan citra tersebut arduino akan mengontrol motor untuk mengejar objek. Saat robot mendapatkan bola dengan indikasi menggunakan sensor sharp gp2d12 maka arduino akan perintah berupa pergantian *thresholding* pada pengolahan citra yang mulanya *thresholding* yang digunakan untuk mendapatkan warna bola maka akan diganti dengan warna gawang. Lalu proses pengolahan citra akan berlangsung seperti sebelumnya hanya saja filter warna berubah menjadi warna gawang. Lalu robot akan membaca kompas untuk menentukan yang mana daerah lawan. Setelah itu kamera membaca gawang dengan output akan tetap sama yaitu berupa posisi yang akan dikirimkan kembali ke arduino. Sistem pengambilan keputusan robot dapat dilihat pada gambar 5.



Gambar 5. Flowchart sistem

III. HASIL DAN DISKUSI

A. Hasil Percobaan

Berikut hasil percobaan yang telah dilakukan, sesuai dengan tabel 1. Berikut ini;

Tabel 1. Pendeteksian bola sensor sharp gp2d12

Nilai	Nilai Sensor	Keterangan	Ouput kamera
1	≥ 500	Dapat Bola	Kamera membaca gawang
0	<500	Tidak Dapat Bola	Kamera membaca bola

1. Pada saat nilai sensor sharp gp2d12 = 0



Gambar 6. Deteksi bola dikarenakan nilai *threshold* warna yang diambil yaitu warna oranye



Gambar 7. Binary image

Blob warna putih pada Gambar 7 merupakan daerah yang dideteksi dan warna hitam diabaikan, tetapi masih banyak noise yang ada terdeteksi maka dari itu dengan metode seleksi maka diambil *blob* yang paling banyak

2. Pada saat nilai sensor sharp gp2d12 = 1



Gambar 8. Deteksi gawang yang berwarna putih, dikarenakan *threshold* nilai warna telah berganti menjadi *threshold* warna putih.



Gambar 9. Binary image yang dihasilkan saat *threshold* yang diambil merupakan *threshold* warna putih

B. Pembahasan

Berdasarkan hasil percobaan yang dilakukan di atas bahwa metode pendeteksian bola dan gawang berdasarkan pergantian *threshold* warna berhasil dilakukan. Namun metode ini hanya mendeteksi gawang berdasarkan warna putih pada gawang, tidak menutup kemungkinan *noise* yang ada akan mengganggu pendeteksian gawang seperti warna putih di luar maupun di dalam lapangan seperti garis.

Metode ini perlu diuji lagi dan di-*fix*-kan lagi *threshold* warna putih antara gawang, garis lapangan maupun warna putih lainnya.

IV. KESIMPULAN

Metode pendeteksian bola dan gawang berdasarkan pergantian *threshold* warna ini dapat diimplementasikan pada robot sepak bola beroda. Hasil percobaan sesuai dengan harapan walaupun masih ada gangguan yaitu berupa *noise* yang ada. Dengan demikian robot sepak bola beroda dapat melakukan pertandingan dan mencapai tujuannya dilapangan.

V. DAFTAR PUSTAKA

- [1] Heru, Endra Pitowarno, dan Kusprasapta Mutijarsa. "Buku Panduan Kontes Robot Sepak Bola Indonesia Beroda". 2017
- [2] Pratomo Hendrianto Awang,dkk.2009,Model dan Teknik Kalibrasi Citra Untuk Sistem Autonomous Robot.Yogyakarta.
- [3] Prabuwo, A.S., Sulaiman, R., Hamdan, A.R., Hasniaty, A., 2006, Development of Intelligent Visual
- [4] Kadir, A. (2013). Panduan Praktis Mempelajari Aplikasi Mikrokontroler & Pemrogramannya Menggunakan Arduino.Yogyakarta: Penerbit Andi
- [5] Kadir, A. (2015). Buku pintar Pemrograman arduino.Yogyakarta: Penerbit Andi.

Penentuan Posisi Bola berdasarkan Sudut Arah Bola terhadap Robot pada Kamera *Omnidirectional*

Yogo Paranto Aji, Syahid Al Irfan, dan Tofik Nurochman
Teknik Elektro, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
E-mail: yogo1500022012@webmail.uad.ac.id

Abstrak— Kontes robot sepak bola divisi beroda adalah salah satu kontes yang diadakan setiap tahunnya oleh Kemenristekdikti. Robot-robot yang ada harus memiliki kemampuan seperti manusia harus dapat melihat bola dan gawang, dalam hal ini robot harus dapat mengenali bentuk dan warna objek yang ada di lapangan. Kamera omnidirectional atau kamera dengan sudut pandang 360 derajat akan sangat memungkinkan robot untuk tidak memerlukan gerakan-gerakan tambahan karena seluruh lapangan pertandingan yang digunakan akan dapat terlihat secara keseluruhan. Pada penelitian ini untuk mendapatkan posisi bola di lapangan menggunakan sudut arah bola terhadap bola dengan cakupan sudut dari 0 derajat sampai dengan 360 derajat

Kata Kunci—Kamera Omnidirectional; BallDetection;

I. PENDAHULUAN

Robot adalah seperangkat alat mekanik yang bisa melakukan tugas fisik, baik dengan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dahulu. Untuk mengembangkan kecerdasan pada robot ada banyak sekali kontes yang diadakan setiap tahunnya dengan tema-tema yang beraneka ragam. Kontes Robot Sepak Bola Indonesia kategori beroda salah satunya.

Salah satu kekurangan pada kamera konvensional adalah terbatasnya sudut pandang ketika berada di atas lapangan, sehingga memerlukan gerakan mekanik tambahan untuk mendapatkan sudut pandang yang diinginkan. Kamera omnidirectional atau kamera dengan sudut pandang 360 derajat tentu akan sangat memungkinkan robot untuk tidak memerlukan gerakan-gerakan tambahan.

Penggunaan Kamera omnidirectional sangat membantu dalam hal sudut pandang yang diinginkan, karena seluruh lapangan pertandingan yang digunakan akan dapat terlihat secara keseluruhan. Salah satu metode yang

II. DASAR TEORI

Sistem deteksi bola yang digunakan memanfaatkan koordinat x dan y pada bola yang kemudian dikonversikan menjadi sebuah sudut dengan memanfaatkan persamaan garis pada hasil pengolahan citra yang dilakukan, sehingga posisi bola dapat diketahui melalui arah sudut terhadap robot.

A. OpenCV.

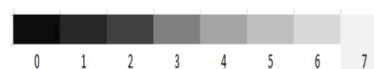
OpenCV adalah suatu library gratis yang dikembangkan oleh developer-developer Intel Corporation. Library ini terdiri dari fungsi-fungsi computer vision dan API (Application Programming Interface) untuk image processing high level maupun low level dan sebagai optimasi aplikasi realtime. OpenCV sangat disarankan untuk programmer yang akan berkecukupan pada bidang computer vision, karena library ini mampu menciptakan aplikasi yang handal, kuat di bidang digital vision, dan mempunyai kemampuan yang mirip dengan cara pengolahan visual pada manusia. Karena library ini bersifat cuma-cuma dan sifatnya yang open source, maka dari itu OpenCV tidak dipesan khusus untuk pengguna arsitektur Intel, tetapi dapat dibangun pada hampir semua arsitektur.

B. Filtering

Bertujuan menanggulangi gangguan pada hasil citra yang kita inginkan, adapun contoh-contoh filtering yaitu *thresholding*, *gaussian blur*, *erode*, *dilasi* dan masih ada lainnya lagi. Berikut ini adalah beberapa filtering yang di terapkan pada robot sepak bola beroda UAD, yaitu :

1. Citra Grayscale

Citra grayscale merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pixelnya, dengan kata lain nilai bagian Red = Green = Blue. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas warnanya. Warna yang dimiliki adalah warna dari hitam, keabuan, dan putih seperti yang ditunjukkan gambar 1 Tingkatan keabuan disini merupakan warna abu dengan berbagai tingkatan dari hitam hingga putih. Citra grayscale memiliki derajat keabuan 8 bit seperti yang dapat dilihat berikut ini :



gambar 1. gradasi warna *grayscale*

2. Thresholding

Proses thresholding atau binerisasi pada prinsipnya adalah melakukan pengubahan nilai derajat keabuan menjadi dua nilai yaitu

0 atau warna hitam dan 255 atau warna putih. Pemilihan nilai threshold yang digunakan berpengaruh terhadap ketajaman suatu citra. Persamaan 1 Secara umum dapat menggambarkan proses threshold citra grayscale untuk menghasilkan citra biner dapat adalah sebagai berikut:

$$g(x,y) = \begin{cases} 0 & \text{jika } f(x,y) \geq T \\ 1 & \text{jika } f(x,y) < T \end{cases}$$

Keterangan:

$g(x,y)$ = citra biner dari citra grayscale

$f(x,y)$

T = nilai threshold

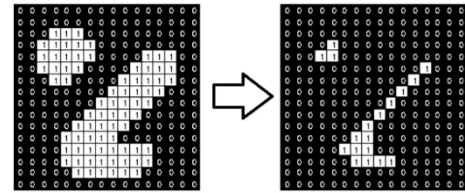
Proses threshold dilakukan dengan memeriksa nilai derajat keabuan pada citra. Jika nilai derajat keabuan kurang dari nilai threshold maka warna piksel berubah menjadi hitam, begitu juga sebaliknya jika piksel lebih dari nilai threshold maka warna piksel akan berubah menjadi putih.

3. Smoothing

Data citra yang telah melalui proses thresholding selalu memberikan beberapa noise, oleh sebab itu dilakukan proses smoothing yang digunakan untuk menghilangkan noise pada citra dengan menggunakan Gaussian filtering untuk mendapatkan efek blur pada citra. Melalui proses thresholding maka akan didapatkan citra biner, namun terdapat noise yang akan mengganggu dalam proses pendeteksian bentuk pada citra. Oleh sebab itu digunakan smoothing citra, yang diharapkan akan mengurangi atau menghilangkan noise[5].

4. Operasi morfologi

Proses erosi merupakan kebalikan dari proses dilasi. Jika dalam proses dilasi menghasilkan objek yang lebih luas maka dalam proses erosi akan menghasilkan objek yang menyempit (mengecil). Lubang pada objek juga akan tampak membesar seiring menyempit-nya batas objek tersebut. Semakin besar kernel yang digunakan maka hasil yang akan didapatkan akan semakin kecil. Begitu jug apabila proses erosi dilakukan berulang-ulang akan terus mengecilkan objek walaupun hanya menggunakan kernel berukuran kecil.



Gambar 2. proses erosi

5. find contours

Pendeteksian bola pada penelitian ini menggunakan metode *findcontours*, selain akurasi pemrosesan citra yang lebih baik pemrosesan menggunakan metode *findcontours* dilakukan dengan menemukan kontur yang telah diatur ambang batasnya. Sehingga akan dapat mendeteksi objek secara akurat, input pada metode *findcontours* merupakan gambar biner.

C. Kamera Omnidirectional

Dalam fotografi, kamera omnidirectional adalah kamera dengan sudut pandang 360 derajat pada bidang horizontal, atau bidang visual yang mencakup seluruh *sphere*. Kamera omnidirectional penting untuk mendapatkan cakupan wilayah bidang visual yang luas, seperti dalam fotografi panorama dan robotika[3]. Pada robot sepak bola beroda UAD menggunakan jenis cermin cembung hiperbola yang ditunjukkan oleh gambar 3.



gambar 3. Kamera *omnidirectional*

III. METODOLOGI PENELITIAN

Penentuan posisi bola berdasarkan sudut arah bola terhadap robot menggunakan metode Gradien persamaan garis lurus ada beberapa proses tahapan yang dilalui.

A. Algoritma penentuan sudut bola

Persamaan garis lurus merupakan persamaan linier yang mengandung satu maupun dua variabel. Pada penelitian ini untuk menentukan besar sudut posisi bola terhadap posisi depan robot digunakan persamaan *trigonometri* sehingga perhitungan persamaan garis lurus menjadi dasar penentuan sudut bola pada penelitian ini. Apabila sebuah titik awal dari sebuah garis yang pada penelitian ini ditentukan pada koordinat

titik tengah pada kamera *omnidirectional* dinyatakan sebagai (x_1, y_1) dan titik akhir yang pada penelitian ini ditentukan pada koordinat titik bola (x_2, y_2) maka nilai gradien m dapat dirumuskan seperti persamaan (2) berikut :

$$m = \frac{y}{x} = \frac{y_2 - y_1}{x_2 - x_1} \quad (2)$$

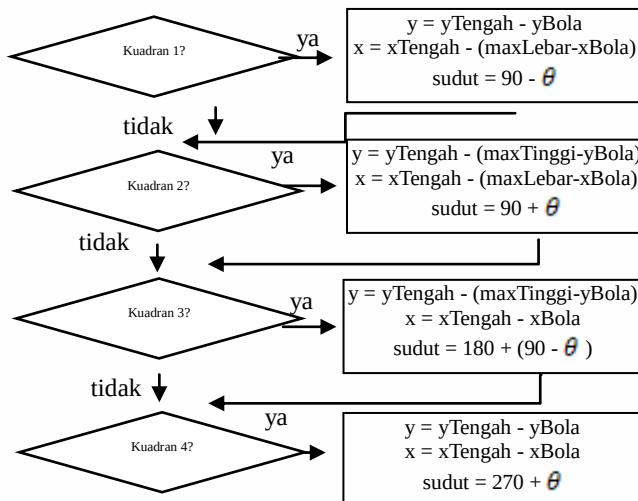
Dari persamaan (2) di atas setelah didapatkan nilai gradien m selanjutnya digunakan rumus persamaan *trigonometri* untuk menentukan sudut yang diinginkan dengan persamaan seperti yang ditunjukkan oleh persamaan (3) berikut :

$$\tan \theta = \left(\frac{y_2 - y_1}{x_2 - x_1} \right) \quad (3.1)$$

sehingga

$$\tan^{-1} m = \theta \quad (3.2)$$

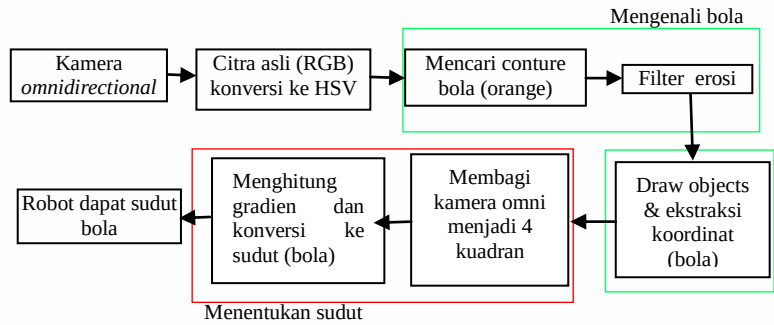
Dari persamaan (3.2) di atas dengan mengkalkulasikan *inverse tan* dari hasil perhitungan gradien m maka didapatkan nilai sudut yang dicari yaitu θ . Walaupun demikian rumus di atas hanya dapat diterapkan hanya pada kuadran 1 sedangkan kuadran lainnya belum bisa langsung diterapkan karena saat daerah kuadrannya berpindah maka perhitungan nilai gradiennya juga akan berubah. Agar dapat diterapkan pada semua gradien pada penelitian ini dibuatkan percabangan untuk menentukan di gradien manakah posisi bola berada. Berikut diagram alur percabangannya yang ditunjukkan oleh gambar 4.



Gambar 4. Diagram alur percabangan penentuan posisi bola

B. Sistem deteksi

Perancangan sistem untuk melakukan proses mendapatkan hasil, dapat digambarkan dalam bentuk blok diagram seperti yang diperlihatkan pada Gambar.5 dan Gambar.6 menunjukkan hasil citra yang dideteksi dan telah mendapatkan posisi arah sudut terhadap robot.



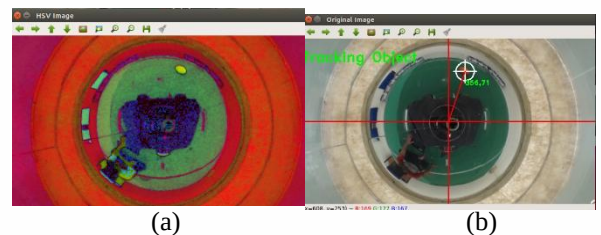
Gambar5. Blok diagram proses penentuan sudut arah bola



Gambar 6. Posisi bola dan besar sudut

C. Konevresi warna RGB ke HSV

Dalam pengolahan citra kamera defaultnya biasanya pertama kalinya akan menangkap citra RGB, yang digunakan untuk dapat mendeteksi objek. Dari citra RGB ini nantinya akan di konversi ke dalam bentuk citra HSV. Gambar 7 menunjukkan model citra HSV dan RGB. Citra RGB lebih susah untuk dibedakan ke kelas-kelas warna karena merupakan campuran dari warna-warna primer [1].



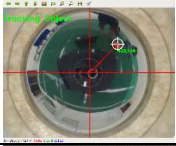
Gambar 7. (a) citra HSV (b) citra RGB

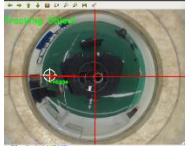
IV. HASIL DAN PEMBAHASAN

A. Posisi sudut bola terhadap robot

pada pengujian ini menggunakan kamera *omnidirectional* dan sebuah laptop dengan OS Linux Ubuntu 16.04 untuk mengolah data citra yang akan diproses pada robot sepak bola beroda UAD. Pada persamaan 2 di rumuskan yaitu untuk mendapatkan posisi sudut bola terhadap robot dibutuhkan koordinat bola dari hasil citra yang dideteksi. Koordinat dan posisi sudut bola terhadap robot ditunjukkan oleh Tabel 1

Tabel 1. Pengujian sudut bola

Data	Sudut	Nilai x bola	Nilai y bola	Pendeteksian sudut
1	4	355	56	
2	44	423	134	
3	116	439	299	
4	158	387	361	
5	230	215	328	
6	284	178	205	
7	323	275	97	
8	359	318	70	
9	86	463	241	

10	272	160	234	
----	-----	-----	-----	---

Dari hasil uji yang dilakukan pada Tabel 1 diperlihatkan data 1- 10 penggunaan metode ini dapat berfungsi dan mendeteksi bola dengan baik sehingga sudut yang dihasilkan dapat diterapkan pada robot, karena selain sudut pandang robot di lapangan tidak terbatas penggunaan metode ini dapat digunakan pada setiap robot yang menggunakan kamera *omnidirectional* jenis hiperbola.

V. KESIMPULAN

1. Robot dapat menentukan posisi bola berdasarkan koordinat x dan y bola melalui arah sudut pada kamera *omnidirectional*.
2. Penelitian menggunakan arah sudut pada kamera *omnidirectional* sangat cocok karena sudut pandang pada kamera *omnidirectional* tidak terbatas.

VI. DAFTAR PUSTAKA

- [1] Tofik Nurochman, Yogo Paranto Aji, Riky Dwi Puriyanto, "Sistem pendeteksian gawang menggunakan metode houghlines dan cornerharris pada robot fire-x," The 5th Indonesian Symposium on Robotic System and Control 2017
- [2] Pambudi Wahyu Setyo, Irma salamah, Alam Novi Tomunu, "Deteksi dan estimasi jarak objek menggunakan *single* kamera dengan model segmentasi HSV ," Seminar nasional teknoin 2011
- [3] Iwan kumianto wibowo, mochamad mobed bachtiar,rizki taufik ulil, "Desain omnidirectional vision pada ERSOW", The 5th Indonesian Symposium on Robotic System and Control 2017
- [4] https://docs.opencv.org/3.4/d2/d96/tutorial_py_table_of_contents_imgproc.html image processing in OPENCV
- [5] 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.
- [6] Yagi , Y. (1999) . Omni directional Sensing And its Applications , IEICE TRANS ,INF. & SYST . Vol. E82-D(No.3), pp. 568-579.

Rancang Bangun Robot Sepak Bola Beroda

Jhoram Beltsezar Parera, Ferdy Tjoanapessy

Teknik Elektro, Universitas Sam Ratulangi Manado, Jl. Kampus Bahu-Unsrat Manado, 95115
yobelparera@gmail.com, ferdytj08@gmail.com

Abstrak - Pada Kontes Robot Indonesia 2018 yang di adakan oleh Ristekdikti, salah satu tema nya adalah KRSBI atau Kontes Robot Sepak Bola Indonesia dan salah satu jenisnya adalah Robot Sepak Bola beroda. Paper ini akan membahas tentang bagaimana di rancanganya sebuah robot sepakbola beroda yang tidak hanya memiliki kemampuan untuk mendeteksi bola tetapi juga kemampuan untuk bergerak dengan efisien, merebut dan menggiring bola, menendang bola serta menentukan arah tendangan yang tepat agar mengarah ke arah gawang lawan. Robot sepakbola beroda menggunakan Motor DC PG-45 dan roda omni agar Robot dapat bergerak maju, mundur, kiri, kanan serta berputar secara bebas, Motor DC PG-45 juga di gunakan sebagai penggerak untuk menendang bola. Untuk menentukan arah gawang, Robot Sepakbola Beroda menggunakan sensor kompas untuk menentukan arah gawang lawan sebelum bola di tendang. Dan mekanisme pengiring juga di gunakan Robot untuk menggiring bola, mengambil bola dan merebut bola dari cengkraman Robot lawan.

Kata Kunci : Kontes Robot Indonesia, Robot, Mendeteksi, Menggiring, Menentukan Arah

Abstract - In the Indonesian Robot Contest 2018 held by Ristekdikti, one of its themes is KRSBI or Football Robot Contest Indonesia and one of its kind is Robot Soccer wheels.

This paper will discuss how to design a wheeled football robot that not only has the ability to detect the ball but also the ability to move efficiently, seize and tackle the ball, kick the ball and determine the right kick direction to point towards the opposing goal.

Wheeled soccer robot using PG-45 DC Motor and omni wheel so that the robot can move forward, backward, left, right and rotates freely, PG-45 DC motors are also used as a movement to kick the ball. To determine the direction of the goal, the Football Soccer Robot uses a compass sensor to determine the direction of the opponent's goal before the ball is kicked. And the companion mechanism is also use in Robot to dribble, take the ball and snatch the ball from the clutch Robot opponent.

Keywords: Indonesian Robot Contest, Robot, Mendeteksi, Dribbling, Determining Direction

I. PENDAHULUAN

A. Latar Belakang

Di tahun 2018, Kementerian Riset Teknologi dan Pendidikan Tinggi (KEMENRISTEKDIKTI) mengadakan Kontes Robot Sepak Bola Indonesia (KRSBI) yang menjadi salah satu kategori Kontes Robot Indonesia (KRI). KEMENRISTEKDIKTI mengundang seluruh Universitas yang ada di Indonesia untuk mengikuti Kontes Robot Indonesia, dengan tujuan meningkatkan kreativitas dan meningkatkan ilmu pengetahuan dan teknologi robotika di Indonesia. Robot sepakbola beroda adalah robot tipe beroda yang di rancang agar dapat bergerak bebas secara maju, mundur, belok kiri, belok kanan, serong kiri, serong kanan serta putar kiri dan putar kanan. Bukan hanya di rancang untuk bergerak, robot sepak bola di harapkan mampu untuk mendeteksi bola serta mendang atau memasukan bola ke dalam gawang secara otomatis. Sebagai robot yang harus bergerak, mendeteksi bola dan menendang atau memasukan bola secara otomatis. Maka robot sepakbola beroda di lengkapi juga dengan sensor untuk membantu pergerakan robot menjadi efisien.

B. Rumusan Masalah

Bagaimana Merancang atau membuat bodi robot yang sesuai untuk di gunakan sebagai robot sepakbola beroda.

Bagaimana membuat robot dapat mengenali objek sebagai bola dan membuat robot memasukan bola kedalam gawang.

Apakah robot akan berfungsi dengan baik sebagai robot sepak bola beroda.

C. Tujuan

Merancang atau membuat body robot yang sesuai dengan ketentuan KRSBI

Merancang dan membuat pergerakan robot untuk perlombaan KRSBI

Membuat robot sepakbola yang berfungsi dengan baik.

II. LANDASAN TEORI

A. Base Robot dan Konfigurasi Roda

Pada robot sepakbola beroda memakai *system omni wheels robot*. *System omni wheels robot* adalah sistem robot yang dirancang dengan menggunakan roda omni sehingga bisa bergerak ke segala arah. Beberapa robot menggunakan 4 roda omni ataupun 3 roda dalam membuat *base robot*.



Gambar 1 Base Robot dilihat dari bawah

B. Motor DC

Robot sepakbola beroda menggunakan empat *Motor DC* untuk pergerakan robot. *Motor DC* yang di gunakan adalah *Motor PG-45*. *Motor PG-45* dilengkapi dengan *encoder*. Motor ini lebih lambat jika dibandingkan dengan motor 24V lainnya. Tetapi motor ini memiliki torsi yang kuat. Spesifikasi dari *Motor PG-4* adalah sebagai berikut.

Nominal Voltage : 24 Volt
 Nominal Power : 22,5 Watt
 Nominal Torque : 3250 Rpm
 Nominal Spee : 6,6 Ncm
 Starting Torque : 38 Ncm



Gambar 2 Motor DC

C. Roda Omni

Roda *omni directional* merupakan roda yang dapat berputar kesegala arah. Hal ini disebabkan rancangan roda khusus yang tidak hanya mempunyai roda tunggal, tetapi banyak roda dalam

satu roda inti. Ada roda inti besar, dan sepanjang tepi ada banyak roda kecil tambahan yang mempunyai sumbu tegak lurus terhadap sumbu roda inti. Roda omni sendiri memiliki badan roda berbahan plastik, dan material dari roller berbahan karet.



Gambar 3 Roda Omni

D. Mekanisme Penendang

Dalam aturan KRSBI 2018, Robot di haruskan untuk menendang saat memulai pertandingan dan memasukan bola kedalam gawang lawan, sehingga mekanisme penendang di dalam robot sepakbola beroda di butuhkan mekanisme penendang bola.



Gambar 4 Mekanisme Penendang

E. Mekanisme Penggiring Bola

Mekanisme penggiring bola sangat penting sebagai bagian dari robot sepakbola beroda, jika ingin robot untuk menahan bola dan menendang bola dengan benar . Pada aturan KRSBI 2018 penggiring bola adalah maksimum area bola yang dapat di sentuh oleh robot adalah 1/3 dari ukuran bola. Umumnya untuk penggiring bola menggunakan 2 buah *Motor DC*. Penggiring bola umumnya di rancang agar mudah dalam menangkap serta menggiring bola.



Gambar 5 Mekanisme Penggiring Bola

F. Pixy CMUCam5

Pixy CMUCam 5 adalah sensor gambar dengan prosesor yang kuat yang dapat di program untuk mengirim informasi yang di cari sehingga mikrokontroler Anda tidak kewalahan oleh data. *Pixy CMUCam* juga mengeksport informasinya dalam berbagai cara yaitu serial *UART*, *SPI*, *I2C*, *digital out*, atau *analog out*. Sehingga mikrokontroler dapat berkomunikasi dengan mudah sambil tetap melakukan tugas lain. Ini sangat tidak mengganggu sehingga dapat menghubungkan beberapa *Pixy CMUCams* ke satu mikrokontroler.



Gambar 6 Pixy Cam CMU5

G. Sensor Kompas

Dalam menentukan arah gawang lawan, robot sepakbola beroda menggunakan sensor *DT-Sense 3 Axis Compass*, *DT-Sense 3 Axis Compass* merupakan suatu modul sensor medan magnet yang menggunakan IC *HMC5883L* produksi *Honeywell*. IC *HMC5883L* merupakan chip yang didesain untuk membaca medan magnet yang cocok untuk aplikasi penunjuk arah.



Gambar 7 Sensor *DT-Sense 3 Axis Compass*

III. HASIL DAN PEMBAHASAN

A. Desain Robot

Robot sepakbola beroda merupakan robot yang dibuat agar dapat bekerja layaknya manusia memainkan bola. Robot sepak bola ini harus didesain sedemikian rupa sehingga harus bisa menahan bola, menggiring bola, menendang bola, serta bisa bermanuver dengan lincah saat mengejar bola. Desain robot sepakbola beroda bisa di pada gambar di bawah.



Gambar 8 Desain Robot Sepakbola Beroda



Gambar 9 Robot Sepakbola Beroda

B. Penendang Bola

Kontes Robot Sepak Bola Indonesia(KRSBI) menggunakan bola no 4. Keliling 63 – 66 cm dan berat sekitar 400 gram. Pada robot sepakbola beroda, mengunkan desain penendang yang di gerakan oleh dua buah *Motor DC PG-45*. Dan untuk memposisikan penendang, robot sepakbola beroda menggunakan *sensor proximity* dan menggunakan dua servo motor untuk mengangkat penahan bola.



Gambar 10 Design Penendang

C. Penggiring Bola

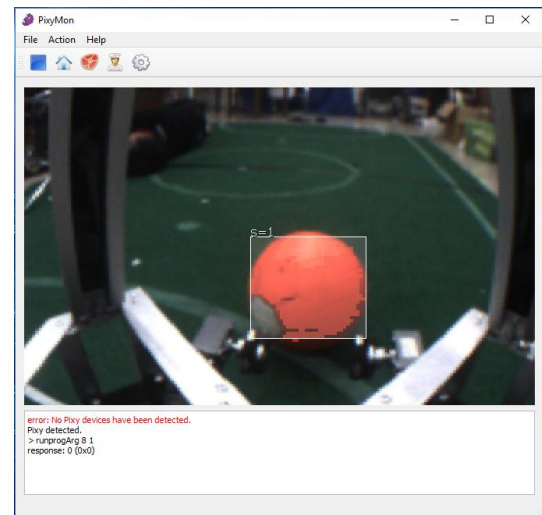
Pada Kontes robot di ijinikan untuk menggiring bola selama bola tidak lebih dari 1/3 diameter bola. Sistem penggiring bola di gunakan robot untuk mengambil bola, merebut bola dari lawan, memposisikan bola dan membawa bola lebih dekat ke arah gawang lawan. Robot sepakbola beroda menggunakan roda karet yang di gerakan oleh *Motor DC PG-45* sebagai penahan bola di bagian atas bola agar tidak lepas dari cengkaman robot dan menggunakan roda bebas di bagian bawah bola agar bola terkunci dengan rapat di penahan bola serta memakai penahan karet pada *Motor DC* agar penahan bola dapat menyesuaikan dengan ukuran bola.



Gambar 10 Design Penahan Bola

D. Sistem Deteksi

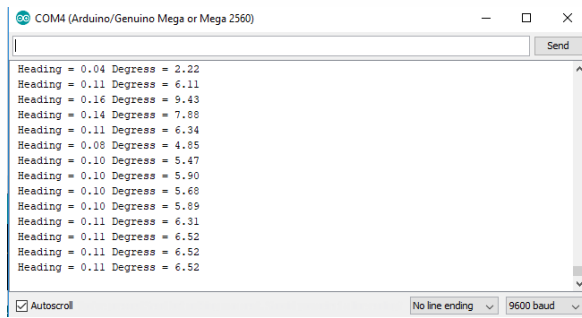
Pada robot sepakbola beroda, robot menggunakan sistem deteksi bola menggunakan sensor kamera *Pixy CMUCam 5*. Sensor kamera ini digunakan untuk mendeteksi bola dengan cara mendeteksi warna bola. Untuk men-scan warna yang akan diingat oleh kamera menggunakan sebuah software *open-source* yaitu *pixymon*, dari *software* ini bisa di atur berapa warna objek yang akan diingat oleh kamera *Pixy CMUcam5*, maksimal 7 warna objek. Selain itu *software* ini juga bisa digunakan untuk mengatur terang atau gelapnya penglihatan kamera. Kamera *pixy CMUcam 5* mendeteksi objek melalui pendeteksian warna. Pendeteksian warna untuk objek bola berwarna dilakukan dengan mengambil nilai X (pembacaan pergerakan bola secara vertikal) dan nilai Y (pembacaan pergerakan bola secara horizontal). Yang diterjemahkan kedalam koordinat agar memudahkan pengguna untuk memakainya.



Gambar 11 Penguncian warna pada Sensor Kamera

E. Sistem Navigasi

Untuk memetakan arah pada robot sepakbola beroda, robot menggunakan Sensor Kompas *HMC5883L* yang mengunkan arah mata angin sebagai panduan arah pada Robot. Pada sensor ini terdapat 3 *axis* data yaitu x, y dan z. Sensor ini memiliki kelebihan yaitu data yang diolah bisa menunjukan semua arah dengan nilai satu putaran atau 360 derajat. Dengan menggunakan Sensor Kompas, robot dapat membedakan arah gawang dan dapat di gunakan untuk strategi lain pada permainan.



Gambar 12 Konfigurasi Data Kompas

IV. Kesimpulan

Berdasarkan hasil pengujian robot, di dapat kesimpulan sebagai berikut.

1. Robot dapat bergerak dengan baik.
2. Robot dapat mendeteksi bola dengan menggunakan penanda warna pada kamera CMUcam 5.
3. Robot dapat mengunci dan menggiring bola.
4. Robot dapat menendang bola ke arah gawang lawan.

V. Daftar Pustaka

- [1] Vicky N. S. Lahimade. Rancang Bangun Robot Penendang Bola Menggunakan Pengendali PID [skripsi]. Manado (ID): Fakultas Teknik Jurusan Elektro Universitas Samratulangi
- [2] M. T. Kantori, Z. Sugianto, M. I. Nugraha, A. Febriansyah. Mobile Robot Pemain Bola: Aplikasi Pada KRSBI 2017.[Online] [diunduh 2018 Jun 10]; Tersedia Pada : <http://abec2017.polman-babel.ac.id/assets/file/PaperE061%20Mobile%20Robot%20Pemain%20Bola%20%20Aplikasi%20Pada%20KRSBI%202017.pdf>.
- [3] Muhammad Hilmi Asyrofi .Penerapan Algoritma Greedy untuk Mendeteksi Bola pada Robot Sepak Bola Beroda [Online] [diunduh 2018 Jun 10]; Tersedia Pada : <http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2016-2017/Makalah2017/Makalah-IF2211-2017-022.pdf>
- [4] Adafruit, Pixy CMU cam 5 [Online], diakses pada 18 Juli, Available: <https://www.adafruit.com/product/1906>

Forward dan Inverse Kinematics pada Robot Sepak Bola Beroda (*Robot Soccer Middle Size*)

La Ode Muhamad Idris*, Daniyah, Faizal Aminuddin Aziz
Magister Teknik Elektro
Universitas Islam Sultan Agung Semarang
Jl.Raya Kaligawe Km.4, Semarang 50112
ode_idris@std.unissula.ac.id

Abstrak – Pada penelitian ini sensor *rotary encoder* dan *gyroscope* digunakan sebagai sensor utama dalam melakukan pergerakan robot sepak bola beroda. *Rotary encoder* digunakan untuk membaca kecepatan dan jarak tempuh di setiap roda robot dan *gyroscope* digunakan untuk membaca orientasi dari robot yang kemudian pembacaan keduanya dinyatakan dalam diagram kartesian (X,Y,θ). *rotary encoder* yang digunakan adalah *rotary encoder* jenis *magnetic incremental encoder* yang dipasang pada setiap roda. *inverse kinematic* dan *forward kinematic* digunakan dalam pengontrolan pergerakan robot saat melakukan *positioning* di lapangan atau dalam pengambilan keputusan saat menendang bola ke arah gawang lawan serta pengambilan keputusan-keputusan lainnya. dari hasil pengujian didapatkan dalam melakukan *positioning* robot dapat bergerak ke segala arah dengan acuan posisi awal ke posisi akhir dengan orientasi robot yang berbeda-beda, serta dapat melakukan eksekusi tendangan ke arah gawang lawan dengan tepat dari segala posisi dalam lapangan. Dari hasil pengujian didapatkan robot mampu mengukur koordinat titik robot dengan *error* rata-rata sebesar $x = 5.371552\%$, $y = 5.200101\%$, dan $\Theta = 0.265778\%$. Robot juga mampu bergerak ke-arah koordinat yang diberikan dengan waktu tempuh 3 – 4 detik.

Kata kunci--*Inverse Kinematic, Forward Kinematic, Robot Sepak Bola, Rotary Encoder.*

I. PENDAHULUAN

Robot adalah salah satu teknologi yang diciptakan untuk membantu meringankan pekerjaan manusia, dalam arti luas robot adalah suatu alat yang pada batasan-batasan tertentu dapat bekerja secara otomatis maupun manual sesuai dengan perintah dari perancangannya. Perkembangan robotika Indonesia berkembang pada setiap tahunnya, perkembangan ini dapat dilihat dari dengan sering diadakannya kontes robot oleh Direktorat Jendral Pendidikan Tinggi (DIKTI)

serta beberapa instansi pendidikan, yang pesertanya baik ditingkat pelajar (SD/SMP/SMA/Sederajat), maupun tingkat Mahasiswa.

Salah satu kontes robot yang baru diadakan di Indonesia tahun 2017 yaitu Kontes Robot Sepak Bola Indonesia devisi ber-roda. Pada Kontes Robot Sepak Bola Indonesia ini mengacu pada aturan Kontes Robot Sepak Bola Internasional yaitu RoboCup. RoboCup adalah merupakan kompetisi robot sepakbola resmi tingkat dunia di bawah organisasi RoboCup (<http://www.robocup.org>), layaknya organisasi FIFA dalam kejuaraan dunia sepak bola manusia. Pada kontes robot ini Layaknya perlombaan sepak bola pada manusia, pada Kontes Robot Sepak Bola juga sangat membutuhkan kecepatan dan ketepatan pergerakan robot sehingga dapat memaksimalkan dalam mencetak gol. Untuk itu sistem kontrol pergerakan sangat penting, salah satu sistem kontrol adalah navigasi atau perencanaan jalur (*inverse kinematic*). Pada aplikasi robot dikenal istilah posisi relatif dan posisi absolut. Posisi relatif adalah sekumpulan data yang digunakan melalui perhitungan untuk menentukan posisi robot (*forward kinematic*), sedangkan posisi relatif merupakan hanya perkiraan saja. Dengan keakuratan posisi sangat mempengaruhi pengambilan keputusan pergerakan robot. Di mana penentuan pergerakan tersebut dapat dipengaruhi oleh data *input* beberapa sensor yang ada pada robot. Perancangan sistem kontrol pergerakan robot sangat penting dalam penentuan pergerakan ini. Salah satunya adalah dengan menggunakan sistem kontrol *kiwi drive three omnidirectional* dan dipadukan dengan *PID controller* sebagai pengontrol utama dalam mengatur kecepatan motornya.

II. TEORI PENUNJANG

A. Kinematika Robot

Dalam ilmu fisika, kinematika merupakan cabang dari matematika yang membahas gerak benda dan sistem benda tanpa mempersoalkan gaya penyebab gerakan. Kata kinematika dicetuskan

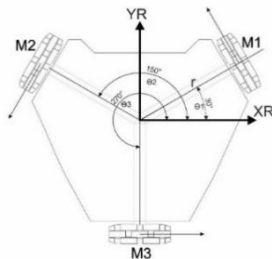
oleh fisikawan Prancis A.M. Ampere cinématique yang di ambil dari kata Yunani kuno, kinema (gerak) yang diturunkan dari kinein. Studi tentang kinematika juga dikenal sebagai geometri gerak [1] [2] [3].

Dalam bidang robotika kinematik sangat penting dalam melakukan pengontrolan pergerakan robot, baik robot dengan konstruksi manipulator, *humanoid*, *animalia*, maupun robot beroda. Hal yang penting dalam kinematika robot adalah pemahaman tentang sistem gerak mekanik dari robot itu sendiri sebelum melakukan perancangan *software* atau program dari robot itu sendiri.

Pada robot beroda *forward kinematic* berarti bagaimana mendapatkan koordinat posisi (x,y,theta) jika diketahui kecepatan di setiap roda robot, sedangkan *inverse kinematic* berarti bagaimana mendapatkan kecepatan di setiap rodanya jika diketahui koordinat posisi (x,y,theta) dari robot tersebut.

Untuk menentukan posisi robot, dibutuhkan referensi global (XG,YG) dan referensi lokal yang berpusat pada robot tersebut (XR, YR). posisi robot tersebut diperoleh dari rumus :

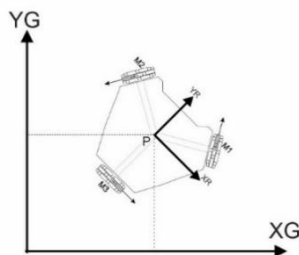
$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (1)$$



Gambar 1. Referensi vektor posisi robot 3 roda

Pemetaan pergerakan di antara referensi lokal dan global hanya dengan rotasi matriks (mengingat referensi lokal ini tetap dan tidak berputar terhadap robot tersebut):

$$R(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

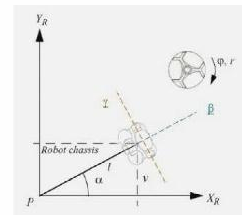


Gambar 2. Referensi vektor rotasi robot 3 roda

Persamaan metrik di atas dapat digunakan untuk pemetaan referensi global (XG, YG) untuk bergerak sesuai dengan referensi lokal (XR, YR). persamaan ini dinyatakan oleh $(\theta)\xi_I$ dikarenakan perhitungan dari persamaan tersebut tergantung pada nilai θ :

$$\xi'_R = R\left(\frac{\pi}{2}\right)\xi'_I \quad (3)$$

Untuk menentukan vektor kecepatan dari roda *omni* robot maka diperoleh dari turunan vektor posisi. Kinematika kecepatan dapat diketahui dengan memperhatikan fungsi dari arah pergerakan setiap roda, sumbu robot, kecepatan putaran sudut dan bentuk geometri dari robot itu sendiri.



Gambar 3. Referensi kinematik omni wheels

Untuk persamaan *forward kinematic* dari robot tiga roda adalah sebagai berikut:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = R \begin{bmatrix} -\frac{2s(\theta)}{3} & \frac{c\theta}{\sqrt{3}} + \frac{s\theta}{3} & \frac{c\theta}{\sqrt{3}} + \frac{s\theta}{3} \\ -\frac{2c\theta}{3} & \frac{c\theta}{3} + \frac{s\theta}{\sqrt{3}} & \frac{c\theta}{3} - \frac{s\theta}{\sqrt{3}} \\ \frac{1}{3L} & \frac{1}{3L} & \frac{1}{3L} \end{bmatrix} \begin{bmatrix} v1 \\ v2 \\ v3 \end{bmatrix} \quad (4)$$

Inverse kinematic dari robot tiga roda adalah sebagai berikut:

$$\begin{pmatrix} v1 \\ v2 \\ v3 \end{pmatrix} = \begin{bmatrix} -C(60 + \theta) & S(60 + \theta) & 1/3 \\ -C(60 + \theta) & -S(60 + \theta) & 1/3 \\ C(60 + \theta) & -S(60 + \theta) & 1/3 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (5)$$

B. Rotary Encoder

Rotary Encoder adalah transduser yang digunakan untuk mengukur posisi sudut dan kecepatan poros berputar. *Rotary encoder* biasa digunakan untuk memonitor dan kontrol di proses industri serta rekayasa robotika [4].

C. Sensor Gyro

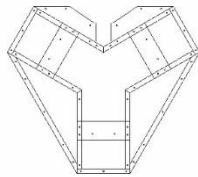
Sensor gyro adalah sensor gerak yang dapat mendeteksi dan mengukur gerak sudut suatu benda. Sebuah giroskop pada dasarnya terdiri dari massa berputar yang berputar mengelilingi porosnya. Secara khusus, ketika massa berputar pada porosnya, ia cenderung tetap sejajar dengan dirinya sendiri dan menentang usaha untuk mengubah orientasinya. Mekanisme ini ditemukan pada tahun 1852 oleh fisikawan Léon Foucault selama studinya tentang rotasi bumi. Jika giroskop

dipasang pada gimbal yang memungkinkan massa menavigasi dengan bebas ke tiga arah ruang, sumbu pemintalnya akan tetap berorientasi ke arah yang sama, bahkan jika ia berubah arah [5].

III. PERANCANGAN SISTEM

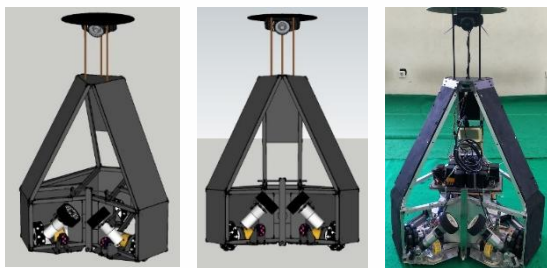
A. Perancangan Mekanik

Mekanik robot didesain menggunakan bahan dasar aluminium persegi 4 dengan tebal sisi $\frac{1}{2}$ inch, dan beberapa bagian robot menggunakan aluminium padat dengan tebal 3mm. Robot pada penelitian ini didesain dengan mengacu pada *rulu RoboCup 2018*, baik dari penentuan ukuran maksimal robot maupun penempatan beberapa penggerak seperti *dribble bola* yang telah ditentukan pada *rulu* tersebut. Pada robot diletakan 3 sensor *rotary encoder* pada setiap motor penggerak dengan jarak antara roda adalah 120° guna mempermudah penempatan dari rotary dan motor penggerak utama robot tersebut. Pada penelitian ini *rotary encoder* menggunakan *rotary encoder* yang telah ada pada motor penggerak utamanya untuk itu penempatan *rotary encoder* sama dengan penempatan motornya yaitu dengan sudut di setiap rotarynya sebesar 120° . Gambar 4 berikut merupakan desain dasaran dari robot:



Gambar 4 Desain dasaran mekanik robot

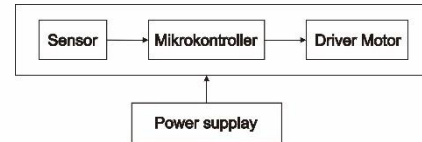
Berikut desain 3D dari robot:



Gambar 5 Desain 3D dan rancangan akhir robot sepak bola

B. Perancangan Elektronik

Elektronik robot terdiri dari *Supplay* tegangan kemudian rangkaian sensor-sensor, rangkaian kontrol dan rangkaian *driver* penggerak robot. Diagram elektronik dari robot pada penelitian ini adalah sebagai berikut:



Gambar 6 Diagram blok elektronik robot

Sensor-sensor yang digunakan oleh robot pada penelitian ini adalah sensor *Rotary encoder*, dan sensor *gyroscope*. Sensor *encoder* menggunakan sensor *encoder magnetik*, sedangkan sensor *gyroscope* yang digunakan adalah sensor modul MPU6050, Sedangkan rangkaian pengontrol atau sistem mikrokontroler yang digunakan adalah modul mikrokontroler Arduino Mega2560, Dan *driver* motor penggerak menggunakan modul *driver* motor BTS7960.

C. Perancangan Program

Perancangan program pada penelitian ini yaitu meliputi perancangan program *basestation* berupa aplikasi desktop dan perancangan program mikrokontroler robot. Program *basestation* digunakan untuk memonitoring posisi robot dan mengontrol posisi robot melalui PC. Program yang digunakan untuk merancang program *basestation* adalah Visual Studio dan bahasa pemrograman menggunakan Visual Basic. Sedangkan program mikrokontroler menggunakan Aplikasi pemrograman Arduino IDE dan bahasa pemrograman menggunakan bahasa standar Arduino.

IV. HASIL PENGUJIAN

A. Pengujian pembacaan jarak dari setiap roda

Pengujian dilakukan dengan menggerakkan robot secara manual dengan cara menggerakkan robot sehingga putaran roda ke arah searah putaran jarum jam dan berlawanan arah jarum jam dari setiap rodanya. Untuk mengukur jarak tempuh di setiap rodanya dibutuhkan persamaan sebagai berikut:

$$\text{Keliling Roda} = \text{Diameter Roda} * \pi$$

$$\text{Jarak per Pulsa} = \frac{\text{Keliling Roda}}{\text{Pulsa per Rotasi}}$$

$$\text{Jarak Tempuh Roda} = \text{Jarak per Pulsa} * \text{Pulsa Rotary}$$

Berikut adalah hasil pengujian jarak tempuh di setiap rodanya:

Tabel 1. Hasil pengujian jarak tempuh roda

No	Jarak Real(cm)	Jarak Pembacaan Roda (cm)		
		Kanan	Kiri	Belakang
1	10	10.01	10.01	10.01

2	20	20.02	20.02	20.02
3	30	30.03	29.94	30.03
4	40	39.93	39.93	40.04
5	50	49.82	50.05	50.05
6	60	60.06	60.06	60.06
7	70	69.96	70.07	70.07
8	80	80.08	80.08	80.08
9	90	89.98	89.98	89.98
10	100	99.99	100.02	99.99
Error rata-rata		0.11289	0.09446	0.03715

Dari percobaan yang dilakukan dapat dilihat bahwa sistem mampu mengukur jarak tempuh di setiap rodanya dengan rata-rata kesalahan pembacaan jarak sangat kecil yaitu untuk roda kiri sebesar 0.11308 cm, roda kanan 0.09450 cm, dan roda belakang 0.03712 cm.

A. Pengujian kinematic robot

Pada percobaan ini terdiri pengujian pembacaan jarak tempuh robot guna menguji *forward kinematic* dan melakukan pengujian memberikan nilai berupa koordinat tujuan robot guna menguji *inverse kinematic* robot. Pertama pengujian *forward kinematic* yaitu dilakukan dengan cara menggerakkan robot ke segala arah dengan posisi koordinat kartesian (x, y, dan theta) dengan satuan jarak *centimeter* (cm). Berikut adalah hasil pengujian *forward kinematic*:

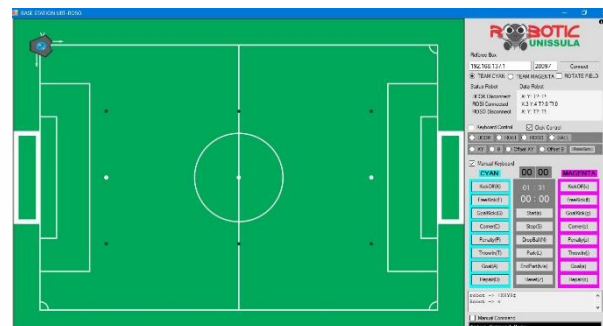
Tabel 2. Hasil pengujian *forward kinematic*

No	Jarak Real (cm)			Jarak Terukur(cm)		
	x	Y	θ	x	y	θ
1	0	10	0	0	10.5	0.52
2	10	10	0	10.11	9.99	0.46
3	10	20	90	11.9	21.09	91.6
4	20	20	90	22.04	21.3	89.9
5	20	30	180	22.1	32.2	180.1
6	30	30	180	32.01	32.5	179.9
7	30	40	270	32.11	43.02	270.2
8	40	40	270	42.06	43.03	271.01
9	40	50	360	42.03	54.19	358.99
10	50	50	360	53.19	54.53	359.33
11	50	60	360	52.99	63.32	359.28
12	60	60	360	61.06	63.45	359.34
13	60	70	270	61.23	73.5	271.45
14	70	70	270	72.6	73.6	271.54
15	70	80	180	72.9	82.99	181.35
16	80	80	180	82.4	83.52	180.34

17	80	90	90	82.6	93.23	91.35
18	90	90	90	92.66	94.12	89.32
19	90	100	0	93.45	102.56	1.25
20	100	100	0	105.6	103.63	1.46
Error rata-rata				5.371552	5.200101	0.265778

Dari hasil pengujian yang dilakukan terlihat bahwa sistem mampu mengukur jarak tempuh robot dalam koordinat (x,y, θ). Terlihat bahwa pada jarak tempuh robot awalnya memiliki sedikit nilai *error* namun setelah menempuh jarak lebih dari 10cm robot memiliki nilai *error* lebih dari 1cm. nilai *error* ini semakin bertambah dengan makin jauhnya jarak tempuh yang telah dilakukan. Hal ini dikarenakan dalam melakukan percobaan terjadi selip pada roda sehingga pembacaan jarak tempuh di setiap rodanya memiliki pembacaan yang lebih. Nilai *error* ini juga dikarenakan pada saat melakukan percobaan hanya menggunakan dorongan manual tanpa jalur khusus sehingga rute dalam perpindahan atau pergeseran tidak begitu akurat.

Pengujian *inverse kinematic* untuk mewakili pengujian mengukur kecepatan disetiap motordilakukan dengan cara melakukan pemberian titik koordinat melalui PC kemudian mengamati seberapa akurat dan cepat respon dari robot menuju titik tersebut. Berikut adalah gambaran program simulasi dari aplikasi yang telah dibuat:



Gambar 7. Basestation robot sepak bola

Berikut adalah hasil pengujian yang dilakukan dengan memberikan beberapa koordinat yang dituju.

Tabel 3. Tabel pengujian *inverse kinematic*

No	awal			tujuan			waktu (s)
	x	y	θ	x	y	θ	
1	0	0	0	0	450	0	4.2
2	0	450	0	0	0	0	4
3	0	0	0	300	0	0	4.8
4	300	0	0	0	0	0	3.1
5	0	0	0	0	450	90	4.3

6	0	450	90	0	0	0	4.2
7	0	0	0	300	0	90	3.8
8	300	0	90	0	0	0	3.4
9	0	0	0	0	450	180	4.3
10	0	450	180	0	0	0	4.2
11	0	0	0	300	0	180	3.5
12	300	0	180	0	0	0	3.2
13	0	0	0	0	450	270	4.2
14	0	450	270	0	0	0	4.1
15	0	0	0	300	0	270	3.8
16	300	0	270	0	0	0	3.7

Dari percobaan yang dilakukan didapatkan bahwa sistem telah mampu melakukan eksekusi *inverse kinematic* dari robot sehingga robot mampu bergerak ke arah titik koordinat yang diberikan dengan waktu yang tercatat seperti pada tabel hasil pengujian *inverse kinematic*. Pada percobaan dengan menguji titik tujuan berupa pergerakan sumbu x yaitu dari titik 0 cm ke 450 cm dan sumbu y 0 cm ke 300 cm dengan beberapa arah yaitu 0°, 90°, 180°, dan 270° yang kemudian dilakukan juga rute yang dibalik yaitu dari 450 cm ke 0 cm untuk sumbu x, dan 300 cm ke 0 cm untuk sumbu y. dari beberapa percobaan tersebut dapat diamati bahwa pergerakan menuju titik akhir 0 cm cenderung lebih lambat dibandingkan dengan pergerakan menuju titik x 300 cm dan menuju titik y 450 cm. Hal tersebut dikarenakan terjadinya pergerakan berupa robot terlebih dahulu melakukan perputaran atau rotasi pada saat mulai melakukan pergerakannya. Dari hasil pengujian juga telah diamati bahwa setelah melakukan pergerakan beberapa kali robot ter-ukur tidak benar-benar mampu berada pada titik yang dituju namun masih memiliki sedikit kesalahan beberapa cm dari koordinat yang dituju, hal tersebut dikarenakan terjadinya selip pada roda sehingga mempengaruhi pembacaan *rotary encoder* di setiap rodanya dan mengakibatkan pembacaan jarak tempuh robot memiliki kesalahan.

V. KESIMPULAN DAN SARAN

Forward kinematic robot mampu memberikan informasi posisi robot dalam koordinat kartesian dalam hal ini titik robot pada lapangan kontes robot sepak bola namun masih memiliki sedikit kesalahan ketika telah melakukan pergerakan dengan terjadinya selip pada roda dan mempengaruhi pembacaan *rotary encoder*.

Inverse kinematic robot mampu memberikan respon dalam melakukan pergerakan dalam lapangan kontes robot sepak bola dengan memberikan titik rujukan robot.

Saran dari penulis untuk pengembangan penelitian serupa adalah penggunaan *rotary*

encoder sebaiknya menggunakan *rotary encoder* yang tidak langsung berada pada motor penggerak utama sehingga memperkecil nilai kesalahan apabila terjadi selip pada roda dalam melakukan pembacaan jarak tempuh dari setiap rodanya. Perancangan mekanik robot sebaiknya memiliki tingkat presisi yang tinggi terutama dalam peletakan *rotary encoder* sehingga memperkecil nilai kesalahan dalam pengaplikasian persamaan-persamaan yang telah ditentukan sebelumnya.

VI. REFERENSI

- [1] E. T. Whittaker, A Treatise on the Analytical Dynamics of Particles and Rigid Bodies, New York: Cambridge University Press, 1904.
- [2] J. S. Beggs, Kinematics, Taylor & Francis, 1983.
- [3] T. W. Wright, Elements of Mechanics Including Kinematics, Kinetics and Statics, E and FN Spon, 1896.
- [4] A. Arslan, "Comparative Analysis of Speed Decoding," Aalto University School of Electrical Engineering, 2017.
- [5] V. M. N. Passaro, "Gyroscope Technology and Applications: A Review in the Industrial Perspective," *Sensors*, 2017.

Pendeteksian Bola Menggunakan Metode YUV *Color Filtering* pada Robot EWS Barracuda

M Hilal Bayu Aji, Alfian Muharromi Adi Nugraha, Sumardi
Departemen Teknik Elektro, Universitas Diponegoro Semarang
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia
Email korespondensi: bayuhilal76@gmail.com

Abstrak - Pendeteksian bola pada robot sepakbola merupakan hal yang sangat penting supaya bisa berjalan dengan baik dalam melakukan misinya. Pada robot EWS Barracuda menggunakan metode *color filtering* yaitu YUV yang sudah tersedia di library *OpenCV*. Untuk perangkat keras yang digunakan adalah *Raspberry Pi 3 B*. Dalam dua kali pengujian dengan menggunakan metode tersebut, robot dapat mendeteksi bola baik jarak dekat maupun jarak yang lebih jauh dengan jangkauan maksimal 3–4 meter.

Kata kunci: YUV, *OpenCV*, robot sepakbola.

I. PENDAHULUAN

Tim EWS Barracuda adalah tim yang mengikuti Kontes Robot Sepakbola Indonesia Beroda (KRSBI Beroda) dari Universitas Diponegoro, Semarang. Di dalam tim ini, mahasiswa diberi kebebasan untuk mengembangkan kemampuannya, di antaranya mendesain mekanik robot dan pcb, membuat algoritma dan memprogramnya.

Salah satu bagian mendasar pada robot sepakbola ini adalah kemampuan untuk mendeteksi keberadaan bola menggunakan kamera. Untuk bisa menggunakan kamera sebagai sensor, digunakanlah library *OpenCV* yang bisa dipasang secara gratis di sistem operasi *windows*, *linux* dan pengembangannya. Di dalam library *OpenCV* terdapat fitur untuk bisa menyaring warna sesuai kebutuhan, diantaranya *HSV*, *YUV*, *RGB* dan lain sebagainya.

Dari beberapa cara untuk menyaring warna diatas, kami menggunakan metode YUV karena dapat membedakan warna bola dan lapangan dengan range nilai setiap komponennya (komponen Y, U, dan V) lebih lebar dari metode yang lainnya.[1]

II. METODE

A. Tinjauan Pustaka

1. *Raspberry Pi 3*

Raspberry Pi 3 Model B adalah model paling awal *Raspberry Pi* generasi ketiga. Ini menggantikan *Raspberry Pi 2 Model B* pada bulan Februari 2016.

Raspberry Pi 3 B memiliki spesifikasi diantaranya sebagai berikut:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU*
- 1 GB RAM*
- 40 Pin I/O*
- 4 Port USB 2*
- Full size HDMI*



Gambar 1. *Raspberry Pi 3 B*

2. *OpenCV*

OpenCV (Open Source Computer Vision Library) dirilis di bawah lisensi BSD dan karenanya gratis untuk penggunaan akademis dan komersial. Library ini bisa digunakan dalam bahasa C++, Python dan antarmuka Java dan mendukung *Windows*, *Linux*, *Mac OS*, *iOS* dan *Android*. *OpenCV* dirancang untuk efisiensi komputasi dan dengan fokus yang kuat pada aplikasi real-time.

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.[2]

3. YUV

Model YUV mendefinisikan ruang warna dalam satu komponen luma dan dua *chrominance*. YUV memodelkan persepsi manusia tentang warna lebih dekat daripada model *RGB* standar yang digunakan pada perangkat keras grafis komputer. Y adalah komponen luma, atau kecerahan. U dan V adalah komponen krominasi (komponen pencahayaan biru dan komponen pencahayaan merah). [2]

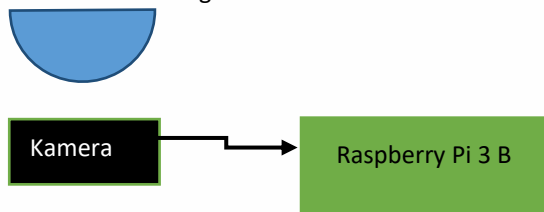
Range nilai dari setiap komponennya yaitu 1 Byte atau dari 0 sampai 255.[3]

B. Perancangan Sistem

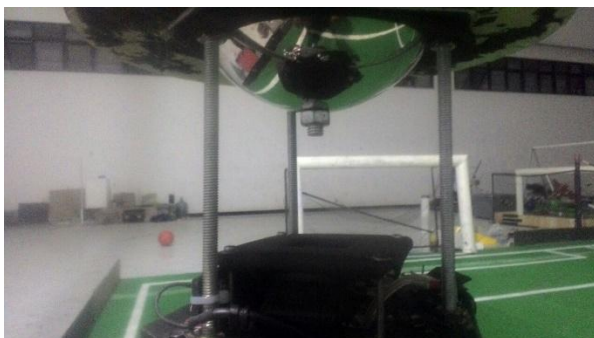
Pada sistem ini terdiri dari kamera yaitu *WebCam Logitech C922*, pemantul yang berbentuk cembung yang bisa dilihat pada gambar 3, dan *Raspberry Pi 3 B*. Untuk bisa lebih memahami, dapat dilihat gambar 2 yaitu diagram blok dari sistem pendeteksian bola.

Pemantul cembung digunakan supaya kamera bisa mendapatkan gambar dengan sudut sampai 360° sehingga pendeteksian lebih efisien. *Raspberry Pi 3 B* digunakan untuk memproses gambar yang didapat dari kamera, sehingga hasilnya sesuai dengan yang diinginkan, yaitu mendeteksi keberadaan bola.

Pemantul Cembung

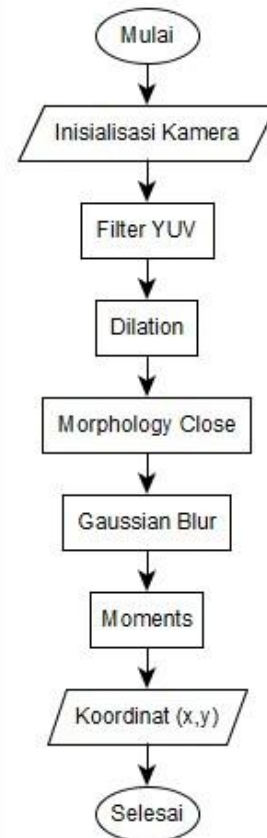


Gambar 2. Diagram blok sistem deteksi bola



Gambar 3. Mekanik kamera dan pemantulnya

Untuk mendeteksi bola, gambar yang dihasilkan oleh kamera, selanjutnya difilter berdasar warnanya dengan metode *YUV*, setelah itu di-*dilation*, kemudian di-*morphology close* dan *gaussian blur*, sehingga mendapatkan koordinat (x,y) melalui proses *Moments*. Untuk lebih jelasnya bisa dilihat *flowchart* program pada gambar 4.



Gambar 4. Flowchart program

Supaya dapat mendeteksi bola, perlu dilakukan kalibrasi terlebih dahulu. Untuk melakukan kalibrasi, digunakanlah *trackbar* untuk mencari nilai minimal dan maksimal untuk setiap komponennya, yaitu Y, U dan V. Dalam penulisan ini, nilai-nilainya sebagai berikut:

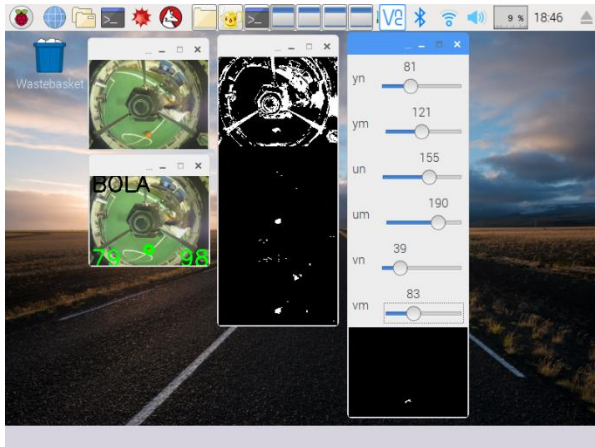
Tabel 1. Nilai hasil kalibrasi

1.	Y-min	81
2.	Y-max	121
3.	U-min	155
4.	U-max	190
5.	V-min	39
6.	V-max	83

III. HASIL DAN PEMBAHASAN

Untuk menilai bahwa hasil sudah sesuai atau belum, maka dilakukan dua kali pengujian dengan nilai-nilai komponen Y, U, dan V sesuai dengan nilai-nilai dari hasil kalibrasi.

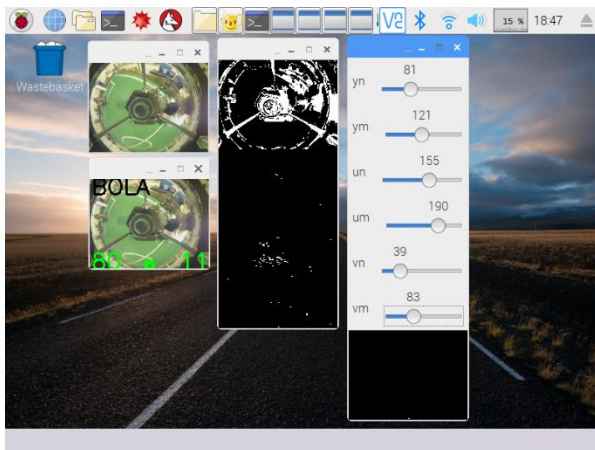
Pada pengujian pertama yaitu dengan meletakkan bola didekat robot. Seperti pada gambar 5.



Gambar 5. Pengujian jarak dekat

Berdasarkan gambar 5, robot bisa mendeteksi bola dengan koordinat $x=79$ dan $y=98$.

Pada pengujian yang kedua, bola diletakkan pada jarak yang lebih jauh yaitu 3-4 meter dari robot, seperti pada gambar 6.



Gambar 6. Pengujian jarak jauh

Berdasarkan gambar 6, robot masih bisa mendeteksi keberadaan bola meski jaraknya bertambah jauh. Pada pengujian ini bola terdeteksi pada koordinat $x=89$ dan $y=110$.

Dalam dua kali pengujian yaitu pada jarak dekat dan jarak yang lebih jauh, robot dapat mendeteksi keberadaan bola. Nilai koordinat yang ada digunakan untuk memberi perintah kepada kinematika gerak, yaitu y , x , dan w (omega). Komponen y untuk gerak maju dan mundur, x untuk gerak kanan dan kiri dan nilai w (omega) untuk berputar. Nilai koordinat x akan dimasukkan ke nilai w (omega) di kinematika gerak dan nilai koordinat y masuk ke nilai y di kinematika gerak.

IV. KESIMPULAN

Berdasarkan hasil pengujian-pengujian yang telah dilakukan dengan menggunakan nilai-nilai setiap komponen dari Y , U dan V sesuai dengan nilai pada saat kalibrasi. Robot dapat mendeteksi

bola baik pada jarak dekat maupun jarak yang lebih jauh yaitu sekitar 3-4 meter.

V. DAFTAR PUSTAKA

- [1] A. K. Mulya, F. Ardilla, D. Pramadihanto, "Ball Tracking and Goal Detection for Middle Size Soccer Robot Using Omnidirectional Camera", International Electronics Symposium (IES), 2016.
- [2] G. B. A. Dewasakti dan Susanto, "Sistem Pendeteksian Bola Secara Real-Time Pada Robot Barelang FC", Indonesian Symposium on Robotic Systems and Control, 2017.
- [3] L. Hai-bo, W. Yu-mei, dan D. Yu-jie, "Fast recognition based on color image segmentation in mobile robot", International Symposium on Computer Science and Computational Technology, 2010.

Implementasi *Omnidirectional Vision Camera* Menggunakan Cermin Cembung

Wahyu Muldayani¹, Mohamad Arif Baihaqiy², Aditty Aprillia A.³, Fitra Elbi A.⁴, Faiq Aprillian R.⁵, Syaiful Komar⁶, Cahya

Mulya A.⁷, Dandi Susanto⁸

¹ Staf Pengajar Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember

^{2,3,4,5,6,7,8} Mahasiswa Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember

Jl. Kalimantan 37, Jember 68121

email: arifbaihaqiy1997@gmail.com

Abstrak -- Pendeteksian warna merupakan hal yang paling penting untuk robot vision pada kontes robot sepak bola beroda. Makalah ini menjelaskan tentang sub-sistem pengaplikasian kamera omnidirectional vision menggunakan pantulan dari cermin cembung untuk mendeteksi bola, lawan dan kawan. Riset yang kami lakukan belum mencapai pendeteksian garis putih yang ada pada lintasan KRSBI Beroda. Pada kondisi di lapangan, ketika digunakan cermin cembung untuk mendeteksi navigasi bola tidak bisa mendeteksi ketika bola berada pada jarak yang cukup jauh, yaitu lebih dari 5 meter dari robot. Dalam tulisan ini kami menyajikan tentang algoritma yang kami gunakan untuk mendeteksi bola dan rintangan yang cukup efisien untuk digunakan pada robot sepakbola beroda khususnya Tim JR EVO. Dalam tulisan ini menunjukkan efektivitas dari algoritme yang kami gunakan sehingga dapat memaksimalkan gerak dan kinerja robot saat bertanding. Dengan metode yang kami kembangkan serta diaplikasikan langsung pada Kontes Robot Indonesia 2018 sehingga membuahkan hasil yang cukup baik. Yaitu Tim JR EVO dari Universitas Jember berhasil menduduki peringkat 4 besar (Juara Harapan) dalam KRI 2018 Regional 4. Selain itu, eksperimen dan pengembangan yang kami gunakan menunjukkan sistem cepat dan akurat memiliki waktu pemrosesan yang konstan serta independen dari lintasan di sekitar robot.

Kata kunci: Robot soccer, Kamera, *Omnidirectional vision*.

I. PENDAHULUAN

Dalam permainan sepak bola dapat dipandang sebagai permasalahan yang mudah dipahami dalam dunia nyata, namun kenyataannya ketika permasalahan ini dibawa kedalam dunia robotika, permasalahan tersebut menjadi sangat kompleks. Yaitu dengan melibatkan algoritma yang sangat kompleks sehingga robot dapat melakukan navigasi, mapping posisi serta dapat berkomunikasi dengan robot lain dalam satu basestation dan dapat mengetahui keberadaan lawan. Skenario sepak bola tersebut menjadi tantangan tersendiri bagi komunitas robotika untuk membuat robot yang mampu memainkan sepak bola seperti layaknya pemain sepak bola pada umumnya, agar menyerupai permainan sepak bola sesungguhnya, robot dituntut bermain secara autonomus, tanpa campur tangan manusia.

Tim robot UNEJ telah mengikuti kompetisi Kontes Robot Sepak Bola Indonesia sejak awal kompetisi ini diadakan, yaitu tahun 2017. Dalam perjalanannya tim robotika UNEJ telah melakukan berbagai modifikasi terhadap robot yang digunakan. Pada tahun ini tim robot UNEJ dengan nama tim JR EVO pada KRSBI divisi Beroda. Kontes robot sepakbola beroda indonesia mengacu pada Middle Size League (MSL) Robocup Soccer League. MSL dari RoboCup adalah tes standar dunia nyata untuk robot sepakbola. meskipun ada perubahan baru-baru ini diperkenalkan, seperti tujuan tanpa warna, mengenali objek berwarna seperti bola oranye, rintangan hitam, bidang hijau dan garis putih adalah kemampuan dasar untuk robot.

Penjejakan Objek atau Object *tracking* merupakan proses mencari letak objek yang akan diamati dari sebuah data video untuk setiap frame pada video tersebut. Fungsi utama dari penjejakan objek adalah mengikuti sebuah objek untuk tujuan tertentu. Penjejakan objek merupakan proses yang dilakukan oleh komputer *vision*. Untuk itu diperlukan algoritma yang dapat menganalisis setiap perubahan frame dalam video sehingga dapat diketahui adanya pergerakan objek beserta lokasi objek tersebut. Selanjutnya penjejakan tersebut akan dilanjutkan ke software visual basic dari warna dengan format RGB. Kemudian dikonversi menjadi gambar biner sehingga lebih mudah dalam proses pengolahan data. Ketika sudah menjadi gambar biner maka akan diperoleh nilai sudut atau nilai *radian* serta nilai jarak yang menjadi acuan posisi dan kemana robot harus bergerak mengikuti bola.

II. METODE PENELITIAN

Penelitian tentang "Implementasi *Omnidirectional Vision Camera* Menggunakan Cermin Cembung" ini dilaksanakan di gedung Aula Lembaga Pengabdian kepada Masyarakat, Universitas Jember. Pada penelitian ini menggunakan alat perekam objek berupa webcam Logitech dengan seri C930e. Dengan memanfaatkan pantulan cermin cembung yang terpasang pada bagian atas robot JR EVO. Dari pantulan cermin cembung tersebut akan diperoleh nilai sudut jika diambil dari titik tengah cermin dan posisi kamera, dengan *range* nilai 0 sampai 180. Sedangkan ketika objek seperti bola bergerak semakin ke kiri, sehingga *range* nilai yang di peroleh adalah 0 sampai -180.

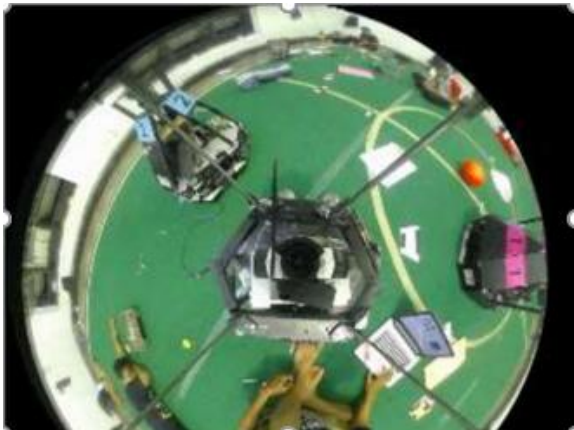
Pengolaha Citra

Pengolahan Citra Gambar yang diperoleh dari kamera pada dasarnya adalah suatu nilai cahaya berbentuk dua dimensi yang diambil pada waktu tertentu. Untuk melakukan proses pengolahan citra, pada komputer diperlukan 4 tahapan antara lain :

1. Kalibrasi warna serta posisi Objek dimana akan dijadikan acuan pengonversian objek ke bentuk biner sehingga memudahkan dalam pengolahan data.
2. Pengubahan dari objek gambar yang telah di kalibrasi dengan jenis gambar dengan nilai *RGB* menjadi gambar dengan nilai biner
3. Kemudian hasil dari pengubahan objek gambar yang telah didapat akan dilakukan proses perhitungan untuk mencari dan menentukan nilai titik tengah dari objek yang telah di kalibrasi dan di *traking*.
4. Hasil dari penghitungan yang berupa nilai akan di kirimkan ke microcontroller kemudian robot dapat bermanuver menuju objek yang telah di tentukan

Jenis Gambar Dan Menentukan Objek

Untuk dapat diproses agar mendapatkan data tertentu dari sebuah gambar, diperlukan jenis gambar yang sesuai. Biasanya digunakan beberapa jenis gambar yaitu : gambar *RGB*, gambar *grayscale*, gambar *binary*, dan *special image*. Gambar *RGB* merupakan gambar berwarna dengan 3 warna dasar yaitu merah, hijau, biru (*Red, Green, Blue*). Sedangkan gambar *grayscale* merupakan gambar dengan warna negative atau hitam putih Masing-masing intensitas warna umumnya ditampung pada channel 8 bit sehingga secara keseluruhan membutuhkan minimal 24 bit.



Gambar 1. Pandangan dari cermin cembung Gambar *RGB* Selanjunya dilakukan kalibrasi object dengan memilih object yang akan dijadikan acuan seperti bola, musuh, dan kawan seperti pada gambar dibawah ini.

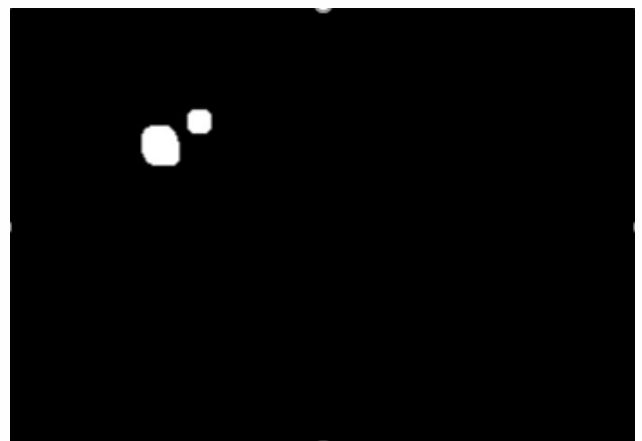


Gambar 2. Kalibrasi object

Setelah diperoleh object yang akan menjadi acuan, Gambar akan dirubah menjadi gambar biner pada masing masing object (bola, kawan, dan lawan) dengan software Visual Basic.



Gambar 3. Hasil Pengolahan Biner Objek Bola



Gambar 4. Hasil Pengolahan Biner Objek Kawan (cyan)



Gambar 5. Hasil Pengolahan Biner Objek Lawan (magenta)

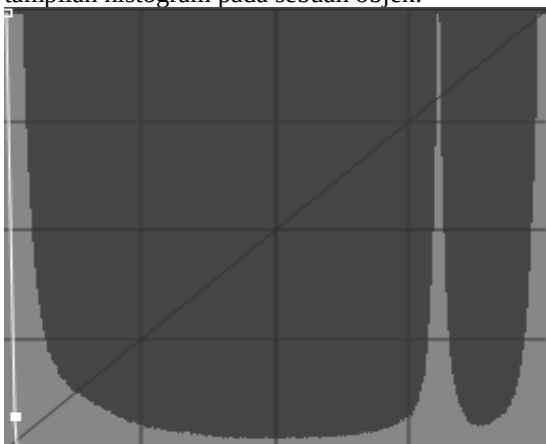
Bola	Lawan	Kawan
Sudut 75	Sudut 75	Sudut -39
R 114	R 111	R 84
X 111	X 108	X -52
Y 29	Y 28	Y 66

Gambar 6. Hasil Pengolahan Objek Biner Secara Keseluruhan

Seluruh data gambar biner tersebut selanjutnya dikirim ke mikrokontroler melalui port komunikasi dengan *type serial port*.

Peningkatan Kualitas Citra

Peningkatan Kualitas Citra secara Histogram adalah sebuah grafik yang menampilkan sebuah frekuensi dengan adanya kemunculan suatu warna pada gambar. Pada grafik dengan menggunakan koordinat jenis kartesian, sumbu X merupakan nilai warna dari gambar dan sumbu Y merupakan jumlah kemunculan nilai warna tersebut. Gambar dibawah merupakan contoh tampilan histogram pada sebuah objek.



Gambar 7. Pengolahan Citra Dengan Histogram

Histogram citra merupakan diagram yang menggambarkan Sebuah frekuensi nilai piksel dalam suatu citra. Sumbu horizontal merupakan nilai intensitas piksel sedangkan sumbu vertikal merupakan frekuensi/jumlah

pikse yang digunakan Dari sebuah histogram dapat diketahui frekuensi kemunculan nisbi (relatif) dari intensitas pada citra tersebut.. Histogram juga dapat menunjukkan banyak hal tentang kecerahan (brightness) dan kontras (contrast) dari sebuahcitra. Karena itu, histogram adalah alat bantu yang berhargadalam pekerjaan pengolahan citra baik secara kualitatif maupun kuantitatif.. Filtering merupakan teknik yang digunakan untuk menekan gangguan atau derau (noise) pada gambar. Gangguan pada gambar dapat merupakan perbedaan suatu piksel dengan piksel sekitarnya. Teknik filtering memiliki beberapa metode diantaranya mean filtering, median filtering, modus filtering, dan gaussian filtering. *Mean filtering* yang digunakan untuk efek *smoothing* ini merupakan jenis *spatial filtering*, yang dalam prosesnya mengikutsertakan piksel-piksel disekitarnya. Piksel yang akan diproses dimasukkan dalam sebuah matrik yang berdimensi $N \times N$. Ukuran N ini tergantung pada kebutuhan, tetapi nilai N haruslah ganjil sehingga piksel yang diproses dapat diletakkantepat ditengah matrik

1	2	3
4	T	5
6	7	8

Gambar 8. Matriks mean filtering

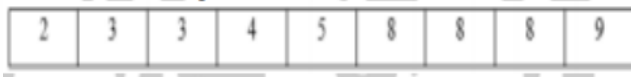
Gambar 8 merupakan mean filtering dengan N bernilai 3. T merupakan piksel yang akan di-filter, sedangkan 1, 2, 3, 4, 5, 6, 7, dan 8 merupakan nilai piksel disekitar piksel T. Nilai T yang baru merupakan nilai rata-rata dari semua piksel. *Modus* adalah nilai *variable* yang memiliki *frekuensi* tertinggi (Usman, 2005). *Modus* dapat ditemukan pada data yang telah diurutkan maupun yang belum terurut. Untuk menentukan modus filter ini, pertama-tama ditentukan piksel utama yang akan diproses dari piksel-piksel disekitarnya. Pada *modus filtering* digunakan matrik berdimensi $N \times N$. Dari matrik tersebut, kemudian data yang ada diurutkan dan dimasukkan dalam sebuah matrik berukuran $1 \times (N \times N)$. Hal ini berguna untuk mempermudah menemukan *modusnya* untuk pencarian nilai yang terbanyak *frekuensinya* dari kumpulan data yang telah urut tersebut. Dari matriks yang telah ditentukan, diambil nilai dengan frekuensi kemunculan terbanyak yang akan menggantikan nilai piksel utama. Karena piksel utama harus diitari oleh piksel lain, maka nilai modus filtering harus bernilai ganjil.

2	3	5
8	8	4
8	3	9

Gambar 9. Matriks modus filtering dengan hasil 8

Median filtering adalah penggantian nilai suatu piksel dengan nilai tengah dari kumpulan data disekitar piksel utama. Untuk median filtering ini, data yang digunakan untuk menghitung median terdiri dari kumpulan data yang ganjil. Hal ini dikarenakan dengan

jumlah data yang ganjil maka piksel yang akan diproses akan berada ditengah.



Gambar 10. Urutan median dari matriks

Gambar 10. merupakan urutan yang dari matriks pada gambar 5 sehingga dapat diketahui nilai piksel barunya adalah 5.

Segmentasi

Segmentasi adalah proses membatasi citra digital menjadi beberapa segmen atau bagian (piksel, juga dikenal sebagai *superpixels*). Tujuan dari segmentasi adalah untuk menyederhanakan dan atau mengubah penyajian gambar ke sesuatu yang lebih bermakna dan lebih mudah untuk di analisis. Dalam proses segmentasi, terdapat beberapa metode yang dapat dilakukan antara lain segmentasi berdasarkan klasifikasi, segmentasi berdasar daerah tepi (*edge*), dan segmentasi berdasarkan daerah (*region*). Segmentasi citra berdasarkan klasifikasi adalah mencari kesamaan kesamaan pada piksel, dapat berupa warna atau ukuran. Segmentasi berdasar tepi adalah mendeteksi suatu garis dengan anggapan setiap garis adalah batas dari tiap objek dengan objek lain atau objek dengan latar belakang. Sedangkan segmentasi daerah adalah segmentasi yang menentukan suatu daerah tertentu pada gambar yang diyakini merupakan objek yang akan dicari berdasarkan kesamaan tekstur.

Metode Tracking

Untuk mendapatkan data lokasi objek yang akan dijejaki, digunakan algoritma mean-shift yaitu merupakan algoritma teknik analisis non-parametric space atau analisis ruang fitur berbasis non-parametrik. Menurut Mahali (2014), Algoritma Mean-Shift klasik tidak dapat mengukur objek dengan perubahan skala dan orientasi sehingga diperlukan modifikasi dari algoritma Mean-Shift klasik. Sebelum menggunakan metode mean-shift diperlukan adanya representasi target model atau pemilihan model target dan representasi target kandidat. Representasi target model merupakan probabilitas dari histogram target yang ditentukan.

$$\hat{q}_u = C \sum_{i=1}^n k(\|x_1\|^2) \delta(b(x_1) - u)$$

{ } merupakan tempat dari piksel yang telah dinormalisasi dari target, dan nilai piksel pada lokasi C adalah merupakan normalisasi yang ditunjukkan oleh persamaan :

$$C = \sum_{i=1}^n k(\|x_1\|^2)^{-1}$$

Representasi target kandidat merupakan probabilitas histogram objek pada tiap frame yang memiliki panjang dan lebar sama dengan panjang dan lebar target model. Probabilitas dari target kandidat dinyatakan dalam persamaan :

Dimana C_h merupakan konstanta yang diperoleh dari persamaan :

$$C_h = \sum_{i=1}^n k(\|x_1\|^2)^{-1}$$

Representasi target model dan representasi target kandidat membantu dalam perhitungan lokasi y agar nilai mendekati nilai (y). Setelah lokasi saat ini diketahui (0) maka lokasi (1) dapat diketahui dengan persamaan berikut :

$$\hat{y}_1 = \frac{\sum_{i=1}^n x_1 \omega_1 g(\|y_0 - x_i\|)^2}{\sum_{i=1}^n \omega_1 g(\|y_0 - x_i\|)^2}$$

Dari persamaan-persamaan yang telah didapat, sebuah algoritma penjejakan objek dapat disusun sebagai berikut (Sulfan dkk, 2014) :

1. Menghitung target model
2. $k=0$
3. Menghitung probabilitas target kandidat
4. Menghitung bobot
5. Menghitung lokasi update
6. Mengatur nilai $d \| - 0 \|$, $1 = 0$, $= + 1$ Perhitungan

kernel density estimation (KDE) digunakan dalam algoritma mean-shift. Perhitungan menggunakan rumus sebagai berikut :

$$f(x) = \frac{1}{n} \sum_{i=1}^n K_H(x - x_i)$$

x = perulangan dari panjang n

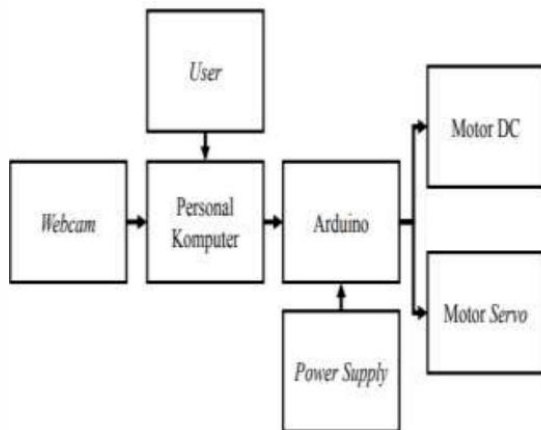
K = kernel simetrikal

H = parameter smooting

Fungsi dari KDE adalah untuk menghitung intensitas warna paling sering muncul pada lingkup tertentu. Pada suatu area yang didominasi warna tertentu, maka semua warna pada area tersebut akan berwarna yang mendominasi. Sehingga pada suatu citra akan terbagi menjadi area-area dengan warna yang homogen.

Diagram Blok Sistem

Diagram blok sistem pada penelitian ini ditunjukkan pada gambar dibawah ini

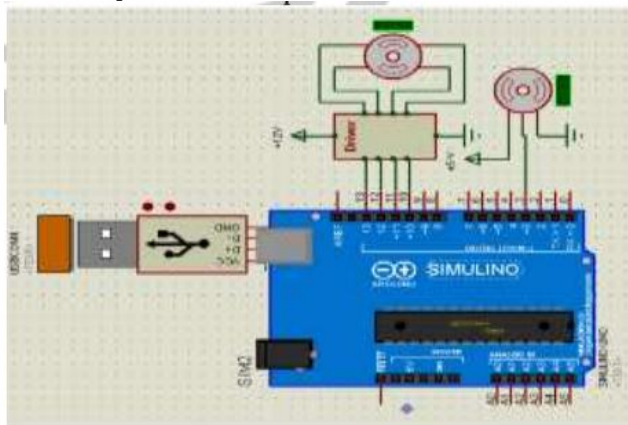


Gambar 11. Diagram blok sistem

Personal komputer yang ada pada robot akan merubah warna masuakn dari webcam yang mendapat pantulan dari cermin cembung. Dalam hal ini, objek yang harus dikalibrasi oleh user yauti warna bola, warna kawan (cyan/magenta) dan warna musuh Hasil pengolahan gambar yang berupa data-data koordinat dll dijadikan masukan oleh arduino untuk menggerakkan motor DC yang menjadi aktuator robot untuk melakukan tracking bola sampai bola berada pada penggiring robot.

Rangkaian Pengendali

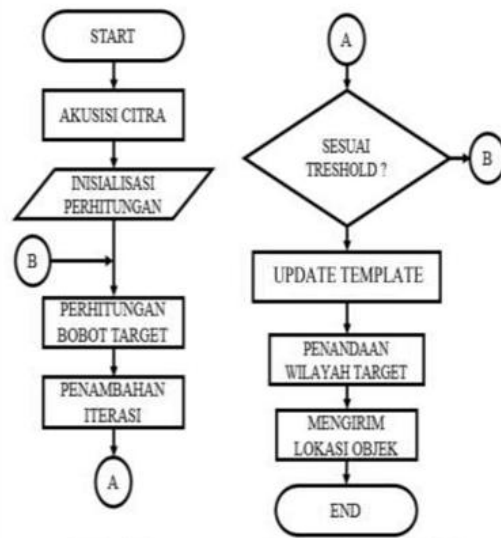
Arduino dihubungkan ke komputer melalui komunikasi serial. Agar dapat menggerakkan motor DC sebagai aktuator pada robot, setiap motor membutuhkan 3 pin pada arduino yang salah satunya harus menggunakan pin PWM sebagai pengatur kecepatan motor DC. Ground pada rangkaian Driver motor seluruhnya terhubung dengan arduino, sumber tegangan dari driver dan mikrokontroler terpisah. Catu daya mikrokontroler diperoleh dari laptop yang terhubung langsung dengan arduino. Hal ini terhubungnya usb arduino dengan laptop tidak hanya diperlukan sebagai catu daya tetapi juga sebagai komunikasi dengan visual basic. Sedangkan catu daya dari driver motor DC serta motor DC langsung bersumber dari baterai tanpa menambah konverter.



Gambar 12. Rangkaian pengendali motor dengan microcontroller

Perancangan Perangkat Lunak

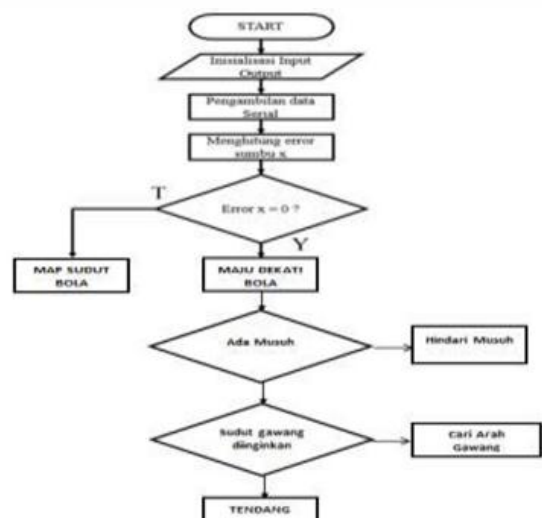
Flowchart utama pengolahan gambar



Gambar 13. Flowchart pengolahan gambar

Pada gambar 13, langkah awal yang dilakukan adalah akusisi citra dimana pengambilan gambar dilakukan secara real-time. Gambar yang diperoleh akan diproses sehingga diperoleh area yang merupakan gambar target. Thresholding dilakukan setelah objek yang dipilih telah ditentukan. Dari area objek yang terpilih, akan dilakukan perhitungan untuk mengetahui bobot objek sebagai acuan. Pada iterasi pertama, bobot target dihitung. Proses perhitungan bobot target dilakukan sampai iterasi telah dilakukan sampai selesai atau sampai nilai bobot target sudah tidak sesuai dengan nilai threshold. Setelah proses iterasi selesai, akan dilakukan perhitungan skala dan orientasi target menggunakan covarian matriks sehingga perluasan wilayah dapat ditemukan. Target yang telah ditemukan akan ditandai pada gambar dan lokasi koordinat objek dikirim ke mikrokontroler.

Flowchart pencarian Bola dan Strategi



Gambar 14. flowchart Tracking bola

Pada gambar diatas, flowchat tersebut merupakan diagram alir dari strategi pencarian bola serta pengambilan keputusan untuk menendang bola. Dimulai dari kalibrasi warna bola, kawan serta warna lawan. Setelah diperoleh warna biner yang akan dikirimkan ke mikrokontroler melalui komunikasi serial berupa nilai sudut bola dan jarak bola. Saat bola sudah berada pada penggiring robot, jika ada musuh maka robot akan menghindari berdasarkan warna biner dari musuh. Ketika tidak ada musuh pada algoritma yang kami gunakan adalah langsung mencari sudut dari gawang musuh dan pada saat sudah menemukan sudut gawang tersebut akan dilanjutkan keputusan untuk menendang bola

III. KESIMPULAN

Dari penelitian yang sudah dilakukan dapat ditarik kesimpulan bahwasannya menggunakan pantulan cermin cembung pada kamera akan menghasilkan pergerakan robot yang lebih baik, karena robot dapat melihat keseluruhan lapangan, namun pada jarak yang cukup jauh yaitu kurang lebih 5 meter kamera dapat menjangkau bola tetapi tidak dapat mendeteksi benda tersebut sebagai bola, karena pengaruh dari warna pada saat kalibrasi bola dan bola menjadi sangat kecil sehingga tidak dapat dirubah menjadi gambar biner.

IV. DAFTAR PUSTAKA

- [1] Faiq Aprilian R.(2018). Implementasi Object Tracking pada Kamera Pengawas Semi Otomatis dengan Image
- [2] Processing Metode Mean-Shift. Universitas Jember
- [3] Tri Bagus Rahmad K.(2013). Desain Pengolahan Citra Autonomus Mobile Robot Penjejak Bola Menggunakan Kamera. Universitas Jember
- [4] Intho Nurshauma.(2018). Robot Soccer Beroda Dengan Metode Behavior Based Control. Universitas Jember
- [5] Muhammad Zidny Naf'an, M Kom (2014) . Teknik Pengolahan Citra Operasi Pixel Dan Histogram.
- [6] Telkom University.
- [7] Aditya Wikan Mahastama. (2012). Pengolahan Citra Digital. Universitas Kristen Duta Wacana.
- [8] K.P. Gerrits (2012) Ball Handling System for Tech United Soccer Robot. University of Technology.
- [9] NuBot Team Description Paper 2009, RoboCup 2009 Graz, CD-ROM, Graz, Austria, July, 2009.
- [10] Tech United Eindhoven, Technische Universiteit Eindhoven: Tech United Eindhoven, <http://www.techunited.nl>, 2009.
- [11] S. Stancli_, Evolution of Active Dribbling Mechanisms in RoboCup, 16-741 Project, April, 2005.
- [12] S. Skogestad and I. Postlethwaite, Multivariable Feedback Control, Analysis and Design (2nd Edition), Wiley, ISBN 13-978-0-470-01168-3, 2007

Centralized System PDI Controller for Balancing of Alfarobi Humanoid Soccer Robot

Gabrielle Satya Mahardika, Bagaskara Primastya Putra, Ikrima Sabri, Tribagus Novandi Winantyo, Muhammad Haritsah Mukhlis, Lintang Erlangga, Teguh Nur Iman, Reynaldo, Rizal Mohamad, Muhammad Hadyan Akbar, Farchan Hakim Raswa, Arifandhi Nur Muhamad, Dimas Dian Pratama, Sahala Wahyu Wardana, Hanif Baskoro, Rizky Agung, Kadek Gemilang Santiyuda, and Dini Nur Anisa

Gadjah Mada Robotics Team

Universitas Gadjah Mada

Bulaksumur, Yogyakarta, Indonesia, 55284

gabrielle_sm_gsm@yahoo.co.id, primastyaputra@gmail.com

Abstract—Humanoid robot is the most difficult type of robot to control its balance. The 20 DoF (Degree of Freedom) joints enable the robot to walk and stabilize itself. The walk pattern of the robot is an open-loop control which can be unstable, so we need a control system to stabilize itself using closed-loop control. This paper explains about how to stabilize the 20 DoF humanoid robot with Centralized System of PDI (Proportional Derivative Integral) Controller. The main goal is to apply PDI controller to maintain the balance from the whole body system angle error for a stable walking motion. The true value of the whole body system angle error is obtained from trigonometry calculation of servo joint angles and links combined with filtered Inertial Measurement Unit (IMU) sensor data. This system angle error is the input for the PDI controller that can control the whole body of the humanoid robot.

Keywords—angle, control, PDI.

I. INTRODUCTION

In RoboCup Kid Size Humanoid League walking is the most important thing to be considered. There are many kinds of models, patterns, and control systems about humanoid robot walking. One of the easiest way to implement the walking of humanoid robot is by implementing DARwIn-OP humanoid robot platform. DARwIn-OP has a good walk pattern calculation of Zero Moment Point (ZMP) from its Center of Mass (CoM) based from its model. Alfarobi's mechanical design has a similar design to DARwIn-OP, but doesn't have the same mass and model, so we need to improve the stability of Alfarobi's system.

Since the new field is made of synthetic grass, it takes more effort to improve the walking ability for the robot to walk properly on the field. This year we developed a new stability control with Centralized System of PDI (Proportional Derivative Integral) controller. This stability control has a benefit of easily modelling and tuning the system.

Centralized System PDI Controller probably isn't the main solution to stabilize humanoid robot when walking on rough terrain, inclined surface, etc. There are many kinds of methods such as Kajita *et al.* 3D-Linear Inverted Pendulum Model as a method of walking control of a biped robot where we can analyze the dynamics of a three-dimensional inverted pendulum in which motion is constrained to move along an arbitrarily defined plane, Kajita *et al.* Preview Controller as the first, the dynamics of a biped robot is modeled as a running cart on a table which gives a convenient representation to treat ZMP, Goswami *et al.* the foot-rotation indicator (FRI) point, which is a point on the foot/ground-contact surface where the net ground-reaction force would have to act to keep the foot stationary. To ensure no foot rotation, the FRI point must remain within the convex hull of the foot support area. Ha *et al.* achieved balance walking based on coupled oscillators with DARwIn-OP robot. Yi *et al.* proposed a method of active stability and successfully implemented on DARwIn-OP robot. Missura *et al.* proposed a hierarchical, omnidirectional gait control framework that is able to counteract strong perturbations using a combination of step-timing, foot placement, and zero-moment-point strategies. Hirukawa *et al.* proposed a universal stability criterion of the foot contact of legged robots that checks if the sum of the gravity and the inertia wrench applied to the COG of the robot, which is proposed to be the stability criterion, is inside the polyhedral convex cone of the contact wrench between the feet of a robot and its environment. But all of the methods above need more modification to the framework and requires complex system modelling. Therefore, we implement a very simple feedback control to the framework.

The main topic of this paper is to introduce the stability control which is implemented on Alfarobi's robot. The stability control with Centralized System

of PDI Controller is used because the benefits of this control are easy whole body system modelling, only three parameter gains to tune, and easy to generate gain.

This stability control must have a good error data. The error data is obtained from the whole body angle of this system, which is combination between the IMU sensor data and the CoM angle error obtained from trigonometry calculation of servo joint angles and links. Before these data can be used on the controller, we need to filter the IMU data and calculate the whole body angle.

II. METHOD

A. True Angle Error Calculation

True angle error must be obtained from all sensors so that we can use it to estimate the angle of the whole body. In this case, we used trigonometry calculation to estimate the body angle and we used filtered IMU data to estimate the steady state error. The general expression of the angle calculation can be expressed as follows:

$$\begin{aligned}\theta_{Pitch}(t) &= \alpha_{Pitch}(t) + \beta_{Pitch}(t) \\ \theta_{Roll}(t) &= \alpha_{Roll}(t) + \beta_{Roll}(t)\end{aligned}$$

Eq. 1: Total body angle from trigonometry calculation & IMU sensor measurement

Where $\alpha(t)$ is the body angle from the trigonometry calculation from its feet-support to the CoM of the system and $\beta(t)$ is the filtered IMU data.

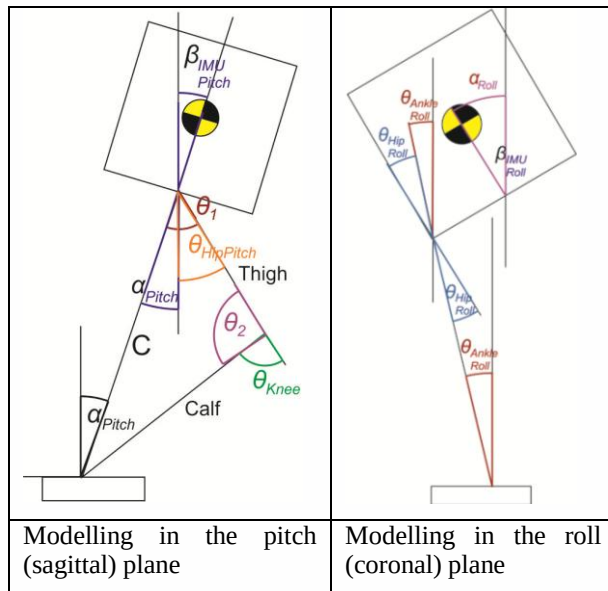


Fig. 1: Modelling the whole body angle of the robot using servo joint angles and links into trigonometry calculation

There are two calculations for the body angle error estimation. The first one is the pitch error of the body and the second one is the roll error of the body. The calculation of each error based from the feet support to the CoM point of the system.

$$\begin{aligned}C^2 &= Thigh^2 + Calf^2 - 2\cos(180^\circ - \theta_{knee}) \\ \frac{C^2}{C} &= \frac{Calf}{\sin(180^\circ - \theta_{knee})} = \frac{Calf}{\sin \theta_1} \\ \theta_1 &= \arcsin\left(\frac{Calf \cdot \sin(180^\circ - \theta_{knee})}{C}\right)\end{aligned}$$

$$\alpha_{pitch} = \theta_1 - \theta_{HipPitch}$$

$$\alpha_{roll} = \theta_{HipRoll} + \theta_{AnkleRoll}$$

Eq. 2: Body angle estimation from servo joint angles and lengths of links into trigonometry equations

For the IMU data, we read the raw accelerometer and gyroscope data from the CM730 Sub-controller board. The raw data of the sensor makes the robot generate some strange movement while walking at higher gain. Besides, we really need a higher gain level to generate an effective response. To obtain an effective and also an efficient response, we need to filter the raw data.

The gyro and the accelero is embedded on CM730 the sub-controller of the DARwIn-OP. The central-axis of the accelerometer and the gyroscope placement is shown in the figure below.

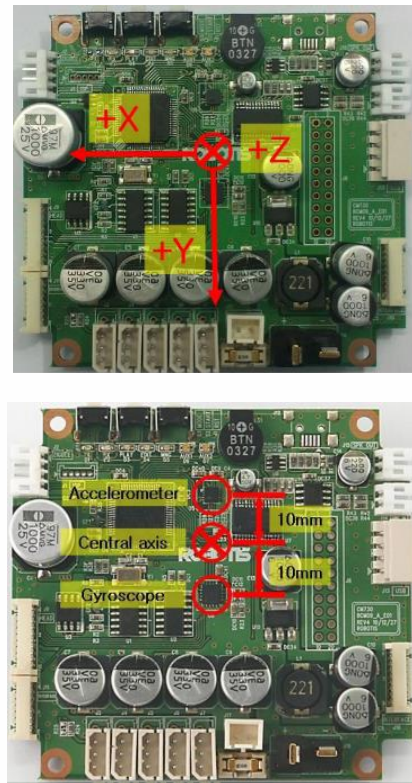


Fig. 2: The CM730 DARwIn-OP Subcontroller with the placement of the gyro and the accelero. The IMU has 6 degree of freedom, 3 at the accelero and 3 at the gyro.

B. PDI Controller

PDI (Proportional Derivative Integral) Controller is based from PID Controller, but we prioritize Derivative over Integral. We change the priority because we need to make the system have a quick response (faster rise time) rather than the stability

precision. We just applied the integrator for some case when steady state error occurs. The block diagram of PDI Controller for Alfarobi's system can be expressed follows:

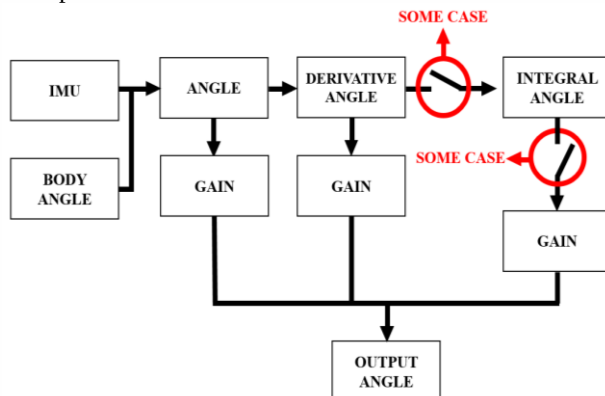


Fig. 3: The Block Diagram of the PDI Controller that is implemented in Alfarobi's system

C. Implementation to System

The output angle of the PDI Controller will be generated for each joint of the robot system. To quickly tune the controller, a terminal user interface has been used.

```
Pitch Kp      :7.2
Pitch Kd      :0.0052
Pitch Ki      :1.2e-05

Roll Kp       :5.3
Roll Kd       :0.002
Roll Ki       :1e-06

Filtered IMU:
Pitch  : 21.62367062
Roll   : -0.089773918

Pitch Correction : -21.6346
Roll Correction  : 0

Pitch Setpoint  :-0
Roll Setpoint   :0

True Error:
Pitch Error     : -0.011074315
Roll Error      : -0.089773918
```

Fig. 4: Terminal to tune gain values and output measured angles and calculated angle errors

Kp, Kd, and Ki are the proportional, derivative, and integral gains respectively. Filtered IMU angles are obtained from the fusion between accelerometer and gyro. Correction angles are obtained from trigonometry equation of the servo angles. Setpoint angles are the angles to be maintained by the controller. True error angles are the difference between sum of IMU & correction with the setpoint angles.

The goal is to obtain a zero true error by setting the setpoint to the most stable open loop position first, then introduce some perturbation by pushing the robot while it is walking to tune the Kp gain. If the

response is not fast enough, then Kd gain is increased. If there are steady state errors, then Ki gain is increased.

III. EXPERIMENT RESULT

We successfully implemented the Centralized PDI Controller while the robot is walking and the robot could maintain its setpoint angle. With this controller, we could flexibly tune the robot to respond to the error signal. In this experiment, we tuned the robot so that it can walk with a maximum speed of 30 mm per step, 47 mm foot step height amplitude, and a walking period of 660 ms. The experimental result is shown in the figure below.

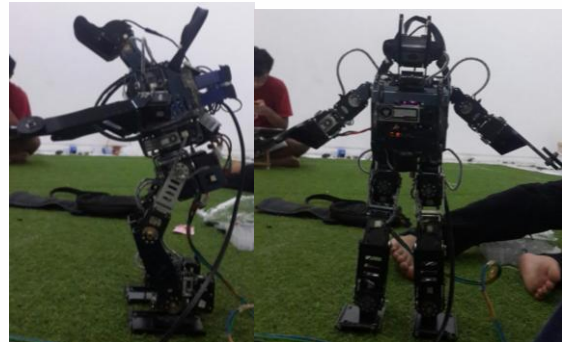
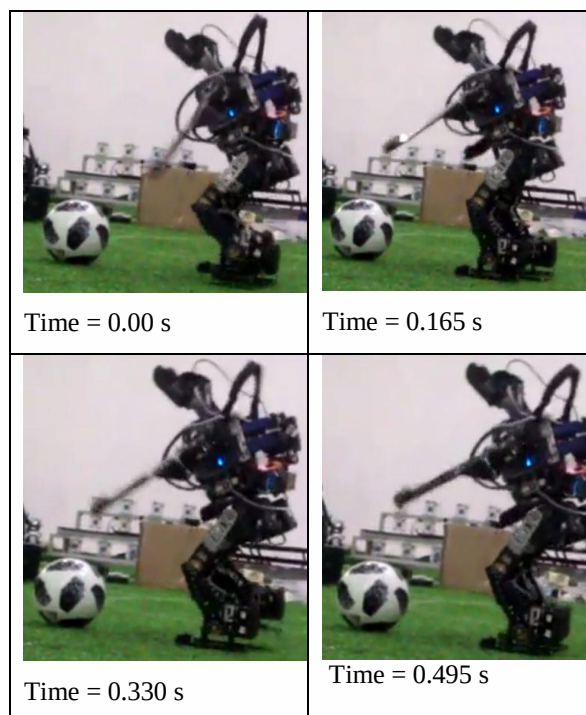


Fig. 5: Reference poses for pitch setpoint (left) and roll setpoint (right) at a standing still position.



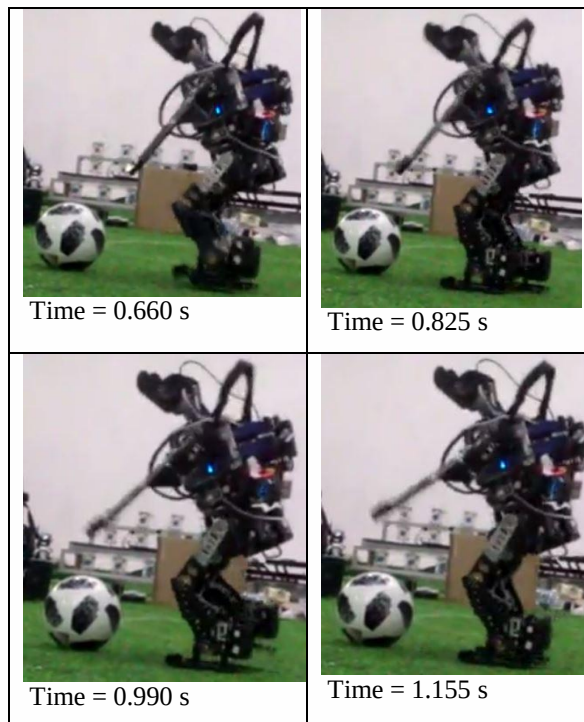


Fig. 6: Response of the robot every 165 ms while walking towards the ball.

In the figure above, the response of the robot while walking towards the ball is shown. It can be seen that the robot maintains its setpoint angle regardless of the synthetic grass terrain. At $t=0.165$ s and $t=1.155$ s pitch error angle is detected in the forwards $x+$ direction, it sways its arms forward and leans its hip and ankle backwards to counteract the error. The reverse action happens at $t=0.660$ s when pitch error is detected in the backwards $x-$ direction.

IV. CONCLUSION

We successfully implemented Centralized System PDI Controller for our robot platform. With the Centralized System PDI Controller, we can easily tune the gains of the feedback controller for walking on synthetic grass.

V. ACKNOWLEDGEMENT

We convey our gratitude to Universitas Gadjah Mada and all Gadjah Mada Robotics Team supervisors and members 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 2017 – 2018.

VI. REFERENCES

- [1] Williams, R. L.. DARwin-OP humanoid robot kinematics. In ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (pp. 1187-1196). American Society of Mechanical Engineers ; 2015, August.
- [2] Ha, I., Tamura, Y., Asama, H., Han, J., & Hong, D. W. Development of open humanoid platform DARwin-OP. In SICE Annual Conference (SICE), 2011 Proceedings of (pp. 2178-2181). IEEE; 2011, September.
- [3] Ha, I., Tamura, Y., & Asama, H. Development of open platform humanoid robot DARwin-OP. Advanced Robotics, 27(3), 223-232; 2013.
- [4] Galdeano, D., Chemori, A., Knit, S., & Fraisse, P. A nonlinear PID stabilizer with spherical projection for Humanoids: From concept to real-time experiments. In Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on (pp. 693-698). IEEE; 2014, November.
- [5] ROBOTIS Co, Ltd. Web Site, <http://support.robotis.com/en/product/darwinop/development/framework/reference/motion/modules/walking.htm>
- [6] ROBOTIS Co, Ltd. Web Site, <http://support.robotis.com/en/product/darwinop/development/framework.htm>
- [7] Ha, I., Tamura, Y., & Asama, H. Gait pattern generation and stabilization for humanoid robot based on coupled oscillators. In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on (pp. 3207-3212). IEEE; 2011, September.
- [8] Yi, S. J., Zhang, B. T., Hong, D., & Lee, D. D. Active stabilization of a humanoid robot for impact motions with unknown reaction forces. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on (pp. 4034-4039). IEEE; 2012, October.
- [9] DARwin-OP – SourceForge Web Site, darwinop.sourceforge.net
- [10] Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K., & Hirukawa, H. The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation. In Intelligent Robot and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on (Vol. 1, pp. 239-246). IEEE; 2001.
- [11] Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., & Hirukawa, H. Biped walking pattern generation by using preview control of zero-moment point. In Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on (Vol. 2, pp. 1620-1626). IEEE; 2003, September.
- [12] Goswami, A. Postural stability of biped robots and the foot-rotation indicator (FRI) point. The International Journal of Robotics Research, 18(6), 523-533; 1999
- [13] Missura, M., & Behnke, S.. Lateral capture steps for bipedal walking. In Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on (pp. 401-408). IEEE; 2011, October.
- [14] Hirukawa, H., Hattori, S., Harada, K., Kajita, S., Kaneko, K., Kanehiro, F., ... & Morisawa, M. A universal stability criterion of the foot contact of legged robots-adios zmp. In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on (pp. 1976-1983). IEEE; 2006, May.

Implementation of Stepping Strategy for Balance Recovery in EROS Humanoid Robot

Aulia Khilmi Rizgi¹, Ilham F. Arifin, Ryan S. Wijaya, M. Ayuf Basthomi, A. Habib Almutawakkil, Rokhmat Febrianto, Cipta Priambodo, Ibrahim Musthofainal, Miftahul Anwar, Anhar Risnumawan, A. Subhan Khalilullah

Laboratorium Robotika

Politeknik Elektronika Negeri Surabaya

Jl. Raya ITS Kampus PENS, Sukolilo, Surabaya

¹Email: auliakhilmirizgi@gmail.com

^{2,3}Email: {anhar,subhankh}@pens.ac.id

Abstract – This paper explains about the implementation of Stepping Strategy for Balance Recovery in EROS (EEPIS Robot Soccer) Humanoid Robot. In previous research, our robot had been able to run stable on a synthetic grass with a certain velocity. In the soccer game, the humanoid robot still can't avoid collision with another robot quickly. So, we need new control method that can remain humanoid in standing condition when the collision happened. One of strategy to realize that is Stepping Strategy. This method is to adjust step position of humanoid robot's leg when getting external perturbation to prevent robot in a standing condition. By reprocessing the inverted pendulum control formula, we get the relation between the angular acceleration and the step that the robot should perform. And the result, with Stepping Strategy EROS can prevent themselves from falling twice than not using the method.

Keyword: *Stepping Strategy, humanoid robot, inverted pendulum*

I. PENDAHULUAN

Pada perkembangan robot humanoid para peneliti berlomba-lomba menemukan cara untuk membuat robot humanoid layaknya seperti manusia. Begitu pula robot humanoid yang dikhususkan untuk bermain sepak bola, kemampuan seperti mengenali objek, menendang, berjalan, dan bangun dari jatuh harus ditanamkan pada robot.

Pada penelitian sebelumnya, robot EROS (EEPIS Robot Soccer) telah banyak mengalami perkembangan dalam metode dan kontrol berjalan. Dalam hal berjalan, kontrol kecepatan sudut telah berhasil diimplementasikan bersama dengan kontrol *trajectory joint*[1][2], optimasi selanjutnya juga telah dilakukan dengan mengontrol akselerasi dan deselerasi yang membuat robot lebih stabil saat melakukan pergerakan menuju kecepatan maksimal[3]. Penambahan optimasi saat melakukan landing juga telah diterapkan untuk mengurangi resiko kerusakan motor *servo* pada robot[4]. Dalam hal gerakan menendang juga telah mengalami pengoptimasian dengan menggunakan *joint trajectory controller*[5].

Dari hasil beberapa penelitian tersebut, robot EROS telah dapat berjalan dan menendang secara

stabil pada medan rumput sintetis dengan kecepatan tertentu. Akan tetapi robot masih belum dapat menjaga keseimbangan jika ada gangguan luar seperti dorongan yang dilakukan oleh robot lain saat bermain sepak bola. Dengan adanya permasalahan ini, kami mencoba memberikan tambahan kontrol yang berupa *Stepping Strategy* untuk mendapatkan kestabilan berjalan saat mendapat gangguan dorongan dari luar.

II. STEPPING STRATEGY

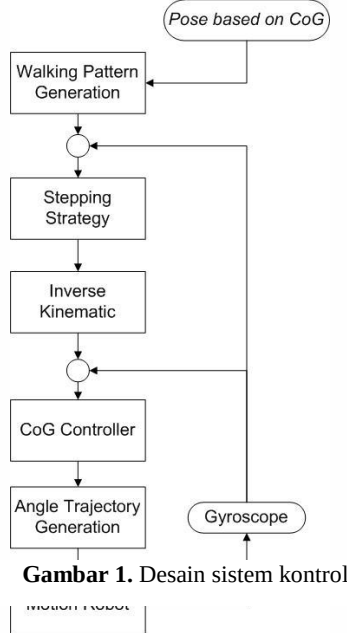
Robot EROS telah memiliki kemampuan menyeimbangkan diri dengan permodelan pendulum terbalik[2]. Robot akan tetap dalam kondisi berdiri dan tetap dalam kondisi seimbang saat pusat gravitasi dari robot masih berada dalam *Support Polygon Area*. Pusat gravitasi merupakan titik tengah antara pusat tekanan saat robot berdiri dengan *Double Support Phase* dan merupakan pusat tekanan saat *Single Support Phase*. Sedangkan *Support Polygon Area* merupakan area yang dimana robot masih dapat mempertahankan diri saat pusat gravitasi masih berada di dalamnya.

Saat robot mengalami gangguan dari luar, pusat tekanan dan pusat gravitasi pada robot berubah-ubah dan memungkinkan posisinya berada hingga keluar dari *Support Polygon Area*. Saat pusat gravitasi berada di luar dari *Support Polygon Area*, robot humanoid berada dalam kondisi sangat tidak seimbang dan memungkinkan untuk jatuh.

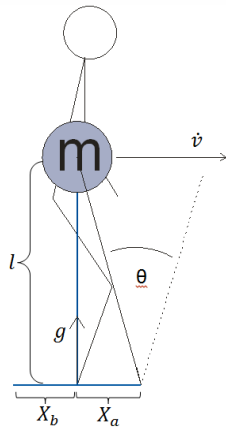
Stepping Strategy yang kami terapkan adalah untuk tetap menjaga keseimbangan robot saat pusat gravitasi berada pada titik kritis *Support Polygon Area* yaitu titik terluar *Support Polygon Area*.

Pada penelitian yang terkait, *Stepping Strategy* telah diterapkan dalam simulasi dengan kombinasi *hip, ankle, and stepping strategy* oleh Z. Aftab, dkk[6]. Penelitian tersebut memerlukan sensor penentu pusat tekanan pada telapak robot sebagai bagian dari formula yang ada. Karena tidak adanya sensor tersebut pada robot EROS, kami melakukan cara lain untuk menerapkan *Stepping Strategy* dengan algoritma yang sama.

Kami menggunakan sensor kecepatan posisi dan kecepatan sudut sebagai input dari *Stepping Strategy* yang kami terapkan. Nilai dari kecepatan posisi dan kecepatan sudut dikonversi melalui kontroler *fuzzy* dan mengeluarkan nilai berupa titik *trajectory* dari langkah kaki robot.



Gambar 1. Desain sistem kontrol



Gambar 2. Permodelan Pendulum Terbalik

Dengan mempertimbangkan kembali model pendulum terbalik yang diterapkan pada robot EROS[2] ketika berjalan, menghasilkan percepatan $\dot{v}$. Pada sudut pendulum dari arah vertikal, menghasilkan percepatan sudut $\ddot{\theta}_g = (g/L) \sin \theta$, dan akselerasi dari pergerakan robot menghasilkan $\ddot{\theta}_z = -(\dot{v}/L) \cos \theta$.

Di mana $L = \sqrt{l^2 + X_a^2}$. Kita dapat menghasilkan fungsi transfer $G(s)$, sebagai berikut :

$$\ddot{\theta} = \ddot{\theta}_g + \ddot{\theta}_z = (g/L) \sin \theta - (\dot{v}/L) \cos \theta.$$

$$L\ddot{\theta} - g\theta = -\dot{v}$$

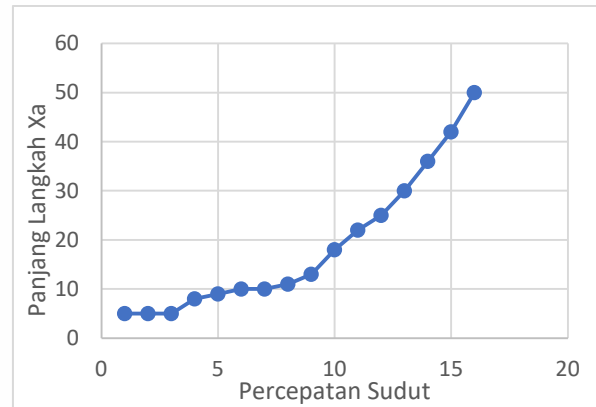
$$G(s) = \frac{\theta(s)}{v(s)} = \frac{-s}{Ls^2 - g} = \frac{-s/g}{(\tau_L s + 1)(\tau_L s + 1)}$$

Dari permodelan kontrol tersebut, didapatkan nilai sudut pendulum θ dan dikontrol agar tetap seimbang sehingga $\theta < \tan^{-1}(X_a/l)$.

Karena dalam konteks ini kami membatasi penelitian hanya untuk gerakan jalan ditempat dimana nilai X_a adalah 0. Sehingga berapapun nilai θ menghasilkan sekitar 0 derajat. Hal ini membuat hanya dengan sedikit gangguan luar dari robot dapat membuat robot menjadi tidak stabil dan jatuh.

Dari persamaan tersebut kami membandingkan percepatan sudut dan langkah yang harus dihasilkan saat mendapat gangguan, dengan nilai l konstan 210mm saat robot berdiri. Didapatkan gambar 2 yang merupakan grafik hasil hubungan $\ddot{\theta}_g$ dan X_a .

Dari grafik pada gambar 2 dilakukan pemetaan nilai untuk masing-masing nilai percepatan sudut dan mengkonversi menjadi nilai *trajectory* untuk mengubah panjang langkah yang dilakukan oleh robot.



Gambar 3. Grafik hubungan percepatan sudut dan panjang langkah

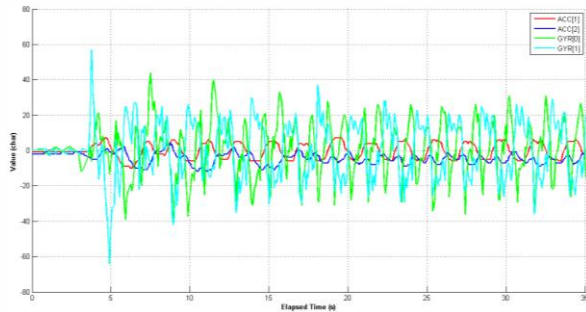
III. HASIL DAN DISKUSI

Dalam percobaannya, kami membagi 4 kondisi yang terjadi pada robot untuk mendapatkan hasil yang dapat dianalisa dengan baik yaitu, 1) robot berjalan di tempat tanpa gangguan dan tanpa *Stepping Strategy*, 2) robot berjalan di tempat tanpa gangguan dan dengan *Stepping Strategy*, 3) Robot berjalan di tempat dengan gangguan dan tanpa *Stepping Strategy*, dan 4) Robot berjalan di tempat dengan gangguan dan dengan *Stepping Strategy*. Gangguan yang diberikan berupa dorongan dengan gaya sekitar 15 m.a. Data yang kami dapatkan adalah kecepatan sudut dan kecepatan linear pada sumbu X dan Y.

Warna cyan menunjukkan kecepatan sudut sumbu X. Warna hijau muda menunjukkan kecepatan sudut sumbu Y. Warna merah menunjukkan kecepatan linear sumbu X dan warna biru menunjukkan kecepatan linear sumbu Y.

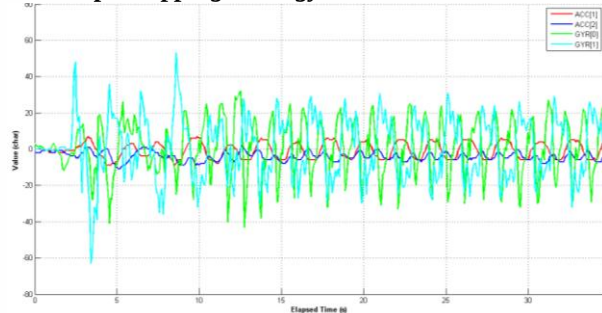
A. Robot berjalan ditempat tanpa gangguan dan tanpa *Stepping Strategy*

Saat robot berjalan ditempat robot bergetar stabil dan grafik percepatan sudut cenderung berulang ulang yang menandakan tidak adanya gangguan (Gambar 4).



Gambar 4. Grafik Sumbu X Y tanpa gangguan dan tanpa *Stepping Strategy*

B. Robot berjalan ditempat tanpa gangguan dan tanpa *Stepping Strategy*

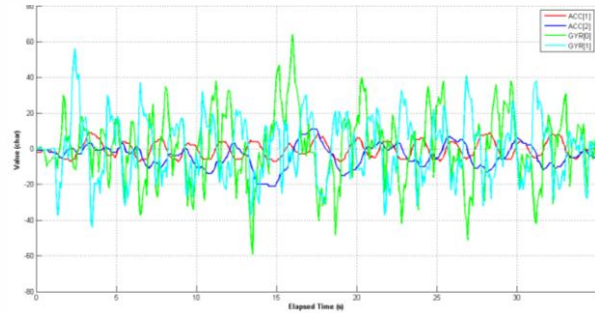


Gambar 5. Grafik Sumbu X Y tanpa gangguan dan dengan *Stepping Strategy*

Grafik yang hampir sama kami dapatkan untuk robot yang menggunakan *Stepping Strategy* dengan keadaan tanpa gangguan (Gambar 5). Grafik percepatan sudut cenderung berulang ulang menandakan sistem stabil saat melakukan gerakan jalan ditempat.

C. Robot berjalan ditempat dengan gangguan dan tanpa *Stepping Strategy*

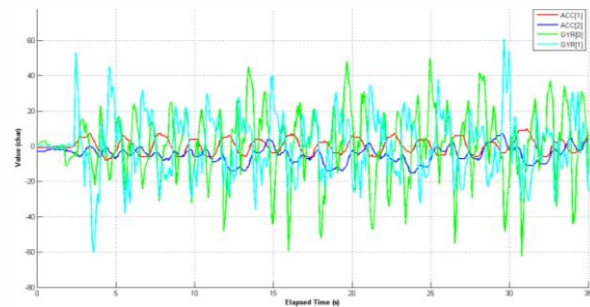
Pada percobaan ini, gangguan diterapkan. Dapat dilihat pada Gambar 6 bahwa perubahan nilai dari kecepatan sudut dan kecepatan linear lebih besar daripada tanpa gangguan dan grafik cenderung tidak berpola karena robot berusaha mempertahankan diri pada posisi yang tetap karena tidak menggunakan *Stepping Strategy*. Gerakan mempertahankan diri tanpa *Stepping Strategy* membuat tidak stabil dan menuju jatuh.



Gambar 6. Grafik Sumbu X Y dengan gangguan dan tanpa *Stepping Strategy*

D. Robot berjalan ditempat dengan gangguan dan dengan *Stepping Strategy*

Berbeda dengan grafik sebelumnya, grafik pada robot yang menggunakan *Stepping Strategy* saat mendapat gangguan cenderung lebih bergetar (Gambar 7). Hal ini dikarenakan robot tidak mempertahankan posisi melainkan berpindah posisi mengikuti gaya dorong yang diterima dengan mengubah langkahnya. Alhasil robot berjalan maju yang membuat nilai kecepatan sudut semakin besar.



Gambar 7. Grafik Sumbu X Y dengan gangguan dan dengan *Stepping Strategy*

Untuk menunjukkan hasil yang lebih signifikan, kami melakukan beberapa percobaan untuk membandingkan penggunaan *Stepping Strategy* dan tidak pada robot yang dalam kondisi mendapatkan gangguan. Dari percobaan tersebut kami mendapatkan hasil berikut.

Tabel 1. Percobaan penggunaan *Stepping Strategy* pada saat robot mendapat gangguan

Percobaan Ke-	Dengan <i>Stepping Strategy</i>	Tanpa <i>Stepping Strategy</i>
1	Maju tidak stabil	Jatuh
2	Maju tidak stabil	Jatuh
3	Maju stabil	Jatuh
4	Jatuh	Jatuh
5	Maju tidak stabil	tidak stabil
6	Maju tidak stabil	tidak stabil
7	Maju tidak stabil	Jatuh
8	Jatuh	Jatuh
9	Jatuh	tidak stabil
10	Maju tidak stabil	Jatuh

IV. KESIMPULAN

Stepping Strategy telah berhasil diterapkan pada robot EROS dengan hasil dua kali lebih baik daripada tidak menggunakan *Stepping Strategy*. Hal ini cukup membantu robot dalam menjaga keseimbangan saat mendapatkan gangguan eksternal.

V. PEKERJAAN SELANJUTNYA

Berdasarkan hasil tabel dapat disimpulkan bahwa *Stepping Strategy* yang kami terapkan masih belum sempurna karena metode yang diterapkan belum sepenuhnya stabil dalam mempertahankan kondisi dari jatuh. Oleh karena itu, kami akan melanjutkan penelitian untuk mendapatkan hasil yang lebih baik.

VI. DAFTAR PUSTAKA

- [1] Saputra, A., Khalilullah, A. (2013). Comparing Pose Control and Angular Velocity Control for Motion Balance Humanoid Robot Soccer EROS, EEPIS Robotic Research Center.
- [2] Saputra, A., Khalilullah, A. (2013). Implementasi Pembangkitan Pola Gerakan Berjalan Berbasis Trajectori Joint pada Robot EROS-1.1st *Indonesian Symposium on Robotics System and Control 2013*.
- [3] Saputra, A., Khalilullah, A., Sulistijono, I. (2014). Acceleration and Deceleration Optimization using Inverted Pendulum Model on Humanoid robot EROS-2. .2nd *Indonesian Symposium on Robotics System and Control 2014*.
- [4] Riananda, D., Zuhri, A., Qohar, H., Aminulloh, L., Ulil A., R., Suryanto, N., Saputra, A. and Khalilullah, A. (2015). Deceleration Optimization Landing for Trajectory Walking on Robot EROS 5. 3rd *Indonesian Symposium on Robotics System and Control 2015*.
- [5] Sucipto, A., Khalilullah, A., Risnumawan, A., Alasiry, A., Riananda, D., Fauzi, I., Rizgi, A., Wijaya, R., Syahputra, K., Arifin, I., Basthomi, M. and Almutawakkil, A. (2017). Increasing Stabiliy of Shooting Motion On EROS (EEPIS Robot Soccer) Using Joint Trajectory Controller. 5th *Indonesian Symposium on Robotic Systems and Control 2017*, 5, pp.208-211.
- [6] Aftab, Z., Robert, T. and Wieber, P. (2012). Ankle, hip and stepping strategies for humanoid balance recovery with a single Model Predictive Control scheme. 12th *IEEE-RAS International Conference on Humanoid Robots Nov.29-Dec.1, 2012*.

Pola Gerak Dinamis Pada Robot Krakatau FC

Achmad Fauzi, Andika Agashi, *Muhammad Reynaldo, Rendy Apriyanto
Universitas Teknokrat Indonesia
Bandar Lampung
krakatau.fc.uti@gmail.com

Absrak – KRSBI Humanoid setiap tahun selalu mengalami peningkatan tantangan. Perubahan lapangan yang semula menggunakan karpet hijau menjadi rumput sintetis pada tahun 2016 membuat robot kesulitan berjalan karena area pijak robot yang tidak datar. Pada tahun 2018 ukuran lapangan kembali diperbesar sehingga semakin menyulitkan robot berjalan. Untuk mengatasi tantangan tersebut, tim Krakatau menerapkan pola gerak dinamis berdasarkan arah kecenderungan gerak robot. Metode ini terbukti efektif untuk mengatasi masalah keseimbangan serta robot tetap dapat menjaga kecepatan berjalan sehingga dapat menyelesaikan misi dengan cepat.

I. PENDAHULUAN

Kontes Robot Sepakbola Indonesia 2016 yang mengadopsi peraturan Robocup 2015 menghadirkan tantangan baru berupa lapangan dengan rumput sintetis dengan ketebalan 3cm dan luas 6x4 meter. Pada tahun 2018 ukuran lapangan diperbesar menjadi 9x6 meter. Keberadaan rumput ini menjadi sebuah tantangan bagi robot terutama dalam hal penjaan keseimbangan robot. Robot menjadi mudah terjatuh karena perubahan tekanan pijak pada telapak kaki akibat dari daya dorong balik rumput. Sebagai baseline, Tim Krakatau menggunakan model gerak Darwin-OP dengan penyesuaian mengikuti struktur robot Krakatau. Model gerak tersebut terbukti berhasil pada tahun kontes 2015 dimana pada tahun 2015 lapangan masih menggunakan karpet berwarna hijau. Namun dengan adanya perubahan struktur fisik robot Krakatau disertai dengan adanya tantangan rumput sintetis, model gerak tersebut tidak lagi cukup. Berdasarkan baseline dan tantangan, Tim Krakatau secara lebih lanjut mengembangkan model gerak Darwin-OP agar mampu bergerak dinamis sesuai kebutuhan sekaligus tetap mempertahankan keseimbangan robot

II. METODOLOGI

A. Platform Robot

Robot sepak bola Krakatau FC dibangun dengan referensi desain DARWIN-OP. Robot memiliki 20 DOF, dimana 10 sendi tubuh bagian bawah digerakkan oleh kombinasi servo MX-64T dan MX-106T serta 10 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 eksternal robot didapat melalui



Gambar 1. Platform Robot Krakatau FC

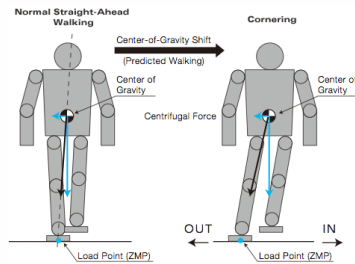
sensor passive berupa digital kamera PS Eye. Robot ini sendiri di otaki oleh komputer dengan processor Intel Cherry Trail yang bekerja pada clock 1.8 Ghz.

a. Model Gerak Darwin-OP

Darwin-OP menerapkan model *Zero-Moment-Point* (ZMP) untuk memprediksi peletakan setiap sendi robot dalam koordinat kartesian [1]. Model gerak dinamis ini memungkinkan cara berjalan yang dapat diatur dengan parameter sembari tetap menjaga titik gravitasi robot (*Center Of Gravity*). Model Darwin-OP juga dilengkapi kemampuan menyeimbangkan tubuh dengan mengubah offset actuator tubuh bagian bawah sesuai dengan galat yang didapat dari sensor gyroscope.

b. Zero Moment Point (ZMP)

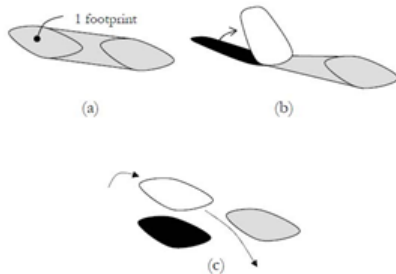
Adalah titik yang memiliki keseimbangan antara momentum yang bekerja pada robot dengan momentum yang dilawan oleh dasar tumpuan. Jika titik ini berada pada support polygon robot maka dapat dipastikan robot tidak akan jatuh [2]



Gambar 2. Ilustrasi Zero Moment Point

c. Support Polygon

Double Support Polygon adalah kondisi pada saat robot bertumpu pada kedua kaki nya tetapi tidak harus kedua permukaan kakinya menempel penuh pada dasar. Sedangkan Single Support Polygon adalah kondisi pada saat robot hanya bertumpu pada salah satu telapak kaki. [3]

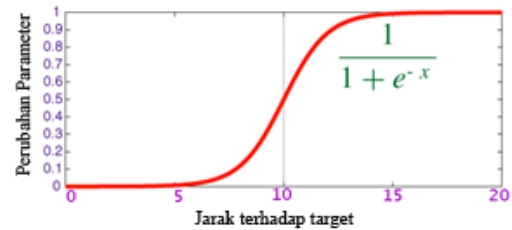


Gambar 3. Support Polygon (a) Double Support Polygon, (b) Double Support Polygon (Pre-Swing), (c) Single Support Polygon

d. Pola Gerak Dinamis

Seperti diulas pada pendahuluan, secara umum model gerak darwin-op bekerja baik pada arena pijak datar. Dengan adanya perubahan tekanan pada alas kaki terus menerus maka galat error menjadi semakin besar sehingga pada akhirnya robot akan terjatuh. Keseimbangan robot pada dasarnya dapat diperbaiki dengan meningkatkan rasio Double-Stances, sehingga robot lebih banyak berada pada tumpuan kedua kaki. Namun penggunaan tumpuan dua kaki mengurangi secara drastis jangkauan langkah robot sehingga memperlambat robot mencapai misi. Mengacu pada masalah diatas, maka tim Krakatau mengusulkan perubahan rasio double-stance yang mengikuti lebar langkah robot. Langkah ini diimplementasikan dengan membuat dua parameter batas minimum dan maksimum rasio double stance serta menerapkan fungsi sigmoid pada perubahan parameter.

Perubahan parameter yang tidak drastis terjadi ketika robot baru berjalan. Sedangkan pada pertengahan, perubahan parameter mulai naik



Gambar 4. Fungsi Sigmoid

drastis secara bertahap. Dan di akhir perubahan parameter kembali seperti semula.

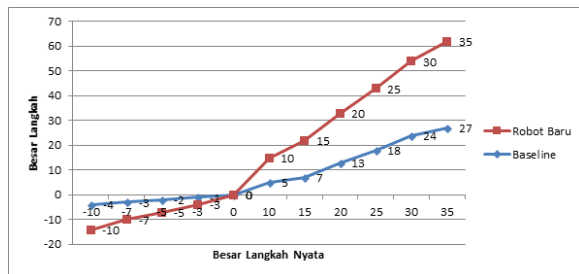
Permasalahan selanjutnya adalah pada parameter body-swing, nilai parameter yang terlalu tinggi membuat robot terlalu condong ke-kanan dan ke-kiri sehingga ketika sulit untuk menjaga robot tetap bergerak lurus, sementara itu nilai yang terlalu kecil akan membuat robot sulit untuk melangkah kesamping (kanan/kiri). Untuk mengatasinya, tim Krakatau kembali menerapkan parameter dinamis yang ditentukan berdasarkan arah gerak robot yang ditentukan dari kecenderungan pergerakan robot yang diinginkan, apakah bergerak ke depan atau bergerak ke samping. Usaha terakhir untuk menjaga keseimbangan robot adalah menggunakan tubuh bagian atas sebagai sarana bantu keseimbangan. Tim Krakatau menerapkan offset pada motor bahu dan lengan untuk membantu menstabilkan robot sesuai dengan galat dari gyroscope.

III. HASIL DAN PEMBAHASAN

Untuk menguji hasil modifikasi model gerak Darwin-OP, dilakukan dua pengujian yaitu untuk menguji kecepatan maksimal robot berjalan, dan efektivitas gerak kanan-kiri dan maju-mundur robot. Kedua pengujian tersebut didalamnya diintegrasikan penyeimbangan robot dengan bahu dan lengan robot. Pengujian pertama dilakukan dengan mengubah besaran langkah kaki untuk melihat kecepatan maksimal yang mampu dicapai robot. Berikut adalah table dan grafik perbandingan pengujian parameter :

Tabel 1. Tabel uji coba perbandingan langkah depan dan belakang (Step forward/back)

No	Besar Langkah(mm)	Besar Langkah Nyata (mm)		Kestabilan	
		Baseline	Usulan	Baseline	Usulan
1	-10	-4	-10	Tidak Stabil	Stabil
2	-7	-3	-7	Tidak Stabil	Stabil
3	-5	-2	-5	Kurang Stabil	Stabil
4	-3	-1	-3	Kurang Stabil	Stabil
5	0	0	0	Stabil	Stabil
6	10	5	10	Stabil	Stabil
7	15	7	15	Stabil	Stabil
8	20	13	20	Kurang Stabil	Stabil
9	25	18	25	Kurang Stabil	Stabil
10	30	24	30	Tidak Stabil	Stabil
11	35	27	35	Tidak Stabil	Stabil



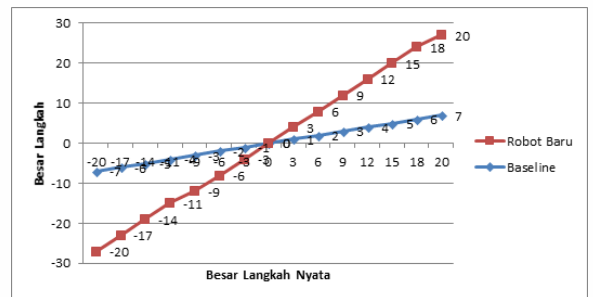
Gambar 5. Grafik uji coba perbandingan langkah depan dan belakang (Step forward/back)

Dari tabel tersebut dapat dilihat bahwa dengan modifikasi yang dilakukan tim Krakatau secara efektif dan efisien meningkatkan lebar langkah yang berimbas pada kecepatan maksimal robot berjalan. Besaran langkah dapat diterjemahkan sistem untuk mengubah kecepatan jalan secara nyata sekaligus menjaga robot tetap seimbang.

Pengujian kedua dilakukan dengan mengubah besaran langkah ke kanan dan kiri robot, berikut adalah tabel dan grafik hasil pengujian parameter :

Tabel 2. Tabel uji coba perbandingan langkah kiri dan kanan (Step left-right)

No	Besar Langkah(mm)	Besar Langkah Nyata(mm)		Kestabilan	
		Baseline	Usulan	Baseline	Usulan
1	-20	-7	-20	Tidak Stabil	Stabil
2	-17	-6	-17	Tidak Stabil	Stabil
3	-14	-5	-14	Kurang Stabil	Stabil
4	-11	-4	-11	Kurang Stabil	Stabil
5	-9	-3	-9	Stabil	Stabil
6	-6	-2	-6	Stabil	Stabil
7	-3	-1	-3	Stabil	Stabil
8	0	0	0	Stabil	Stabil
9	3	1	3	Stabil	Stabil
10	6	2	6	Stabil	Stabil
11	9	3	9	Stabil	Stabil
12	12	4	12	Kurang Stabil	Stabil
13	15	5	15	Kurang Stabil	Stabil
14	18	6	18	Tidak Stabil	Stabil
15	20	7	20	Tidak Stabil	Stabil



Gambar 6. Grafik uji coba perbandingan langkah kiri dan kanan (Step left/right)

Hasil diatas menunjukkan perubahan signifikan pada gerak kanan-kiri robot. Robot dapat melangkah kesamping dengan tetap menjaga keseimbangan robot sekaligus menjaga pergerakan maju-mundur.

IV. KESIMPULAN

Berdasarkan pengujian dapat disimpulkan bahwa usulan perubahan model gerak oleh tim Krakatau terhadap baseline terbukti berhasil sesuai dengan kebutuhan sehingga robot dapat bergerak dinamis dengan tetap menjaga keseimbangan robot.

V. REFERENSI

- [1] J. Rudy, "Zero Moment Point Walking Controller for Humanoid Walking Using Darwin-OP," University of Notre Dame, 2014.
- [2] T. Sugihara, Y. Nakamura, and H. Inoue, "Realtime Humanoid Motion Generation through ZMP Manipulation based on Inverted Pendulum Control."
- [3] A. S. Z. Dimas Pristovani Riananda, Achmad Subhan Kh, "Deceleration Optimization Landing For Trajectory Walking On Robot EROS 5," *ISRSC 2015*, vol. 1, 2015.

Sistem Penendang Menggunakan Pegas Pada Robot Sepak Bola

Muhammad Ivan , Andrea Simon, Jepi Dwiansyah, Imam Jurjawi, Chairul Fahmi Nasution, Indra Aulia, Samuel Simorangkir, Abadi Sanjaya, Muhammad Shobirin Irsan
TIM Si-Hatop, Universitas Sumatera Utara, Medan
muhammadivan111@gmail.com

Abstrak – Robot Sepak Bola merupakan robot yang mampu menendang bola ke arah gawang lawan sehingga menciptakan gol. Untuk dapat menendang dengan baik maka diperlukan sistem penendang yang baik pula. Salah satunya yaitu menggunakan sistem pegas. Pegas dihubungkan ke tungkai atas penendang agar tungkai dapat bergerak bebas saat menendang bola. Agar tungkai penendang dapat bergerak saat ditarik maupun didorong maka digunakanlah *bearing* sebagai poros tungkai lengan agar penendang dapat bergerak mulus. *Wiper motor* yang terhubung dengan gear digunakan untuk memutar gear. Gear dan tungkai penendang dihubungkan oleh lengan sehingga tungkai penendang tertarik oleh gear. Karena gear telah dihubungkan pegas maka pegas akan merenggang karena tertarik oleh tungkai atas penendang dan pegas akan menyimpan energi potensial. Energi potensial ini nantinya akan dikonversi menjadi energi gerak sehingga penendang dapat menendang bola dengan kuat. Semakin besar konstanta pegas maka energi potensial yang disimpan pegas akan semakin besar .

Kata kunci: Robot Sepak bola; Penendang; Pegas

I. PENDAHULUAN

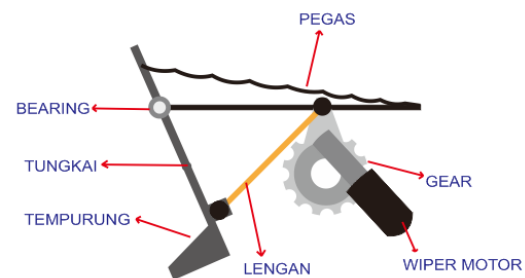
Kontes Robot Indonesia (KRI) 2018 divisi Sepakbola Beroda (KRSBI Beroda) merupakan ajang kompetisi yang diselenggarakan oleh Kementerian Riset Teknologi dan Pendidikan Tinggi (KEMENRISTEKDIKTI). Kontes ini diadakan untuk meningkatkan skill, kemampuan, dan kreatifitas para mahasiswa dalam bidang robotika khususnya divisi sepak bola beroda.

Robot Sepak Bola Beroda merupakan robot yang didesain dapat bermain bola dengan ketentuan KRI. Untuk memenangkan pertandingan dalam kontes robot sepak bola maka robot harus mencetak gol lebih banyak dari lawan. Agar tujuan robot tercapai maka diperlukan sistem penendang yang baik.

Sistem penendang menggunakan pegas digunakan karena kekuatan tendangan yang dihasilkan hampir sama setiap kali menendang.

II. METODE

A. Konstruksi Penendang



Gambar 1 Konstruksi robot penendang

Gambar 1 menunjukkan konstruksi robot penendang pada robot. Konstruksi robot ini terdiri atas tungkai penendang, *wiper motor*, dan gear.

B. Material yang Terdapat di Konstruksi Penendang

B.1. Tungkai Penendang



Gambar 2 Tungkai Penendang

Tungkai penendang pada robot digunakan untuk menghubungkan antara tempurung penendang dengan pegas. Tungkai penendang menggunakan 2 bahan besi hollow yang dikopel menjadi satu yang ujungnya diberi besi padat. Ujung atas tungkai terhubung dengan pegas dan ujung bawahnya terhubung dengan plat besi yang terhubung dengan gear.

B.2. Tempurung Penendang



Gambar 3 Tempurung Penendang

Tempurung penendang merupakan material yang bertumbukan langsung dengan bola. Tempurung penendang berbentuk setengah trapesium dan menggunakan 1 plat besi yang dilaskan ke tungkai penendang.

B.3. Wiper Motor



Gambar 4 wiper motor

Wiper yang merupakan motor DC digunakan untuk menarik penendang ke posisi sedia. Wiper yang digunakan bertegangan 24 volt. Wiper dihubungkan di Gear.

B.4. Gear



Gambar 5 Gear

Gear pada robot berguna untuk mentransfer putaran wiper motor untuk menarik penendang ke posisi sedia menendang. *Gear* yang digunakan pada robot adalah jenis *gear* tarik sepeda. *Gear* ini didesain untuk bergerak satu arah tetapi tidak dapat bergerak dengan arah sebaliknya.

B.5. Pegas



Gambar 6 Pegas

Pegas pada robot digunakan sebagai penyimpan energi potensial sebelum robot menendang bola. Pada Gambar 6 pegas dalam keadaan merenggang berarti pegas bersiap-siap melepaskan tendangan sedangkan apabila pegas merapat berarti pegas sudah melepaskan tendangan pada bola.

B.6. Bantalan (*bearing*)



Gambar 7 Bearing

Bantalan digunakan sebagai poros putar tungkai penendang, bantalan juga berfungsi untuk tungkai penendang dapat bergerak mulus tanpa adanya gesekan dan membuat tendangan stabil pada porosnya.

C. Prinsip Kerja

Mekanisme penendang bola pada robot terdiri dari dua kondisi yaitu pada saat posisi sedia dan posisi setelah menendang.

Mula-mula *wiper motor* akan memutar *gear* sehingga membuat lengan tertarik, kemudian pegas akan meregang dan akan berhenti sampai pegas tertarik maksimum. Pada saat ini pegas akan menyimpan energi sebesar :

$$E_{Pegas} = \frac{1}{2} \times K \times \Delta x^2 \quad (1.1)$$

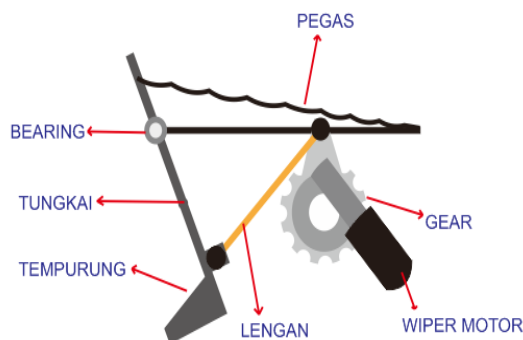
Ukuran kekuatan pegas diukur berdasarkan besarnya konstanta pegas. Untuk mengukur konstanta pegas, maka digunakan persamaan:

$$F = k \cdot \Delta x$$

$$k = \frac{F}{\Delta x}$$



Gambar 8 posisi Penendang Sedia



Gambar 9 posisi penendang sedia

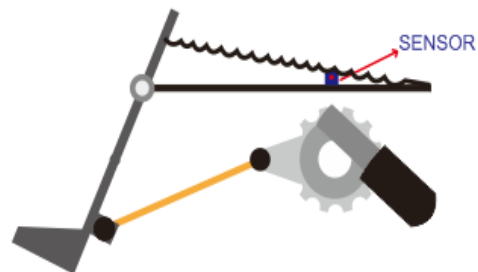
Pada gambar 7 dan 8 menunjukkan kondisi sedia yaitu saat robot belum menendang bola. Karena *gear* hanya berputar satu arah sehingga *gear* mampu menahan gaya dari pegas agar tidak terputar balik.

Ketika robot sudah mendapatkan bola dan berada pada posisi untuk menendang maka *wiper motor* akan memutar *gear* sampai melewati sensor. Setelah itu energi pada pegas akan dilepas dan menarik tungkai atas penendang sehingga tempurung menumbuk bola yang menyebabkan bola bergerak sesuai arah tendangan. Pada gambar 9 dan 10

menunjukkan penendang sudah menendang bola.



Gambar 10 posisi penendang setelah menendang



Gambar 11 posisi penendang setelah menendang

Dalam penggunaannya sistem penendang ini menggunakan input tegangan pada motor sebesar 24 volt. Untuk menandakan posisi pegas maksimum sistem ini menggunakan sensor inframerah yang datanya akan dikirim ke mikrokontroler dimana mikrokontroler akan memberi instruksi kepada motor untuk berhenti.

Saat robot siap untuk menendang mikrokontroler akan memberikan intruksi kepada motor untuk berputar sampai mikrokontroler menerima data kembali dari sensor.

III. HASIL DAN DISKUSI

A. Analisa Pengukuran Konstanta Pegas

Pengukuran konstanta pegas diperlukan untuk mengetahui karakteristik suatu pegas seperti yang ditunjukkan pada tabel 1 menggunakan rumus:

$$k = \frac{mg}{\Delta x}$$

Karakteristik pegas ini perlu diketahui agar bisa mengetahui seberapa kuat pegas apabila pegas merenggang atau merapat.

Tabel 1 Pengujian Konstanta Pegas

m(kg)	$\Delta x(m)$	$k(N/m)$
1,33	0,11	118,49
1,420	0,12	115,96
1,625	0,14	115,5
Rata-rata konstanta pegas		116,55

B. Analisa Pengukuran Banyak Pegas Terhadap Jarak Jangkauan Tendangan Bola

Percobaan ini diperlukan untuk menghitung seberapa jauh tendangan robot oleh bola berdasarkan banyaknya jumlah pegas yang digunakan dan disesuaikan dengan kekuatan *motor wiper*.

Dalam percobaan ini, yang dilakukan adalah membandingkan jarak awal bola dengan robot sebelum menendang dengan jarak bola dengan robot setelah bola ditendang. Di percobaan ini juga membandingkan kekuatan jumlah pegas terhadap jarak jangkauan bola setelah robot menendang bola.

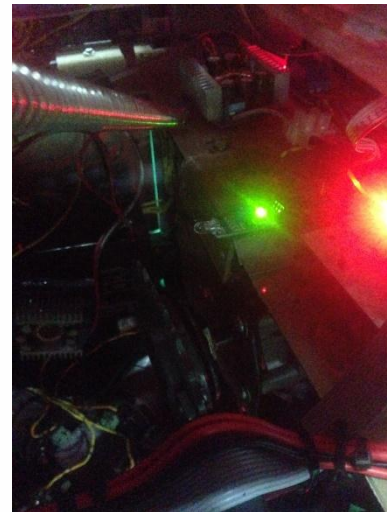
Tabel 2 Pengukuran Jarak Jangkauan Tendangan Bola

PERCOBAAN	2 Pegas		3 Pegas	
	Panjang Awal (m)	Panjang Setelah Ditendang (m)	Panjang Awal (m)	Panjang Setelah Ditendang (m)
1	2,93	2,78	2,94	3,16
2	2,93	2,72	2,94	3,16
3	2,93	2,73	2,94	3,24

C. Sensor Inframerah

Pengujian sensor inframerah pada penendang diperlukan untuk mendeteksi apakah penendang dalam posisi sedia menendang atau tidak. Sensor inframerah diletakkan di dekat gear kayuh sepeda. Pembacaan sensor inframerah ditunjukkan pada Tabel 2.

Pada sensor inframerah terdapat 2 lampu yang memberitahukan pada penendang apakah sudah dalam posisi sedia atau melakukan tendangan ke bola. Apabila ada 2 lampu yang hidup pada sensor inframerah maka penendang dalam posisi sedia. Sedangkan apabila hanya 1 lampu yang hidup maka penendang sudah melakukan tendangan ke bola.

**Gambar 12** keadaan sensor dalam posisi sedia**Gambar 13** Keadaan sensor saat melakukan tendangan ke bola**Tabel 2** Pembacaan sensor inframerah

Nilai	Pembacaan Sensor	Keterangan
0	Detect	Penendang sudah di posisi sedia
1	No-detect	Penendang melakukan tendangan ke bola

IV. KESIMPULAN

Paper ini menjelaskan tentang sistem penendang mekanis dengan menggunakan pegas pada robot sepak bola beroda.

Tujuan dari studi ini adalah untuk mendapatkan kekuatan tendangan yang besar dan konstan. Berdasarkan studi semakin besar konstanta pegas maka akan semakin besar energi yang dapat disimpan oleh pegas. Namun gaya pegas yang terlalu kuat juga

dapat menyebabkan motor tidak mampu untuk menarik pegas.

V. DAFTAR PUSTAKA

- [1] RISTEKDIKTI,(2018), "Buku Panduan Kontes Robot Sepakbola Indonesia Divisi Beroda (Krsbi Beroda) 2018," Jakarta:RISTEKDIKTI
- [2] Vitaloka,Dwi Candra,Moh Hasan,Rusli Hidayat,(2013)," Simulasi Sistem Pegas Massa,"Kadikma, 4(1): 115-128
- [3] Surya, Yohanes, 2009 "Mekanika dan Fluida Edisi 1," PT. Kandel,
- [4] Sears dan Zemansky, 2002, "Fisika Universitas Edisi Sepuluh Jilid I," Erlangga,
- [5] Cohen Michael, 2011, "Classical Mechanics," University of pennsylvania,

Implementasi *Hough Circle, Contours and Convex Hull Method* untuk *Humanoid Robot Soccer* dalam Deteksi Bola Putih Berdasarkan Rule KRSBI Humanoid 2018

Lukas Septa Hernanda⁽¹⁾, Chrysna Aji Prastio⁽²⁾, Ruli Refpalda⁽³⁾, Yudi Aprilianto⁽⁴⁾, Eki Faqih Akbaruddin⁽⁵⁾,
Faqih Dinulloh Nurul Anam⁽⁶⁾, Riska Serlia Kurniawati⁽⁷⁾, Silvia Ramadhani⁽⁸⁾, Putri Wulandari⁽⁹⁾ dan
Muslikhin⁽¹⁰⁾

Universitas Negeri Yogyakarta
Kampus Karangmalang, Jl. Colombo No 1 Yogyakarta
⁽¹⁾lukas.septa2016@student.uny.ac.id

Abstract—*The Difficulty of ball detection for humanoid robot soccer is noise, light intensity, white lines and shapes. Therefore, in order to get a better accuracy, it needs to develop the detection method based on image processing. The alternative detection method that is seen accurately with Hough Circle, Contours and Convex Hull Algorithm. This journal will be explained about combination of the three methods. The algorithm is tested in Humanoid Robot Soccer with the rule of KRSBI Humanoid Indonesia 2018. Generally, the combination of implementation of the three algorithms can detect the white ball with a minimum error result.*

Keyword : *OpenCV, Hough Circle, Contours, Convex Hull, Robot*

I. PENDAHULUAN

Prinsip dasar pendeteksian bola pada robot humanoid soccer yakni dengan menggunakan pencitraan kamera. OpenCV merupakan sebuah API (*Application Programming Interface*) library yang sudah sangat familiar pada Pengolahan Citra Computer Vision yang dapat digunakan dalam bahasa pemrograman C, C# ataupun C++, yang di desain sebagai aplikasi real-time yang memiliki fungsi-fungsi akuisisi yang baik untuk *image* ataupun *video*. OpenCV sebagai Inovasi yang lebih baik akan membantu *humanoid robot soccer* untuk mengoptimalkan performa dalam mencetak gol di lapangan. Oleh karena itu, diperlukan metode yang akurat untuk mendeteksi dan mengenali bola pada *humanoid robot soccer*. Terdapat berbagai macam metode yang ada dan support bagi library OpenCV dalam berbagai macam bahasa pemrograman. Sehingga diperlukan kejelian dalam pemilihan metode dalam melakukan filterisasi pendeteksian dan pengenalan bola.

Tujuan dari pembuatan paper ini adalah untuk membahas bagaimana tingkat keakuratan pendeteksian bola melalui penggunaan metode pada sistem pencitraan kamera. Penggunaan metode ini

dapat menentukan algoritma program robot, sehingga *humanoid robot soccer* dapat melakukan pendeteksian bola secara lebih optimal.

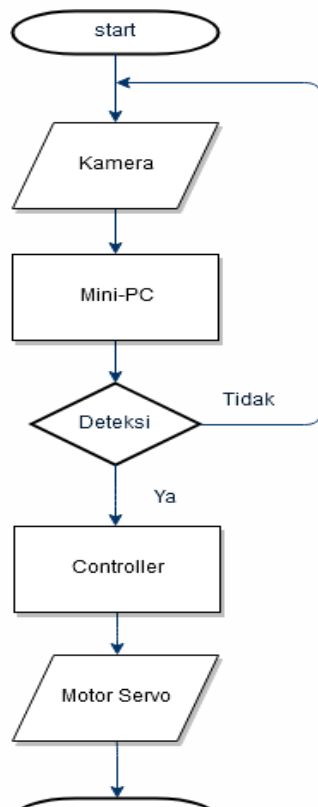
II. METODE

Sistem perancangan yang digunakan pada robot Al-‘Aadiyaat merupakan sebuah sistem yang dibuat dengan tujuan untuk mengendalikan robot dalam menyelesaikan misi perlombaan sepak bola pada Kontes Robot Sepak Bola Indonesia. Secara umum sistem yang dirancang terdiri sebagai berikut :

A. Sistem Kontrol

Susunan bagian komponen yang berfungsi sebagai sistem kontrol yang ada pada robot Al-‘Aadiyaat ditunjukkan pada gambar 1.

Berdasarkan Gambar. 1, Diagram Sistem Kontrol Robot tersebut menjelaskan bahwa, robot mengenali bola melalui kamera, kemudian gambar dari kamera tersebut diolah dan dikirimkan pada MINI-PC didalam MINI-PC gambar tersebut di proses dan diolah menggunakan Library OpenCV sehingga didapatkan hasil algoritma bola putih yang dapat terdeteksi. Hasil algoritma yang telah diolah di MINI-PC tersebut kemudian diteruskan ke Controller untuk menggerakkan motor servo pada robot sehingga robot melakukan gerakan yang dikehendaki.

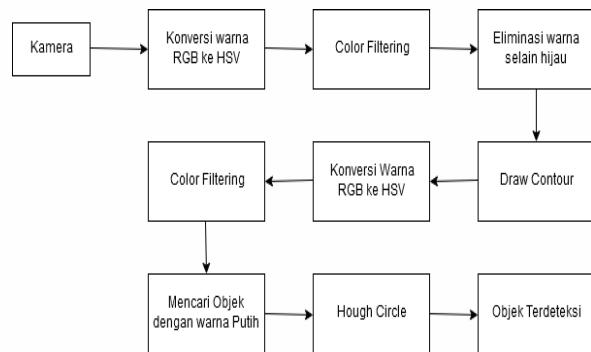


Gambar. 1 Diagram Sistem Kontrol Robot

A. Sistem Navigasi

Sistem navigasi pada robot ini menggunakan sensor kamera yang berfungsi menyerupai indera penglihatan manusia untuk mendeteksi benda berdasarkan warna dan bentuk. Kamera yang digunakan dalam robot ini adalah webcam dengan seri Logitech C525. Webcam keluaran Logitech ini hadir dengan dukungan video High Definition full HD 720P layar lebar untuk panggilan video. Webcam ini dilengkapi dengan lensa tajam buatan Carl Zeiss-Tessar dengan fungsi fokus otomatis. Dengan adanya fungsi ini, penglihatan robot akan lebih jernih sehingga dapat membedakan objek dengan lebih baik.

Pada sistem navigasi robot, bola didefinisikan sebagai benda dengan warna dan bentuk tertentu, dengan algoritma khusus, maka robot akan terus mencari kesegala arah, apabila belum mendeteksi keberadaan bola putih. Setelah bola terdeteksi, selanjutnya robot akan mendekati bola, mengarahkan bola, kemudian menendangnya ke arah gawang lawan. Sistem navigasi robot Al-'Aadiyaat dapat disajikan dalam bentuk diagram dibawah ini.



Gambar. 2 Sistem Navigasi Pengenalan dan Pendeteksian Bola Putih.

OpenCV

OpenCV adalah sebuah *open source library* untuk *computer vision* yang dibuat dengan menggunakan bahasa pemrograman C, C# ataupun C++ yang dapat dijalankan di berbagai sistem operasi mulai dari Linux, Windows dan Mac OS X. Arsitektur dan manajemen memori yang dimiliki oleh OpenCV memungkinkan untuk menjalankan proses dan perhitungan yang kompleks dengan sumberdaya yang terbatas sehingga memberi keleluasaan bagi penggunaanya untuk membangun algoritma pengolahan citra baik itu dengan menggunakan masukan berupa gambar ataupun video tanpa harus khawatir dengan proses alokasi dan dealokasi memori.^[1]

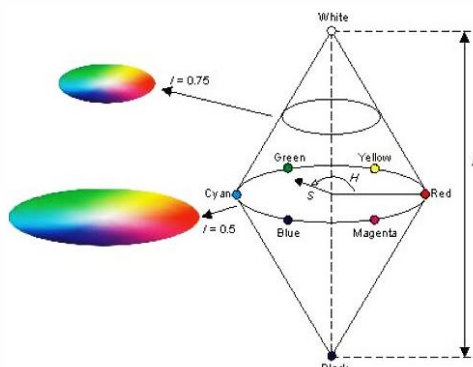
Ruang Warna RGB

Ruang warna RGB (Red, Green dan Blue) merupakan kombinasi dari warna primer yaitu merah, hijau dan biru, yang biasa digunakan oleh monitor komputer atau televisi. Warna yang dihasilkan berasal dari kombinasi tiga warna dan masing – masing memiliki nilai 8 bit merah, 8 bit hijau dan 8 bit biru. Campuran ketiga warna primer tersebut dengan proporsi seimbang akan menghasilkan nuansa warna kelabu. Sedangkan, jika ketiga warna ini disaturasikan penuh akan menghasilkan warna putih (Pratt, 2007).^[2]

Ruang Warna HSV

Model HSV terdiri dari tiga komponen warna yaitu hue, saturation dan value yang dapat membedakan value dari hue dan saturation. Hue mewakili gradasi dari warna dalam spektrum optik atau spektrum cahaya yang dapat dilihat. Saturation merupakan intensitas dari hue tertentu yang berdasarkan pada kemurnian sebuah warna. Kepraktisan dari model HSV dapat dilihat berdasarkan dua faktor yaitu value yang terpisah dari warna citra, dan hue dan saturation berhubungan dengan penglihatan manusia sehingga untuk membuat sistem berbasis visi manusia, model HSV adalah metode yang ideal. (R.C Gonzales dan R.E Woods, 1992).^[3]

Komponen chroma (hue dan saturation), dan komponen luminance (value) dari model warna HSV ditentukan dengan gambar RGB sebagaimana yang ditunjukkan pada Gambar. 3, yang terdiri dari tiga warna utama yaitu merah (red), hijau (green), dan biru (blue) yang berada dalam lingkaran kromatik. Hue atau H adalah sudut vektor yang sesuai dengan axis merah. Ketika $H = 0^\circ$, maka warna yang dihasilkan adalah merah. Saturation atau S adalah derajat warna yang belum ditambahkan dengan putih. Jarak dari lokasi hingga ke titik tengah sebanding dengan Saturation. Semakin jauh jaraknya dari pusat lingkaran kromatik, semakin tinggi pula saturation. Value atau V diukur dengan garis yang tegak lurus terhadap lingkaran kromatik dan melewati pusat lingkaran. (W. Z Chen, 2011).^[3]



Gambar 3. Model Warna HSV (W. Z Chen, 2011).

Binerisasi Citra

Pada proses binerisasi, file citra digital dikonversi menjadi sebuah citra biner yang hanya memiliki dua nilai derajat keabuan, yaitu hitam dan putih. Pixel-pixel objek bernilai satu dan pixel-pixel latar belakang bernilai nol. Pada gambar akan ditampilkan bahwa nol merupakan putih dan satu merupakan hitam. Sehingga pada citra biner, latar belakang akan didapatkan berwarna putih sedangkan objek berwarna hitam.^[4]

Hough Transform

Hough Transform telah dikenal sebagai sebuah *treatment* yang kuat untuk pendeteksian bentuk kurva-kurva yang bersifat parametrik di dalam citra. *Circle Hough Transform* didesain untuk menemukan lingkaran-lingkaran yang terbentuk oleh sebuah titik pusat (x_0, y_0) dan sebuah radius r . Dalam menentukan sebuah lingkaran, sangat penting untuk mengakumulasi nilai – nilai parameter pada tiga ruang dimensi (x_0, y_0, r) .^[5]

Contour

Kontur (*contour*) dapat didefinisikan sebagai suatu keadaan yang terjadi karena adanya perubahan intensitas pada piksel-piksel yang bertetangga. Dengan adanya perubahan intensitas ini, maka titik-titik tepi (*edge*) pada citra dapat terdeteksi. Kontur

juga dapat didefinisikan sebagai urutan titik yang dapat menguraikan bentuk atau region.^[6]

Convex Hull

Convex hull adalah suatu himpunan yang terdiri dari beberapa titik dimana sebarang dari titik tersebut akan terhubung membentuk sebuah garis yang melingkupi semua. *Convex Hull* mampu meng-cover semua himpunan titik dengan jumlah garis yang paling minimal. *Convex hull* digambarkan secara sederhana dalam sebuah bidang sebagai pencarian subset dari himpunan titik pada bidang tersebut, sehingga jika titik-titik tersebut dijadikan poligon maka akan membentuk poligon yang *konveks*. Suatu poligon dikatakan *konveks* jika garis yang menghubungkan antar kedua titik dalam poligon tersebut tidak memotong garis batas dari poligon.^[6]

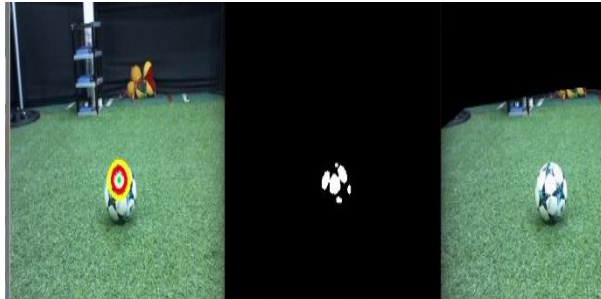
Berdasarkan peraturan KRSBI Humanoid dengan mengacu kepada RoboCup Soccer Humanoid League, dengan tingkat kesulitan yang bertambah sulit dari tahun ke tahun. Hal tersebut mendorong peserta KRSBI untuk semakin giat dalam menghadirkan pengembangan inovasi yang baru dan lebih baik pada robot humanoid soccer khususnya dalam pendeteksian bola yang dari sebelumnya berwarna orange, menjadi berwarna putih. Pada peraturan beberapa tahun sebelumnya pendeteksian dilakukan terhadap bola berwarna orange yang dapat dideteksi dengan melakukan pengenalan warna dan bentuk lingkaran saja akan tetapi pada peraturan tahun ini, diwajibkan untuk melakukan deteksi terhadap bola putih yang menjadikan tantangan tersendiri bagi peserta untuk mendeteksi tidak hanya warna saja, akan tetapi, bentuk maupun ciri khusus bola putih menjadikan tantangan tersendiri dalam kompetisi ini.

Berdasarkan Gambar. 2, pada pendeteksian bola putih, diawali dengan pengambilan gambar dari kamera. Apabila sistem telah mendeteksi objek, selanjutnya gambar objek yang terdeteksi dikonversikan dari warna RGB menjadi HSV. Setelah dihasilkan konversi image dalam warna HSV, dilanjutkan dengan proses *color filtering* dengan binerisasi, yakni mengubah warna gambar dalam bentuk biner, sehingga gambar yang dihasilkan hanya memiliki dua warna saja, yakni hitam dan putih. Dengan dilanjutkan eliminasi warna selain warna hijau. Pada proses inilah, bola putih dapat dikenali. Selanjutnya, agar kamera mengenali bentuk dan ciri bola, maka dilakukan konversi warna RGB menjadi HSV dan dilanjutkan dengan *color filtering*. Filterisasi menggunakan metode *Hough Circle*, *Contour*, dan *Convex Hull* juga dilakukan setelah proses tersebut sehingga bola dapat terdeteksi sesuai ciri dan bentuk yang diharapkan.

III. HASIL DAN DISKUSI

Pada tahap ini, Percobaan perlu dilakukan untuk mendapatkan hasil dan analisa apakah dengan menggunakan metode OpenCV pada citra kamera dapat mengenali dan mendeteksi bola putih sesuai dengan yang diharapkan.

Percobaan ke-1



Gambar. 4 Percobaan pertama

Pada percobaan pertama, diberikan metode *contour* dan *hough circle* dengan sebuah bola putih yang diletakkan di sembarang tempat. Pada percobaan ini dapat dianalisa bahwa bola dapat di deteksi dan dikenali oleh camera pada robot.

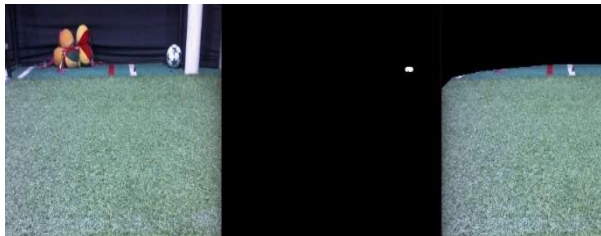
Percobaan ke-2



Gambar. 5 Percobaan ke-dua

Pada percobaan ke-dua, diberikan metode *contour* dan *hough circle* pada *syntax program* dengan bola putih yang diletakkan tepat didepan kamera dengan diberi objek tambahan sebagai *noise* yang menyerupai warna bola tersebut sehingga dapat dianalisa bahwa bola dapat terdeteksi dan dikenali meskipun terdapat objek yang berwarna putih di sekitar bola.

Percobaan ke-3



Gambar. 6 Percobaan ke-tiga

Pada percobaan ke-tiga, diberikan tambahan metode *convex hull* pada *syntax program* dengan bola putih yang ditempatkan disembarang tempat di luar lapangan berwarna hijau. Berdasarkan percobaan ke-

tiga ini, dapat dianalisa bahwa bola tidak dapat terdeteksi apabila tidak berada di dalam lapangan berwarna hijau.

IV. KESIMPULAN

Berdasarkan hasil percobaan dengan menggunakan beberapa metode dalam pengolahan citra, maka penggunaan *treatment* atau metode segmentasi objek dengan *Contour* dan *Hough Circle* pada robot dapat mendeteksi bola putih. Akan tetapi, dengan adanya penambahan metode *Convex Hull* dihasilkan deteksi bola putih yang lebih akurat dengan ciri, bola putih yang berada di luar lapangan berwarna hijau tidak akan terdeteksi oleh kamera, sedangkan bola putih yang terdapat di dalam lapangan berwarna hijau dapat terdeteksi dengan baik. Oleh karena itu, dapat disimpulkan bahwa, penggunaan ketiga metode tersebut terbukti dapat meningkatkan keakuratan pendeteksian bola putih dalam pertandingan KRSBI humanoid yang mengacu pada peraturan KRSBI Humanoid 2018.

DAFTAR PUSTAKA

- [1] A. Antonius D. Triyanto dan I. Ruslianto. "Penerapan Pengolahan Citra Dengan Metode Adaptive Motion Detection Algorithm Pada Sistem Kamera Keamanan Dengan Push Notification Ke Smartphone Android," Jurnal Coding Sistem Komputer Untan. 2015. 03(2): 54-65
- [2] Rulaningtyas R, Suksmono AB, Mengko TLR, Saptawati GAP. Segmentasi Citra Berwarna dengan Menggunakan Metode Clustering Berbasis Patch untuk Identifikasi Mycobacterium Tuberculosis. 2015.
- [3] Muqtadir A. Pengendali Fungsi Pointer Berbasis Hand Gesture Menggunakan Algoritma Convex Hull. Universitas Halu Oleo; 2016.
- [4] Munir R. Pengolahan Citra Digital dengan Pendekatan Algoritmik. Bandung; 2004.
- [5] Y. P. Prayitno Harianto dan M. C. Wibowo. "Rancang Bangun Aplikasi Pendeteksi Bentuk Dan Warna Benda Pada Mobile Robot Berbasis Webcam," Makalah. STIKOM. 2012.
- [6] A. Dhawan dan V. Honrao. "Implementation of hand detection based techniques for human computer interaction," International Journal of Computer Applications. 2013. 72(17): 6-13

Prediksi Arah Datang Bola Menggunakan Triangulasi Pada Robot Penjaga Gawang

Fadhlul Azmi, Hendawan Soebhakti, S.T., M.T, Eko Rudiawan Jamzuri, S.ST.,
Susanto, S.ST., M.Sc.

¹Email korespondensi: fadhlul.4211401056@students.polibatam.ac.id

Abstrak -- Selayaknya pemain sepak bola, robot sepak bola memiliki beberapa peran dalam bertanding, salah satunya penjaga gawang. Robot penjaga gawang bertugas menjaga gawang dari serangan robot lawan. Beberapa robot penjaga gawang yang telah diciptakan sudah memiliki algoritma yang baik dalam menjaga gawang. Contohnya menggunakan metode Triangulation agar robot dapat mengestimasi posisi bola disekitar robot[1]. Sistem ini sangat baik diterapkan ke robot, tetapi pada aplikasinya robot sering keliru dalam pengambilan keputusan. Hal ini dapat meningkatkan resiko servo rusak disebabkan robot sering membanting diri. Untuk itu penulis bermaksud mengembangkan suatu sistem agar robot semakin baik dalam menjaga gawang. Sistem yang ditambahkan yaitu memprediksi arah datang bola ketika bola bergerak. Proses prediksi ini digunakan untuk mengestimasi jarak robot dengan bola ketika bola sejajar dengan robot terhadap sumbu horizontal. Sehingga robot lebih efektif dalam melakukan penyelamatan. Dari berbagai percobaan yang dilakukan, sistem ini memiliki akurasi 83.2% dalam mengestimasi arah bola dan akurasi 96% dalam memprediksi jarak horizontal bola.

Kata kunci: *Triangulation*, prediksi, robot penjaga gawang.

I. PENDAHULUAN

Robot penjaga gawang memiliki tugas khusus yaitu mengusahakan agar gawang sendiri tidak dibobol oleh pemain lawan. Robot penjaga gawang dituntut dapat melakukan gerakan yang dapat menjangkau seluruh gawang, dan juga dapat melakukan intersepsi atau gerakan yang cekatan jika bola tidak dapat dijangkau robot tersebut.

Sistem mekanik dan sistem algoritma yang baik sangat berpengaruh terhadap keputusan robot penjaga gawang dalam melakukan tugasnya. Sistem mekanik berpengaruh terhadap pergerakan robot dalam menyelamatkan bola.

Untuk sistem algoritma, terdapat beberapa algoritma robot penjaga gawang yang sudah diterapkan. Algoritma yang telah diterapkan pada robot Bareleng yaitu memanfaatkan pergerakan koordinat bola dalam mengambil keputusan robot[1]. Cara ini cukup baik diterapkan ke robot penjaga gawang.

Contoh algoritma lain yang diterapkan yaitu Decision Tree Analysis, merupakan pengambilan keputusan dengan menghitung peluang dari parameter parameter yang di inputkan[2]. Untuk sistem ini,

performa robot penjaga gawang dapat bekerja dengan maksimal hanya pada jarak 40 s/d 80 cm, selebihnya presentase keberhasilan dari sistem ini 60%.

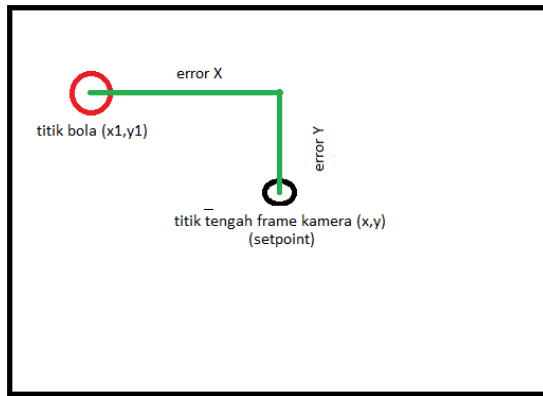
Dari permasalahan di atas penulis memiliki suatu sistem tambahan yang dapat digunakan pada robot penjaga gawang yaitu memprediksi jarak horizontal robot dengan bola sebelum bola tersebut sampai ke robot. Sistem ini memanfaatkan sudut yang terbentuk sesaat ketika bola bergerak ke arah tertentu. Sudut yang terbentuk dikombinasikan dengan jarak bola terhadap robot sehingga didapatkan perkiraan jarak horizontal ketika bola sejajar dengan robot. Sistem ini dikombinasikan dengan mendeteksi pergerakan bola agar robot dapat melakukan intersepsi. Dengan ditamhkannya sistem prediksi ini diharapkan robot penjaga gawang dapat menjaga gawang lebih baik lagi.

II. METODE

A. Tracking Bola

Kompetensi utama yang mesti dimiliki setiap robot sepak bola yaitu dapat mencari kemudian mengikuti pergerakan objek yang ingin dideteksi, hal ini dapat disebut tracking objek. Tracking merupakan usaha robot untuk meyelaraskan titik tengah (x,y) kamera dengan titik tengah (x1,y1).

Konsep dari tracking bola yaitu kamera memiliki koordinat x dan koordinat y dengan satuan pixel, begitu pula dengan objek bola yang dideteksi. Koordinat bola dapat pula diketahui oleh kamera. Titik tengah koordinat x dan y dapat diperoleh dengan membagi dua frame pada kamera. Ketika posisi titik tengah kamera tidak sesuai dengan titik tengah bola, terdapat nilai selisih antara koordinat tengah kamera dengan bola. Nilai selisih ini yang akan menggerakkan kepala robot dengan perhitungan yang sudah dirumuskan agar titik tengah kamera sesuai dengan titik tengah bola. Diperlukan sistem kontrol agar proses tracking bola berjalan dengan baik. Konsep tracking bola digambarkan pada gambar 1.



Gambar 1. Tampilan kamera dengan bola

B. Kalkulasi Jarak Bola

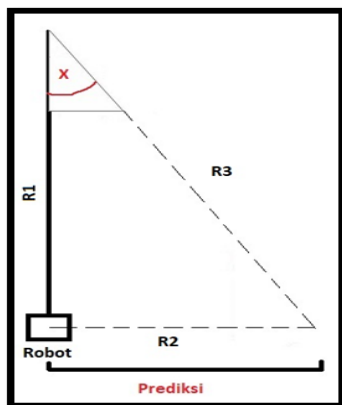
Setelah robot mampu mengikuti pergerakan bola, langkah selanjutnya untuk memprediksi arah datang bola yaitu mengetahui jarak antara bola dengan robot. Metode yang dapat digunakan untuk menghitung jarak bola adalah Triangulation. Metode ini merupakan metode pengukuran dengan menggunakan aturan segitiga[3]. Aturan ini digunakan karena posisi robot saat berdiri mengikuti pergerakan bola menyerupai bentuk segitiga siku-siku. Adapun rumus dari aturan sinus adalah sebagai berikut :

$$\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C} \quad (1)$$

Untuk mendapatkan resultan jarak robot dengan bola, hal yang mesti kita cari yaitu jarakY, jarakX kemudian jarak resultan.

C. Sudut Pergerakan Bola

Selain Jarak bola terhadap robot, parameter lainnya yang dibutuhkan yaitu sudut yang terbentuk saat bola bergerak. Saat bola bergerak, jika diteliti terdapat segitiga yang terbentuk. Sebesar apapun perubahan ukuran pada segitiga, nilai sudut yang terbentuk pada segitiga selalu mendekati konstan, asalkan perubahan sisi – sisi pada segitiga mendekati linier.



Gambar 2. Pembentukan segitiga saat prediksi bola

Untuk mendapatkan sudut pergerakan bola, langkah yang mesti dilakukan yaitu mencari panjang sisi – sisi

segitiga yang terbentuk saat bola bergerak. Persamaan untuk mendapatkan panjang sisi – sisi segitiga yang terbentuk yaitu :

$$R = \sqrt{(X_{awal} - X_{akhir})^2 + (Y_{awal} - Y_{akhir})^2} \quad (2)$$

Persamaan untuk mendapatkan sudut yang terbentuk dari sisi segitiga yaitu :

$$X = \cos \left(\frac{R1^2 + R3^2 - R2^2}{2 \times R1 \times R3} \right) \quad (3)$$

Dari parameter jarak bola dan sudut pergerakan bola, dapat dikalkulasikan prediksi jarak horizontal robot terhadap bola ketika bola sejajar dengan robot. Persamaannya menggunakan aturan sinus yaitu :

$$\text{Prediksi jarak } X = \frac{\text{Jarak } Y \times \sin X}{\sin (180 - (X + 90))} \quad (4)$$

D. Teorema GLB

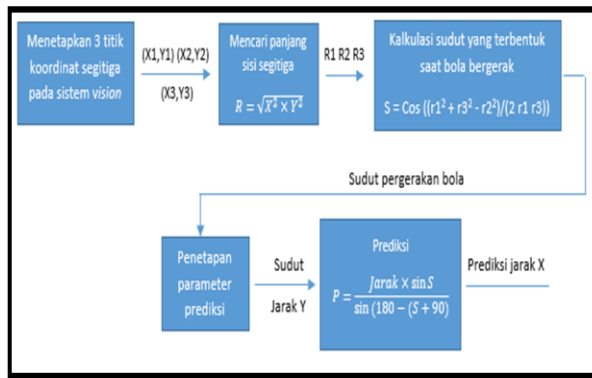
Saat robot menjaga gawang, tentunya robot diharapkan dapat melakukan gerakan penyelamatan saat bola tak mampu dihadang robot. Seperti halnya pemain bola saat menjaga gawang, ketika bola bergerak dengan cepat, maka hal yang harus dilakukan yaitu membanting badan menghadang bola.

Robot mesti mengetahui apakah bola sedang bergerak atau tidak. Agar dapat mengindikasikan perubahan posisi bola, dapat menggunakan prinsip gerak lurus beraturan (GLB) yang dialami bola. GLB merupakan kondisi dimana suatu benda bergerak dengan kecepatan yang tetap pada suatu waktu[4]. Contoh GLB yang mudah ditemui yaitu rotasi bulan terhadap bumi.

Pada kasus ini, perubahan posisi pada bola diambil kecepatan rata-rata, dengan mencari perubahan posisi bola terhadap selang waktu. Saat bola di tendang oleh robot, bola mengalami perpindahan. Kecepatan perpindahan bola diasumsikan tetap karena lintasan gerak bola yang pendek. Kecepatan rata – rata suatu benda yang bergerak merupakan akumulasi dari perpindahan bola dibagi selang waktu Δt selama perpindahan terjadi[4]. Seperti pada persamaan (5) berikut :

$$v = \frac{\Delta x}{\Delta t} = \frac{x_2 - x_1}{t_2 - t_1} \quad (5)$$

Dari metode-metode yang digunakan diatas, dapat disusun perancangan perannngkat lunak robot penjaga gawang dalam memprediksi arah bola

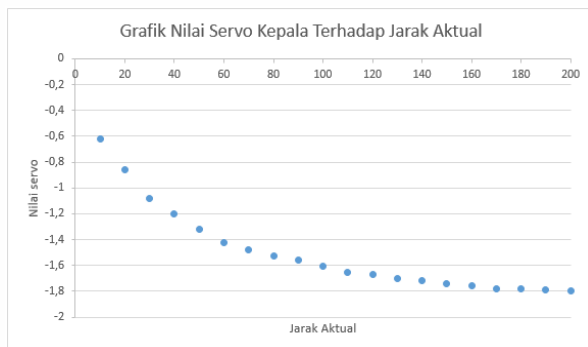


Gambar 3. Sistem Prediksi Bola

III. HASIL DAN DISKUSI

A. Pengujian Estimasi Jarak Bola

Dalam mengestimasi Jarak Bola, robot menggunakan servo kepala robot sebagai indikasi panjang jarak bola terhadap robot, dilakukan pengambilan data hubungan nilai servo kepala terhadap jarak aktual dengan hasil seperti pada gambar 4.



Gambar 4. Grafik hubungan nilai servo terhadap jarak aktual

Berdasarkan gambar 4, dapat dilihat semakin jauh jarak objek (bola) yang dilihat, maka perubahan nilai dari servo kepala semakin kecil. Agar robot menghasilkan jarak perhitungan berdasarkan nilai servo kepala, dapat menggunakan cara regresi polinomial pada microsoft excel. Dengan memasukkan nilai servo sebagai masukan sumbu X dan jarak aktual sebagai masukan sumbu Y, hasil regresi polinomialnya adalah sebagai berikut :

$$y = -933,9x^5 - 5340,8x^4 - 12018x^3 - 13183x^2 - 7050,2x - 1454,3 \quad (6)$$

Persamaan (6) digunakan sebagai rumus untuk mendapatkan jarak bola terhadap robot. Persamaan ini dapat dimasukkan ke pemograman dengan menggunakan nilai servo kepala sebagai masukan.

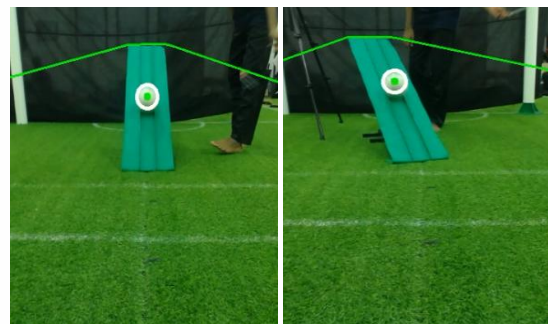


Gambar 5. Bola diletakkan sejajar dengan robot (a), hasil pembacaan jarak aktual (b)

Gambar 5 merupakan pengujian perhitungan jarak bola terhadap robot dilakukan dengan meletakkan bola dihadapan robot, kemudian robot mencari bola. Saat bola terdeteksi, robot mulai memperhitungkan jarak bola menggunakan persamaan 6. Pengujian ini dilakukan untuk mengetahui akurasi dari nilai jarak perhitungan yang dihasilkan robot terhadap jarak aktual dari pengukuran menggunakan meteran. Pengujian estimasi jarak ini dilakukan karena jarakY merupakan salah satu parameter dalam memperdiksi jarak horizontal bola.

B. Pengujian Pergerakan Bola

Pengujian pergerakan bola dilakukan dengan meletakkan bola pada lintasan miring dengan jarak dan sudut tertentu. Kemudian bola digelindingkan menuju kearah robot. Pada pengujian ini diperhatikan respon robot terhadap bola yang bergerak mendekati robot.



Gambar 6. Pengujian Pergerakan Bola, lurus terhadap robot(a) dan berbeda arah(b).

Gambar 6a merupakan pengujian respon robot ketika bola bergerak lurus dihadapan robot. Gambar 6b merupakan pengujian respon robot ketika bola bergerak ke arah tertentu. Pengujian ini dilakukan untuk memperhatikan respon robot terhadap arah pergerakan bola.

Dalam mengindikasikan pergerakan bola, robot memanfaatkan perpindahan posisi koordinat bola pada kamera dengan interval waktu yang telah ditentukan. Saat bola mengalami perpindahan, sistem akan menghasilkan deltaY dan deltaX. DeltaY mengindikasikan perpindahan bola secara vertikal, sedangkan deltaX mengindikasikan perpindahan horizontal. Berdasarkan perubahan delta, robot dapat

mengindikasikan kecepatan bola. Semakin besar nilai delta, maka semakin tinggi kecepatan bola.

Dari hasil deltaY dan deltaX, robot akan melakukan pergerakan banting tengah (M), kiri (L) dan kanan (R) ataupun diam (D). Pengujian respon robot terhadap pergerakan bola terdapat pada tabel 1 dan 2.

Tabel 1. Respon robot pada jarak 2 M

Kecepatan (m/s)	SUDUT				
	-20°	-10°	0°	10°	20°
3,1 m/s	LLLLL	LDLDL	MMMMM	MMMMM	RDRRD
2,6 m/s	LLLLL	LLMLL	MMMMM	MMMMM	RRRDR
2,48 m/s	MLLDD	MLDLL	MMMLL	MMMRM	RRRRR
2,15 m/s	LLLLL	MLLMM	MMMMM	RMMMM	RRRDR
1,5 m/s	LLLDD	LLLLL	MMMMM	MMMMM	RRRRR

Tabel 2. Respon robot pada jarak 1.5 M

Kecepatan (m/s)	SUDUT				
	-20°	-10°	0°	10°	20°
3,1 m/s	DLLML	LLDDL	MMMMM	RRDRD	RDRRD
2,6 m/s	DLLLL	LDLDD	MMDDM	RRDRD	RRRDD
2,48 m/s	LLLDL	LLLDD	MMMMM	DRRMR	RDRRD
2,15 m/s	LLLLL	MLLLL	MMMMM	RRRMR	RRMRR
1,5 m/s	LLLDL	LLLLL	MMMMM	DRRRR	RRRRR

Dari hasil tabel 1 dan 2, respon robot dikatakan berhasil jika :

- Pada sudut 0°, robot hanya banting tengah (M).
- Pada sudut -10°, robot dapat banting tengah (M). atau banting kiri (L).
- Pada sudut -20°, robot hanya banting kiri (L).
- Pada sudut 10°, robot dapat banting tengah (M). atau banting kanan (R).
- Pada sudut 20°, robot hanya banting kanan (R).
- Selain dari ketentuan diatas, maka respon robot dikatakan gagal.

Berdasarkan tabel 1 dan 2, persentase keberhasilan respon robot terhadap pergerakan bola yaitu :

$$P = \frac{208}{250} \times 100\% = 83,2\% \quad (7)$$

Dapat diperhatikan bahwa persentase keberhasilan respon robot terhadap pergerakan bola sebesar 83,2%. Hal ini karena ada beberapa kondisi dimana sistem vision kehilangan bola. Hal ini berpengaruh saat robot mengkalkulasi deltaY dan deltaX.

deltaY: 0	deltaX: 0	p1: 0	p3: 0	p2: 0	p4: 0
deltaY: 0	deltaX: 0	p1: 0	p3: 0	p2: 0	p4: 0
deltaY: 0	deltaX: 0	p1: 0	p3: 0	p2: 0	p4: 0
deltaY: 0	deltaX: 0	p1: 0	p3: 0	p2: 0	p4: 0
deltaY: 0	deltaX: 0	p1: 0	p3: 321	p2: 209	p4: 0
deltaY: 0	deltaX: 0	p1: 0	p3: 321	p2: 231	p4: 0
deltaY: 0	deltaX: 0	p1: 0	p3: 321	p2: 231	p4: 0
deltaY: 9	deltaX: 5	p1: 240	p3: 321	p2: 231	p4: 316
deltaY: 9	deltaX: 5	p1: 240	p3: 321	p2: 231	p4: 316
deltaY: 14	deltaX: 12	p1: 245	p3: 321	p2: 231	p4: 309
SAVE TENGAH.....					

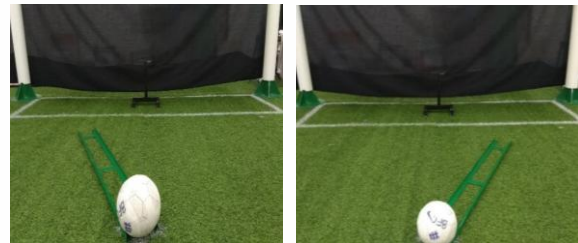
Gambar 7. Hasil Delta Saat Bola Hilang

Terdapat empat parameter pada proses kalkulasi deltaY dan deltaX, yaitu p1, p2, p3, p4 seperti pada gambar 7. Ketika salah satu dari parameter ini sama

dengan nol (bola hilang) maka sistem tidak akan mengkalkulasi deltaY dan deltaX.

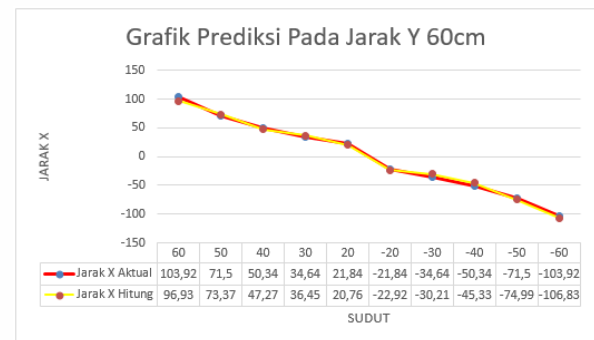
C. Pengujian Proses Prediksi

Proses Pengambilan data prediksi dengan meletakkan bola pada suatu jarak tertentu, kemudian bola di gilir pada sudut tertentu menggunakan alat agar sudut aktual dapat terukur dengan baik.



Gambar 8a dan 8b. Pengujian prediksi bola

Pengujian prediksi bola dilakukan menggunakan lintasan. Bola ditelakkan diatas lintasan agar dapat bergerak kekiri (gambar 8a) dan kekanan (gambar 8b) dengan terarah. Hasil prediksi jarak horizontal robot dapat diamati pada gambar 9.



Gambar 9. Grafik prediksi

Dari gambar 9, dapat diketahui bahwa semakin besar sudut yang terbentuk saat bola bergerak maka semakin jauh pula jarak horizontal bola. Maka dalam hal ini robot dapat mengestimasi dengan baik jarak horizontal robot ketika bola bergerak.

Hasil prediksi ini diperoleh dari kombinasi antara sudut pergerakan bola dengan jarak vertikal bola. Pengujian dilakukan sebanyak 80 kali berupa delapan posisi jarak dan sepuluh posisi sudut. Dari pengujian yang dilakukan, persentase akurasi prediksi jarak horizontal yaitu:

$$P = \frac{\Sigma \text{persentase prediksi}}{\Sigma \text{data}} = \frac{7680,96}{80} = 96\% \quad (8)$$

IV. KESIMPULAN

Berdasarkan eksplorasi sistem secara keseluruhan, terdapat beberapa kesimpulan sebagai berikut:

A. Pada pengujian jarak robot terhadap bola, selisih hasil perhitungan jarak bola dengan jarak aktual semakin tinggi ketika bola semakin jauh. hal ini disebabkan perubahan nilai servo kepala semakin kecil ketika bola semakin jauh. Perubahan ini dapat dilihat dari grafik hubungan antara nilai servo terhadap jarak aktual yang semakin tidak linear saat jarak bola semakin jauh. Namun hal ini tidak berpengaruh pada proses prediksi bola. Karena pembacaan jarak dilakukan ketika jarak bola di area pertahanan.

B. Tingkat keberhasilan respon robot terhadap arah pergerakan bola sebesar 83,2%. Hal yang berpengaruh pada hasil persentase respon robot tersebut yaitu sistem vision. Saat bola bergerak dengan cepat, terkadang vision kehilangan bola, hal ini menyebabkan perhitungan ΔY dan ΔX terkendala sehingga robot tidak merespon ataupun salah respon.

C. Pada pengujian prediksi jarak horizontal robot. Hasil prediksi sudah mendekati nilai aktual. Jika hasil prediksi bola sangat jauh dengan nilai aktual, hal ini karena kedua parameter yang digunakan untuk memprediksi jarak horizontal merupakan hasil perhitungan juga, sehingga saat kedua parameter menghasilkan nilai yang kurang presisi, akan berpengaruh besar terhadap hasil prediksi.

V. UCAPAN TERIMA KASIH

Dalam menyusun Jurnal ini, Penulis berterima kasih kepada pihak Politeknik Negeri Batam yang telah memberikan dukungan dari segi finansial dalam meneliti penelitian ini.

VI. DAFTAR PUSTAKA

- [1] Widodo, Wisnu Ananto, "Pengambilan Keputusan Robot Terhadap Bola Dengan Satu Kamera Menggunakan Metode Triangulation," Tugas Akhir D-4, Politeknik Negeri Batam, Indonesia, 2015
- [2] Sungkono, Silvester Kristian, Banu, Daniel, "6th International Annual Engineering Seminar (InAES)," Yogyakarta, 1998
- [3] Yenni, Rika Firma, "Desain Aturan Sinus dan Aturan Cosinus Berbasis PMRI.," Jurnal Matematika dan Sains, vol.2, 97-108, Desember. 2014.
- [4] Young dan Freedman, "Gerak Pada Garis Lurus," dalam FISIKA UNIVERSITAS EDISI KESEPULUH Jilid 1. Jakarta : Erlangga, 2002.
- [5] www.kontesrobotindonesia.org, Buku Panduan KRSBI Humanoid 2017 Versi 21.12.16
- [6] Susanto, dkk., "Bareleng FC Team Description Paper Humanoid Kid Size League of RoboCup 2017", <https://www.robocuphumanoid.org/hl-2017/teams/>, 2017

Penerapan *Google's Speech API* sebagai Antarmuka Interaksi Robot Dan Manusia.

Anton Hidayat, S.T., M.T.¹, Marta Zuriadi², Teguh P. Nugraha³

¹Teknik Elektro, Politeknik Negeri Padang, Padang

²Teknik Elektro, Politeknik Negeri Padang, Padang

³Teknik Informatika, Politeknik Negeri Padang, Padang

¹antonramiati@gmail.com

Abstrak – Kehadiran robot dalam kehidupan manusia semakin hari semakin disadari manfaatnya. Hanya saja dalam perkembangannya robot belum begitu mampu berinteraksi dengan manusia. Untuk bisa berinteraksi dengan manusia robot harusnya memiliki beberapa kemampuan seperti berbicara, mendengar dan mengenali objek lawan ia berinteraksi. Oleh karenanya pemanfaatan *Google's Speech API (Application Programming Interface)* sebagai antarmuka interaksi robot dan manusia dengan menggunakan teknologi *Speech Recognition*, *Speech to Text*, dan *Text to Speech*. Disamping itu citra penglihatan robot di-monitoring secara *wireless* untuk memastikan apakah robot mampu mengenali objek dengan benar. Dengan menambahkan fitur ini pada robot diharapkan dapat meningkatkan kemampuan robot dalam berinteraksi dengan manusia. Sehingga pemanfaatan robot semakin tepat guna dikemudian hari.

Kata kunci: *Speech Recognition*, *Text to Speech*, *Object Monitoring*.

I. PENDAHULUAN

Kehadiran robot dalam kehidupan manusia makin hari dirasakan manfaatnya. Robotik tidak lagi dipandang sebagai ilmu yang berkembang hanya dalam konteks teknologi (fisik) saja, namun juga dalam hal yang berkaitan dengan lingkungan hidup manusia yang perlu juga diambil perhatian.

Kehadiran robot di lingkungan yang bersifat lebih fleksibel, seperti misalnya rumah sakit, rumah tangga, perkantoran, eksplorasi hutan, dan pembangunan kawasan-kawasan berbahaya (plant nuklir, kimia, dsb.), telah membuat manusia harus menata ulang definisi, konstruksi dan fungsi robot. Keadaan ini telah menempatkan robot sebagai kehidupan keseharian sehingga dikenal istilah human-robot interaction.

Interaksi antara manusia dengan robot atau mesin (human-machine interactions) dapat dinyatakan dalam 3 tingkatan, yaitu:

1. Manusia sebagai kontroler robot sepenuhnya,
2. Manusia sebagai manager dari operasi robot, dan
3. Manusia dan robot berada dalam kesetaraan.^[1]

Dalam perkembangannya interaksi robot dan manusia lebih sering dikembangkan dengan menggunakan *remote control* baik itu secara *non-wireless* maupun secara *wireless*. Dalam hal ini memposisikan manusia sebagai operator maupun sebagai manager dari operasi robot. Sehingga robot

belum sepenuhnya setara dengan manusia dalam hal berinteraksi.^[4]

Untuk membuat robot memiliki kemampuan yang setara dengan manusia, maka robot harus memiliki alat indera yang sama dengan manusia. Seperti halnya manusia memiliki mata untuk melihat, telinga untuk mendengar, dan mulut untuk berbicara dalam hal ini semua indera itu dapat diterapkan pada robot dengan memanfaatkan kamera sebagai indera penglihatan, mic sebagai indera pendengaran, dan speaker sebagai alat untuk berbicara.

Namun yang menjadi permasalahan ialah bagaimana membuat robot mengerti dengan bahasa yang diucapkan oleh manusia dan bagaimana cara robot untuk menyampaikan suatu kalimat dengan bahasa yang bisa dimengerti oleh manusia. Oleh karena itu kita perlu menerapkan suatu metadata yang berisi kumpulan bahasa pada robot supaya bisa paham dengan apa yang disampaikan oleh manusia pada saat berinteraksi dengan robot.

Google's Speech API merupakan suatu metadata yang terdiri atas *Google's Speech Recognition*, *Google Speech to Text*, dan *Text to Speech API* yang terintegrasi kedalam *Google's Cloud API* dapat difungsikan sebagai penerjemah bahasa manusia dari berbagai bahasa didunia kedalam bentuk teks. Kemudian dari teks ini di proses untuk menentukan apa respon dari robot jika mendapat instruksi tertentu dari manusia. Respon dari robot yang masih dalam bentuk tekstual kemudian diterjemahkan kembali kedalam bentuk suara melalui *Google's Text to Speech API*. Dengan demikian robot dapat berinteraksi dengan manusia dan dapat mengerti perintah yang disampaikan oleh manusia, serta mampu berbicara dalam bahasa tertentu layaknya manusia.

II. METODE.

A. *Google's Speech API*.

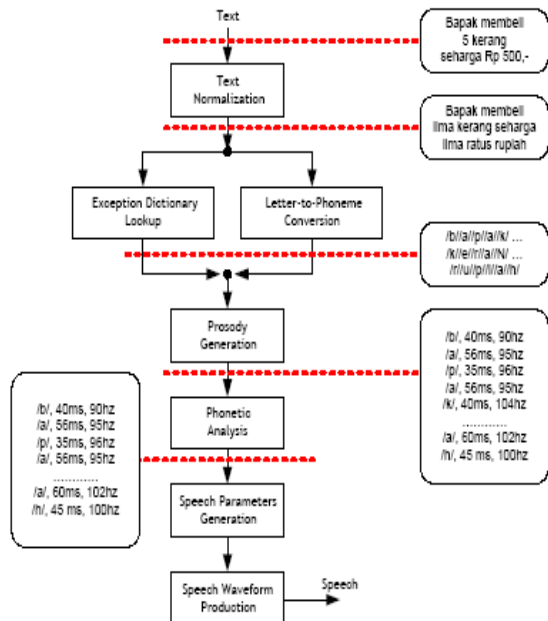
1) *Speech Recognition*

Sistem *Speech Recognition* atau Sistem pengenalan suara adalah sistem yang berfungsi untuk mengubah bahasa lisan menjadi bahasa tulisan. Masukan sistem adalah ucapan manusia, selanjutnya sistem akan mengidentifikasi kata atau kalimat yang diucapkan dan menghasilkan teks yang sesuai dengan apa yang diucapkan.

2) Text to Speech

Pada dasarnya *Teks to Speech* adalah suatu sistem yang dapat mengubah text menjadi ucapan. Suatu sistem pensintesa ucapan atau *Text to Speech* pada prinsipnya terdiri dari dua sub sistem, yaitu :

1. Bagian Konverter Teks ke Fonem (Text to Phoneme), serta
2. Bagian Konverter Fonem ke Ucapan (Phoneme to Speech).



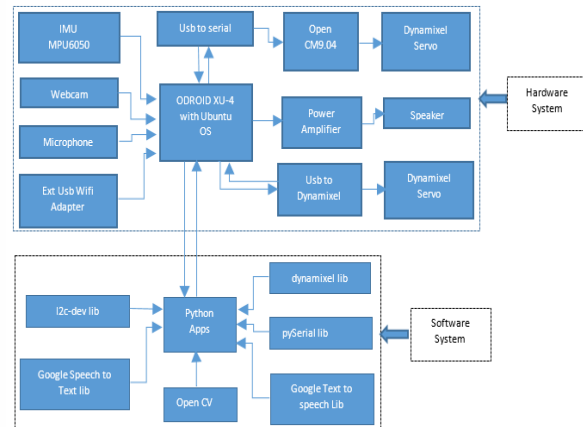
Gambar 1 : Bagan proses konversi *Text to Speech*.^[2]

B. Google's Speech API dengan Python

Python adalah bahasa pemrograman 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. Python mendukung multi paradigma pemrograman, utamanya; namun tidak dibatasi; pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada python adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis.^[3]

Untuk menjalankan *Google's speech API* dengan python maka harus menginstal librarynya terlebih dahulu. Adapun library *Google's speech API* terdiri atas dua bagian, yaitu : *Speech Recognition* dan *Text to Speech*.

C. Blok Diagram Sistem Robot



Gambar 2 : Blok Diagram Robot Midun.

Pada dasarnya sistem yang bekerja pada robot ini terbagi atas dua bagian, yaitu :

1. Hardware System

Hardware system atau sistem perangkat keras yang terdiri atas beberapa perangkat dengan fungsi-fungsi tertentu yaitu diantaranya sebagai pemroses data (*processing unit*), menerima input dari dunia luar (*input devices*), merespon kedunia luar (*output devices*), ataupun sebagai media penyimpanan data (*storage devices*).

2. Software System

Software system atau sistem perangkat lunak berfungsi sebagai sarana interaksi yg menghubungkan atau menjembatani input dari dunia luar dengan perangkat keras, selain itu perangkat lunak juga berfungsi sebagai *analyzer*, pengolah data, dan manajemen data, serta beberapa fungsi-fungsi lainnya.

D. Algoritma Interaksi Robot dan Manusia

Pada dasarnya algoritma robot dalam menemukan jawaban saat berinteraksi ialah dengan cara mengkonversi sinyal suara yang di sampaikan menjadi text dengan menggunakan metode *speech to text*. Kemudian pertanyaan ini dicocokkan dengan jawaban yang telah tersedia dalam database. Jika jawaban dari pertanyaan ditemukan maka kemudian text jawaban dikonversi kembali menjadi sinyal suara dengan menggunakan metode *text to speech*. Namun jika jawaban dari pertanyaan tidak ditemukan maka robot secara otomatis meminta untuk diinputkan jawabannya kedalam *database*.



Gambar 3 : Algoritma robot dalam menemukan jawaban saat berinteraksi

III. HASIL DAN DISKUSI

A. Hasil Pengujian Interaksi dengan Robot

Berikut adalah hasil interaksi manusia dan robot dengan menerapkan *Google's speech API* menggunakan Python.

```

Python 2.7.12 (default, Dec 4 2017, 14:50:18)
[GCC 5.4.0 20160609] on linux2
Type "copyright", "credits" or "license()" for more
>>>
----- RESTART: /home/tpn/tts.py -----
Hi Teguh, what can I do for you?
Say something!
You said: how are you
I am fine
Say something!
Google Speech Recognition could not understand audio
Say something!
You said: what time is it
Tue Jun 12 18:41:07 2018
Say something!
You said: hands up
hands up
Say something!
You said: heads up
Say something!
You said: how are you
I am fine
Say something!
You said: hands up
hands up
Say something!

```

Gambar 4 : Hasil Percakapan Manusia dengan robot.

Didalam percakapan singkat tersebut terlihat bagaimana robot mampu berinteraksi dengan manusia. Selain mampu berinteraksi dalam percakapan robot juga mampu menerima perintah secara verbal dari manusia. Dari beberapa percobaan memberi perintah secara verbal kepada robot maka didapatkan data hasil pengujian sebagai berikut ini.

Tabel 1. Hasil pengujian interaksi robot dengan satu orang manusia.

No.	Perintah / Pertanyaan	Respon Robot
1.	"hands up"	Robot mengangkat kedua tangannya.
2.	"go a head"	Robot berjalan maju
3.	"move left"	Robot begeser kekiri
4.	"move right"	Robot bergeser kekanan
5.	"Who are you ?"	Robot menjawab "I am midun"
6.	"how are you ?"	Robot menjawab "I am fine"
7.	"What time is it ?"	Robot memberi tahu jam berapa saat itu.

Tabel 2. Hasil pengujian kepekaan robot dalam mendengarkan perintah dari suara yang berbeda.

No.	Perintah / Pertanyaan	Intensitas Suara	Hasil
1.	"hands up"	keras	Terdengar "heads up" oleh robot.
2.	"hands up"	keras	Terdengar dengan baik
3.	"hands up"	sedang	Terdengar dengan baik
4.	"hands up"	Sedang	Terdengar dengan baik.
5.	"hands up"	Pelan	Terdengar dengan baik.
6.	"hands up"	Pelan	Terdengar dengan baik.
7.	"hands up"	pelan sekali	Tidak terdengar, Robot meminta mengulangi pertanyaan.
8.	"hands up"	pelan sekali	Tidak terdengar, Robot meminta mengulangi pertanyaan.
9.	"hands up"	Sedang (suara dari speaker)	Terdengar dengan baik.
10.	"hands up"	Sedang (suara dari speaker)	Terdengar dengan baik.

Berdasarkan tabel 2. dapat dianalisa bahwa robot lebih peka dalam mendengarkan suara dengan intensitas sedang dan pelan. Disamping itu robot juga mampu mendengarkan suara yang berasal dari speaker dengan intensitas suara sedang.

IV. KESIMPULAN

Untuk bisa berinteraksi dengan manusia, robot harus memiliki kemampuan yang setara dengan manusia seperti mendengar, berbicara, serta mampu menerjemahkan perkataan ataupun bahasa yang disampaikan oleh manusia. Melalui penerapan *Google's speech* API sebagai antarmuka, yang terdiri atas *speech recognition* dan *text to speech*, dapat membantu robot berinteraksi dengan manusia.

V. DAFTAR PUSTAKA

- [1] <http://atepbz.blogspot.com>. Interaksi Manusia dan Robot.
- [2] Pelton. Gordon E. (1993). "**Voice Processing**", McGraw-Hill, New York.
- [3] <https://id.wikipedia.org>. Python_(bahasa_pemrograman)
- [4] Manuela Veloso, Mindy Aisen, Ayanna Howard, et al. Human Robot Interaction Japan, South Korea, and China. *World Technology Evaluation Center (WTEC) panel report*, 2012.

Dago Hoogeschool dalam Kontes Robot Sepak Bola Indonesia Humanoid 2018

Hisham Lazuardi Yusuf¹, Ahmad Habbie Thias², Salman Al Farisi³, Tjia, Johan Setiawan⁴, Kevin Putra Adiwijna⁵, Muhammad Rafly Arya Putra⁶, Tessya Rismonita⁷, Achmad Fahrurrozi Maskur⁸, Hermawan Phanjaya⁹, Fildah Ananda¹⁰, Fernanda Rinia¹¹

Tim Dago Hoogeschool, Unit Robotika ITB

hishamlazuardi@s.itb.ac.id¹, ahmad_habbie@s.itb.ac.id², salman.hiro@s.itb.ac.id³, tjiajohansetiawan@s.itb.ac.id⁴, kpadiwijna@s.itb.ac.id⁵, raflyap@s.itb.ac.id⁶, tessyarismonita@s.itb.ac.id⁷, um@s.itb.ac.id⁸, hermawanp@s.itb.ac.id⁹, fildahananda@s.itb.ac.id¹⁰, fernandarinia@s.itb.ac.id¹¹

Abstrak – Kontes Robot Indonesia (KRI) merupakan kompetisi robot tahunan yang dilaksanakan oleh Menristek Dikti. Salah satu cabang kompetisi dalam KRI adalah Kontes Robot Sepak Bola Indonesia (KRSBI) Humanoid. Robot humanoid memiliki bentuk seperti manusia, yang memiliki 2 kaki, 2 tangan, batang tubuh serta kepala. Peraturan KRSBI Humanoid mengacu kepada peraturan kompetisi internasional Robocup Humanoid Kid Size League. Robot harus memiliki kemampuan untuk melakukan permainan sepak bola secara otomatis. Pada paper ini akan dijelaskan pengembangan robot sepak bola humanoid tim Dago Hoogeschool ITB dalam satu tahun terakhir, dalam persiapan untuk menghadapi KRI 2018.

Kata kunci: *Balancing, Hough Transform, Humanoid, Sobel*

I. PENDAHULUAN

Robot humanoid, memiliki bentuk seperti manusia, dengan 2 tangan, 2 kaki, batang tubuh serta kepala. Pada Kontes Robot Sepak Bola (KRSBI) Humanoid, robot harus dapat bermain sepak bola, dengan tujuan memasukkan bola ke gawang lawan.

KRSBI Humanoid merupakan kompetisi tahunan yang diadakan oleh Menristek Dikti. Peraturan yang digunakan dalam KRSBI Humanoid mengacu kepada aturan kompetisi internasional Robocup Kidsize League. Peraturan pada KRSBI Humanoid mengalami peningkatan kesulitan setiap tahunnya. Pada tahun 2016 untuk pertama kalinya digunakan rumput sintetis dengan ukuran 4x6 m². Kemudian pada tahun 2017, bola yang digunakan dalam KRSBI Humanoid mengalami perubahan dari bola tenis berwarna oranye, menjadi bola FIFA Size 1 dengan diameter 13 cm yang berwarna putih¹. Sedangkan pada tahun 2018, ukuran lapangan pertandingan KRSBI Humanoid menjadi 6x9 m²[2]^f.

Tim Dago Hoogeschool merupakan peserta dari KRSBI Humanoid tingkat nasional tahun 2018. Dago Hoogeschool merupakan bagian dari Unit Robotika ITB. Tim ini berdiri sejak 2008 dengan menggunakan platform Bioid. Sejak tahun 2013, Dago Hoogeschool menggunakan Darwin-OP dalam riset dan pengembangan robot humanoid. Saat ini, Dago Hoogeschool menggunakan platform Alfassung,

dengan Odroid XU-4 untuk menjalankan proses komputasi deteksi bola, komunikasi, serta keseimbangan dinamis robot.

Untuk menghadapi berbagai tantangan dalam pertandingan dilakukan beberapa pengembangan pada robot. Pengembangan pengolahan citra robot untuk mendeteksi objek di dalam lapangan secara optimal dilakukan dengan menggunakan operator Sobel dan Hough Circle seperti yang dijelaskan pada jurnal International Conference on Control, Electronics, Renewable Energy, and Communications⁹. Selain itu, pengembangan robot juga telah dilakukan pada artikel Humanoid Robot Locomotion System With Balancing Feedback Using Leg and Arm Strategy and Stepping Strategy⁵ untuk mengatasi kondisi lapangan yang menggunakan rumput sintetis.

Pengembangan yang dilakukan oleh Dago Hoogeschool diharapkan dapat memberikan kontribusi berupa robot humanoid, yang mampu bermain sepak bola secara otomatis. Melalui kemampuan membedakan objek di dalam lapangan berdasarkan bentuk objek. Selain itu, robot yang dikembangkan oleh Dago Hoogeschool, juga mampu melakukan dynamic balancing sehingga dapat berjalan secara stabil di atas rumput sintetis. Untuk ke depannya, pengembangan robot humanoid ini diharapkan semakin menyerupai kemampuan manusia dalam bermain sepak bola.

II. OVERVIEW DESAIN ROBOT

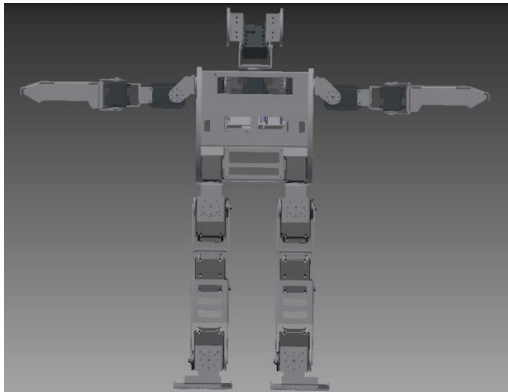
Robot Dago Hoogeschool didesain agar dapat memasukkan bola ke gawang lawan. Untuk mencapai tujuan tersebut perlu dilakukan desain sistem robot baik dari sisi mekanik, elektrik, maupun software dari robot. Adapun desain robot Dago Hoogeschool adalah sebagai berikut.

A. Desain Mekanik

Secara umum robot Dago Hoogeschool memiliki 2 kaki yang masing-masing menggunakan 6 buah servo Dynamixel MX-64. Serta menggunakan 3 buah servo MX-28 pada masing-masing tangan. Sedangkan pada bagian kepala digunakan 2 buah servo Dynamixel MX-28 untuk gerakan pan dan tilt pada kepala robot saat melakukan object tracking.

Rangka robot menggunakan alumunium yang melalui proses *laser cutting*. Kemudian alumunium tersebut di-*bending* sehingga terbentuk pola rangka robot. Selain menggunakan alumunium, Dago Hoogeschool juga menggunakan bahan *hard-nylon* untuk bagian ujung lengan robot.

Pada bagian batang tubuh robot, terdapat kompartemen untuk meletakkan komponen elektrik robot. Selain itu, pada bagian batang tubuh robot ini juga terdapat pelindung robot yang terbuat dari alumunium untuk mencegah terjadinya kerusakan rangka tubuh robot saat terjatuh akibat benturan.



Gambar 1 Desain Mekanik robot Dago Hoogeschool

B. Desain Elektrik

Dago Hoogeschool menggunakan *Odroid XU-4* sebagai *main controller* dari robot humanoid. *Odroid XU-4* memiliki prosesor *octa core*, sehingga mampu melakukan proses komputasi yang lebih tinggi dibandingkan dengan menggunakan *fitPC2i*. Adapun Spesifikasi dari *Odroid XU-4* adalah sebagai berikut³.

Tabel 1 Spesifikasi *Odroid XU4*

Processor	Samsung Exynos5422 Cortex™-A15 2Ghz and Cortex™-A7 Octa core CPUs
GPU	Mali-T628 MP6(OpenGL ES 3.1/2.0/1.1 and OpenCL 1.2 Full profile)
Memory	2Gbyte LPDDR3 RAM PoP stacked
Hard-disk	eMMC5.0 HS400 Flash Storage
Peripheral	2 x USB 3.0 Host, 1 x USB 2.0 Host, Gigabit Ethernet port, HDMI 1.4a for display
Power	5V/4A input

Odroid XU-4 juga melakukan komunikasi serial dengan *slave* berupa CM-730 untuk mengontrol 20 *Servo* yang digunakan pada robot. CM 730 memiliki arsitektur *ARM Cortex 32-bit CPU*. Selain itu, CM-

730 juga memiliki *3-axis gyroscope* dan *3-axis accelerometer*, serta 5 buah *TTL port* yang digunakan untuk menghubungkan CM-730 dengan *servo* pada robot.



Gambar 2 *Odroid XU4*

Dago Hoogeschool menggunakan 2 buah baterai *Lithium Polymer* 3 sel 2200mAh. Namun, *input* tegangan *Odroid XU-4* sebesar 5V sehingga digunakan *UBEC* sebagai regulator tegangan dengan *output* baterai menjadi 5V.



Gambar 3 Modul *CMPS11*

Untuk pengambilan citra pada lapangan, digunakan kamera *Logitech C920*. Selain itu, digunakan juga *CMPS 11* sebagai sensor orientasi dari robot. Selain memiliki magnetometer, *CMPS11* juga memiliki *gyro* dan *accelerometer*. Adapun Spesifikasi dari *CMPS* adalah sebagai berikut⁴.

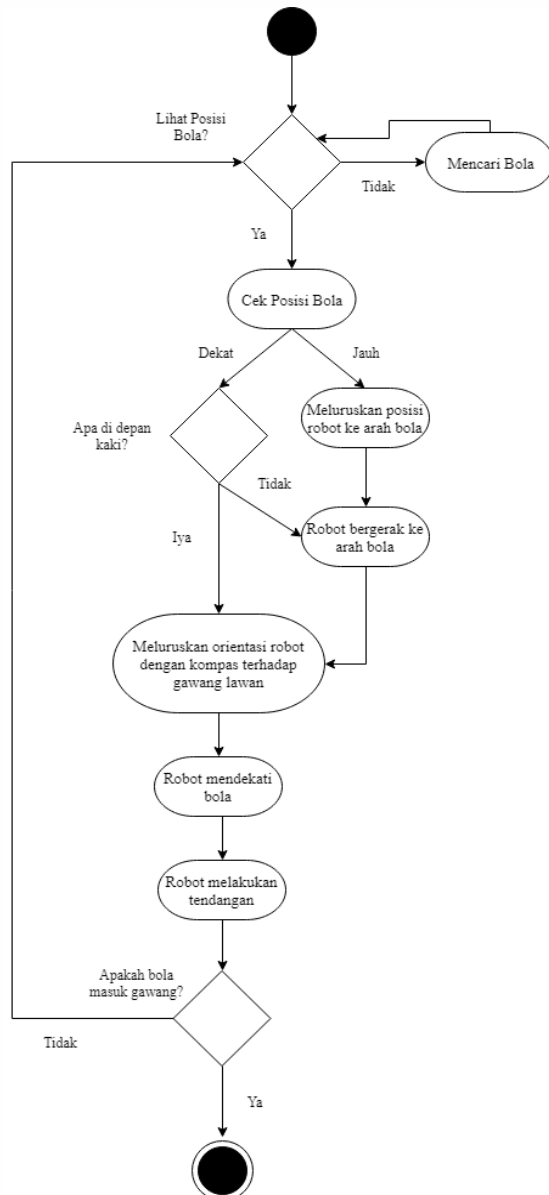
Tabel 2 Spesifikasi *CMPS11*

Input Tegangan	3.6V-5V
Arus	35 mA
Resolusi	0.1°
Akurasi	2%
Mode I2C	Up to 100 kHz
Mode Serial	9600, 19200, 38400(baud)

C. Arsitektur Software

1. Behaviour

Robot Dago Hoogeschool secara umum memiliki suatu *behavior* yang hampir sama untuk semua robot. Secara sederhana *behavior* robot dijelaskan melalui *activity diagram* di bawah ini. Robot akan melakukan pendeteksian lokasi bola, mendekatinya, lalu langsung menendang bola yang berada di depan kaki robot.



Gambar 4 Activity Diagram behavior umum robot

2. Dynamic Balancing

Keseimbangan robot merupakan sebuah tantangan yang cukup besar dalam pengembangan robot humanoid. Keseimbangan robot dapat diimplementasikan dengan menggunakan *invers kinematics*, yaitu posisi akhir suatu gerakan ditentukan oleh nilai sudut dari masing-masing *servo*⁵.

Dynamic Balancing dilakukan dengan menggunakan sensor *gyroscope* dan *accelerometer* untuk mendeteksi perubahan sudut pada robot. *Output* dari *gyroscope* dan *accelerometer* akan di-filter dengan menggunakan *moving average filter* untuk mereduksi *noise* pada sinyal *output*. Nilai *output gyroscope* dan *accelerometer* tersebut dijadikan parameter *feedback* untuk menyeimbangkan robot.

3. Vision

Untuk melakukan pendeteksian objek bola, robot Dago Hoogeschool menggunakan beberapa metode.

Salah satu metodenya, yaitu menggunakan transformasi *Hough* dengan deteksi tepi menggunakan *Sobel Filter*.

Sobel Operator atau disebut juga *Sobel Filter*, merupakan suatu metode yang digunakan untuk melakukan pengukuran gradien sebuah gambar pada ruang 2 dimensi. Operasi ini menitikberatkan pada penentuan daerah dengan frekuensi spasial tinggi, dimana daerah tersebut adalah batas yang berupa segmen garis, ditandai dengan perubahan warna secara drastis atau diskontinu.

Pertama akan dilakukan konvolusi. Konvolusi adalah operasi matematis sederhana dengan melakukan operasi perkalian pada dua buah matriks bilangan. Matriks harus memiliki dimensi yang sama, untuk menghasilkan matriks ketiga yang akan digunakan pada pengolahan citra.

Ketika menerapkan operasi ini pada dua buah matriks, *filter* akan bekerja seperti turunan orde pertama dan menghitung perbedaan intensitas piksel pada daerah tepi. Operator akan berisi *kernel*, atau matriks operator. *Kernel* tersebut berukuran 3×3 . Akan digunakan dua *kernel*, *Gx* dan *Gy*, dimana *Gy* adalah hasil rotasi 90° dari matriks *kernel* tersebut.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Gambar 5 Kernel konvolusi sobel⁶

Kernel tersebut berfungsi untuk merespons secara maksimal daerah-daerah tepi (*edges*) pada gambar secara vertikal dan horizontal, relatif terhadap *grid* pikselnya, dengan orientasi yang saling tegak lurus antar *kernel Gx* dan *Gy*. *Kernel* tersebut dapat dikombinasikan untuk menghasilkan besaran atau *magnitudo* mutlak dari gradien pada tiap titik dan orientasi.

Pada gambar 5 dapat dilihat bahwa nilai-nilai, misalnya -1, -2, +1, +2, dan 0, adalah pembobotan pada *mask* untuk mendapatkan tingkat pendeteksian tepi yang berbeda. Semakin besar nilai mutlaknya maka semakin banyak tepian yang terdeteksi. Nilai 0 pada *kernel Gx* tersebut tersusun secara vertikal, sehingga justru akan mendeteksi tepi dalam arah vertikal.

Besarnya *magnitudo* gradien dapat dinyatakan dengan:

$$|G| = \sqrt{Gx^2 + Gy^2} \quad \dots(1)$$

Secara umum, pendekatan *magnitudo* dinyatakan dengan:

$$|G| = |Gx| + |Gy| \quad \dots(2)$$

sehingga proses komputasinya akan lebih cepat.

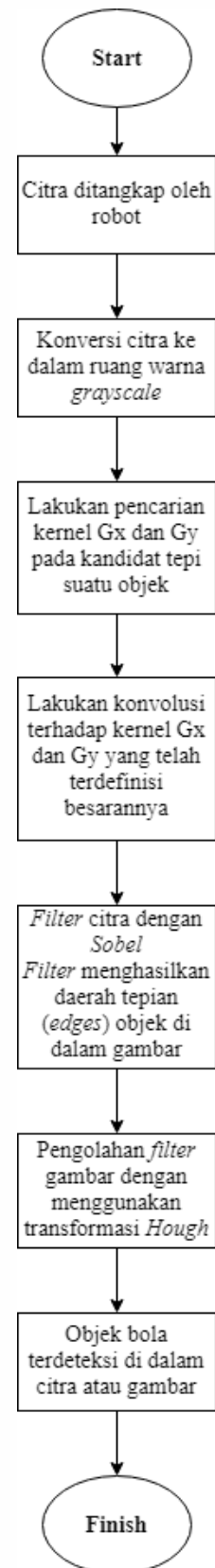
Sudut orientasi dari bagian tepi (*edge*) relatif terhadap *grid* pikselnya yang menimbulkan gradien spasial dan dinyatakan dengan:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad \dots(3)$$

Pada kasus ini, orientasinya nol berarti arah kontras maksimumnya dari kiri ke kanan gambar (*horizontal*) dan sudutnya diukur berlawanan jarum jam dari orientasi ini.

Dalam penggunaan *Sobel Operator* untuk melakukan pengolahan citra pada robot, citra yang telah ditangkap dalam ruang warna *RGB* harus diubah terlebih dahulu menjadi ruang warna *grayscale* sebelum diolah dengan *Sobel Operator* untuk menentukan titik tepi dari sebuah objek yang terdeteksi di dalam citra yang ditangkap oleh robot. Selanjutnya, dilakukan pencarian *kernel Gx* dan *Gy* yang akan digunakan untuk menentukan apakah tepi suatu objek telah terdeteksi di dalam sebuah citra dengan melakukan pencarian pada piksel gambar secara vertikal dan horizontal.

Setelah tepian suatu objek telah terdeteksi di dalam citra atau gambar, selanjutnya dilakukan *filtering* bentuk objek dengan menggunakan salah satu cara yaitu transformasi *Hough* untuk mendeteksi lingkaran atau biasa disebut dengan *HoughCircle*. Di dalam transformasi *Hough*, *filter* gambar yang terdeteksi tepian sebuah objek akan diolah dan dari lingkaran-lingkaran yang merupakan kandidat *blob* bola yang telah terdeteksi tepiannya akan ditentukan sebuah lingkaran yang berdasarkan parameter dan pembobotan tertentu diprediksi merupakan *blob* bola yang berada di dalam lapangan pertandingan.



Gambar 6 Flowchart pengolahan citra dengan Sobel Filter

III. HASIL DAN DISKUSI

Untuk melakukan evaluasi dari desain robot Dago Hoogeschool, dilakukan implementasi pada program *DARWIN-OP* yang dimodifikasi. Dengan menggunakan *dynamic balancing* robot Dago Hoogeschool dapat menjadi lebih stabil dan mampu berjalan diatas rumput sintetis dengan ketebalan 2,75 cm. Gambar 7 menunjukkan *streaming vision* tanpa penerapan Sobel filter. Pada gambar 8, menunjukkan implementasi *vision* dengan menggunakan Sobel filter.



Gambar 7 Streamer robot tanpa Sobel Filter



Gambar 8 Streamer robot dengan implementasi Sobel Filter

Sedangkan pada hasil implementasi *vision* memiliki hasil sebagai berikut.

Tabel 3 Data pendeteksian bola dengan Sobel Filter dan HoughCircle

No.	Jarak Bola terhadap Robot	Keterangan
1	20 cm	Terdeteksi
2	40 cm	Terdeteksi
3	60 cm	Terdeteksi
4	80 cm	Terdeteksi
5	100 cm	Terdeteksi
6	120 cm	Terdeteksi

Melalui tabel 3 diperoleh hasil bahwa robot dapat melakukan pendeteksian bola menggunakan Sobel

Filter dan HoughCircle pada jarak bola terhadap robot hingga 120 cm.

IV. KESIMPULAN

Dalam paper ini, kami menyajikan deskripsi tim Dago Hoogeschool yang akan menjadi kontestan dalam KRSBI Humanoid tingkat nasional 2018. Robot Dago Hoogeschool mampu mendeteksi bola dengan jarak hingga 120 cm, serta mampu membedakan objek berupa bola, dan objek berwarna putih lainnya di dalam lapangan. Selain itu, robot juga mampu melakukan *dynamic balancing* sehingga dapat berjalan dengan kecepatan maksimum 31.5 cm/s secara stabil di atas rumput sintetis setebal 2.75 cm.

DAFTAR PUSTAKA

- [1] Rule KRSBI Humanoid 2017. 21 Desember 2016.
- [2] Rule KRSBI Humanoid 2018. 8 Desember 2018.
- [3] https://www.hardkernel.com/main/products/prdt_info.php, Hardkernel Products : Orroid-XU4 Feature, diakses pada 9 Juni 2018 pukul 20.30.
- [4] <https://en.manu-systems.com/DEV-CMPS11.shtml>, CMPS11-Tilt : Overview. Diakses pada 8 Juni 2018 Pukul 19.30.
- [5] Luqman, et. al. Final Project Document Humanoid Robot Soccer League, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2015.
- [6] <https://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm>, Sobel Edge Detector, diakses pada 8 Juni 2018 pukul 21.50.
- [7] Fisher, R, et al. "Sobel Edge Detector". The University of Edinburgh School of Informatics, 2003.
- [8] https://www.tutorialspoint.com/dip/sobel_operator.htm, Sobel Operator, diakses pada 10 Juni 2018, pukul 15.26.
- [9] Sanusi, et. al. Humanoid Robosoccer Goal Detection using Hough Transform. *International Conference on Control, Electronics, Renewable Energy, and Communications*, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2015.

Rancangan Robot Humanoid Tim Devata 2018

Anthony Susanto^{a,1}, Mulyanti^{b,2}, Alvan Prayogo^a, Susanto^b, Janson Hendryli^b

^aProgram Studi Teknik Industri, Fakultas Teknik, Universitas Tarumanagara,
Jalan Letjen S. Parman No.1, DKI Jakarta

^bProgram Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara
Jalan Letjen S. Parman No.1, DKI Jakarta

¹anthonysusanto080@gmail.com; ²yangthing22@gmail.com

Abstrak - Para peserta KRSBI-H 2018 senantiasa dituntut untuk dapat mengembangkan kemampuan robotnya dalam mekanika, elektronika, hingga algoritma seiring dengan perkembangan teknologi yang diterapkan dalam pertandingan. Rancangan perangkat mekanik dan proses pengolahan sistem yang tepat akan mempengaruhi kemampuan robot dalam melakukan setiap algoritma dan perintah yang dihasilkan. Pada makalah ini, akan dibahas tentang pengembangan robot DEVATA 2018, serta sistem deteksi bola dan gawang pada robot DEVATA 2018. Rancangan Robot DEVATA 2018, merupakan pengembangan dari robot DEVATA A 2016, dan DEVATA NEW GENERATION 2017. Sebagai robot untuk bermain sepak bola, sangatlah penting setiap robot dilengkapi dengan kemampuan untuk dapat mendeteksi bola dan gawang. Deteksi bola dan gawang dilakukan melalui proses pengolahan citra yang diperoleh dari kamera yang terdapat pada setiap robot. Hasil gambar yang diperoleh diproses melalui pengolahan citra menggunakan library tambahan OpenCV dalam beberapa proses pengolahan tersebut, seperti *thresholding*, *morphology*, dan *hough line transform*.

Kata Kunci – KRSBI-H 2018, DEVATA, Deteksi Bola, Deteksi Gawang, Image processing.

I. PENDAHULUAN

Kontes Robot Sepak Bola Indonesia Humanoid (KRSBI-H) 2018 adalah penyelenggaraan KRSBI-H tahun ke-enam semenjak KRSBI-H diadakan pertama kali di tahun 2013. Cikal bakal KRSBI-H adalah Kontes Robot Cerdas Indonesia (KRCI) *Robot 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 lapangan yang menjadi lebih besar dari tahun lalu yaitu 9x6 meter.

Universitas Tarumanagara (UNTAR) kembali berpartisipasi dalam Kontes Robot Sepak Bola Indonesia *Humanoid* (KRSBI-H) 2018. Dalam kontes tahun ini, Tim KRSBI-H UNTAR merancang robot sebanyak 4 buah yang terdiri dari DEVATA 1, DEVATA 2 sebagai penyerang, DEVATA 3 sebagai

pemain bertahan, dan DEVATA 4 sebagai penjaga gawang.

Sebagai robot yang memiliki tujuan untuk bermain sepakbola, robot dari tim DEVATA 2018 harus memiliki perangkat keras (*hardware*) dan perangkat lunak (*software*) yang dapat mendukung robot untuk dapat berjalan, serta memiliki kemampuan untuk mengenali bola dan gawang. Perangkat keras dan perangkat lunak yang digunakan harus dapat bekerja secara berdampingan sehingga dapat menghasilkan fungsi yang sesuai untuk menjadi robot yang dapat bermain sepakbola.

Robot DEVATA 2018 dilengkapi dengan kamera WebCam yang digunakan untuk mengambil gambar. Setiap gambar yang diperoleh dari kamera, melalui proses pengolahan citra untuk melakukan pengenalan dan pendeteksian suatu objek. Sistem deteksi bola DEVATA 2018 menggunakan pengenalan warna pada ruang warna YCbCr serta koordinat posisi bola dan area lapangan [2]. Sedangkan untuk mendeteksi gawang menggunakan pengenalan warna pada ruang warna HSV dan pengenalan tepian objek berupa garis lurus dengan *Hough Line Transform* [3].

Pada makalah ini akan dibahas tentang pengembangan robot DEVATA pada KRSBI-H 2018, serta sistem deteksi bola dan gawang pada robot DEVATA 2018.

II. TINJAUAN PUSTAKA

DEVATA 2018 adalah tim KRSBI *Humanoid* Universitas Tarumanagara yang terdiri dari 4 buah robot, yaitu DEVATA 1 dan 2 yang berperan sebagai robot penyerang, DEVATA 3 sebagai robot bertahan, dan DEVATA 4 sebagai robot penjaga gawang. DEVATA 1 dan 2 merupakan robot humanoid pengembangan dari hasil rancangan DEVATA A 2016 [4] dan DEVATA NEW GENERATION 2017 [5]. DEVATA 3 merupakan robot hasil rancangan pada KRSBI

2016, dan DEVATA 4 adalah hasil rancangan pada KRSBI 2017.

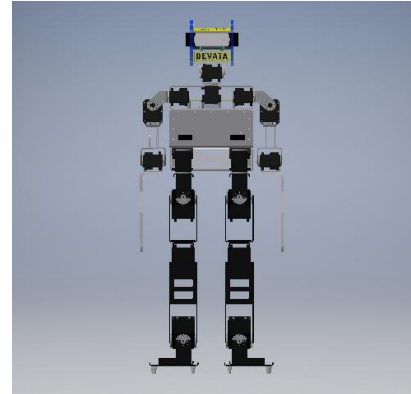
DEVATA 1 dan DEVATA 2 merupakan robot rancangan terbaru yang dimiliki oleh tim KRSBI-H UNTAR pada tahun 2018. Secara menyeluruh, robot DEVATA 1 dan DEVATA 2 memiliki perbedaan antara lain tipe aktuator yang dipilih, serta dimensi mekanik yang digunakan. Spesifikasi, rancang bangun robot, rancangan diagram blok sistem yang digunakan pada DEVATA 1 dan DEVATA 2 dapat dilihat pada **Tabel 1**, **Gambar 1**, dan **Gambar 2**.

Tabel 1. Spesifikasi Rancangan Robot DEVATA 1 dan DEVATA 2

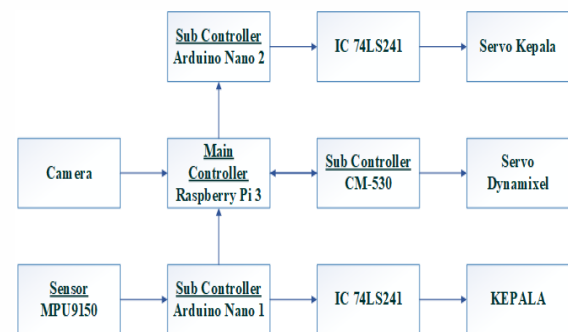
		DEVATA 1	DEVATA 2
Aktuator	Kepala	2 x MX-28	2 x MX-28
	Tangan	6 x MX-28	6 x MX-28
	Kaki	12 x MX-64	8 x MX-64 4 x MX-28
CPU	Main Controller	Raspberry-Pi 3 Model B	
	Sub Controller	CM-530, Arduino Nano	
Sensor	Kamera	Logitech C-930	Logitech C-920
	Gyro, Accelero, dan Compass	MPU-9150	
Power		Battery LiPo 3 Cell 2200 mAh	
Dimensi	Tinggi	69.5 cm	62.5 cm
	Telapak Kaki	60 mm x 100 mm	
	Tangan	205 cm	197 cm
	Lebar Tubuh	205 cm	

Robot DEVATA 2018 dilengkapi dengan beberapa perangkat pemroses yang terbagi menjadi *main controller* dan *sub controller*. *Main Controller* yang digunakan pada robot DEVATA 2018 adalah Raspberry Pi 3 Model B [6] yang berfungsi sebagai pusat pengolahan data, membaca dan mengolah citra, serta menghasilkan perintah untuk melakukan gerakan pada robot. Sedangkan *sub controller* yang digunakan adalah CM-530 [7] sebagai pengendali servo untuk melakukan gerakan robot, dan Arduino Nano [8] sebagai pemroses pembacaan sensor keseimbangan

dan arah. Sensor keseimbangan dan arah yang digunakan pada robot DEVATA 2018 adalah MPU-9150 yang terdiri dari *gyroscope*, *accelerometer* dan *compass* [9]. Untuk menggerakkan servo Dynamixel kepala dan tangan, dibutuhkan IC 74LS241 untuk merubah komunikasi arduino secara *full duplex* ke *half duplex* sesuai cara komunikasi servo Dynamixel MX-28 [10].



Gambar 1. Rancang Bangun Robot DEVATA 2018



Gambar 2. Diagram Blok Arsitektur Sistem Robot DEVATA 2018

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. Proses pendeteksian objek dapat dilakukan dengan melihat setiap ciri khusus pada setiap objek target. Ciri khusus yang dapat digunakan untuk pengenalan suatu objek seperti pengenalan warna, pengenalan bentuk atau kontur objek, dan pengenalan pola. Untuk mendapatkan hasil yang maksimal, proses pengenalan suatu objek dilakukan melalui beberapa tahapan proses pengenalan. Seperti untuk melakukan deteksi pada objek bola dapat dilakukan dengan melakukan proses pengenalan warna dan pengenalan kontur tepian objek. Sedangkan untuk proses pengenalan objek gawang dapat

dilakukan dengan proses pengenalan warna dan garis lurus pada setiap tepi objek.

Visi pada robot DEVATA 2018 diproses menggunakan bahasa pemrograman C++ dengan *library* tambahan yaitu OpenCV. OpenCV (*Open Source Computer Vision Library*) adalah sebuah pustaka perangkat lunak yang ditujukan untuk pengolahan citra dinamis secara *real-time*, yang dibuat oleh Intel, dan sekarang didukung oleh Willow Garage dan Itseez [12]. *Image* merupakan hasil citra yang diperoleh dari suatu kamera yang diproses pada suatu komputer atau sistem tertanam melalui suatu proses pengenalan dan pengolahan.

Untuk mendapat melakukan pengenalan warna pada *image*, terdapat beberapa proses yang dilalui seperti proses binerisasi (*thresholding*) dan *morphology* (proses *erode* dan *dilate*). Proses binerisasi dilakukan untuk melakukan proses filterisasi warna yang akan dicari sesuai dengan ruang warna yang digunakan seperti RGB, HSV, atau YCbCr. Proses *morphology* dilakukan untuk menghilangkan *noises* yang terdapat pada *image* melalui proses *erode* dan *dilate* [13]. Proses *erode* berfungsi untuk memperkecil atau mengikis tepian objek pada suatu *image* atau citra, sedangkan proses *dilate* merupakan lawan dari proses *erode* yaitu proses untuk memperluas atau memperbesar tepian objek pada suatu *image* atau citra.

Untuk melakukan proses pengenalan bentuk tepian pada objek, dapat dilakukan dengan mengenali bentuk tepian pada citra yang telah melalui proses pengenalan warna. Untuk mengenali objek berbentuk persegi, maka tepian pada objek berjumlah 4, sedangkan segitiga berjumlah 3, dan bentuk lainnya.

Untuk melakukan proses pengenalan pada garis lurus tepian objek, proses *hough line transform* dapat digunakan untuk melakukan proses pengenalan. Hough line transform adalah proses pengenalan berupa tepian objek dengan mengenali setiap titik dan panjang pada garis objek [14].

III. MEKANIK ROBOT DEVATA 2018

Mekanik pada robot DEVATA 2018 menggunakan hasil rancangan dari tim DEVATA yang mengacu dari kerangka mekanik DARwIn-OP. Rancangan kerangka mekanik robot DNG mengacu pada konsep robot DARwIn-OP dan hasil rancangan robot DEVATA A.

DEVATA 1 dan DEVATA 2 menggunakan bahan aluminium pada bagian kerangka kaki, badan, dan tangan sedangkan pada kepala menggunakan bahan akrilik untuk memperingan beban pada kepala.

Hasil dari pengembangan robot DEVATA 2018 dapat dilihat pada **Gambar 3**.

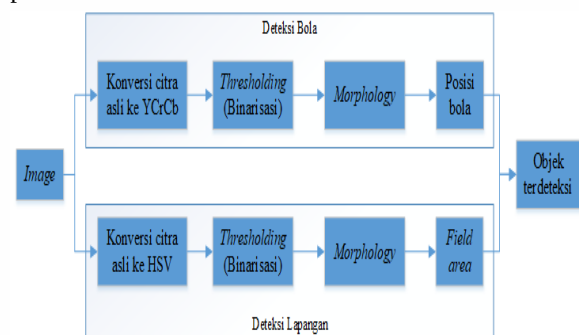


Gambar 3. Robot DEVATA 2018

IV. SISTEM PENDETEKSI BOLA ROBOT DEVATA 2018

Terdapat perangkat lunak dan keras yang digunakan dalam sistem pendeteksi bola robot DEVATA 2018. Perangkat lunak yang digunakan adalah OpenCV sebagai *library* pendukung pengolahan citra (visi dari robot), dan C++ sebagai bahasa pemrogramannya. Sedangkan perangkat keras yang digunakan adalah kamera Logitech C-930, C-920, dan C-525, Raspberry Pi 3 sebagai pemrosesannya, dan laptop sebagai perangkat pembantu dalam menjalankan proses yang lebih berat.

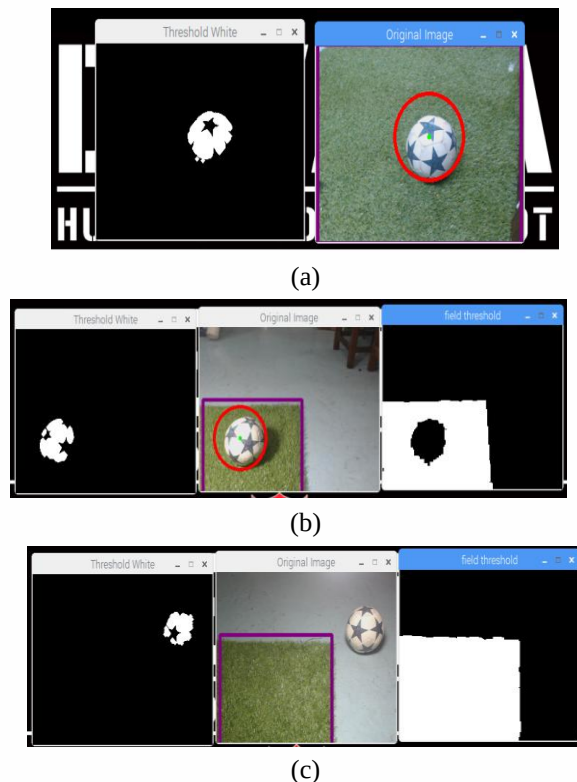
Sistem pendeteksi bola pada robot DEVATA 2018 dapat dilihat pada diagram blok pada **Gambar 4**.



Gambar 4. Diagram Blok Sistem Deteksi Bola pada Robot

Proses awal pada sistem pendeteksi bola robot DEVATA 2018 adalah melakukan pengambilan *image* menggunakan kamera, setelah itu terdapat 2 proses yang dipisah, dilakukan konversi dari *image* ke YCbCr agar dapat mendeteksi warna bola, dan konversi *image* ke HSV untuk mendeteksi warna lapangan, diikuti dengan proses binerisasi untuk melakukan *filter* warna bola dan lapangan. Pada proses *morphology* terjadi proses *erode* dan *dilate* untuk menghilangkan *noises* dari *image*. Dari proses-proses tersebut ditemukan posisi bola dan juga *field area* sehingga objek

terdeteksi. Hasil dari pengolahan citra dari robot DEVATA 2018 dapat dilihat pada **Gambar 5**.



Gambar 5. Hasil Binerisasi dan Deteksi Bola (a), dalam lapangan (b) dan luar lapangan (c)

Dilakukan pengujian pada sistem deteksi bola DEVATA 2018 dengan cara meletakkan bola di dalam lapangan dan luar lapangan pada jarak yang berbeda. Pengujian dilakukan dengan melihat keberhasilan sistem untuk mendeteksi bola di dalam lapangan dan luar lapangan. Hasil pengujian sistem deteksi bola DEVATA 2018 dapat dilihat pada **Tabel 2**.

Tabel 2. Hasil Pengujian Sistem Deteksi Bola DEVATA 2018

Jarak (cm)	Dalam Lapangan	Luar Lapangan
20	Terdeteksi	Tidak Terdeteksi
40	Terdeteksi	Tidak Terdeteksi
60	Terdeteksi	Tidak Terdeteksi
80	Terdeteksi	Tidak Terdeteksi
100	Terdeteksi	Tidak Terdeteksi
200	Terdeteksi	Tidak Terdeteksi
300	Terdeteksi	Terdeteksi
400	Terdeteksi	Terdeteksi

500	Tidak Terdeteksi	Tidak Terdeteksi
-----	------------------	------------------

V. SISTEM PENDETEKSI GAWANG ROBOT DEVATA 2018

Pada robot DEVATA, sistem deteksi gawang menggunakan algoritma *hough line* yang bekerja dengan mencari garis untuk diklasifikasikan sebuah objek gawang. Hasil algoritma *hough line* dipengaruhi oleh sudut pengambilan citra sebab perbedaan ukuran proyeksi tiang gawang mempengaruhi proses seleksi citra.

Pada proses *color thresholding*, semua objek dengan warna putih dideteksi. Kemudian algoritma *hough line* menyeleksi semua objek warna putih dan menentukan garis yang vertikal yang diyakini sebagai tiang gawang dengan melakukan deteksi gawang. Pada proses deteksi, robot melakukan mencari tiang gawang pertama dan mencari tiang gawang kedua, sehingga robot dapat menentukan titik arah gawang tersebut.

Sistem pendeteksi gawang pada robot DEVATA 2018 dapat dilihat pada diagram blok pada **Gambar 6**.

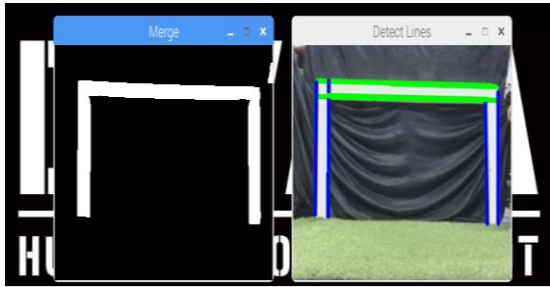


Gambar 6. Diagram Blok Sistem Deteksi Gawang pada Robot

Proses awal pada sistem pendeteksi Gawang robot DEVATA 2018 adalah melakukan pengambilan *image* menggunakan kamera dan dilakukan konversi *image* ke HSV untuk mendeteksi gawang dengan proses binerisasi untuk *filter* warna gawang. Pada proses *morphology* terjadi proses *erode* dan *dilate* untuk menghilangkan *noises* dari *image*.

Proses *canny* bertujuan untuk mendapatkan garis tepi gawang dan melakukan transformasi ke *hough line*. Proses ini digunakan untuk mendeteksi garis lurus yang dihasilkan oleh *filter canny*. Hal itu dilakukan agar dapat dideteksi lebih spesifik sehingga robot akan lebih jelas dan cepat dalam mengenali gawang. Setelah melakukan transformasi *hough line*, robot dapat mencari titik sudut gawang secara vertikal agar mudah membedakan objek tiang gawang dan objek lainnya. Kemudian titik sudut gawang tersebut disatukan secara vertikal menjadi garis lurus dan objek terdeteksi sebagai gawang. Hasil

deteksi gawang dari robot DEVATA 2018 dapat dilihat pada **Gambar 7**.



Gambar 7. Hasil Binerisasi dan Deteksi Gawang

VI. KESIMPULAN

Tim DEVATA 2018 mengembangkan robot DEVATA 1 dan DEVATA 2 sehingga memiliki rancangan mekanik menggunakan bahan aluminium. Pada pengembangan dari sistem pendeteksi bola, robot DEVATA melakukan konversi *image* ke YCbCr dan HSV sehingga dapat mendeteksi bola serta dapat membedakan antara bola dengan lapangan. Sedangkan pada pengembangan sistem pendeteksi gawang, gawang berhasil dideteksi menggunakan metode penyatuan titik sudut dari gawang secara vertikal.

VII. REFERENSI

- [1] Buku Panduan KRSBI Humanoid 2018 versi 8.12.17. Direktorat Kemahasiswaan, Direktorat Jenderal Pembelajaran dan Kemahasiswaan, Kementerian Riset, Teknologi dan Pendidikan Tinggi. Jakarta, Indonesia. 2017. pp. 1-6
- [2] Susanto, dkk. "Bareleng FC Team Description Paper Humanoid Kid Size League of RoboCup 2017" [online], Available: <https://www.robocuphumanoid.org/hl-2017/teams/>
- [3] Susianto, Budi. "Deteksi Fitur Gawang Pada Robot Berbasis Hough Line Transform", Institut Bisnis dan Informatika STIKOM, Surabaya, 2018
- [4] Danieal, J. Chayadi, K. William, dan A.B.Dharmawan. "Rancang bangun sistem komunikasi data robot DEVATA A dalam KRSBI 2016", The 4th Indonesia Symp. on Robot Soccer Competition, PENS, Surabaya, Juni 2016. pp. 119-125
- [5] Anthony Susanto, J. Chayadi, Danieal, dan A. B. Dharmawan. "Rancang bangun sistem komunikasi data robot DEVATA NEW GENERATION", The 5th Indonesia Symp. on Robotics System and Control, UPI, Bandung, Juli 2017. pp. 198-202
- [6] Raspberry Pi Foundation. (2016). What is a raspberry pi?[online]. Available: <https://www.raspberrypi.org/>
- [7] ROBOTIS. (2010). ROBOTIS e-manual v.1.29.00 CM-530[online]. Available: <http://support.robotis.com/en/product/controller/cm530.htm>
- [8] Arduino. (2017). Arduino Nano[online]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardNano>
- [9] MPU-9150 Product Specification Revision 4.3. InvenSense Inc. California. USA. 2013. pp. 6-10
- [10] Y.Y. Joeferie, A. Yuliansyah, W.R. Nugraha, Felisyia, and R.A.Silva, "Sistem pergerakan servo dynamixel AX series berbasis Arduino pada robot NAMA EKA", in The 3rd Indonesian Symp. on Robot Soccer Competition, UMY, Yogyakarta, Juni 2015. pp. 130-133
- [11] Basuki, Ahmad. 2005. *Pengolahan Citra Digital Menggunakan Visual Basic*. Graha Ilmu: Yogyakarta.
- [12] Itseez leads the development of the renowned computer vision library OpenCV. Available: <http://itseez.com/>
- [13] Anggri Y.P. (2017). Operasi Morfologi pada Pengolahan Citra. Available: <https://devtrik.com/opencv/operasi-morfologi-pada-pengolahan-citra/>
- [14] Ivan J. (2018) Mendeteksi Garis dengan Hough Line Transform. Available: <https://www.ivanjul.com/mendeteksi-garis-hough-line-transform/>

Implementation of De Casteljau's Algorithm for Pattern Generation on BIOLOID Based Humanoid Robot's Leg Movement

Dimas Fajrian Nugroho, Andi Dharmawan, Mufti Irawan Fikri, Iham Nur Ahmad, Muhammad Ridho Fuadin
Universitas Gadjah Mada Bulaksumur, Yogyakarta, Indonesia, 55284
dimas.fajrian.n@mail.ugm.ac.id

Abstract – This paper represents a method to generate a pattern of an robot end-effector's movement based on *De Casteljau's Algorithm*. The method generates Bézier curves which is equivalent to linear interpolation as the reference of the end-effector's coordinate in certain plane. The Bézier curves are widely used because it is compact and intuitive. From that point of view, this method is quiet implementable for the pattern generation of humanoid robots leg. This method is easy to compute since it only used repeated linear interpolation.

Keywords: Interpolation, Pattern generation, Robot Motion

I. INTRODUCTION

BIOLOID based robots is by its default configuration only using a sort of "record and play" mechanism in generating its motion pattern. The pattern that generated by the default system only specifies the initial point and the end point of the robot end-effector's motion path. Based on these behavior, it is difficult to control the path between the initial point and the end point of the robot motion path. For more advance task such as walking is more complex and "record and play" method in most of BIOLOID robots could not make the robot generate a desired walking pattern since an uncertainty could turns out and make the pattern generation more difficult. Hence, a methods to generate or to control the path between the starting point and the end point is required.

De Casteljau's Algorithm is method that offers flexibility in generating patterns with a compact method to be implemented [1]. Casteljau's Algorithm generates Bézier curve which is used in previous researches [2][3]. This method also optimize the robot programs faster and easy to implemented [4]. This curve generates points which play a role to control the robot's end-effector movements. This method has also solved problems in CNC tool path planning such as saving machining time, reducing number of CNC blocks and reducing acceleration and deceleration jerks [5]. It was also used as the dynamic obstacle avoidance and trajectory learning for autonomous mobile robots [6].

In this research, the Casteljau's algorithm is used

to generate the motion pattern of the robot's leg. The motion is generated using three control points with adjusted interval to determine the accuracy of the pattern.

II. METHODS

The Robot's end-effector or the tip of the robot's manipulator [7] moves in a 2 dimensional axis. To enable the robot's tip moves, the angle of the robot's joints must be calculated in sequence of time. The robot joints values are determined using inverse kinematic equations. After the joint angles are determined, the Casteljau's Algorithm is used to determine the pattern of the robot's end-effector movements in sequence of time.

A. Inverse Kinematic of The Robot

The inverse kinematic method determines the joint angles of the robot using analytical approach [8][9]. Figure 1 describes the sagittal view of the robot body. y and x is the determined coordinate reference, L_1 and L_2 are the length of the leg links. In this case to determine the θ_{3b} value, it is considered equal as θ_{1b} and θ_{1b} value is generated using the $\text{atan2}(x, y)$ function. θ_1 is the angle of the hip pitch joint, θ_2 is the angle of the knee joint, and θ_3 is the angle of the ankle pitch joint. L_{xy} is the distance between the hip and the ankle of the robot leg.

Firstly, we need to determine the L_{xy} length using Equation (1).

$$L_{xy} = \sqrt{X^2 + Y^2} \quad (1)$$

Then, the angle value of the hip pitch is determined by the sum of θ_{1a} and θ_{1b} .

$$\theta_1 = \theta_{1a} + \theta_{1b} \quad (2)$$

θ_{1a} is an angle between y and L_{xy} and also one of the angle of the right triangle which also consists of the x length value. Therefore, we can generate the θ_{1a} value by using the $\text{atan2}(y, x)$ function. θ_{1b} is an angle between L_{xy} and L_1 and contained in a scalene triangle. Hence, we use Equation (3) to determine the

value of the angle which is convenient with the principal of the scalene triangle.

$$\theta_{1b} = \text{acos} \left(\frac{L_1^2 + L_{xy}^2 - L_2^2}{2 \cdot L_1 \cdot L_{xy}} \right) \quad (3)$$

This procedure is applied to the Equation (4) and (5) to generate the value of θ_{2a} and θ_{3a} respectively, because those angles is contained in the same scalene triangle.

$$\theta_{2a} = \text{acos} \left(\frac{L_1^2 + L_2^2 - L_{xy}^2}{2 \cdot L_1 \cdot L_2} \right) \quad (4)$$

$$\theta_{3a} = \text{acos} \left(\frac{L_{xy}^2 + L_2^2 - L_1^2}{2 \cdot L_{xy} \cdot L_2} \right) \quad (5)$$

θ_{2a} and θ_{3a} is required to generate θ_2 as the knee joint and θ_3 as the ankle pitch value respectively. Later on, θ_2 value is generated by subtracting the $\frac{\pi}{2}$ by θ_{2a} which is described in Equation (6).

$$\theta_2 = \frac{\pi}{2} - \theta_{2a} \quad (6)$$

For the ankle pitch value, θ_3 is determined by decreasing the θ_{3a} by θ_{3b} .

$$\theta_3 = \theta_{3a} - \theta_{3b} \quad (7)$$

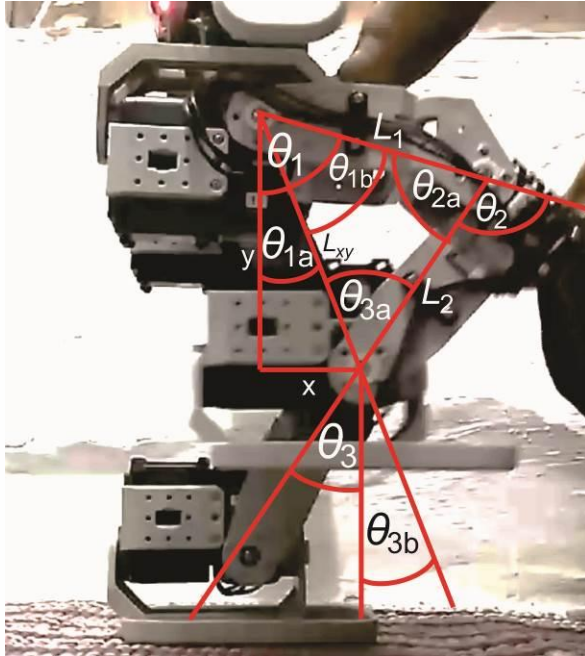


Figure 1. Kinematic and joints angle description

B. De Casteljau's Algorithm

In the robot system, as illustrated in Figure 2, the implementation of the algorithm only takes three control points [10]. Equation (8) is the general equation to generate the Bézier curves with A and B as the control points.

$$P(t) = (1-t)A + tB \quad (8)$$

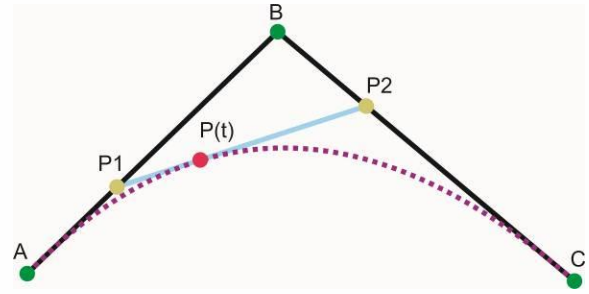


Figure 2. Bézier curve description using De Casteljau's Algorithm

Points in the curve described in Figure 2 are considered as position vector, which means that each of the point contains a coordinate in two dimensional axis (x and y). A, B and C are the main control points. P1 and P2 is the control point of P(t). t is the resolution of the end-effector point and its value goes from 0 to 1. Respectively, P1 and P2 value generated using Equation (9) and Equation (10).

$$P1 = (1-t)A + tB \quad (9)$$

$$P2 = (1-t)B + tC \quad (10)$$

By using the general equation to connects P1 and P2 to determine the P(t) value, the Equation (11) is used.

$$P(t) = (1-t)P1 + tP2 \quad (11)$$

Equation (11) could also be determined as below by substituting the Equation (9) and (10) into Equation (11).

$$P(t) = (1-t)^2 A + 2(1-t)tB + t^2 C \quad (12)$$

Finally, to take the complete forms of the Equation (12) into account, the Equation (13) describes the pattern generator equation.

$$\begin{bmatrix} P_x(t) \\ P_y(t) \end{bmatrix} = (1-t)^2 \begin{bmatrix} A_x \\ A_y \end{bmatrix} + 2(1-t)t \begin{bmatrix} B_x \\ B_y \end{bmatrix} + t^2 \begin{bmatrix} C_x \\ C_y \end{bmatrix} \quad (13)$$

As further explanations, $P_x(t)$ and $P_y(t)$ are the points result of the motion pattern. A_x and A_y is the first control point. B_x and B_y is the second control point and finally C_x and C_y is the final control points where the end-effector would stop moving at this position.

III. RESULT AND DISCUSSION

As the result, the algorithm implemented in the system described in Figure 3.

To adjust the pattern of the movements, by which

is implemented, there are three variables to consider such as the initial control point (**P1**), the second control point (**P2**), the final control point (**P3**) and t .

The control initial control point show the first position of the feet and the final position (1,0) is discovered in the final control point coordinate. Based on the Figure 4 the initial control point coordinate is (0,0). Afterwards, the second control point takes role in determining the swing height of the robot feet.

In the equation, t goes from 0 to 1, which means we need convert it to the desired time range to control the velocity of the system. To convert the t value into the time format, it is done by converting the ratio of each value where the highest value of t is 1 and the highest value of the time sequence is n or as has been applied, the value of n is 3 seconds. Therefore, the end-effector moves along the generated pattern in 3 seconds.

The increment value of t takes role in determining the pattern accuracy. The more iteration in the algorithm the more points generated by the system and the more points generated by the equations, the more accurate the motion will be.

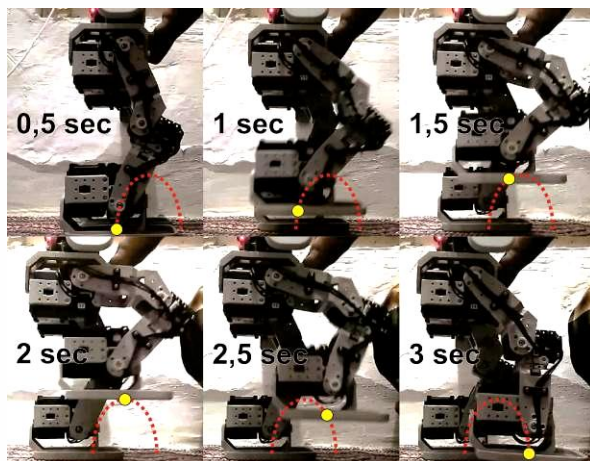
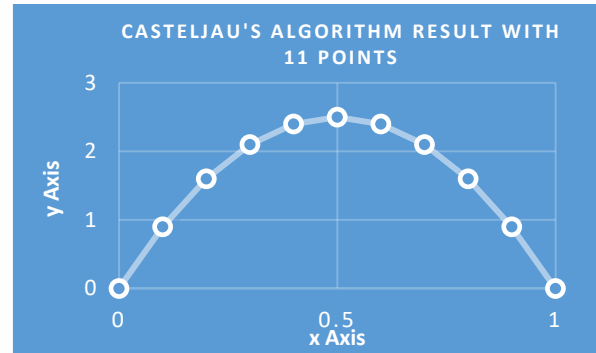
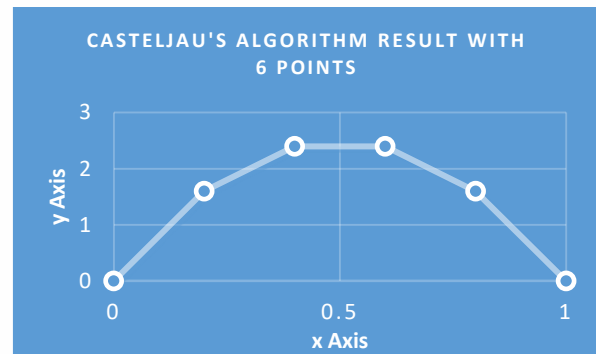


Figure 3. The pattern generator result in Bioid Humanoid Robot

The comparison of the pattern accuracy is described in Figure 4 (a) and (b). It is shown that Figure 4 (a) generate more points than the one in the Figure 4 (b). The result in Figure 4 (a) shows a smooth movement along the plane.



(a)



(b)

Figure 4. Casteljau's algorithm result comparison

The high level of accuracy requires more computation speed to make the robots movements realtime. Simply it means that the more points made by the generator, more computation load will be required or there might be an odd behavior generated by the system if the load is too heavy. The result shows that using Casteljau's Algorithm, the operator of the robot could set the convenient precision level for the robot's movement.

IV. CONCLUSION

The Casteljau's Algorithm offers a flexibility in determining the accuracy of the robot movement patterns. This method is implementable in motion pattern generation. This method is very intuitive since we could exactly describe the pattern by adjust the position of the control points.

V. FUTURE WORKS

This method is implemented in two dimensional space. The implementation of the method in three dimensional space is extremely possible and would be a complement for the current system. The potential of the Casteljau's algorithm shows that the method could be implemented in many branches of robotics especially motion planning.

VI. REFERENCES

- [1] C. Alta, "The De Casteljau algorithm on SE (3)," no. October, 2016.
- [2] T. Popiel and L. Noakes, "Bézier curves and C² interpolation in Riemannian manifolds," vol. 148, pp. 111–127, 2007.
- [3] T. Popiel and L. Noakes, "C² spherical Bézier splines," vol. 23, pp. 261–275, 2006.
- [4] T. Horsch and B. Jiittler, "Cartesian spline interpolation for industrial robots," vol. 30, no. 3, pp. 217–224, 1998.
- [5] H. N. Fitter, A. B. Pandey, D. D. Patel, and J. M. Mistry, "12th GLOBAL CONGRESS ON MANUFACTURING AND MANAGEMENT, GCMM 2014 A review on approaches for handling Bezier curves in CAD for Manufacturing," *Procedia Eng.*, vol. 97, pp. 1155–1166, 2014.
- [6] T. Chaudhry, T. Gulrez, and A. Zia, "Bézier Curve based Dynamic Obstacle Avoidance and Trajectory Learning for Autonomous Mobile Robots," pp. 1059–1065, 2010.
- [7] R. N. Jazar, *Theory of Applied Robotics: Kinematics, Dynamics, and Control*. 2010.
- [8] S. Kajita, F. Kanehiro, K. Kaneko, and K. Fujiwara, "Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point," pp. 1620–1626, 2003.
- [9] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi, *Introduction to Humanoid Robotics*. Heidelberg: Springer Berlin Heidelberg, 2014.
- [10] P. Crouch, G. Kun, and F. S. Leite, "THE DE CASTELJAU ALGORITHM ON LIE GROUPS AND SPHERES," vol. 5, no. 3, pp. 397–429, 1999.

GPU-Accelerated Monte Carlo Localization for Mobile Robot Soccer with Omnidirectional Camera

Fahri Ali Rahman^{a,1}, Resha Dwika Hefni Al-Fahsi^{a,2}, Igi Ardiyanto^a

^aDepartment of Electrical Engineering and Information Technology,
Faculty of Engineering,
Universitas Gadjah Mada

¹fahri.ali.r@mail.ugm.ac.id

²resha.dwika.hefni.alfahsi@mail.ugm.ac.id

Abstract -- A real time, GPU-accelerated Monte Carlo Localization for Mobile Robot Soccer with Omnidirectional Camera is presented in this paper. Three kinds of sensors are used in the proposed method to achieve robust robot localization on soccer field. Two observation sensors are used in this research. Omnidirectional camera is used for line detection in the soccer field. Digital compass is used for determining the robot orientation. An incremental encoder is equipped in each robot's wheel acts as action sensor to determine the motion of the robot. To achieve real-time performance, we proposed parallelized algorithm using NVIDIA CUDA[®]. The experimental result shows that the proposed algorithm could achieve 4 x speed-up over sequential algorithm.

Keywords: GPU Programming, Robot Localization, Omnidirectional Vision, Sensor Fusion.

I. INTRODUCTION

RoboCup is an international robotic competition and academic event to promote research on intelligent robot. This competition includes robotic soccer competition, service robot competition, and rescue robot competition. The football competition divided into small-size league (SSL), middle-size league (MSL), humanoid robot and standard platform league (SPL). In RoboCup MSL, teams of five fully autonomous robots play soccer with a regular size FIFA soccer ball [1]. The main challenge of RoboCup MSL is its highly dynamic environment where robots and ball are moving fast in soccer field. Furthermore, the robots in the team should be able to work in a distributed multi-robot environment.

Kontes Robot Indonesia is an annual robotic competition in Indonesia [2]. The competition includes robot soccer competition, firefighting robot competition, artistic dancing robot competition and ABU-Robocon Indonesian region. The robot soccer competition itself divided to humanoid soccer robot league which is adopting the rule of RoboCup humanoid soccer league and wheeled-soccer robot league which is adopting the rule of RoboCup Middle Size League. In wheeled soccer robot league of *Kontes Robot Indonesia*, the field size is 9 m wide and 6 m height with maximum three robots for each team. Team fukuRO is the wheeled-robosoccer team of

Universitas Gadjah Mada competing for the wheeled soccer robot league of *Kontes Robot Indonesia*. The robots of team fukuRO is a three-wheeled omnidirectional mobile robot equipped with an omnidirectional camera.

In order to achieve fully-autonomous robot in robot soccer competition, it is important for the robots to know its position in the field. This position information could enable higher level of the robot's intelligence such as decision making, robots coordination, cooperation and path planning. The act of robot to estimate its own position in a known map is known as self-localization. However, as the robot moves, the error from odometry sums up and it is necessary to compensate this uncertainty of robot motion.

Monte Carlo Localization (MCL) is usually used to estimate the robot's position given the motion uncertainty using observation from sensor. This method requires $O(n)$ computational complexity to update robot's position. As the number of particles increase the time required to update is linearly increases. Since the accuracy of the method relies on the number of particles, it is necessary to improve the computation speed of the method.

In this paper, we presents a faster MCL by relieve the computational complexity of the method using GPU technology yields $O(1)$ sometimes $O(\log n)$ computational complexity independent from the number of particles. The rest of this paper is organized as follow, in section II, we briefly review the standard Monte Carlo Localization method. In section III we present our approach to speed up the computation using NVIDIA CUDA[®]. In section IV we report our experiment and section V concludes this paper.

II. MONTE CARLO LOCALIZATION

Monte Carlo Localization is a robot localization method capable of solving global localization problems [3]. The method uses particles filter to estimate posteriors over robot poses. The key idea of the particle filter is to represent the posterior by a set of random state samples drawn from this posterior [3]. In MCL, a probabilistic motion model and sensor

measurement model is applied to the particle filters.

The robot pose is encoded to three-dimensional state-vector:

$$s = [x \ y \ \theta]^T \quad (1)$$

where s is the particle's pose, x and y are robot's position in global x-directional axis and global y-directional axis, respectively, and θ is the robot orientation.

The estimated posture of the robot at time t is defined by:

$$bel(s_t) = p(s_t | z_{1:t}, u_{1:t}) \quad (2)$$

where z is the measurement of robot's sensor and u is the robot's motion. In particle filter algorithm, the belief at time t is estimated recursively from the belief at $t-1$ one step earlier [3].

A. Motion Model for Omnidirectional Mobile Robot

When the motion event is detected, the particle positions are updated. The update procedure should consider the motion uncertainty since in the reality, the robot's motions are subject to noise. Thus, we add a zero-centered random variable with finite variance to the robot's velocity [4].

$$\begin{bmatrix} \hat{v}_x \\ \hat{v}_y \\ \hat{\omega} \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} \quad (3)$$

where v_x and v_y are robot velocity in x and y axis, respectively and ω is the robot's rotational velocity, ϵ is zero-mean error, thus the true velocity is the robot's velocity added with some small additive noise.

B. Sensor Measurement Model

The sensor measurement model is used for computing the importance weights of the particles. For RoboCup MSL, an omnidirectional camera is usually implemented to detect line points transitions on the soccer field [5]. For each detected line points, the probability of a single line transition is computed [5]. The probability value of the line transition points is the sum over the minimum distance of the points to true lines on the soccer field. Since this computation is has to be done for every line transition and every particle, a pre-calculated two-dimensional lookup-table is used to effectively determine the probability of the line points with $O(1)$ computational cost. The visualization of this pre-calculated lookup-table is shown in the figure 1, where brighter points represent smaller distance to field lines.

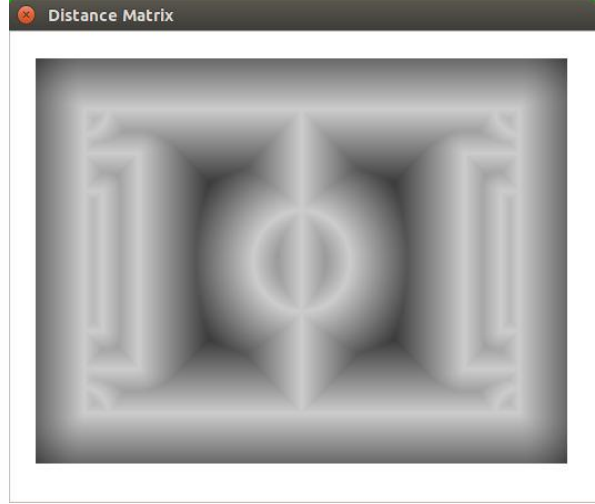


Figure 1. Distance Lookup-Table

Since the field is symmetric, it is hard for the robot to correctly solve the global localization problem without additional information. Hence, we use a digital compass to give heading information to the robot. The final importance weight for each particle is computed as:

$$w_i = \alpha \frac{j}{\sum_j \|f_{i,j}\|} + (1 - \alpha)(\|\phi_i - \phi\|) \quad (4)$$

Where w is the weight particle, α is a normalization factor, f is nearest distance to field lines for detected line transition in global coordinate transformed by particle i , ϕ_i is particle orientation and ϕ is the reading of digital compass.

Additionally, the MCL needs particles resample procedure given the particles importance weight. Before resample procedure is executed, the particles weight need to be normalized.

III. MONTE CARLO LOCALIZATION ON GPU

To achieve high performance MCL algorithm, we implement GPU parallel computation to compute the particle distribution on motion model and sensor measurement model update. The pipeline of motion update with particle data parallelism could be easily implemented on NVIDIA CUDA[®]. Since GPU has many cores and the motion updates between particles are independent, this allows straightforward implementation on GPU and results a constant computation time independent from the number of particles. Since the sensor measurement model is a sum operation over the detected line transition points, the probability value computation is dependent between line points. Furthermore, since the probability value needs to be normalized, the particles become dependent on each other. To cope with this problem, we implement parallel reduction techniques to calculate the probability value for each particle given the sensor measurements.

C. NVIDIA CUDA®

General Purpose computing on Graphical Processing Unit (GPGPU) allows GPU to compute numerical operations that is usually handled by CPU. The major advantages of GPGPU programming is data parallelism that offers computation performance speed-up. One of the most popular GPGPU technology is CUDA®, introduced by NVIDIA. CUDA offers computation on GPU using C as a high-level programming language [6].

D. Motion Update on GPU

The overview of GPU implementation for motion model is depicted in figure 2. In the following figure, every time odometry event is received, the motion update is performed. At initialization, particles are copied to GPU memory from the CPU memory so the particle update routine could be performed parallel on GPU threads. Given odometry data from rotary encoder, robot velocity is calculated and small white noise is generated in each thread. Then each particle is updated given the augmented robot velocity, parallel in each GPU thread. For the subsequent motion update, the copy operation from CPU to GPU is not needed.

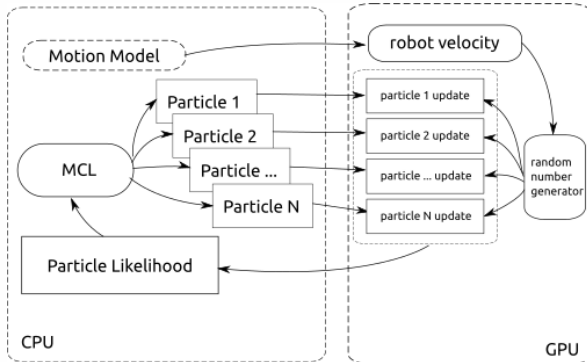


Figure 2. Overview of MCL motion update

E. Measurement/Sensor Update on GPU

The overview of measurement/sensor model computation on GPU is depicted in figure 3. In the following figure, the computation pipeline on the GPU is consist of weight computation for detected line points, angle error computation for measured heading, weight normalization, angle error normalization and final particle's weight calculation.

The line points weight calculation computes each point error calculation on GPU, requiring $N \times S$ threads, where N is the number of particles and S is the number of detected line points. For each detected line points, its position on global coordinate system is calculated using corresponding particle's rotation matrix, and the distance error to nearest line on the field is determined. The weight for the particles is the mean squared error of the distance errors of the transformed points. This calculation is performed using parallel reduction techniques. Then, the weight

between particles is normalized, using the sum of particles weight calculated using parallel reduction.

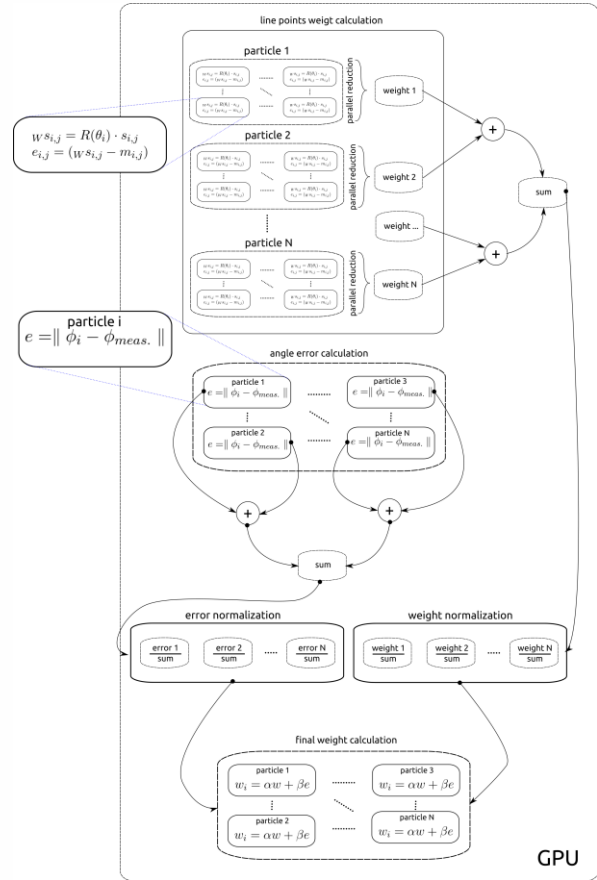


Figure 3. Overview of MCL sensor update

The angle error calculation determines each particle importance weight based on the particle's orientation difference with digital compass measurement. The absolute error value is considered for determining this weight. The angle error calculation is independent between particles so this leads to straightforward implementation on GPU, requiring N threads, where N is the number of particles. After the angle error is calculated, these error needs to be normalized before computing the final weight. Firstly, the sum of error angle is computed using parallel reduction techniques, requiring $N/2$ threads. Second, the error angle for each particle is normalized using this sum.

The final weight computation is weighted sum of normalized angle error and normalized line points weight. This results the final importance weight for each particle. Since this operation is independent between particles, again, this leads to only N threads required.

IV. RESULTS

In this section we report our results. We analyzed the computation needed for motion update and sensor update for CPU and GPU version of MCL. The result is shown in Table 1. The reported computation time

unit is in milliseconds. We use laptop computer with NVIDIA CUDA[®] 9.0 software platform on Intel Core i3 CPU and NVIDIA GEFORCE 920MX GPU hardware platform to perform the experiment.

Table 1. Computation Time Results

N	Average (ms)					
	motion update			sensor update		
	CPU	GPU	speed up	CPU	GPU	speed up
50	2,16157	0,18169	11,8965	8,29674	1,77362	4,6778
100	3,26833	0,26402	12,3789	12,3749	3,68858	3,3549
200	5,71251	0,25684	22,2409	17,8761	5,85965	3,0507
500	14,3046	0,90872	15,7413	36,716	8,96627	4,0949
1000	37,867	1,44686	26,1718	66,0203	15,3347	4,3052
1500	51,6456	2,38706	21,6356	95,864	25,1778	3,8074
2000	94,8268	5,4194	17,4976	119,903	32,0352	3,7428

In the table above, the computation time is averaged over 200 computations based on software and hardware specification above. In the table above, it is shown that the GPU version of MCL is up to 26 x faster than the CPU version for the motion update and up to 4 x faster than the CPU version for sensor update.

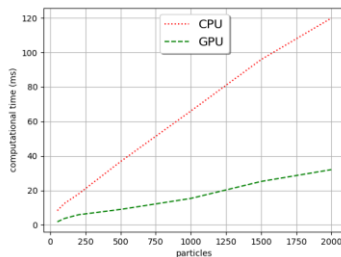
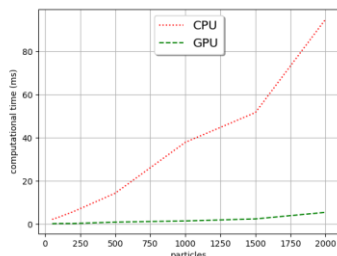


Figure 4. MCL Motion Update Computation Time

Figure
MCL
Motion
Update



Computation Time

From the two pictures above, CPU version of MCL computation time is increasing linearly, $O(n)$ time complexity, for both motion update and sensor update while the GPU version approach to constant time, $O(1)$ time complexity, for the motion update and

logarithmic for the sensor update, $O(\log n)$. Since MCL sensor update consist of resampling particle step, this step is the bottleneck that gives the time complexity of the algorithm [7].

V. CONCLUSION

In this paper, we presented our approach to implements parallel version of MCL. We implement our approach using NVIDIA CUDA[®]. We presented our design of GPU computation pipeline for motion update and sensor update for MCL. We also report the computation time for GPU and CPU version on laptop computer. We show that GPU version of MCL is up to 26 x faster for motion update and up to 4 x faster for sensor update. Both CPU version of MCL time complexity are $O(n)$ while GPU version time complexity of MCL motion update and sensor update yield $O(1)$ and $O(\log n)$ respectively.

VI. REFERENCES

- [1] URL: <http://www.robocup.org/leagues/6>
- [2] URL: <https://kontesrobotindonesia.id/index.html>
- [3] Thrun, Sebastian, Wolfram Burgard, and Dieter Fox. "Probabilistic Robotics". MIT press, 2005.
- [4] Menegatti, Emanuele, Alberto Pretto, Alberto Scarpa, and Enrico Pagello. "Omnidirectional vision scan matching for robot localization in dynamic environments." In *IEEE Transaction on Robotics*, vol. 22, no. 3, pp. 523-535, June 2006.
- [5] Merke, Artur, Stefan Welker, and Martin Riedmiller. "Line based robot localization under natural light conditions." *ECAI 2004 Workshop on Agents in Dynamic and Real Time Environments*. 2004.
- [6] NVIDIA Corporation. "CUDA C Programming Guide." NVIDIA Corporation, 2017.
- [7] Hendeby, Gustaf, et al. "A graphics processing unit implementation of the particle filter." *Signal Processing Conference, 2007 15th European*. IEEE, 2007.
- [8] Rud, Maxim N., and Alexander R. Pantiykhin. "Development of GPU-accelerated localization system for autonomous mobile robot." *Mechanical Engineering, Automation and Control Systems (MEACS), 2014 International Conference on*. IEEE, 2014.
- [9] Chao, Min-An, et al. "Efficient parallelized particle filter design on CUDA." *Signal Processing Systems (SIPS), 2010 IEEE Workshop on*. IEEE, 2010.
- [10] Chitichian, Mehdi, et al. "Distributed computation particle filters on GPU architectures for real-time control applications." *IEEE Transactions on Control Systems Technology* 21.6 (2013): 2224-2238.
- [11] Lenz, Claus, Giorgio Panin, and Alois Knoll. "A GPU-accelerated particle filter with pixel-level likelihood." *VMV. 2008*.
- [12] Liu, Bo, et al. "A self-localization method through pose point matching for autonomous soccer robot based on omni-vision." *Electronic Measurement & Instruments, 2009. ICEMI'09. 9th International Conference on*. IEEE, 2009.
- [13] 14.EEE, 2014. Conference. 2007.

Segmentasi *Circle Hough Transform* untuk Deteksi Bola pada Robot Sepak Bola Beroda Menggunakan Kamera ov7670

Mohamad Kharis^{a,1}, Adrin Thamrin^b

^{a,b}STMIK Adhi Guna, Jl. Undata No. 3, Palu

¹Email korespondensi: moh.kharis85@gmail.com

²Email korespondensi: adrin.adhiguna@gmail.com

Abstrak – Kemampuan penting yang harus dimiliki robot sepak bola selain bernavigasi adalah kemampuan deteksi bola. Kemampuan ini didukung dengan kamera sebagai sensor untuk menghasilkan citra. Pada beberapa penelitian, citra yang diperoleh diolah dengan metode yang berbasis pada warna. Dengan demikian, metode-metode yang dihasilkan tidak dapat digunakan pada kamera-kamera dengan hasil berbentuk citra *grayscale*. Penelitian ini menggunakan metode deteksi objek lingkaran *circle hough transform* pada citra *grayscale* yang dihasilkan kamera ov7670. Metode ini menghasilkan jari-jari dan koordinat pusat lingkaran objek bola pada citra *grayscale* hasil tangkapan kamera ov7670. Lingkaran dihasilkan dengan menerapkan deteksi tepi *canny* untuk mendapatkan titik-titik tepi sebagai titik referensi kemudian menerapkan *circle hough transform* pada setiap titik referensi tersebut.

Kata kunci: circle hough transform, deteksi bola, robot sepak bola

I. PENDAHULUAN

Kontes Robot Sepak Bola Indonesia (KRSBI) merupakan kontes tahunan yang dilaksanakan Direktorat Jenderal Pendidikan Tinggi. Pada kontes tersebut, robot dituntut untuk bisa mengarahkan bola ke *goal* layaknya olah raga sepak bola sebenarnya. Selain kemampuan navigasi robot untuk bergerak mengarahkan robot, kemampuan dasar pendeteksian posisi bola pada setiap robot sangat dituntut pada lomba ini. Untuk mendukung kemampuan ini, robot didukung dengan kamera sebagai sensor untuk mengambil gambar. Gambar yang diperoleh akan diolah dengan algoritma tertentu untuk memastikan pada gambar tersebut terdapat bola atau tidak. Pada beberapa penelitian, deteksi bola pada dilakukan berdasarkan warna. Seperti pada penelitian [1] oleh Nur Khamdi, et al., dan penelitian [2] oleh Satrimo, et al., yang memanfaatkan HSV untuk mendeteksi posisi bola. Penelitian seperti ini hanya mendukung sensor kamera dengan output gambar berwarna. Saat ini di pasaran juga terdapat sensor-sensor dengan kemampuan menghasilkan citra dengan skala abu-abu (*grayscale*), sehingga metode-metode deteksi bola berdasarkan warna tidak dapat digunakan. Pada penelitian ini, diterapkan metode deteksi objek lingkaran menggunakan algoritma *circle hough*

transform untuk menemukan bola. Dengan metode ini, deteksi bola dapat dilakukan pada robot dengan kamera yang menghasilkan gambar *grayscale* seperti kamera ov7670. *Output* yang dihasilkan berupa koordinat pusat dan jari-jari objek bola pada citra hasil tangkapan kamera ov7670.

II. PENELITIAN SEBELUMNYA

Penelitian-penelitian sejenis yang pernah dilakukan sebelumnya menggunakan deteksi lingkaran dengan basis warna. Penelitian[1] menerapkan metode *filtering color* HSV (*hue, saturation, value*). Penelitian ini menggunakan *filter maximum* dan *minimum* terhadap seluruh parameter HSV untuk menemukan bola berdasarkan warna. Dengan demikian harus dilakukan kalibrasi untuk menyesuaikan nilai *filter* terhadap warna bola yang digunakan.

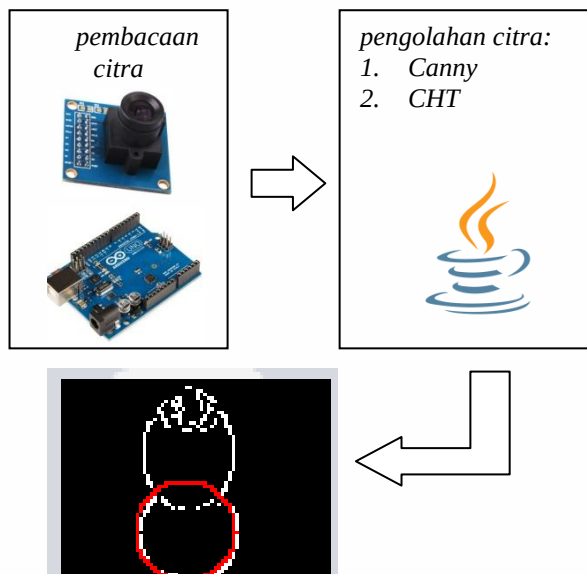
Penelitian lain yang memanfaatkan HSV adalah Penelitian[2]. Penelitian ini memanfaatkan HSV untuk menemukan bola yang berwarna jingga. Warna jingga ditentukan dengan segmentasi warna kuning menggunakan kisaran nilai HSV. Sama seperti penelitian[1], penelitian ini membutuhkan kalibrasi untuk menemukan bola berwarna jingga.

Penelitian lain terhadap deteksi bola juga dilakukan pada penelitian[3] oleh M. T. Kantori, et al., penelitian ini menggunakan kamera Pixy CMUcam5 yang juga membutuhkan kalibrasi warna untuk menemukan bola. Kelemahan kamera Pixy CMUcam5 menurut penelitian ini adalah tidak mampunya kamera ini menghasilkan filter terhadap bentuk bola.

III. METODE

Seperti yang telah dijelaskan sebelumnya, hasil dari eksperimen ini adalah koordinat dan jari-jari objek bola pada citra. Untuk menghasilkan nilai-nilai tersebut, dibutuhkan beberapa metode pengolahan citra seperti penghalusan citra *gaussian*, operasi deteksi tepi *canny* dan operasi-operasi geometri untuk *circle hough transform*. Selain menerapkan metode-metode tersebut, digunakan kamera ov7670 yang dijalankan dengan arduino uno. Tahap-tahap utama eksperimen penelitian ini digambarkan pada blok

diagram berikut:



Gambar 1. Blog Diagram Deteksi Lingkaran

Tahap utama eksperimen ini adalah pembacaan citra dan pengolahan citra. Pembacaan dilakukan dengan menggunakan camera ov7670 dan arduino uno. Pengolahan citra dilakukan dengan operasi deteksi tepi dengan metode *canny* dan penentuan posisi objek lingkaran berdasarkan titik-titik tepi menggunakan *circle hough transform*.

A. Kamera ov7670

Kamera ov7670 merupakan sensor citrabertegangan rendah yang menghasilkan frame citra 8-bit. Kamera ini menghasilkan *array* citra hingga diatas 30 frame per detik (fps). Pada penelitian ini, kamera ov7670 digunakan untuk mengambil gambar yang akan diolah menggunakan bahasa pemrograman java melalui komunikasi serial arduino uno.



Gambar 2. Kamera ov7670

B. Arduino uno

Arduino uno adalah sebuah board mikrokontroler dengan komponen utamanya adalah ATmega328. Arduino memuat fitur cukup lengkap sebagai kontroler antara lain, 14 pin yang dapat difungsikan sebagai input atau output, 6 pin input *analog to digital*

converter, komunikasi serial TX/RX via USB. Kelebihan arduino uno adalah pemrogramannya yang mudah melalui arduino IDE yang telah memiliki berbagai library pendukung. Pada penelitian ini, arduino uno digunakan sebagai kendali operasi kamera ov7670 dan mengirimkan array citra yang dihasilkan kamera untuk diolah untuk menghasilkan koordinat dan jari-jari lingkaran objek bola pada citra.



Gambar 3. Board Arduino Uno

C. Deteksi tepi Canny

Deteksi tepi *Canny*, dikemukakan oleh John Canny pada tahun 1986. Metode deteksi tepi ini dikenal sebagai metode deteksi tepi paling optimal[4]. Langkah-langkah deteksi tepi *canny* adalah penghalusan citra, deteksi tepi dengan operator sobel horizontal dan vertical, penentuan magnitude dan arah tepi, penghilangan piksel non-maximum dan pengembangan hysteresis untuk menghubungkan titik-titik yang putus pada tepi.

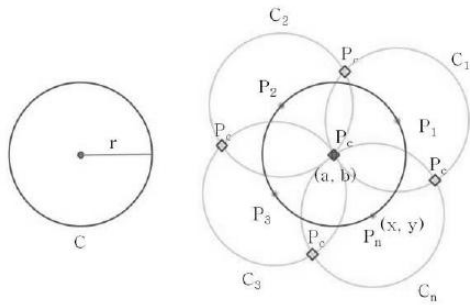
D. Circle Hough Transform

Hough Transform (HT) dicetuskan pertama kali oleh P.V.C hough (1862) yang kemudian diimplementasikan oleh Rosenfeld (1969) untuk mendeteksi garis pada pengolahan citra. umumnya, metode ini digunakan untuk menemukan garis, lingkaran atau *elips* pada citra[4]. Persamaan *Circle Hough Transform* dilakukan dengan menerapkan persamaan geometrik, yaitu persamaan (1) dan (2) disetiap titik-titik tepi yang ditemukan dari operasi deteksi tepi.

$$x = a + R \times \cos\theta \dots\dots\dots (1)$$

$$y = b + R \times \sin\theta \dots\dots\dots (2)$$

Rumus di atas menghasilkan titik x dan y dengan dari pusat (a,b) dan jari-jari (R). Dimana titik a,b merupakan titik-titik tepi yang dihasilkan oleh operasi deteksi tepi *canny*. Untuk setiap titik tepi, rumus tersebut diterapkan dengan nilai sudut (θ) dari 0 hingga 359. Dengan operasi tersebut, kombinasi titik x dan y yang sering dihasilkan akan dianggap sebagai pusat lingkaran. Ilustrasi operasi ini dapat dilihat pada gambar 3.



Gambar 4. Ilustrasi Operasi Circle Hough Transform

IV. HASIL DAN DISKUSI

A. Pembacaan Citra

Hasil pembacaan citra menggunakan kamera ov7670 dapat dilihat pada gambar 4. Citra dihasilkan dari array yang diterima melalui komunikasi serial dengan arduino. Citra pada gambar dibawah merupakan hasil view citra menggunakan java.



Gambar 5. Citra hasil pengambilan gambar

B. Filter Citra

Filter citra merupakan proses penghalusan citra untuk meminimalisir titik-titik yang tidak diperlukan (*noise*). *Noise* mengakibatkan lambatnya proses deteksi lingkaran karena *noise* ikut menjadi titik referensi saat proses pencarian lingkaran berlangsung.

C. Deteksi Tepi Canny

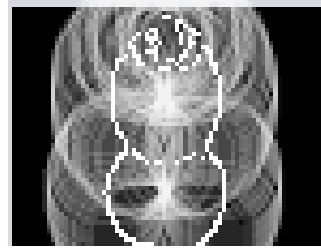
Deteksi canny merupakan proses untuk menghasilkan titik-titik referensi untuk menerapkan persamaan geometri.



Gambar 6. Gambar hasil deteksi tepi

D. Proses Pencarian titik Pusat

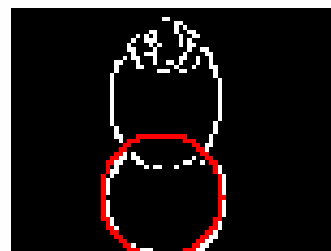
Pada proses ini, persamaan geometri diterapkan pada setiap titik tepi yang ditemukan dengan sudut dari 0 hingga 359. Proses pencarian titik pusat dapat dilihat pada gambar 7.



Gambar 7. Proses Pencarian Titik Pusat

E. Hasil Deteksi Lingkaran

Deteksi lingkaran dihasilkan dengan asumsi titik yang sering muncul dari proses sebelumnya adalah titik pusat objek bola pada citra. hasil deteksi lingkaran dapat dilihat pada gambar 8.



Gambar 8. hasil deteksi Citra bola

Eksperimen dilakukan dengan 4 kali pengambilan gambar dengan posisi dan pencahayaan berbeda. dari keempat gambar tersebut, dilakukan pengujian metode *circle hough transform* untuk menentukan posisi lingkaran yang dibentuk oleh bola pada citra. hasil pengujian dapat dilihat pada tabel 1 dimana metode deteksi bola menghasilkan koordinat, jari-jari lingkaran dan citra baru.

Tabel 1. Hasil Deteksi Lingkran

Citra Asli	Deteksi Tepi	Deteksi Circle			
		Gambar	X	Y	R
			45	37	14
			27	33	22
			42	28	23
			22	39	30

V. KESIMPULAN

Metode *circle hough transform* yang digunakan dapat mendeteksi lokasi lingkaran dengan baik tapi tidak dapat membedakan objek bola asli dengan objek lingkaran yang dibentuk oleh bayangan bola tersebut. Hal ini dapat terjadi jika bayangan yang dihasilkan mebuat lingkaran sempurna. Meskipun demikian,

bayangan berbentuk lingkaran sempurna sangat jarang terjadi. Proses deteksi lingkaran akan menjadi semakin lama jika ditemukan banyak titik referensi hasil deteksi tepi.

VI. DAFTAR PUSTAKA

- [1] Sutrimo, Muhamis, M. Afridon, Khairudinsyah, “Penerapan Metode HSV pada Robot Soccer,” Seminar Nasional Teknologi Informasi, Komunikasi dan Industri (SNTIKI) 7, 2014
- [2] Ach Hadi D., Muhtadin, Mauridhi P, Ilham B, Ilham H, “Deteksi Fitur dan Penentuan Lokasi Robot Pemain Sepak Bola Berbasis Penanda Yang Tidak Unik,” Indonesian Symposium on Robot Soccer Competition, 2014
- [3] M. T. Kantori, Z. Sugianto, M. Nugraha, A. Febriansyah , “Mobile Robot Pemain Bola: Aplikasi Pada KRSBI 2017”, 2017
- [4] Abdul Kadir, Adhi Susanto, “Teori dan Aplikasi Pengolahan Citra”, Jogjakarta, 2012
- [5] Riwinoto, “Penggunaan Algoritma Hough Transforms Untuk Deteksi Bentuk Lingkaran pada Ruang 2D”, Batam

Pengembangan Aplikasi Pendeteksi Objek Lingkaran Menggunakan Metode *Houghcircle* Sebagai Media Pembelajaran Penginderaan Visual Robot

Muhammad Rafii Naufal, Herlambang Sigit Pramono, S.T., M.Cs.

Pendidikan Teknik Mekatronika, Fakultas Teknik, Universitas Negeri Yogyakarta
muhammad.rafi@student.uny.ac.id, herlambangpramono@yahoo.com

ABSTRACT -- *Purposes of this study are to know: (1) performance and (2) feasibility level reviewed by media and material experts and user test results of detecting circle object software use HoughCircle method as learning media for visual robot imaging course. This study is a Research and Development model with ADDIE (Analyze, Design, Develop, Implement, Evaluate) method according to Robert Maribe Branch. Subjects of this study are students of Mechatronics Engineering Education, Faculty of Engineering, Yogyakarta State University. Results of this study are: (1) performance of detecting circle object software use HoughCircle method as learning media for visual robot imaging course is the software can only detect any circle objects from diameter of 5.5 centimetres to 12 centimetres. The 5.5 cm, 6.8 cm, and 7 cm diameter circle objects can be detected by software in range of 10 cm to 70 cm from webcam to the objects. The 10 cm diameter circle objects can be detected by software in range of 15 cm to 70 cm from webcam to the objects. The 12 cm diameter circle objects can be detected by software in range of 20 cm to 70 cm from webcam to the objects. Overall, the mean percentages of distance detection errors by the software reaches a rate of 20%, (2) feasibility level of detecting circle object software use HoughCircle method as learning media for visual robot imaging course include in the very feasible category by media expert with the average scores of 67.5, while by material expert include in the very feasible category with the average scores of 85.5. From the user test results can be obtained with the average scores of 57.9, which means include in the feasible category.*

Keywords: *learning media, software, detect circle object, houghcircle, visual robot.*

I. PENDAHULUAN

Pesat dan majunya dunia industri seharusnya diimbangi juga dengan majunya dunia pendidikan, dikarenakan sumber daya manusia yang berkualitas tinggi akan dihasilkan dari pendidikan yang berkualitas baik juga. Namun hal tersebut tampaknya belum tercapai di negara Indonesia. Salah satu persoalan besar yang saat ini sedang melanda yakni para lulusan yang sudah terampil dan terdidik justru jadi pencetak pengangguran.

Berdasarkan data Badan Pusat Statistik (BPS) pada Agustus 2017, di Indonesia ada 12,3 persen (861.695 orang) merupakan alumni perguruan tinggi,

dari total seluruh pengangguran pada tahun itu. Mereka punya ijazah diploma tiga atau ijazah strata satu (S-1). Dari jumlah itu, pengangguran paling tinggi merupakan lulusan universitas bergelar S-1 yang mencapai hingga angka 618.758 orang. Angka pengangguran terdidik pada 2017 persentasenya meningkat dibanding jumlah pengangguran lulusan perguruan tinggi pada 2016 yang hanya 11,19 persen (786.971 orang) dan pada 2015 sebesar 11,97 persen (905.127 orang) [1].



Gambar 1. Persentase Jumlah Pengangguran Lulusan Perguruan Tinggi di Indonesia

Salah satu faktor penyebab masih banyaknya pengangguran lulusan sarjana yakni belum sesuai kemampuan sumber daya manusia (SDM)-nya dengan kebutuhan perusahaan, dikarenakan para lulusan masih banyak yang belum menguasai beberapa kompetensi yang sudah dicanangkan pemerintah bersama industri untuk setiap jurusannya. Mahasiswa belum bisa menguasai sebuah kompetensi inti atau dasar ini dikarenakan mereka pada saat proses belajar mengajar, ilmu yang diberikan para dosen tidak benar-benar atau belum bisa dipahami secara matang, sehingga mereka (mahasiswa) menjadi tidak tahu dan belum menguasai sebuah kompetensi yang seharusnya mereka kuasai.

Sumber muasal mengapa ilmu yang diberikan para dosen tidak benar-benar sampai dipahami para mahasiswa terjadi karena di saat proses pembelajaran,

metode yang digunakan para dosen masih mengikuti cara lama yang monoton dan kurang variatif, apalagi dengan metode pembelajaran otodidak dimana mahasiswa harus lebih banyak belajar sendiri di luar ruang dan waktu jam perkuliahan, sehingga membuat para mahasiswa menjadi kurang tertarik dan bersemangat saat mengikuti pembelajaran sebuah mata kuliah.

Universitas Negeri Yogyakarta (UNY) adalah salah satu institusi terkemuka di Indonesia yang berkembang dari Institut Keguruan dan Ilmu Pengetahuan (IKIP), telah berhasil menghasilkan banyak tenaga kependidikan dan juga yang non-kependidikan. Dalam rangka mengikuti kemajuan perkembangan teknologi saat ini yang sudah semakin pesat, maka dari itu UNY telah membuka beberapa program studi yang mempelajari teknologi sehingga dapat memberikan manfaat bagi kelangsungan hidup umat manusia. Teknologi yang banyak manfaatnya dan juga turut andil besar dalam penambahan keefektifan dan keefisienan industri saat ini dan juga banyak membantu dalam kehidupan manusia yakni teknologi robot. Program studi di UNY yang benar-benar mempelajari seluk-beluk dunia robotika ialah program studi Pendidikan Teknik Mekatronika yang dinaungi oleh Jurusan Pendidikan Teknik Elektro, Fakultas Teknik. Mekatronika sendiri merupakan ilmu pengetahuan bidang keteknikan yang mempelajari teknik pengolahan sistem otomasi dan robotika, sehingga, lulusan program studi ini diharapkan dapat menjadi insan yang benar-benar paham dan bisa menjadi pakar terkemuka di dunia robotika.

Salah satu mata kuliah yang penting dan berhubungan dengan ilmu robotika ialah mata kuliah Penginderaan Visual Robot. Mata kuliah ini merupakan pengembangan dari mata kuliah Praktik Robotika. Mata kuliah ini mempelajari teori dan praktik pengolahan citra digital yang nantinya dapat diaplikasikan langsung dengan berbagai jenis robot, baik robot manipulator (robot yang dapat melakukan pekerjaan yang dapat dilakukan manusia), maupun robot humanoid (robot yang bentuknya menyerupai manusia). Besar harapan dari mata kuliah ini nantinya para mahasiswa dapat mengenal dan memahami prinsip penginderaan visual dalam sebuah robot.

Dalam mendukung jalannya kegiatan pembelajaran, perlu adanya media pembelajaran. Media pembelajaran merupakan sarana dan prasarana yang dapat menunjang suatu proses pembelajaran. Berdasar pengalaman serta observasi yang dilakukan, mata kuliah Penginderaan Visual Robot ini masih terhitung baru di program studi ini, terhitung baru dua tahun, sehingga muncul sebuah permasalahan utama di sini, yakni belum tersedianya media pembelajaran yang memadai dalam mata kuliah ini. Dampak buruknya ialah sebagian besar mahasiswa merasa sangat jenuh dan bosan dalam mengikuti mata kuliah ini, dikarenakan sebagian besar yang diajarkan dosen hanyalah teori-nya saja. Tentunya ini juga secara tidak langsung berdampak pada kurang berkembangnya

kemampuan mahasiswa dalam bidang penginderaan visual robot.

Mengacu pada permasalahan tersebut, peneliti bermaksud melakukan penelitian yang mengarah pada uji coba media pembelajaran aplikasi pendeteksi objek lingkaran menggunakan metode *HoughCircle*. Objek yang dideteksi merupakan objek berbentuk lingkaran dikarenakan bentuk lingkaran adalah bentuk yang paling mudah dan paling banyak ditemui di lingkungan sekitar kita, dan juga bentuk yang paling sering dideteksi dalam dunia citra visual. Hasil akhir yang diharapkan dengan adanya media ini, ketertarikan dan kompetensi mahasiswa pada mata kuliah Penginderaan Visual Robot dapat meningkat pesat.

Rumusan masalah yang dibahas dalam penelitian ini adalah: (1) bagaimana unjuk kerja media pembelajaran aplikasi pendeteksi objek lingkaran menggunakan metode *HoughCircle*? (2) bagaimana tingkat kelayakan media pembelajaran aplikasi pendeteksi objek lingkaran untuk mahasiswa Pendidikan Teknik Mekatronika UNY ditinjau oleh para ahli dan hasil pada pengujian lapangan? Berdasarkan rumusan masalah, maka tujuan dari penelitian ini adalah (1) mengetahui unjuk kerja pada media pembelajaran aplikasi pendeteksi objek lingkaran menggunakan metode *HoughCircle*, (2) mengetahui tingkat kelayakan media pembelajaran aplikasi pendeteksi objek lingkaran untuk mahasiswa Pendidikan Teknik Mekatronika UNY, ditinjau oleh para ahli dan pengguna.

Media pembelajaran meliputi alat yang secara fisik digunakan untuk menyampaikan isi materi pembelajaran. Media pembelajaran merupakan komponen sumber belajar yang mengandung materi instruksional di lingkungan siswa yang memotivasi siswa untuk belajar [2].

Software aplikasi adalah program-program yang dibuat oleh pabrik *software* atau *programmer* untuk pengguna yang dipakai atau beroperasi dalam bidang-bidang umum. Beberapa contoh dari *software* aplikasi seperti: *word processor*, *spreadsheet*, pengolah *database*, desain grafis, *desktop publishing*, CAD/CAM, *antivirus*, *utilities*, komunikasi, multimedia, dan *game* [3].

Kata deteksi memiliki artian usaha untuk menemukan dan menentukan keberadaan, anggapan, ataupun kenyataan [4]. Pengolahan citra, proses pendeteksian objek dapat dilakukan dengan memisahkan *foreground* (objek yang dideteksi) dengan *background* (latar belakang)-nya. Deteksi objek merupakan sebuah proses dalam pengolahan citra yang mana merupakan sebuah citra bergerak (video). Citra video pada dasarnya merupakan gabungan beberapa citra gambar yang saling berganti pada durasi waktu tertentu. Citra gambar yang saling berganti ini sering disebut dengan *frame*. Proses pergantian *frame* pada citra video berlangsung sangat cepat sehingga sampai-sampai kita tidak menyadarinya. Dalam sebuah citra video, *frame* kedua

tidak selalu sama (identik) dengan *frame* pertama, begitu pula dengan *frame-frame* selanjutnya. Hal inilah yang menjadi dasar pemikiran terhadap teknik deteksi objek pada pengolahan citra dengan menganalisa nilai selisih antara *frame* pertama yang dapat juga disebut dengan latar belakang atau *background* dengan *frame-frame* selanjutnya [5].

Transformasi *Hough* merupakan salah satu teknik atau metode utama transformasi citra yang peneliti gunakan untuk mengisolasi suatu objek pada citra dengan menemukan batas-batasnya (*boundary detection*). Dinamakan transformasi *Hough* karena penemu metode transformasi ini bernama Paul Hough. Pada tahun 1962, awalnya ia hanya bertujuan untuk mendeteksi adanya garis lurus dalam suatu citra. Tujuan utama dari sebuah transformasi yakni untuk mendapatkan suatu fitur yang lebih spesifik, maka dari itu *classical Hough transform* diciptakan sebagai teknik yang paling umum digunakan untuk mendeteksi objek yang berbentuk kurva, layaknya garis, lingkaran, elips, dan parabola. Keuntungan utama dengan adanya transformasi *Hough* ini yakni dapat mendeteksi sebuah tepian dengan celah pada batas fitur dan secara relatif tidak terpengaruh dengan adanya *noise* atau derau [6].

Penginderaan Visual Robot merupakan mata kuliah baru yang terdapat pada program studi Pendidikan Teknik Mekatronika. Terbilang baru dikarenakan mata kuliah ini baru diajarkan pada tahun ajaran 2016/2017. Dalam pelaksanaan kuliah, mahasiswa lebih diarahkan pada latihan-latihan pembuatan algoritma mengolah sebuah citra visual, baik berupa gambar maupun video yang melalui kamera *webcam*. Metode yang digunakan dosen masih menggunakan cara lama, menjelaskan teori terlebih dahulu, lalu langsung mempraktikkan mencoba mengolah salah satu citra visual, dan di akhir perkuliahan memberi penugasan kepada para mahasiswa. Namun, setelah peneliti melakukan observasi, pembelajaran pada mata kuliah ini masih perlu memerlukan media pembelajaran yang mendukung sehingga peserta didik dapat memahami lebih luas tentang visual robot. Media pembelajaran penginderaan visual robot yang baik dapat memberikan rangsangan kepada peserta sebagai wahana praktikum dan penambah ilmu khususnya di bidang penginderaan visual robot. Berbagai jenis algoritma metode perubahan gambar yang terdapat pada media pembelajaran penginderaan visual robot tersebut dapat mendukung pengetahuan peserta didik pada saat bekerja di dunia industri maupun dalam bidang lainnya. Oleh karena itu, media pembelajaran penginderaan visual robot yang memadai sangat diperlukan guna mendukung kompetensi yang diharapkan pada bidang penginderaan visual robot, sehingga dapat tercapai dengan baik [7].

II. MODEL PENELITIAN

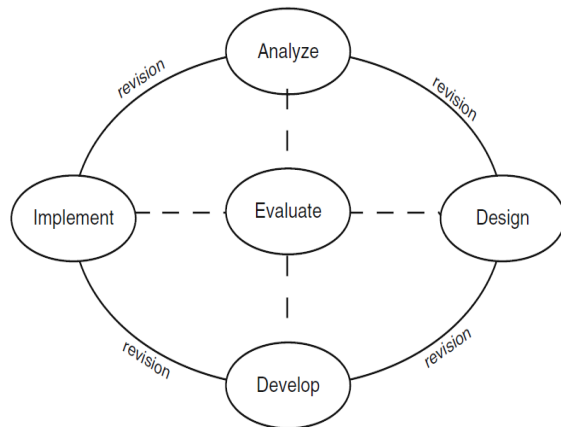
Metode yang digunakan dalam penelitian ini menggunakan pendekatan penelitian pengembangan (*Research and Development*). Model pengembangan yang digunakan dalam penelitian ini adalah ADDIE menurut Robert Maribe Branch. Pengembangan ini nantinya membuat sebuah aplikasi pendeteksi objek lingkaran sebagai media pembelajaran penginderaan visual robot yang dilengkapi *jobsheet* untuk menunjang proses pembelajaran [8].

Penelitian ini dilakukan di Jurusan Pendidikan Teknik Elektro, Fakultas Teknik, Universitas Negeri Yogyakarta. Penelitian ini dilakukan mulai pada bulan April 2018. Subjek penelitian ini adalah mahasiswa Program Studi Pendidikan Teknik Mekatronika, Fakultas Teknik, Universitas Negeri Yogyakarta, serta dosen sebagai ahli materi dan ahli media. Uji pengguna dilakukan pada 20 mahasiswa yang mengikuti mata kuliah penginderaan visual robot. Ahli materi dan ahli media diambil dari dosen Jurusan Pendidikan Teknik Elektro, UNY yang menguasai bidang visual robot.

Prosedur Pengembangan

Prosedur pengembangan yang digunakan dalam penelitian ini mengadopsi langkah-langkah penelitian dan pengembangan ADDIE oleh Robert Maribe Branch dengan berbagai penyesuaian. Adapun tahapan model pengembangan ADDIE yaitu: *analyze*, *design*, *develop*, *implement*, dan *evaluate*. Secara singkat berikut merupakan penjabaran tahapan-tahapan tersebut. Tahap *analyze* yaitu mengumpulkan informasi dengan melakukan observasi langsung dengan mengikuti pembelajaran penginderaan visual robot di Jurusan Pendidikan Teknik Elektro. Selanjutnya dilakukan analisis kebutuhan untuk menentukan jenis media yang akan dikembangkan apakah sudah sesuai atau belum. Tahap *design* yakni: (1) mengidentifikasi komponen dalam aplikasi yang akan digunakan untuk membuat rancang bangun media pembelajaran aplikasi pendeteksi objek lingkaran, (2) merancang desain media pembelajaran aplikasi pendeteksi objek lingkaran sehingga mudah dioperasikan dan dipahami, (3) merancang tata letak komponen dalam aplikasi yang digunakan pada media pembelajaran aplikasi pendeteksi objek lingkaran, (4) membuat program yang akan digunakan pada media pembelajaran aplikasi pendeteksi objek lingkaran, (5) merancang urutan kerja dan petunjuk penggunaan aplikasi. Tahap *develop* yakni: (1) membuat perangkat lunak media pembelajaran aplikasi pendeteksi objek lingkaran, (2) menyusun komponen aplikasi yang digunakan pada media pembelajaran aplikasi pendeteksi objek lingkaran, (3) membuat program sehingga media pembelajaran aplikasi pendeteksi objek lingkaran bisa menjadi sebuah aplikasi yang layak digunakan, (4) pengujian aplikasi pendeteksi objek lingkaran sebagai media pembelajaran penginderaan visual robot, (5) membuat materi dan tugas-tugas dalam *jobsheet* yang dapat membuat

peserta didik mencapai tujuan pembelajaran, (6) melakukan uji *blackbox*, (7) melakukan uji kelayakan media dan materi kepada ahli media dan ahli materi, (8) melakukan perbaikan. Tahap *implement* yaitu, (1) menyiapkan pengajar, dan (2) menyiapkan peserta didik. Tahap *evaluate* yaitu melakukan evaluasi setelah melakukan tahapan uji validasi oleh ahli media dan materi, dan uji pengguna.



Gambar 2. Model Pengembangan Media

Instrumen Pengumpulan Data

Dalam penelitian ini instrumen penelitian yang digunakan untuk melihat tingkat kelayakan aplikasi pendeteksi objek lingkaran sebagai media pembelajaran penginderaan visual robot yakni instrumen non tes berupa kuesioner. Kuesioner ini menggunakan skala pengukuran *Likert* dengan empat pilihan jawaban: sangat setuju, setuju, tidak setuju, dan sangat tidak setuju.

Instrumen yang diberikan kepada ahli media meliputi kriteria yang sesuai dengan elemen media, yakni: kesesuaian dengan tujuan, karakteristik pembelajar atau siswa, materi pembelajaran, teori, gaya belajar siswa, dan kondisi lingkungan. Instrumen yang diberikan kepada ahli materi meliputi kriteria yang sesuai dengan elemen materi yaitu relevansi materi dengan tujuan pembelajaran, penyajian, bahasa, dan kegrafikaan. Instrumen untuk uji pengguna meliputi kriteria kualitas isi dan tujuan, kualitas instruksional atau pembelajaran, teknis, lalu efektivitas dan efisiensi.

Teknik Analisis Data

Teknik analisis data yang digunakan dalam penelitian ini adalah teknik analisis deskriptif. Teknik analisis deskriptif dilakukan dengan menggunakan statistik deskriptif. Data yang diperoleh dari kuesioner berupa data kuantitatif yang ditafsirkan menjadi nilai kualitatif.

III. HASIL PENELITIAN DAN PEMBAHASAN

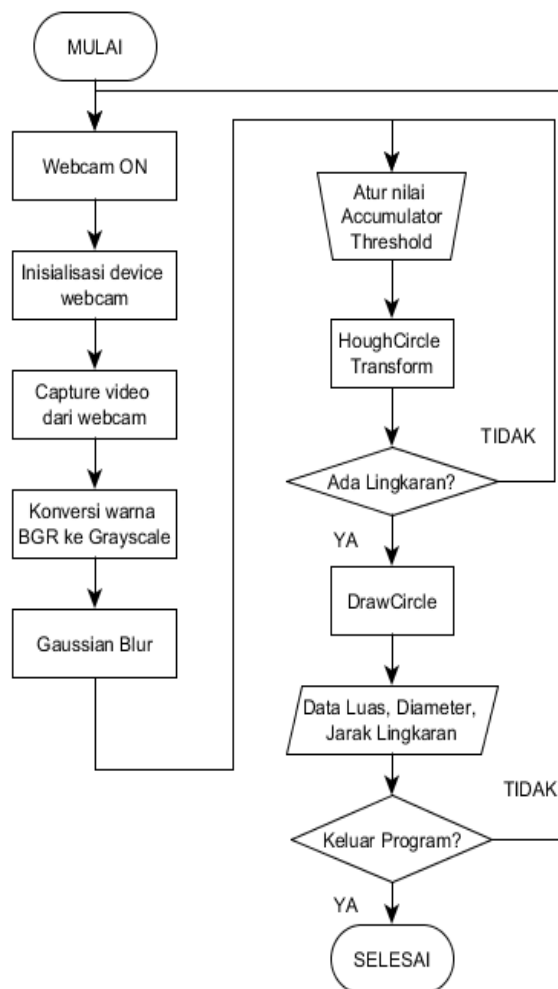
Prosedur pengembangan yang digunakan dalam penelitian ini mengadopsi langkah-langkah penelitian dan pengembangan ADDIE oleh Robert Maribe Branch dengan berbagai penyesuaian. Hasil dari proses analisis media pembelajaran yang ada belum lebih dikembangkan sehingga keaktifan dan minat peserta didik masih kurang terhadap mata kuliah penginderaan visual robot. Selanjutnya dilakukan analisis kebutuhan untuk menentukan jenis media yang akan dikembangkan apakah sudah sesuai atau belum.

Tabel 1. Komponen Utama Media di Visual Studio

No.	Nama Komponen	Jumlah	Keterangan
1.	<i>Label</i>	18	Penampil keterangan
2.	<i>ImageBox</i>	4	Penampil gambar <i>webcam</i> dan proses
3.	<i>Button</i>	3	Pengatur <i>webcam</i>
4.	<i>Trackbar</i>	9	Pengatur variabel dalam rumus

Hasil dari analisis yakni perlu adanya media pembelajaran penginderaan visual robot. Hasil dari proses perancangan yaitu menentukan komponen aplikasi yang akan digunakan dan membuat desain aplikasi. Komponen yang digunakan dapat dilihat pada Tabel 1. *Flowchart* aplikasi pendeteksi objek lingkaran sebagai media pembelajaran penginderaan visual robot pada Gambar 2. Aplikasi dibuat dengan *software* Visual Studio 2012 dengan menggunakan library *EmguCV*. Metode yang dipakai yakni *HoughCircle*, dikarenakan akan mendeteksi objek berbentuk lingkaran. Objek yang dideteksi berbentuk lingkaran, dikarenakan banyak objek yang berbentuk lingkaran mudah dijumpai di sekitar kita, contohnya: bola, kipas angin, topi, helm, dan lain sebagainya.

Hasil dari proses pengembangan yakni: (1) pembuatan dan perakitan media pembelajaran, pembuatan program dan tampilan GUI dengan menggunakan bantuan *software* Visual Studio 2012 dengan bahasa pemrograman C# dan library *EmguCV*, program dibuat untuk pengolahan citra digital dalam aplikasinya, (2) Pengujian digunakan untuk mengetahui kinerja media pembelajaran apakah sudah sesuai dengan rancangan yang telah dibuat atau belum, (3) Pembuatan materi bahan ajar, panduan penggunaan, dan *jobsheet*, dan (4) uji *blackbox* yang dapat dilihat di Tabel 2.



Gambar 2. Flowchart Aplikasi Pendeteksi Objek Lingkaran sebagai Media Pembelajaran

Tabel 2. Hasil Pengujian Blackbox.

No.	Keterangan	Fungsi	
		Ya	Tidak
1	Fungsi <i>Tool strip</i> menu pemilih <i>port</i> kamera	√	
2	Fungsi <i>imageBox</i> (Citra Asli Kamera)	1 √	
3	Fungsi <i>imageBox</i> (Grayscale)	2 √	
4	Fungsi <i>imageBox</i> (Deteksi Lingkaran)	3 √	
5	Fungsi <i>imageBox</i> (Grayscale GaussianBlur)	4 + √	
6	Fungsi <i>domain UpDown</i> Gaussian Blur	√	
7	Fungsi <i>trackbar</i> Min Radius	√	
8	Fungsi <i>trackbar</i> Max Radius	√	
9	Fungsi <i>trackbar</i> Canny Threshold	√	
10	Fungsi <i>trackbar</i>	√	

11	Fungsi <i>Accumulator Threshold</i>	√
12	Fungsi <i>trackbar Radius</i>	√
13	Fungsi <i>trackbar Ketebalan</i>	√
14	Fungsi <i>trackbar Biru</i>	√
15	Fungsi <i>trackbar Hijau</i>	√
16	Fungsi <i>trackbar Merah</i>	√
17	Fungsi Deteksi Jumlah Lingkaran	√
18	Fungsi Deteksi Luas Lingkaran	√
19	Fungsi Deteksi Diameter Lingkaran	√
20	Fungsi <i>Library</i> EmguCV	√
21	Fungsi kinerja media secara keseluruhan	√
22	Fungsi <i>button</i> Start	√
23	Fungsi <i>button</i> Pause	√
24	Fungsi <i>button</i> Stop	√
25	Fungsi <i>button</i> viewCode	√

Tabel 3. Keakuratan bentuk dideteksi aplikasi

Bentuk	Terdeteksi?
Lingkaran	Iya
Segi tiga	Tidak
Persegi	Tidak
Bujur Sangkar	Tidak
Segi enam	Tidak

Tabel 4. Pengukuran jarak yang bisa dideteksi objek d=5,5 cm

Dapat Mendeteksi?	Jarak Objek Sebenarnya (cm)	Jarak pada Aplikasi (cm)	Persentase Kesalahan
Ya	10	22,2	122%
Ya	15	19,8	32%
Ya	20	18,5	7,5%
Ya	25	17,6	29,6%
Ya	30	17,1	43%
Ya	35	16,5	52,86%
Ya	40	16,2	59,5%
Ya	50	15,8	68,4%
Ya	70	15,6	77,71%
Rata-rata Persentase Kesalahan			20,5%

Tabel 5. Pengukuran jarak yang bisa dideteksi dengan objek berdiameter 6,8 cm

Dapat Mendeteksi?	Jarak Objek Sebenarnya (cm)	Jarak pada Aplikasi (cm)	Persentase Kesalahan
-------------------	-----------------------------	--------------------------	----------------------

Ya	10	22,1	121%
Ya	15	20,4	36%
Ya	20	19,3	3,5%
Ya	25	18,5	26%
Ya	30	17,8	40,67%
Ya	35	17,4	50,29%
Ya	40	17,2	57%
Ya	50	16,4	67,2%
Ya	70	15,9	77,29%
Rata-rata Persentase Kesalahan			18,33%

Tabel 6. Pengukuran jarak yang bisa dideteksi dengan objek berdiameter 7 cm

Dapat Mendeteksi?	Jarak Objek Sebenarnya (cm)	Jarak pada Aplikasi (cm)	Persentase Kesalahan
Ya	10	24,1	141%
Ya	15	21,5	43,33%
Ya	20	19,6	2%
Ya	25	19,2	23,2%
Ya	30	18,2	39,33%
Ya	35	17,8	49,14%
Ya	40	17,3	56,75%
Ya	50	16,8	66,4%
Ya	70	16	77,14%
Rata-rata Persentase Kesalahan			14,4%

Sebelum masuk tahap implementasi, terlebih dahulu dilakukan uji validasi media dan materi oleh dua orang ahli atau *expert judgment*. Tujuannya yakni untuk mengetahui apakah media pembelajaran yang dikembangkan sudah layak atau belum untuk digunakan. Tahapan selanjutnya yakni implementasi. Pada proses ini, aplikasi diuji pada pengguna, yakni mahasiswa yang mengikuti perkuliahan penginderaan visual robot.

Tabel 7. Pengukuran jarak yang bisa dideteksi dengan objek berdiameter 10 cm

Dapat Mendeteksi?	Jarak Objek Sebenarnya (cm)	Jarak pada Aplikasi (cm)	Persentase Kesalahan
Tidak	10	-	-
Ya	15	26,2	74,67%
Ya	20	22,6	13%
Ya	25	22,2	11,2%
Ya	30	19,7	34,33%
Ya	35	19,4	44,57%
Ya	40	18,8	53%
Ya	50	17,9	64,2%
Ya	70	16,8	76%
Rata-rata Persentase Kesalahan			24,45%

Tabel 8. Pengukuran jarak yang bisa dideteksi dengan objek berdiameter 12 cm

Dapat Mendeteksi?	Jarak Objek Sebenarnya	Jarak pada Aplikasi	Persentase Kesalahan
-------------------	------------------------	---------------------	----------------------

	(cm)	(cm)	
Tidak	10	-	-
Tidak	15	-	-
Ya	20	26,1	30,5%
Ya	25	24	4%
Ya	30	22,4	25,33%
Ya	35	21,4	38,86%
Ya	40	20,2	49,5%
Ya	50	18,5	63%
Ya	70	17,4	75,14%
Rata-rata Persentase Kesalahan			32,19%

Tabel 9. Pengukuran diameter objek oleh aplikasi pada jarak 20 cm

Diameter Lingkaran Asli (cm)	Diameter Lingkaran Pada Aplikasi (cm)	Persentase Kesalahan
12	10,4	13,33%
10	8,5	15%
7	6	14,29%
6,8	7,2	5,88%
5,5	5	9,09%
Rata-rata Persentase Kesalahan		9,17%

Tabel 10. Pengukuran jarak webcam terhadap media yang dideteksi oleh aplikasi

Diameter Objek Lingkaran (cm)	Jarak Sebenarnya (cm)	Jarak yang Dideteksi Aplikasi (cm)	Persentase Kesalahan
12	20	24,4	36,67%
10	20	22,5	25%
7	20	21,2	17,14%
6,8	20	20	0%
5,5	20	19	18,18%
Rata-rata Persentase Kesalahan			12,13%

Mahasiswa mencoba media pembelajaran yang telah dikembangkan dan pada akhir perkuliahan mahasiswa diminta untuk mengisi kuesioner terkait dengan aplikasi pendeteksi objek lingkaran.

Evaluasi dilakukan setelah melalui tahap uji ahli media, ahli materi, serta uji pengguna. Hasil dari evaluasi dari segi fitur, yaitu pembacaan ukuran, perlu ditambahkan pembacaan jarak webcam ke media yang dideteksi. Dari segi materi, perlu penambahan objek yang *real* seperti bola, dan pada saat pendeteksian perlu dicoba juga menggunakan objek yang tak berbentuk lingkaran.

Hasil Validasi Ahli Media

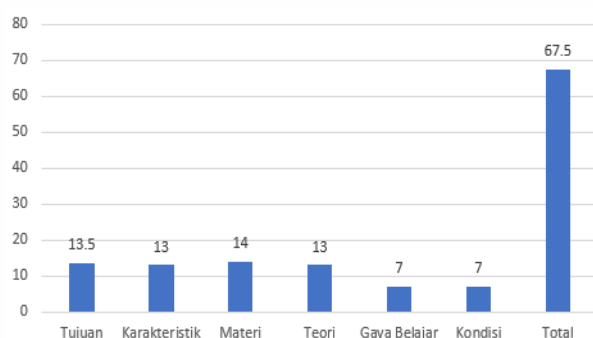
Pada validasi media ini, ahli media menilai media aplikasi pendeteksi objek lingkaran sebagai media pembelajaran penginderaan visual robot berdasarkan enam kesesuaian dengan: tujuan, karakteristik, materi, teori, gaya belajar, dan kondisi. Validator merupakan dosen Jurusan Pendidikan

Elektro, Fakultas Teknik, Universitas Negeri Yogyakarta yang menguasai bidang media pembelajaran visual robot. Data hasil validasi ahli media dapat dilihat pada Tabel 11.

Tabel 11. Hasil Analisis Data Ahli Media

No.	Penilaian Kesesuaian dengan	Rerata	Persentase
1	Tujuan	13,5	84%
2	Karakteristik	13	81%
3	Materi	14	88%
4	Teori	13	81%
5	Gaya Belajar	7	88%
6	Kondisi	7	88%
Rata-rata Total		67,5	84%
Kategori		Sangat Layak	

Berdasarkan data yang diperoleh, untuk penilaian kesesuaian dengan tujuan dari dua ahli media mendapatkan nilai rata-rata 13,5 dari nilai skor maksimal 16 dan skor minimal 4, yang berarti masuk dalam kategori sangat layak dengan persentase 84%. Kesesuaian dengan karakteristik dari dua ahli media mendapatkan nilai rata-rata 13 dari nilai skor maksimal 16 dan skor minimal 4, yang berarti masuk dalam kategori sangat layak dengan persentase 81%. Kesesuaian dengan materi memperoleh nilai rata-rata 14 dari nilai skor maksimal 16 dan skor minimal 4, yang berarti masuk dalam kategori sangat layak dengan persentase 88%. Kesesuaian dengan teori memperoleh nilai rata-rata 13 dari nilai skor maksimal 16 dan skor minimal 4, yang berarti masuk dalam kategori sangat layak dengan persentase 81%. Kesesuaian dengan gaya belajar memperoleh nilai rata-rata 7 dari nilai skor maksimal 8 dan skor minimal 2, yang berarti masuk dalam kategori sangat layak dengan persentase 88%. Kesesuaian dengan kondisi memperoleh nilai rata-rata 7 dari nilai skor maksimal 8 dan skor minimal 2, yang berarti masuk dalam kategori sangat layak dengan persentase 88%. Dari hasil tersebut dapat diperoleh skor rerata total uji kelayakan ahli media adalah 67,5 dari nilai maksimal 80 dan nilai minimal 20, yang berarti masuk dalam kategori sangat layak dengan persentase 84%. Grafik analisis data ahli media pada Gambar 3.



Gambar 3. Grafik Analisis Data Ahli Media

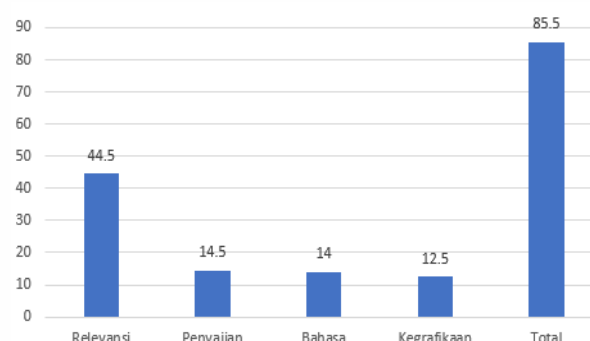
Hasil Validasi Ahli Materi

Pada validasi materi ini, ahli materi menilai materi tentang aplikasi pendeteksi objek lingkaran sebagai media pembelajaran penginderaan visual robot berdasarkan empat aspek: relevansi, penyajian, dan bahasa. Validator merupakan dosen Jurusan Pendidikan Elektro, Fakultas Teknik, Universitas Negeri Yogyakarta yang menguasai materi tentang visual robot. Data hasil validasi ahli materi dapat dilihat pada Tabel 12.

Tabel 12. Hasil Analisis Data Ahli Materi

No.	Penilaian Kriteria	Rerata	Persentase
1	Relevansi	44,5	93%
2	Penyajian	14,5	91%
3	Bahasa	14	88%
4	Kegrafikaan	12	78%
Rata-rata Total		85,5	89%
Kategori		Sangat Layak	

Berdasarkan data yang diperoleh, untuk penilaian kriteria relevansi dari dua ahli materi mendapatkan nilai rata-rata 44,5 dari nilai skor maksimal 48 dan skor minimal 12, yang berarti masuk dalam kategori sangat layak dengan persentase 93%. Kriteria penyajian dari dua ahli materi mendapatkan nilai rata-rata 14,5 dari nilai skor maksimal 16 dan skor minimal 4, yang berarti masuk dalam kategori sangat layak dengan persentase 91%. Kriteria bahasa memperoleh nilai rata-rata 14 dari nilai skor maksimal 16 dan skor minimal 4, yang berarti masuk dalam kategori sangat layak dengan persentase 88%. Kriteria kegrafikaan memperoleh nilai rata-rata 12,5 dari nilai skor maksimal 16 dan skor minimal 4, yang berarti masuk dalam kategori sangat layak dengan persentase 78%. Dari hasil tersebut dapat diperoleh skor rerata total uji kelayakan ahli materi adalah 85,5 dari nilai maksimal 96 dan nilai minimal 24, yang berarti masuk dalam kategori sangat layak dengan persentase 89%. Grafik analisis data ahli materi dapat dilihat pada Gambar 4.



Gambar 4. Grafik Analisis Data Ahli Materi

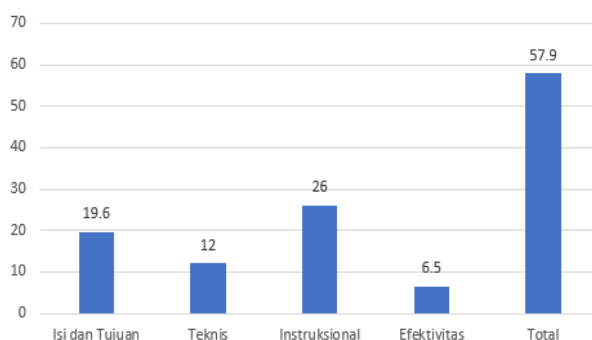
Hasil Uji Pengguna

Uji pengguna dilakukan untuk mengetahui tanggapan peserta didik terhadap aplikasi pendeteksi objek lingkaran sebagai media pembelajaran penginderaan visual robot yang dikembangkan. Instrumen yang digunakan memiliki beberapa kriteria penilaian antara lain: kriteria kualitas isi dan tujuan, teknis, instruksional, dan efektivitas. Data hasil uji pengguna dapat dilihat pada Tabel 13.

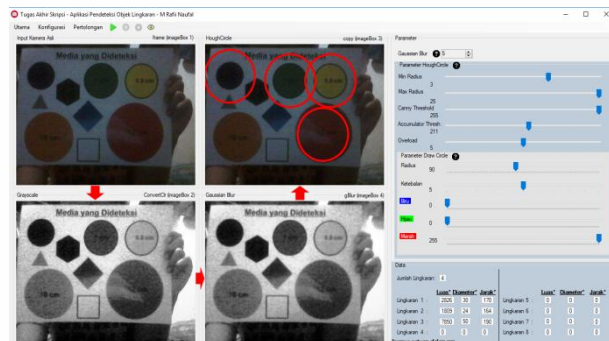
Tabel 13. Hasil Analisis Data Uji Pengguna

No	Penilaian Kriteria	Rerata	Persentase
Kualitas			
1	Isi dan tujuan	19,7	81%
2	Teknis	12	77%
3	Instruksional	26	81%
4	Efektivitas	6,5	81%
Rerata total		57,9	72%
Kategori		Layak	

Berdasarkan data yang diperoleh, untuk penilaian kriteria kualitas isi dan tujuan mendapatkan nilai rata-rata 19,6 dari nilai skor maksimal 24 dan skor minimal 6, yang berarti masuk dalam kategori sangat layak dengan persentase 81%. Kriteria kualitas teknis mendapatkan nilai rata-rata 12 dari nilai skor maksimal 16 dan skor minimal 4, yang berarti masuk dalam kategori sangat layak dengan persentase 77%. Kriteria kualitas instruksional memperoleh nilai rata-rata 26 dari nilai skor maksimal 32 dan skor minimal 8, yang berarti masuk dalam kategori sangat layak dengan persentase 81%. Kriteria kualitas efektivitas memperoleh nilai rata-rata 6,5 dari nilai skor maksimal 8 dan skor minimal 2, yang berarti masuk dalam kategori sangat layak dengan persentase 81%. Dari hasil tersebut dapat diperoleh skor rerata total uji pengguna adalah 57,9 dari nilai maksimal 80 dan nilai minimal 20, yang berarti masuk dalam kategori layak dengan persentase 72%. Grafik analisis data uji pengguna dapat dilihat pada Gambar 5.



Gambar 5. Grafik Analisis Data Uji Pengguna



Gambar 6. Aplikasi yang Telah Dikembangkan Peneliti

IV. KESIMPULAN

Unjuk kerja aplikasi pendeteksi objek lingkaran sebagai media pembelajaran penginderaan visual robot yakni aplikasi hanya dapat mendeteksi objek berupa lingkaran, untuk objek berbentuk selain lingkaran tidak dapat terdeteksi. Berdasarkan hasil ujicoba, aplikasi pendeteksi objek lingkaran dapat menampilkan data ukuran objek berupa luas, diameter, dan jarak objek terhadap *webcam*. Aplikasi dapat mendeteksi objek lingkaran dari ukuran 5,5 cm hingga 12 cm. Objek lingkaran berdiameter 5,5 cm, 6,8 cm, dan 7 cm dapat dideteksi oleh aplikasi, dengan jangkauan jarak antara 10 cm hingga 70 cm dari *webcam* ke objek. Sedangkan untuk objek lingkaran berdiameter 10 cm dapat dideteksi oleh aplikasi, dengan jangkauan jarak antara 15 cm hingga 70 cm dari *webcam* ke objek. Dan untuk objek lingkaran berdiameter 12 cm dapat dideteksi oleh aplikasi, dengan jangkauan jarak antara 20 cm hingga 70 cm dari *webcam* ke objek. Secara keseluruhan, rata-rata persentase kesalahan deteksi jarak oleh aplikasi mencapai angka sebesar 20%.

Tingkat kelayakan aplikasi pendeteksi objek lingkaran menggunakan metode *HoughCircle* sebagai media pembelajaran penginderaan visual robot termasuk dalam kategori sangat layak ditinjau dari para ahli media dengan skor rerata adalah 67,5 dari nilai maksimal 80 dan nilai minimal 20 dengan persentase 84%, sedangkan ditinjau dari para ahli materi termasuk dalam kategori sangat layak dengan skor rerata yang didapat adalah 85,5 dari nilai maksimal 96 dan nilai minimal 24 dengan persentase 89%. Ditinjau dari hasil uji terhadap pengguna dapat diperoleh skor rerata total adalah 57,9 dari nilai maksimal 80 dan nilai minimal 20, termasuk dalam kategori layak dengan persentase 72%.

V. DAFTAR PUSTAKA

- [1] Badan Pusat Statistik. (2017). *Pengangguran Terbuka Menurut Pendidikan Tertinggi yang Ditamatkan 1986-2017*. Diakses pada 21 Februari 2018 dari <https://www.bps.go.id/statictable/2009/04/16/972/pengangguran-terbuka-menurut-pendidikan-tertinggi-yang-ditamatkan-1986---2017.html> .
- [2] Wati, E. R. (2016). *Ragam Media Pembelajaran: Visual-Audio Visual-Komputer-Power Point-Internet-Interactive Video*. Jakarta: Kata Pena.
- [3] Indra P., R. C., dkk. (2012). *Mengenal Software for Beginners*. Yogyakarta: Penerbit ANDI.
- [4] Badan Pengembangan dan Pembinaan Bahasa, Kementerian Pendidikan dan Kebudayaan Republik Indonesia. (2016). *Kamus Besar Bahasa Indonesia V*. Diakses pada 31 Maret 2018 dari <http://kbbi.kemdikbud.go.id> .
- [5] Prabowo, M. R. (2017). *Deteksi Objek Bergerak Dalam Air Menggunakan Metode Gaussian Mixture Model Berbasis Action-Cam*. Skripsi, Universitas Lampung, Bandar Lampung.
- [6] Putra, D. (2010). *Pengolahan Citra Digital*. Yogyakarta: Penerbit ANDI.
- [7] Jurusan Pendidikan Teknik Elektro UNY. (2014). *Buku Peraturan Kurikulum 2014 Jurusan Pendidikan Teknik Elektro Universitas Negeri Yogyakarta*. Yogyakarta: Universitas Negeri Yogyakarta.
- [8] Branch, R. M. (2009). *Instructional Design of ADDIE*. New York: Springer.

Penerapan *Web Base* Sebagai Kontroller dan *NodeMCU 1.0 (Esp-12E)* Sebagai Media Komunikasi Pada Kendali Robot Soccer

Kiki Reski R^a, Mimi Hasmidar^b, Hamdan^c, Bagas Yudhito^d Adrin Thamrin^e, Fitri^f

^{abcde}STMIK Adhi Guna, Jl.Undata no.3, Palu

Email: kikireski1299@gmail.com, mimihasmidar@gmail.com, rmhamdan2@gmail.com, yuditob9@gmail.com, adrin.adhiguna@gmail.com, fitrilad01@gmail.com

Abstrak – *Web Base Controller* merupakan sebuah basis web yang berfungsi mengontrol sebuah Robot sepak bola dalam melakukan aksi dilapangan dengan menggunakan modul *NodeMCU 1.0 (Esp-12E)* sebagai media komunikasi data. *Web Base* tersebut dapat memicu agar robot bisa aktif (on) dan dapat menentukan pergerakan robot yang diinginkan melalui penekanan tombol yang ada pada *Web Base Controller*. Untuk menghubungkan *Web Base* ke Robot dibutuhkan sebuah Modul *NodeMCU 1.0 (Esp-12E)* agar Robot dapat bermanuver sesuai dengan intruksi *Web Base*. Modul *NodeMCU 1.0 (Esp-12E)* adalah alat yang dapat menggerakkan robot melalui *Web Base Control* yang terhubung ke jaringan SSID yang telah tersedia pada *acces point*. Hasil yang diperoleh, *Web Base Controller* dapat bekerja layaknya referee box dalam mengendalikan robot soccer dan dapat digunakan sebagai basis kendali jarak jauh pada robot soccer. Maka dari hasil uji coba *Web Base Controller* dapat bekerja 100%.

Kata kunci: *Web Base Controller*, *NodeMCU 1.0 (Esp-12E)*, *Robot Soccer*.

I. PENDAHULUAN

Robot merupakan sebuah alat mekanik yang bekerja tanpa lelah yang dikontrol oleh manusia (*Manually*) ataupun tanpa kontrol manusia (*Automatic*). Robot dibuat untuk mempermudah suatu pekerjaan yang dilakukakan secara terus menerus atau robot biasanya dibuat untuk ajang-ajang bergengsi seperti Kontes Robot Indonesia yang diadakan oleh RISTEK DIKTI setiap tahunnya. Pada Kontes Robot Indonesia (KRI) tersebut terdapat beberapa divisi atau beberapa jenis robot yang dipertandingkan salah satunya yaitu Kontes Robot Sepak Bola Indonesia (KRSBI) beroda.

Robot sepakbola beroda yaitu robot yang dituntut dapat bermain bola layaknya manusia dan atau robot dapat bermain tanpa bantuan manusia. Akan tetapi robot yang digunakan pada tim peneliti masih menerapkan sistem manual dalam bertanding yang mana operator akan menyentuh dan memindahkan robot ketempat start awal yang telah ditentukan sesuai dengan aturan pertandingan.

Maka untuk itu salah satu solusi yang digunakan adalah dengan menggunakan *Web Base Controller*

sebagai basis kontrol, karena web base dapat digunakan untuk mengontrol robot agar robot bermanuver secara otomatis tanpa sentuhan tangan operator secara langsung ke robot. Penerapan *Web Base Controller* dapat mengontrol robot secara otomatis menggunakan mikrokontroller yang dilengkapi dengan modul wifi sehingga robot dapat bekerja sesuai dengan ketentuan yang telah ditentukan oleh *rule of the game*.

Web Base Controller sebagai sistem kendali, robot dapat bekerja karena terdapat sebuah mikrokontroller yang dilengkapi dengan modul wifi yaitu *NodeMCU 1.0 (Esp-12E)* yang berfungsi sebagai penghubung antara web base controller dengan robot. Sehingga web base dapat mengirim perintah ke robot.

II. METODE

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

Metode *experimental* ini penulis lakukan dengan tujuan untuk melakukan uji coba dari setiap perintah yang dikirim dari *Web Base* ke Robot seperti perintah *KickOff*, *FreeKick* dan seterusnya.

A. Tahapan Penelitian

Dari alat dan bahan yang digunakan oleh penulis. Maka penulis menerapkan tahapan penelitian Sebagai berikut:

1. Menghubungkan *Web Base Controller* ke *NodeMCU 1.0 (Esp-12E)*.
2. Melakukan pengujian program *WebBase* ke *nodeMCU 1.0 (Esp-12E)*.
3. Melakukan pengujian program pada *NodeMCU 1.0 (Esp-12E)* ke *Arduino* sebagai *Master Controller* pada robot.
4. *IP NodeMCU 1.0 (Esp-12E)* ditambahkan ke dalam listing program *Web Base Controller*.

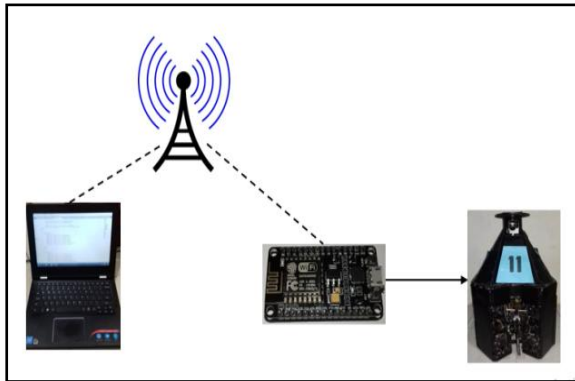
B. Alat dan Bahan

Kebutuhan bahan dan alat yang akan digunakan untuk menghubungkan *Web Base* ke Robot sebagai

berikut:

1. Satu buah Robot sepak bola beroda
2. Satu buah laptop *Lenovo* yang dilengkapi dua *software* yaitu:
 - a. *Arduino IDE 1.8.5*,
 - b. *sublime text 2*.
3. *NodeMCU 1.0 (Esp-12E)* dan *Arduino* sebagai pusat kendali Robot.
4. *Web* sebagai pengontrol Robot. Pemrograman web terdiri dari:
 - a. *HTML 5*,
 - b. *CSS 3*,
 - c. *Java Script ES5*.
5. Kabel yang digunakan untuk menyambungkan antara *NodeMCU 1.0 (Esp-12E)* dan *Arduino* sebagai *Slave Master*.
6. Jaringan lokal/*access point*.

C. Komunikasi Web Base Controller, NodeMcu dan Robot Soccer



Gambar 1. Komunikasi Antar Device

Ada beberapa tahapan untuk membangun *Web Base Control* yaitu dengan beberapa *device* yang saling terhubung, adapun *device* yang dimaksud adalah:

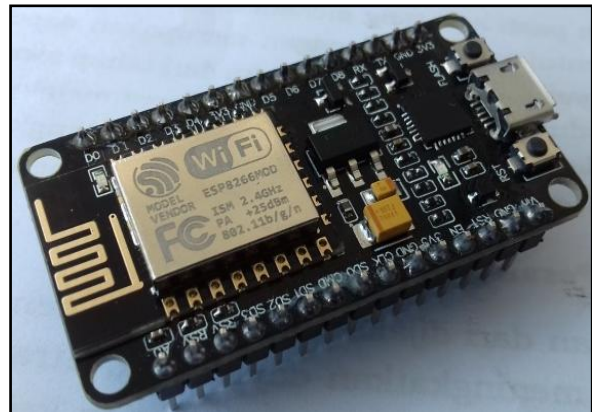
1. Laptop *lenovo*,
2. *Access point*,
3. Modul *NodeMCU 1.0 (Esp-12E)*,
4. Robot sepak bola beroda.

Dalam komunikasi laptop dan modul *NodeMCU 1.0 (Esp-12E)* ke duanya saling terhubung dengan *access point* yang sama ke *SSID* yang telah tersedia pada *access point* tersebut. Kemudian *Web Base Controller* dapat mengirim data ke *NodeMCU 1.0 (Esp-12E)* dan robot menjalankan perintah dari *Web Base Controller*.

D. NodeMCU 1.0 (Esp-12E)

1. Board NodeMcu

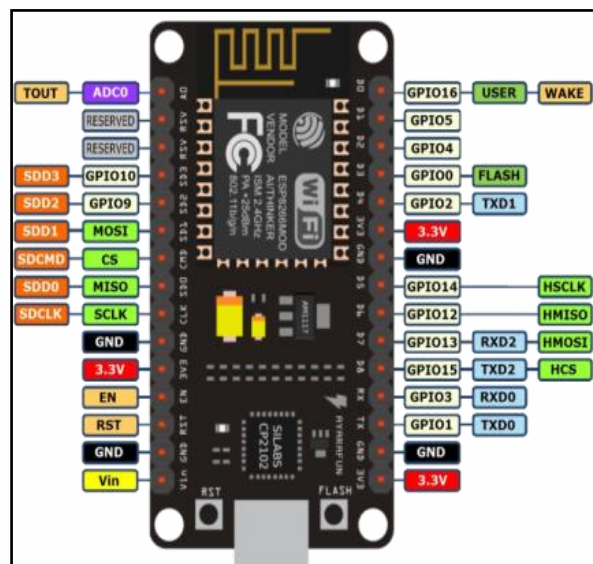
Modul *Wifi NodeMCU* adalah *firmware interaktif* berbasis *LUA Espressif Esp8266 Wifi Soc*. Bahasa pemrograman yang digunakan dalam pemrograman *NodeMCU 1.0 (Esp-12E)* adalah Bahasa *LUA* dan dapat juga diprogram menggunakan Bahasa *C* menggunakan *Arduino IDE*[3].



Gambar 2. Modul NodeMCU 1.0 (Esp-12E)

2. Deskripsi Pin NodeMcu

Beberapa fitur dan akses input output pada *nodemcu* diperlihatkan pada gambar 3 dibawah ini. Beberapa pin terhubung ke bagian master kontrol pada robot.



Gambar 3. Datasheet NodeMCU 1.0 (Esp-12E)
(Sumber : <https://iotbytes.wordpress.com>)

3. Spesifikasi NodeMcu 1.0

Tabel 1. Spesifikasi Modul NodeMCU 1.0 (Esp-12E)

SPESIFIKASI	NodeMCU V1
Mikrokontroler	ESP8266
Ukuran Board	57 mm x 30 mm
Tegangan Input	3.3 ~ 5V
GPIO	13 Pin
Kanal PWM	10 Kanal
10 bit ADC pin	1 Pin
Flash Memory	4 MB
Clock Speed	40/26/24 MHz
Wifi	IEEE 802.11 b/g/n
Frekuensi	2.4 GHz – 22.5 GHz
USB Port	Micro USB
USB to Serial Converter	CH340G

E. Robot Soccer

Pada penelitian ini robot yang digunakan adalah robot soccer yang digunakan pada Kontes Robot Sepak Bola (KRSBI) Beroda pada tahun 2018. Dimana robot tersebut dituntut untuk bermain sepak bola layaknya manusia dan bermanuver secara otomatis.



Gambar 4. Robot Soccer Yang Digunakan

F. Web Base Controller

Web Base Controller adalah sebuah aplikasi yang dapat berjalan menggunakan basis teknologi web atau Browser dengan menggunakan pemrograman HTML, CSS, Java Script [7]. Menjalankan Web Base Controller harus terhubung dengan jaringan Wifi yang sama antara NodeMCU 1.0 (Esp-12E) dan laptop sebagai pusat kontrol.

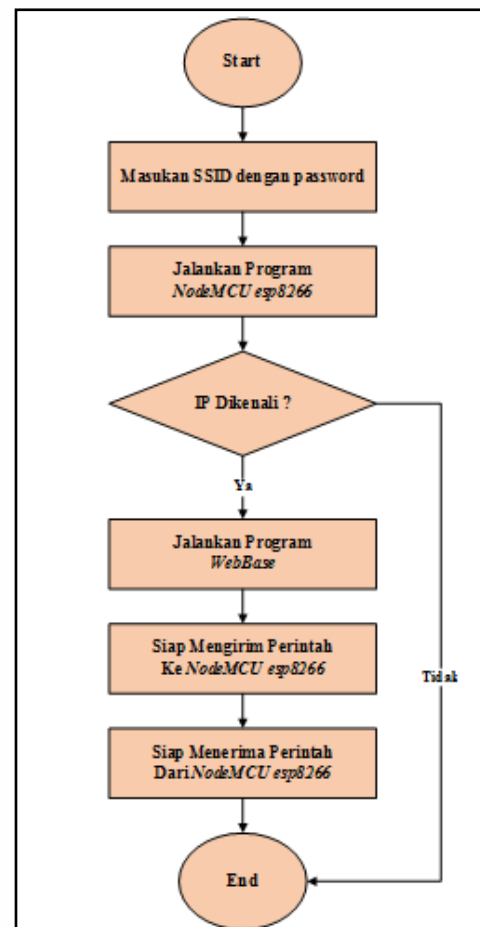
G. Web Base sebagai kontroler dan NodeMCU 1.0 (Esp-12E) sebagai media komunikasi pada kendali robot soccer

Sistem control robot sepak bola beroda

menggunakan Web Base Controller dapat mempermudah operator dalam menjalankan robot karena robot dikontrol secara otomatis dengan bantuan jaringan local dan menggunakan modul NodeMCU 1.0 (Esp-12E) .

H. Algoritma Program

Pada saat menggunakan sistem kendali robot soccer yang pertama dilakukan yaitu memasukan SSID dan password pada program NodeMCU 1.0 (Esp-12E) dan jalankan NodeMCU 1.0 (Esp-12E), ketika selesai dilakukan maka akan muncul IP jika terhubung ke jaringan. Langkah selanjutnya input IP yang dikenali NodeMCU 1.0 (Esp-12E) ke listing program Web Base Controller. Hal ini akan berfungsi untuk dikenali ketika terhubung ke jaringan yang sama antara NodeMCU 1.0 (Esp-12E) dan laptop yang digunakan sebagai sistem kendali robot soccer. Web Base Controller akan mengirim perintah ke NodeMCU 1.0 (Esp-12E) dan meneruskan perintah tersebut ke robot. Berikut Flowchart sistem Web Base Controller pada robot soccer :



Gambar 5. Flowchart Sistem Web Base Controller Pada Robot Soccer

III. HASIL DAN DISKUSI

A. Tampilan Web Base Controller

Dibawah ini adalah tampilan *Web Base Controller* yang mana tampilannya hampir mirip dengan aplikasi *Refree Box* internasional. Aplikasi *Refree Box* digunakan pada saat Kontes Robot Sepak Bola Indonesia (KRSBI) untuk digunakan juri dalam mengontrol pertandingan. Sedangkan *Web Base Controller* ini digunakan oleh peserta dalam mengontrol robotnya sendiri pada saat bertanding.



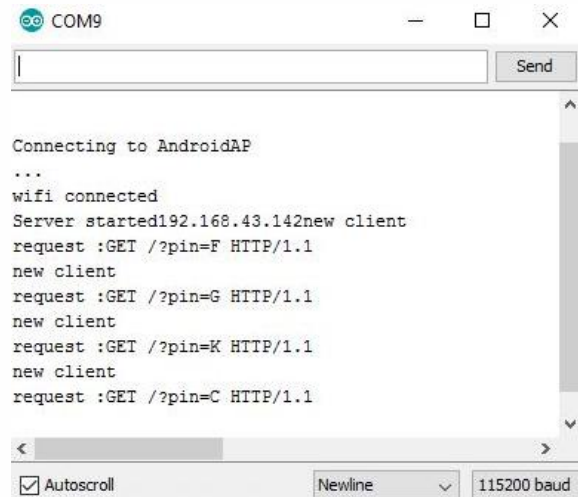
Gambar 6. Tampilan Web Base Controller

B. Hasil dan Pengujian Sistem

Pengujian sistem yang dilakukan peneliti ada beberapa langkah untuk memastikan apakah sistem bekerja dengan baik dan sesuai dengan keinginan. Berikut beberapa langkah pengujian data dari *Web Base Controller* ke Robot sepak bola sebagai berikut:

1. Pengujian IP

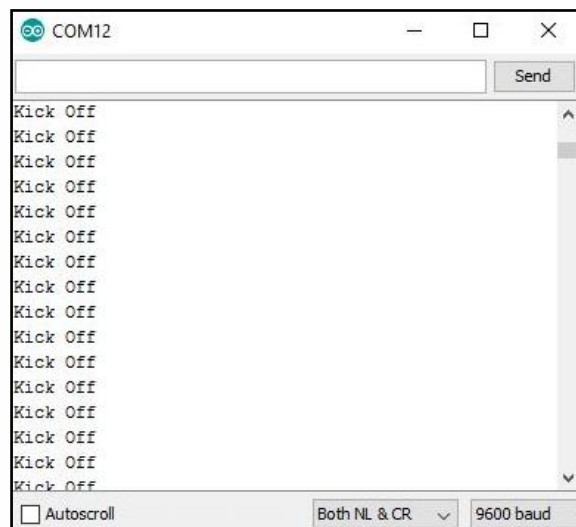
Pengujian ini bertujuan untuk memastikan bahwa IP *NodeMCU 1.0 (Esp-12E)* ke listing program *Web Base Controller* sudah benar. Hasil pengujian ini ditunjukkan pada gambar 7.



Gambar 7. Hasil Pengujian Web Base ke NodeMCU 1.0 (Esp-12E)

2. Hasil Penerimaan Data Pada NodeMcu

Pengujian ini dilakukan untuk memastikan bahwa *NodeMCU 1.0 (Esp-12E)* dapat menerima data yang dikirim dari *Web Base Controller*. Salah satu pengujian yang dilakukan dengan menekan tombol *kickOff* pada *Web Base Controller*. Hasil penerimaan data dari *NodeMCU 1.0 (Esp-12E)* dapat dilihat pada gambar 8.



Gambar 8. Hasil Penerimaan Data Dari Nodemcu 1.0 (Esp-12E)

3. Hasil Pengujian Web Base Controller

Dalam pengujian ini dilakukan percobaan sebanyak 22 kali Magenta dengan frekuensi percobaan satu kali untuk setiap tombol, hasil percobaan yang dilakukan adalah sebagai berikut :

Tabel 2. Pengujian Data Dari Web Base Controller

Pengujian ke-	Data Yang Dikirim Web Base	Data yang diterima nodeMCU esp8266	Keterangan
1	Start (s)	Start (s)	Data Valid
2	Stop (S)	Stop (S)	Data Valid
3	Dropball (N)	Dropball (N)	Data Valid
4	Park (L)	Park (L)	Data Valid
5	End part (h)	End part (h)	Data Valid
6	Reset (Z)	Reset (Z)	Data Valid
7	Cyan Goal (A)	Cyan Goal (A)	Data Valid
8	Cyan Kick off (K)	Cyan Kick off (K)	Data Valid
9	Cyan free kick (F)	Cyan free kick (F)	Data Valid
10	Cyan Goal kick (G)	Cyan Goal kick (G)	Data Valid
11	Throw In (T)	Throw In (T)	Data Valid
12	Cyan Corner (C)	Cyan Corner (C)	Data Valid
13	Cyan Penalty (P)	Cyan Penalty (P)	Data Valid
14	Repair Out cyan (O)	Repair Out cyan (O)	Data Valid
15	Magenta Goal (a)	Magenta Goal (a)	Data Valid
16	Magenta Kick off (k)	Magenta Kick off (k)	Data Valid
17	Repair Out magenta (o)	Repair Out magenta (o)	Data Valid
18	Magenta Free kick (f)	Magenta Free kick (f)	Data Valid
19	Magenta Goal kick (g)	Magenta Goal kick (g)	Data Valid

20	Magenta Throw In (t)	Magenta Throw In (t)	Data Valid
21	Magenta Corner (c)	Magenta Corner (c)	Data Valid
22	Magenta penalty (p)	Magenta penalty (p)	Data Valid

Keterangan:

Diketahui: uji coba yang dilakukan 22 kali dan tombol yang ada pada *Web Base Controller* berjumlah 22

$$\begin{aligned}\text{Persentasi keberhasilan} &= \frac{\text{Berhasil}}{\text{jumlah pengujian}} \times 100\% \\ &= \frac{22}{22} \times 100\% \\ &= 100\%\end{aligned}$$

Jadi persentase keberhasilan pengujian tombol pada *Web Base Controller* pada Robot adalah 100%

$$\begin{aligned}\text{Rata-rata keberhasilan} &= \frac{100}{1} \\ &= 100\%\end{aligned}$$

Jadi rata-rata keberhasilan pengujian tombol pada *Web Base Controller* adalah 100%

IV. KESIMPULAN

Berdasarkan penelitian yang dilakukan *Web Base Controller* bekerja 100%, karena dapat mengirim data ke modul *NodeMcu 1.0 (Esp-12E)* dengan baik, kemudian modul *NodeMCU 1.0 (Esp-12E)* mengirim data ke Robot, dan Robot dapat *bermanuver* sesuai dengan intruksi dari *Web Base Controller*. Sehingga memudahkan operator dalam mengontrol robot, baik mengontrol jarak dekat ataupun mengontrol jarak jauh.

V. UCAPAN TERIMA KASIH

Syukur Alhamdulillah senantiasa penulis panjatkan kehadiran Allah SWT atas nikmat, kesehatan dan kekuatan dalam penyusunan karya ilmiah ini. Shalawat serta salam senantiasa tercurahkan kepada muhammad SAW, keluarga dan para sahabatnya dan penegak sunahnya hingga akhir Zaman.

Pada kesempatan ini penulis mengucapkan rasa terimakasih sebesar-besarnya kepada kedua orang tua atas dukungan dan doa, serta teman-teman Adhi Guna

Robotech Club (AGRC) yang selalu mendukung dan membantu dalam menyelesaikan karya ilmiah ini.

VI. DAFTAR PUSTAKA

- [1] Mimi Hasmidar, dkk, "Pemanfaatan Sensor Jarak GP2D12 Sebagai Navigator Pada Robot Madikarobo ,"The 4th Indonesia Symposium on Robot Soccer Competition, 2016
- [2] Fitri, dkk, "Penggunaan Bahasa Pemrograman python sebagai Base Station pada kontes Robot sepak Bola Beroda,"5th Indonesia Symposium on Robotic Systems and Control, 2017
- [3] Mochamad Fajar Wicaksono, "Implementasi Modul Wifi NodeMCU Esp8266 untuk smart Home," jurnal teknik computer Unikom, 2017.
- [4] Thomas Agung, dkk. "Pengendalian Robot Mobile Berbasis Web dan Internet Protocol Melalui Jaringan WIFI," Jurnal tele volume 13 Nomor 2 Edisi oktober, 2015
- [5] H. Wibisono, dkk. "Rancang Bangun Sistem Komunikasi Data Game Controller Menggunakan Bluetooth pada Robot Humanoid soccer,"The 3th Indonesia Symposium on Robot 2015
- [6] Ricky Firmansyah, dkk. "Perancangan Web Based learning sebagai media pembelajaran berbasis ICT," Informatika, 2016
- [7] Muhammad jimmy. "Pengertian Web Based dan desktop Based,"Sistem Komputer, 2014

Pemanfaatan Sensor Jarak Untuk Mengidentifikasi Ruangan Pada Map KRPAI (Studi Kasus: Robot KRPAI 2018 STIKOM DB)

Agus Siswanto^a, M.Irwan Bustami^b, Abdul Fatah Kurniawan^c

^{abc} STIKOM Dinamika Bangsa Jambi

Jl. Jendral Sudirman Thehok – Jambi, telp (0741) 35095

^cabdulfatah557@hotmail.com

Abstrak – Dalam Kontes robot pemadam api indonesia memiliki 4 buah ruangan yang salah satunya terdapat titik api. Robot memiliki tujuan utama untuk memadamkan api tersebut yang diletakan secara acak di salah satu ruangan. Robot harus dapat melakukan pengecekan api terhadap ruangan tersebut. Untuk dapat melakukan pengecekan yang efektif Robot harus bisa membedakan saat posisi keluar ruangan dan posisi masuk ke ruangan agar tidak terjadi pengecekan api di luar ruangan. Solusinya dapat Solusinya dapat digunakan sebuah metode yaitu logika fuzzy. dari hasil penelitian yang telah dilakukan logika fuzzy dapat mengetahui bahwa robot memasuki ruangan dan melakukan pengecekan api di ruangan tersebut dan saat keluar dari ruangan robot tidak akan melakukan pengecekan api.

Kata kunci: logika fuzzy, robot berkaki

I. PENDAHULUAN

Saat ini perkembangan robotika sangatlah pesat. Berbagai macam robot pintar semakin banyak diciptakan untuk dapat membantu aktifitas manusia. Banyak perusahaan industri yang sudah memakai teknologi robotika dalam pembuatan produk mereka, selain dalam bidang industri robot juga dibuat untuk melakukan sesuatu yang penuh resiko saat manusia yang melakukannya seperti robot penjinak bom ataupun robot pemadam kebakaran.

Di Indonesia, salah satu bukti berkembangnya teknologi robotika dengan adanya ajang kontes robot tahunan yang dikenal dengan nama Kontes Robot Indonesia. Dalam kontes tersebut terdapat beberapa kategori yaitu[1]:

1. Kontes Robot ABU Indonesia
2. Kontes Robot Pemadam Api Indonesia
3. Kontes Robot Sepak Bola Indonesia
 - a. Divisi beroda
 - b. Divisi humanoid
4. Kontes Robot Seni Tari Indonesia

Setiap kategori akan dipertandingkan di regional untuk melakukan seleksi agar dapat mengikuti pertandingan di nasional. Juara pertama nasional setiap kategori akan mewakili Indonesia di tingkat internasional.

Dari beberapa kategori yang dipertandingkan

tersebut, ada satu kategori yang ingin penulis bahas yaitu Kontes Robot Pemadam Api Indonesia. Dalam Kontes tersebut terdapat 4 buah ruangan yang salah satunya terdapat titik api. Robot memiliki tujuan utama untuk memadamkan api tersebut yang diletakan secara acak di salah satu ruangan. Robot harus dapat bernavigasi menelusuri setiap ruangan yang ada dan melakukan pengecekan api terhadap ruangan tersebut. Untuk dapat melakukan pengecekan yang efektif Robot harus dapat membedakan saat posisi keluar ruangan dan posisi masuk ke ruangan agar tidak terjadi pengecekan api di luar ruangan. Solusinya dapat menggunakan sebuah metode yaitu logika fuzzy.

Penelitian yang dilakukan oleh Muhammad bagus bintang timur, dkk yaitu “menerapkan Desain dan Implementasi Kendali Cerdas untuk Robot Quadpod (Berkaki Empat) – Studi Kasus Robot Pemadam Api (RPA)”[2]. Pada penelitian tersebut menggunakan logika fuzzy untuk menentukan posisi robot ketika berada didalam arena. Pengujian penelitian tersebut dilakukan menggunakan software matlab yang data dari sensor diambil secara acak, kemudian data tersebut diproses oleh fuzzy dan didapatkanlah hasil berupa posisi robot. Posisi robot inilah yang akan diambil untuk menentukan keputusan apa yang akan diambil oleh robot.

Dari penelitian yang masih berupa simulasi dalam bentuk software matlab tersebut, maka penulis tertarik untuk menerapkannya langsung ke robot KRPAI 2018 dan mengembangkannya untuk dapat mengambil keputusan pengecekan api pada saat posisi masuk dan keluar ruangan. Berdasarkan uraian tersebut penulis tertarik melakukan penelitian dengan judul “Pemanfaatan Sensor Jarak Untuk Mengidentifikasi Ruangan Pada Map KRPAI”.

II. METODOLOGI

A. Robot Pemadam Api

Robot pemadam api merupakan robot yang dibuat dengan tujuan memadamkan api pada daerah yang sulit untuk dijangkau manusia. Robot Pemadam Api harus memiliki kemampuan navigasi yang handal dalam menyelesaikan misinya dengan menerapkan metode yang baik.[3].

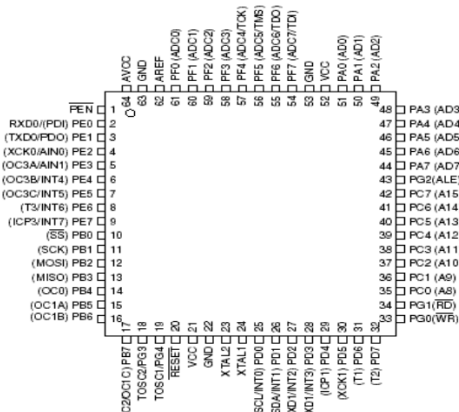
B. Robot Berkaki Enam

Robot hexapod setidaknya dibutuhkan minimal enam buah motor servo, oleh karena itu pergerakan tiap kaki robot tersebut tidak bisa bergerak sendiri-sendiri tetapi ada yang secara bersamaan serta robot tersebut memiliki kecepatan langkah kaki yang konstan. Kemampuan bergerak dan bernavigasi merupakan modal yang harus dimiliki robot untuk dapat menjalankan tugasnya tersebut[4].

C. Mikrokontroler

Mikrokontroler ATmega128 digunakan sebagai pemroses utama yang bertugas memproses seluruh masukan, serta mengatur seluruh keluaran.

Mikrokontroler ATmega 128 merupakan mikrokontroler keluarga AVR yang mempunyai kapasitas flash memori 128KB. AVR (Alf and Vegard's Risc Processor) merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (Reduced Instruction Set Computer). Secara umum, AVR dapat terbagi menjadi 4 kelas, yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga ATmega, dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, peripheral, dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan, bisa dikatakan hampir sama[5].



Gambar 1. ATmega 128

D. Devantech SRF04 Ultrasonic Range Finder

Sensor yang dapat mendeteksi halangan dalam jangkauan 3 cm – 3 m menggunakan pantulan gelombang ultrasonik[6].

Sensor ini bekerja dengan cara memancarkan gelombang ultrasonik lalu menangkap pantulan gelombang tersebut dalam rentang waktu tertentu.



Gambar 2. Sensor Jarak Srf04

E. Motor Servo

Jumlah Servo yang digunakan berjumlah 18 yang berjenis Dynamixel AX-18. Servo berfungsi sebagai aktuator dari robot. Masing-masing kaki robot memiliki 3 buah servo sehingga membentuk kaki yang memiliki tiga derajat kebebasan[7].

F. Codevision AVR

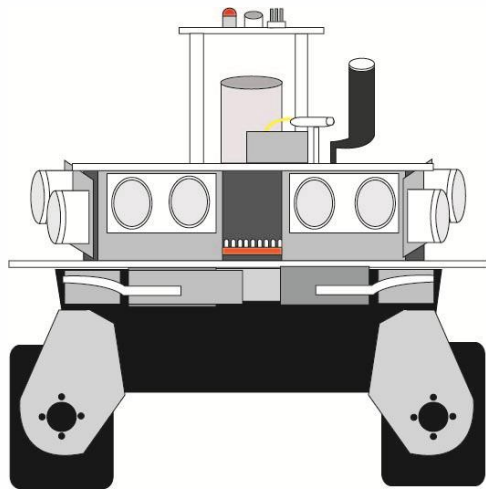
Code Vision AVR C Compiler (CVAVR) merupakan kompilator bahasa C untuk AVR. Kompilator ini cukup memadai untuk belajar AVR, karena selain mudah penggunaannya juga didukung berbagai fitur yang sangat membantu dalam pembuatan software untuk keperluan pemrograman AVR[8].



Gambar 3. Codevision AVR

G. Desain Robot

Robot pemadam api ini memiliki 6 kaki atau hexapod yang dilengkapi 10 sensor jarak bertipe srf04 di setiap sisi atasnya sebagai pendeteksi dinding maupun halangan didepanya.



Gambar 4. Desain Robot Berkaki

H. Logika Fuzzy

Metode ini dikembangkan untuk membantu pengambil keputusan dalam melakukan pengambilan keputusan terhadap beberapa alternatif keputusan untuk mendapatkan suatu keputusan yang akurat dan optimal. Logika fuzzy adalah salah satu cabang dari AI (Artificial Intelligence). Logika fuzzy merupakan modifikasi dari teori himpunan dimana setiap anggotanya memiliki derajat keanggotaan yang bernilai kontinu antara 0 sampai 1[9].

I. Aturan Fuzzy If - Then

Jika terdapat dua variabel maka implikasi dengan teknik fuzzy adalah sebagai berikut :

If x is A Then Y is B

atau

$$y=f((x,A),B)$$

Dari sistem fuzzy diatas, maka sistem tersebut dapat langsung berjalan tanpa harus melalui komposisi dan dekomposisi fuzzy. Nilai output dapat diperkirakan secara langsung dari nilai keanggotaan yang ada.

Aturan *Fuzzy If-Then* (atau disebut juga aturan fuzzy, atau pernyataan kondisional Fuzzy) adalah aturan yang digunakan untuk merumuskan relasi kondisi antara dua atau lebih himpunan fuzzy. Bentuk umum :

If (X_1 is A_1) $\square$ (X_2 is A_2) ... (X_n is A_n) Then Y is B; x_i, y_i skalar, dan A, B

himpunan Fuzzy terdapat 2 fungsi yang dapat dipergunakan yaitu :

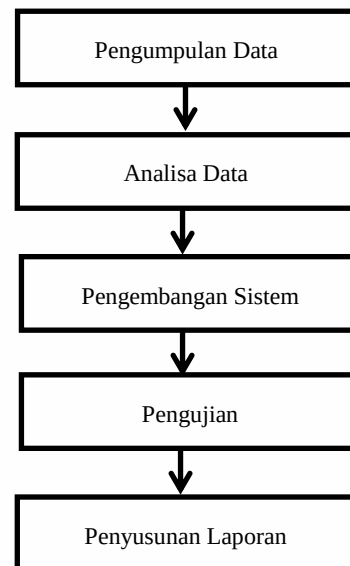
1. Min (*minimum*), fungsi ini akan memotong output (konsekuen) himpunan fuzzy.
2. Dot (*product*), fungsi ini akan menskala output himpunan fuzzy.

[10]

J. Kerangka Penelitian

Pada suatu penelitian kita perlu membuat suatu kerangka kerja penelitian agar apa saja yang kita butuhkan untuk merancang suatu sistem dapat tergambar dengan jelas dan dapat diperoleh suatu logika, baik didalam melakukan pengujian maupun

dalam membuat kesimpulan. Kerangka kerja (*framework*) penelitian merupakan sebuah bagan yang terdiri atas tahapan-tahapan yang tersusun secara sistematis yang kemudian akan digunakan dalam proses penyelesaian penelitian. Adapun kerangka kerja penelitian yang akan digunakan adalah sebagai berikut:



Gambar 5. Kerangka Kerja Penelitian

III. HASIL DAN DISKUSI

A. Bentuk Fisik Robot Berkaki

Hasil fisik robot merupakan kegiatan merubah dari rancangan desain robot menjadi bentuk fisik robot yang dapat dieksekusi. Kegiatan ini memadukan rancangan fisik antar muka dan algoritma menjadi satu kesatuan. Panjang robot yang dibuat adalah 26cm, sedangkan lebar dari robot adalah 21cm. Berikut ini adalah hasil dari bentuk fisik robot.

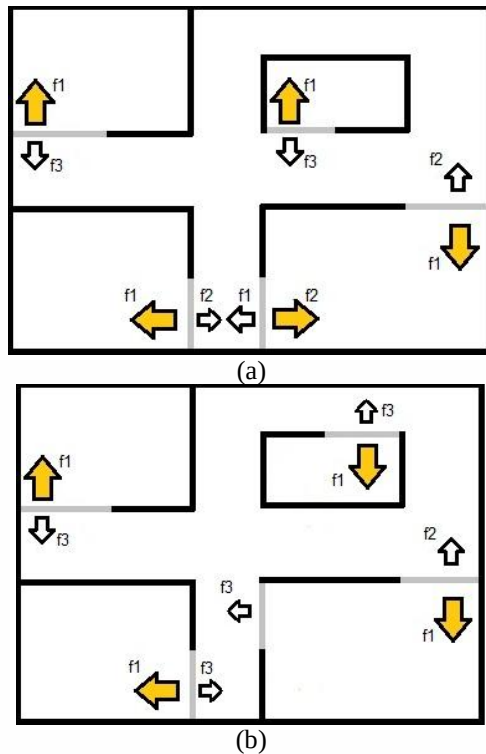


Gambar 6. Hasil bentuk fisik robot berkaki

B. Posisi Fuzzy Dalam dan luar Ruangan

Di dalam ruangan yang sudah ditentukan ukurannya, penulis membagi posisi fuzzy untuk robot menjadi 3 bagian yaitu F1 (robot berada di dalam ru

ngan), F2 (robot berada di luar ruangan) dan F3 (robot berada di luar ruangan). Adapun pembagiannya dapat dilihat sebagai berikut :



Gambar 7. Posisi fuzzy (a) dan (b)

C. Fuzzy If – Then

Aturan Fuzzy If-Then atau disebut juga aturan fuzzy atau pernyataan kondisional Fuzzy adalah aturan yang digunakan untuk merumuskan relasi kondisi antara dua atau lebih himpunan fuzzy. Adapun Fuzzy If-Then pada penelitian ini dapat dilihat pada tabel 1.

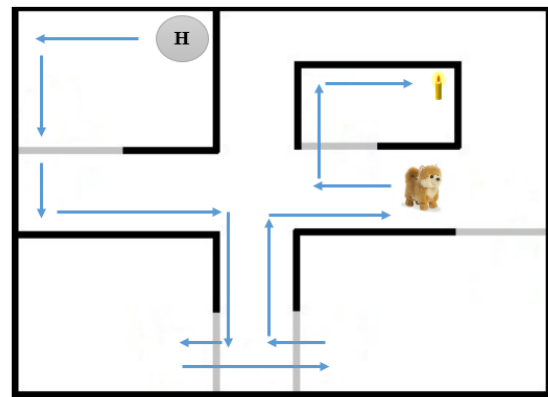
Tabel 1. Posisi fuzzy dengan aturan if-then

No	Aturan fuzzy if then	Posisi
1	if (jrk_kr1<=c_kr)	F1
2	Else if (jrk_dpn1 >= c_dp jrk_dpn2>=c_dp)	F2
3	Else if (jrk_kn1 + jrk_kn2 + jrk_kr1 + jrk_kr2 <= 150 && jrk_dpn1 <= 45 && jrk_dpn2<=45)	F1
4	else	F3

D. Pengujian Robot Dalam Mencari Ruangan Api

Pengujian ini bertujuan untuk melihat kemampuan robot dalam mengetahui posisi robot yang berada di dalam ruangan atau berada di luar ruangan setelah robot mendeteksi adanya garis. Jika robot berada di luar ruangan robot tidak akan melakukan pengecekan adanya api, selain itu jika robot berada di dalam ruangan robot akan melakukan pengecekan api di ruangan tersebut.

Jika di ruangan robot mendeteksi adanya titik api maka robot akan mencari api di dalam ruangan tersebut, jika di ruangan tersebut robot tidak mendeteksi adanya titik api maka robot tidak akan menelusuri ruangan tersebut dan akan berputar menuju ruangan selanjutnya. Arena pengujian dan jalur lintasan pengujian ditunjukkan pada gambar 8.



Gambar 8. Bentuk arena pengujian dan jalur lintasan robot

IV. KESIMPULAN

1. Penelitian ini menghasilkan robot pemadam api berkaki enam yang memiliki 10 sensor jarak untuk mendeteksi penghalang.
2. Robot ini memiliki system logika fuzzy, sehingga robot mengetahui bahwa dirinya berada didalam ruangan maupun diluar ruangan dan melakukan pengecekan api hanya ketika memasuki ruangan.
3. Robot belum mampu membedakan tiap-tiap ruangan yang ada karena hanya mengandalkan sensor jarak sehingga ada beberapa ruangan yang memiliki karakteristik yang sama.

V. DAFTAR PUSTAKA

- [1] K. R. Indonesia, "Kontes Robot Indonesia Kri 2018," 2018.
- [2] M. Bagus, B. Timur, A. Fanany, O. Gaffar, and A. Wajiansyah, "Desain dan Implementasi Kendali Cerdas untuk Robot Quadpod (Berkaki Empat) – Studi Kasus Robot Pemadam Api (RPA)," vol. 5, no. 2, 2017.
- [3] A. Azhar, "Perancangan Fuzzy Logic Model Sugeno untuk Wall Tracking pada Robot Pemadam Api," *Politek. Caltex Riau*, vol. 1, no. 1, pp. 1–11, 2015.
- [4] M. F. Abdi, "Model Kendali Berbasis Perilaku Pada Robot Berkaki Hexapod 3 DOF," pp. 1–8.
- [5] F. Astria, M. Subito, and D. W. Nugraha, "Rancang Bangun Alat Ukur pH Dan Suhu Berbasis Short Message Service (SMS) Gateway," *J. Metr.*, vol. 1, no. 1, pp. 47–55, 2014.
- [6] D. Santoso, D. Susilo, and J. Wasesa, "Pengembangan Robot Berkaki Enam yang dapat Mengidentifikasi Ruang pada Map Kontes Robot Pemadam Api Indonesia menggunakan Algoritma Pengenalan Karakter Ruang," 2016.
- [7] A. M. Fathan, F. T. Elektro, and U. Telkom, "Desain Dan Implementasi Algoritma Mapping Menggunakan Sensor Ultrasonik Dan Kompas Pada Autonomous Quadroped Robot Design and Implementation Mapping Algorithm Using Ultrasonic," vol. 3, no. 2, pp. 2203–2210, 2016.
- [8] A. Aryanto and T. I. Tjendrowasono, "Speed Journal – Sentra Penelitian Engineering dan Edukasi – Volume 11

- No 4 - 2014 - ijns.org,” *Journal, Speed Eng. Sentra Penelit.*, vol. 11, no. 4, pp. 33–40, 2014.
- [9] S. Kusumadewi, I. Guswaludin, K. Sistem, P. Keputusan, and D. Support, “Fuzzy Multi-Criteria Decision Making,” *Media Inform.*, vol. 3, no. 1, pp. 25–38, 2005.
- [10] Helfi Nasution. (2012). Implementasi Logika Fuzzy pada Sistem Kecerdasan Buatan. *Jurnal ELKHA Vol.4*, 4-7.

Implementasi Interpolasi Linier Pada Gerakan Tari Robot Humanoid

Yanuar Ari Setiawan¹, M. Taufiqurrohman²

Fakultas Teknik dan Ilmu Kelautan Universitas Hang Tuah Surabaya
Jalan Arif Rahman Hakim No.150, Keputih-Sukolilo, Surabaya 60111
Email : ari.yanuar26@gmail.com

Abstrak: Paper ini berisi tentang implementasi metode interpolasi linier pada gerakan tari robot humanoid. Implementasi gerakan tari yang digunakan pada penelitian ini adalah gerakan tindak, gerakan gedrug, dan gerakan iket yang merupakan gerak dasar tari remo. Penelitian ini merupakan implementasi metode interpolasi linier dalam melakukan gerakan tari pada robot humanoid. Tujuan penelitian ini adalah untuk mempermudah dalam memetakan gerakan pada robot karena hanya memberikan nilai sudut bukan data 10bit lagi. Hasil pengujian menunjukkan perhitungan manual dengan interpolasi linier pada gerakan tindak dengan persentase error 0,002%, pada gerakan gedrug dengan persentase error 0,0007%, pada gerakan iket dengan persentase error 0,001%.

Kata kunci: gerak tari, robot humanoid, interpolasi linier

I. PENDAHULUAN

Kontes Robot Seni Tari Indonesia (KRSTI) merupakan suatu ajang kompetisi perancangan, pembuatan dan pemrograman robot yang disertai dengan unsur-unsur seni dan budaya bangsa Indonesia khususnya seni tari yang telah terkenal di bumi pertiwi. KRSTI pertamakali diadakan pada tahun 2010 dengan tema “Robot Penari Jaipong”, tahun 2011 dengan tema “Robot Penari Pendet”, tahun 2012 dengan tema “Robot Penari Klono Topeng”, tahun 2013 mengangkat tema “Robot Penari Piring”, tahun 2014 mengangkat tema “Robot Penari Hanuman Duto”, pada tahun 2015 mengangkat tema “Robot Penari Bambang Cakil”, pada tahun 2016 mengangkat tema “Robot Penari Remo”, pada tahun 2017 mengangkat tema “Robot Penari Gending Sriwijaya”. Setiap tim peserta yang terdiri dari 3(tiga) mahasiswa dengan seorang dosen pembimbing, diwajibkan untuk membuat satu atau beberapa robot yang terkoordinasi untuk menampilkan seni budaya yang diinginkan sesuai tema kontes [1].

Robot humanoid yang dirancang ini tersusun dari 24 servo dynamixel ax 12 yang terdiri bagian tubuh seperti manusia yaitu ada bagian kepala, bagian lengan dan bagian kaki. Robot yang dirancang bertujuan untuk dapat bergerak melakukan gerakan tari mengikuti alunan musik. Robot ini dibuat dalam rangka lomba Kontes Robot Indonesia untuk divisi Kontes Robot Seni Indonesia (KRSTI). Divisi ini

merupakan sebuah divisi baru dalam lomba KRI, dimana robot yang dibuat harus mampu bergerak melakukan gerakan tari mengikuti alunan lagu dengan tema musik yang disediakan panitia. Untuk dapat menari mengikuti alunan irama, maka diperlukan sebuah system identifikasi terhadap ciri ciri apa saja yang terkandung dalam sebuah musik, yang meliputi tempo, irama, notasi alat instrumentasi dan keserasian. Pembuatan system identifikasi musik yang terintegrasi secara hardware merupakan kesulitan tersendiri.

Pada perlombaan KRI robot pada divisi seni tari diharuskan mampu bergerak melakukan gerakan tari yang sudah ditentukan. Salah satu metode yang dapat digunakan adalah metode interpolasi linier. Metode ini digunakan untuk mempermudah dalam melakukan gerakan tari. Keunggulan menggunakan metode interpolasi linier adalah dapat memetakan sudut derajat yang diinginkan untuk menggerakkan robot dalam melakukan gerakan tari.

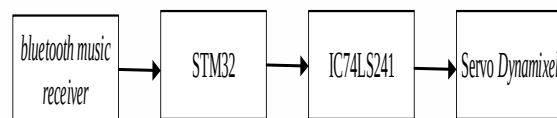
Penelitian ini akan mengembangkan teknologi tentang implementasi interpolasi linier untuk robot humanoid dalam melakukan gerakan tari. Metode ini diharapkan dapat mempermudah dalam pemrograman robot humanoid dalam membuat gerakan tari.

II. METODE PENELITIAN

Pada penelitian ini menggunakan metode interpolasi linier, dengan melakukan perancangan hardware dan perancangan software. Adapun susunan metode penelitian diantaranya :

A. Perancangan Perangkat Keras

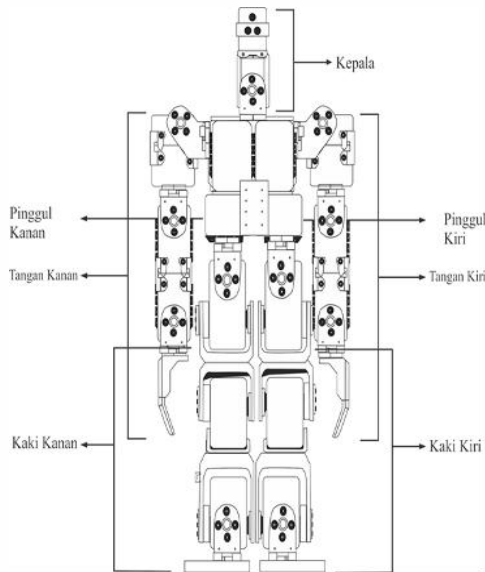
Adapun diagram blok keseluruhan alat yang ditunjukkan pada gambar dibawah ini :



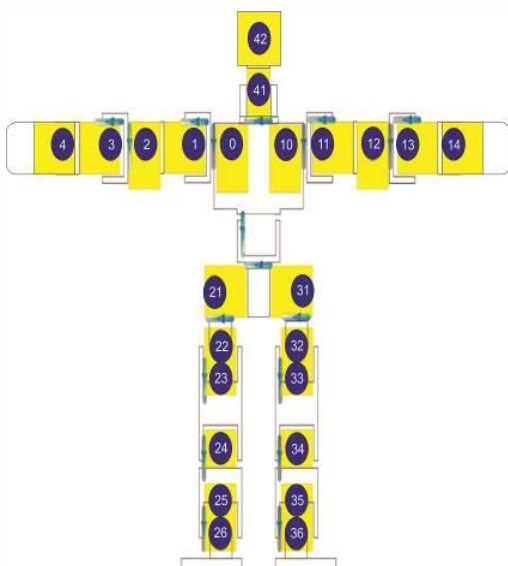
Gambar 1. Diagram Blok Sistem

Secara keseluruhan, system ini terdiri dari masukan, mikrokontroler, dan keluaran. Berdasarkan diagram blok pada Gambar 1 sistem ini memiliki

beberapa komponen yaitu *bluetooth music receiver*, modul STM32, ic 74ls241 dan servo dynamixel. *Bluetooth music receiver* untuk menerima data suara (musik) lalu data suara akan diproses oleh mikrokontroler untuk mengontrol servo dynamixel. Untuk mengontrol servo dynamixel di perlukan memasukkan data 10 bit yaitu antara 0 – 1023 yang mempresentasikan sudut antara 0 derajat sampai dengan 300 derajat [2]. Desain robot dalam penelitian ini adalah sebuah robot humanoid yang tersusun dari 24 servo dynamixel ax 12 yang terdiri bagian tubuh seperti bagian kepala, bagian lengan dan bagian kaki.



Gambar 2. Rancangan Mekanik Robot



Gambar 3. Penomoran ID Servo

B. Interpolasi linier

Pada penelitian ini metode interpolasi linier digunakan untuk membuat nilai tengah pada servo dynamixel. Pada Metode interpolasi dijelaskan bahwa Perbandingan jarak ($X - X_1$) dengan jarak ($X_2 - X_1$) sama dengan perbandingan jarak ($Y - Y_1$) dengan jarak ($Y_2 - Y_1$). [3]

Dengan cara ini setiap titik yang berada di antara dua titik diketahui memiliki hubungan linear akan dapat ditentukan dengan perhitungan menggunakan rumus interpolasi linear. Adapun persamaan interpolasi linier yang digunakan pada sistem ini adalah :

$$\frac{y_2 - y_1}{x_2 - x_1} (x - x_1) + y_1 \dots \dots \dots (1)$$

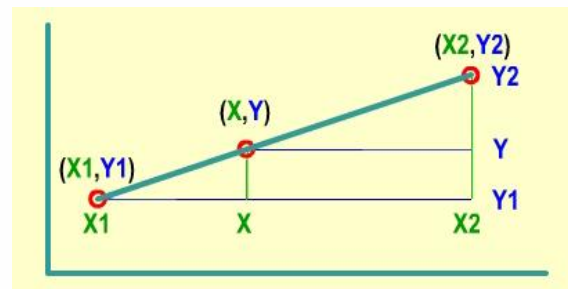
X_1 = Nilai batas bawah

Y_1 = Derajat batas bawah

X_2 = Nilai batas atas

Y_2 = Derajat Batas atas

X = Nilai derajat inputan yang di inginkan



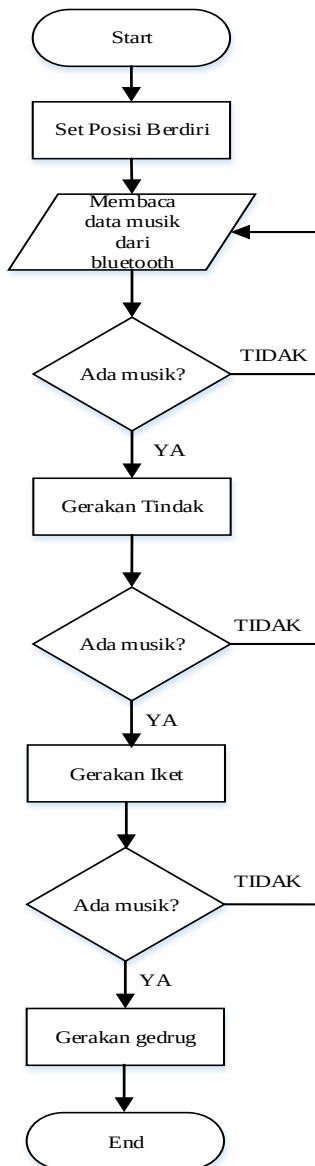
Gambar 3. Ilustrasi Interpolasi Linier

Pada penelitian ini data yang sudah diketahui adalah nilai batas bawah dan atas dari servo dynamixel yaitu antara 0 – 1023 dan jika dikonversi dengan derajat adalah 0 – 300 derajat. Data derajat nanti akan petakan menjadi -150 sampai 150, minus pada derajat menandakan perbedaan arah putaran servo. Dengan menggunakan rumus interpolasi diatas akan mempermudah untuk pemetaan gerakan tari pada robot humanoid.

C. Perancangan Perangkat Lunak

Adapun *flowchart* dari sistem yang digunakan pada robot humanoid dalam melakukan gerakan tari ditunjukkan pada Gambar 4.

Program dimulai dengan kondisi awal robot sudah berdiri. Kemudian Bluetooth music receiver akan membaca data musik yang dikirim melalui jaringan Bluetooth lalu data akan diproses pada mikrokontroler. Apabila ada data musik yang terbaca maka robot akan melakukan gerakan tari dan jika tidak ada data musik yang terbaca robot akan diam tidak melakukan gerakan tari dan akan bergerak melakukan gerakan tari apabila ada data musik yang diterima.



Gambar 4. Flowchart

III. HASIL DAN DISKUSI

Setelah melakukan perancangan perangkat keras (*hardware*) dan juga perangkat lunak (*software*). Penulis melakukan pengujian alat. Pengujian sistem ini dilakukan untuk mengetahui dan menguji kehandalan dari sistem yang telah dibuat. Berikut merupakan hasil pengujian :

A. Pengujian Bluetooth Music Receiver

Bluetooth music receiver ini digunakan untuk menerima sinyal suara yang dihasilkan dari bunyi musik melalui jaringan *bluetooth*, diaman pin VCC akan dihubungkan ke pin VCC baterai dengan tegangan 5V, *ground bluetooth music receiver* dihubungkan ke *ground* baterai, pin output *bluetooth music receiver* dihubungkan dengan pin A5 pada stm32 untuk membaca tegangan analog dari *bluetooth*

music receiver. Berikut tabel percobaan *bluetooth music receiver*:

Tabel 1. Data pengujian bluetooth music receiver

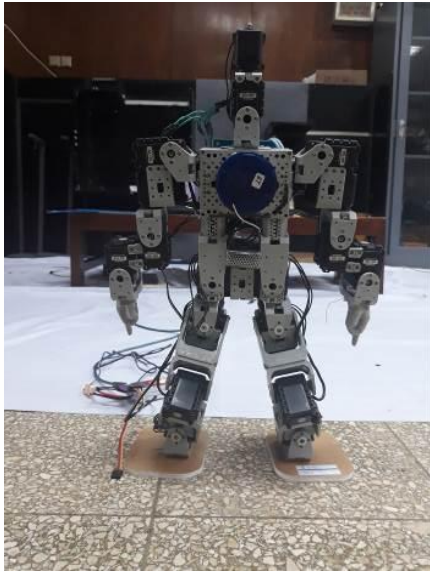
Percobaan	Tidak Ada Musik (des)	Ada Musik (des)
1	0	2856
2	1	1600
3	0	2654
4	1	2035
5	0	2856
6	0	2398
7	0	2187
8	0	1987
9	0	2098
10	1	2698
Rata -rata	0.3	2336.9

Dari hasil percobaan pembacaan nilai tegangan analog *bluetooth music receiver* pada Tabel 1, diperoleh hasil ketika tidak ada suara musik bernilai 0,3 dan ketika ada suara musik bernilai 2336.9.

B. Interpolasi Linier Pada Gerak Tari

Pada robot humanoid terdiri dari Servo dynamixel ax 12 ini digunakan untuk penggerak robot yang disusun berbentuk seperti manusia, memiliki beberapa bagian yaitu 2 tangan, 2 kaki dan kepala.

Dari datasheet pada servo dynamixel yaitu servo dynamixel bekerja ketika diberi nilai antara 0 – 1023 dengan besar derajat dari 0 derajat sampai 300 derajat. Dari data tersebut bisa dihitung ketika ingin menambah derajat servo yaitu membagi nilai maksimal dengan derajat maksimal sehingga didapat $1023 / 300 = 3,41$.



Gambar 4. Robot Pose Berdiri

Tabel 2. Data servo dynamixel pose berdiri

No	Nomor servo	Data nilai servo (des)
1	0	512
2	1	512
3	2	499
4	3	508
5	4	512
6	10	512
7	11	512
8	12	509
9	13	528
10	14	512
11	21	502
12	22	524
13	23	556
14	24	553
15	25	517
16	26	504
17	31	502
18	32	500
19	33	468
20	34	471
21	35	507
22	36	520
23	41	512
24	42	512

Dari Tabel 2 maka didapatkan data untuk tiap-tiap servo hal ini digunakan untuk menentukan nilai derajat atas dan derajat bawah pada tiap tiap servo

selanjutnya akan dicari nilai batas bawah dan batas atas seperti Tabel 3.

Tabel 3. Data pemetaan servo dynamixel

Nomor servo	batas bawah(des)	batas atas(des)	derajat batas bawah	derajat batas atas
0	0	1023	-150.147	149.8534
1	0	1023	-150.147	149.8534
2	0	1023	-146.334	153.6657
3	0	1023	-148.974	151.0264
4	0	1023	-150.147	149.8534
10	0	1023	-150.147	149.8534
11	0	1023	-150.147	149.8534
12	0	1023	-149.267	150.7331
13	0	1023	-154.839	145.1613
14	0	1023	-150.147	149.8534
21	0	1023	-147.214	152.7859
22	0	1023	-153.666	146.3343
23	0	1023	-163.05	136.9501
24	0	1023	-162.17	137.8299
25	0	1023	-151.613	148.3871
26	0	1023	-147.801	152.1994
31	0	1023	-147.214	152.7859
32	0	1023	-146.628	153.3724
33	0	1023	-137.243	162.7566
34	0	1023	-138.123	161.8768
35	0	1023	-148.68	151.3196
36	0	1023	-152.493	147.5073
41	0	1023	-150.147	149.8534
42	0	1023	-150.147	149.8534

Dari Tabel 3 dapat dimasukkan dalam metode interpolasi linier yaitu dengan rumus interpolasi linier

Untuk membuat gerakan tari penulis terlebih dahulu memperkirakan sudut tiap servo lalu dilihat apakah gerakan sudah sesuai. Jika belum sesuai maka tinggal menambahkan sudut tanpa harus menghitung manual .



Gambar 4. Robot Pose Gerakan Awal Tari Gedrug

Tabel 4. Data Percobaan Pose Gerakan awal tari gedrug

Nomor servo	Sudut (derajat)	Perhitungan manual	Perhitungan dengan interpolasi	Error
10	121	924.61	924.60	0,001%
11	14	559.74	559.73	0,001%
12	8	536.28	536.28	0%
13	38	657.58	657.57	0,001%
14	35	631.35	631.34	0,001%
0	121	924.61	924.6	0,001%
1	13	556.33	556.32	0,001%
2	9	529.69	529.68	0,001%
3	37	634.17	634.16	0,001%
4	40	648.4	648.3	0,001%
	Rata-rata	6602.76	6602.58	0.002%

Perhitungan Manual : Sudut servo x 3.41 + nilai pose robot berdiri

$$\text{Error} = \frac{(\text{Data perhitungan interpolasi} - \text{data perhitungan manual})}{\text{data perhitungan manual}} \times 100 \%$$



Gambar 4. Robot Pose Gerakan Awal Tari Tindak

Tabel 5. Data Percobaan Pose Gerakan Awal Tari Tindak

Nomor servo	Sudut (derajat)	Perhitungan manual	Perhitungan dengan interpolasi	Error
10	90	818.9	818.89	0,001%
11	14	559.74	559.73	0,001%
12	-80	272.8	272.8	0%
13	38	657.58	657.57	0,001%
14	70	750.7	750.69	0,001%
0	121	924.61	924.6	0,001%
1	-100	853	852.99	0,001%
2	9	529.69	529.68	0,001%
3	627.35	627.34	627.35	0,001%
4	648.4	648.39	648.4	0,001%
	Rata-rata	6642.75	6642.7	0.0007 %

Data tabel 4 merupakan gerakan awal ketika melakukan gerakan tari gedrug. Servo yang bergerak untuk melakukan gerakan ini hanya 10 buah(bagian tangan) dari 24 buah servo pada robot. 14 servo (bagian kaki) yang lainnya tidak ada perhitungan karena penelitian ini tidak menggerakkan bagian kaki.

Dari data percobaan Gerakan tari gedrug didapatkan hasil , diperoleh rata rata error sebesar 0.002%.

Data tabel 5 merupakan gerakan awal ketika melakukan gerakan tari tindak. Servo yang bergerak untuk melakukan gerakan ini hanya 10 buah dari 24 buah servo pada robot. 14 servo (bagian kaki) yang lainnya tidak ada perhitungan karena penelitian ini tidak menggerakkan bagian kaki

Dari data percobaan Gerakan tari tindak didapatkan hasil , diperoleh rata rata error sebesar 0.0007%.



Gambar 4. Robot Pose Gerakan Awal Tari Iket

Tabel 6. Data Percobaan Gerakan tari iket

Nomor servo	Sudut (derajat)	Perhitungan manual	Perhitungan dengan interpolasi	Error
10	30	614.3	614.29	0,001%
11	14	559.74	559.73	0,001%
12	-120	99.80	99.80	0%
13	38	657.58	657.57	0,001%
14	70	750.7	750.69	0,001%
0	121	924.61	924.6	0,001%
1	-100	853	852.99	0,001%
2	9	529.69	529.68	0,001%
3	35	627.35	627.34	0,001%
4	60	716.6	716.59	0,001%
	Rata-rata	6333.37	6333.28	0.001%

Data tabel 6 merupakan gerakan awal ketika melakukan gerakan tari tindak. Servo yang bergerak untuk melakukan gerakan ini hanya 10 buah dari 24 buah servo pada robot. 14 servo (bagian kaki) yang lainnya tidak ada perhitungan karena penelitian ini tidak menggerakkan bagian kaki.

Dari data percobaan Gerakan tari iket didapatkan hasil , diperoleh rata rata error sebesar 0.001%.

IV. KESIMPULAN

Berdasarkan analisis dan pembahasan hasil penelitian didapat kesimpulan sebagai berikut:

1. Data dari gerakan tari tindak jika menggunakan interpolasi linier terdapat eror 0,002% .
2. Data dari gerakan tari gedrug jika menggunakan interpolasi linier terdapat eror 0,0007% .
3. Data dari gerakan tari iket jika menggunakan interpolasi linier terdapat eror 0,001% .
4. Jika error terlalu besar maka akan berpengaruh pada gerakan robot .
5. Sistem dapat berjalan baik dan robot mampu melakukan gerakan tari sesuai yang diinginkan
6. Kelebihan menggunakan metode interpolasi linier adalah lebih mempermudah user dalam pemrograman karena hanya memberi masukan berapa sudut yang diinginkan.
7. Kekurangan dalam paper ini adalah masih belum adanya sensor pendukung untuk keseimbangan robot dalam melakukan gerakan tari.

V. DAF

VI. TAR PUSTAKA

- [1] <http://kontesrobotindonesia.id>. Panduan KRSTI
- [2] <http://support.robotis.com>. Specification dynamixel ax-12
- [3] S.Setia Budi.2010.Metode Numerik Dengan Scilab.
- [5] <http://jadipaham.com>. Gambar ilustrasi interpolasi linier
- [6] Ariyus, Doni & K.R, Rum Andri. 2008. Komunikasi Data. Yogyakarta: ANDI.
- [7] Ltd, Hitex (UK). 2009. The Insider's Guide To The STM32 ARM Based Microcontroller: An Engineer's Introduction To The STM32 Series Version 1.8. United Kingdom: University Of Warwick Science Park Coventry.

Rancang Bangun Sistem *Omnidirectional Vision* Cermin Spherical pada Robot KRSBI Beroda

Muliady¹, Hadiananto

^aProgram Studi Teknik Elektro Universitas Kristen Maranatha,
Jl. Prof. Drg. Surya Sumantri, M.P.H. No.65, Bandung.

¹Email korespondensi: mld_ang@yahoo.com

Abstrak -- Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda merupakan salah satu divisi dari Kontes Robot Indonesia dengan robot *autonomous* yang mampu bermain bola di lapangan seperti dalam olah raga sepak bola. Salah satu masalah yang dihadapi adalah jika robot hanya menggunakan kamera depan, maka tidak dapat mendeteksi bola yang berada di belakang robot. Ada beberapa cara untuk mengatasi masalah ini, salah satunya menggunakan *omnidirectional vision*. Pada penelitian ini, sistem *omnidirectional vision* dibangun dengan menggunakan webcam Logitech c930e dan sebuah cermin *spherical*. Dengan dilakukan eksperimen untuk menentukan jarak cermin dengan kamera dan jari jari cermin *spherical* terbaik dari yang ada. Serta dicari hubungan pixel terdeteksi frame webcam terhadap posisi dan jarak bola dengan kajian empiris. Proses pengolahan citra pada *omnidirectional vision*, digunakan metoda *thresholding*, *morphological transformation*, *findContours*, dan *minEnclosingCircle* untuk mengenal dan mendeteksi bola. Sistem *Omnidirectional vision* yang dibangun mampu mendeteksi bola pada ruang dengan intensitas cahaya 55lux sampai 191lux dengan radius 0cm (jika tidak terhalang konstruksi robot) sampai terdekat 347cm saat bola berada di belakang robot, terjauh 452cm saat bola berada di depan robot.

Kata kunci: cermin *spherical*, *thresholding*, *morphological transformation*, *minEnclosingCircle*, *findContours*

I. PENDAHULUAN

Kebutuhan kemampuan robot yang cerdas untuk dapat melakukan tugas di lingkungan yang sama dengan manusia, dapat dilihat dari perkembangan robot yang digunakan dalam penelitian ataupun dalam perlombaan. Salah satu perlombaan adalah Kontes Robot Indonesia (KRI) yang merupakan kegiatan kontes bidang robotika yang diadakan oleh Kementerian Riset, Teknologi, dan Pendidikan Tinggi dan dapat diikuti oleh mahasiswa dari institusi maupun Perguruan Tinggi Negeri dan Swasta.

Salah satu kategori dari KRI ini adalah Kontes Robot Sepak Bola Indonesia Beroda (KRSBI Beroda). Tujuan dari KRSBI Beroda adalah robot dapat bermain bola dan memasukkan bola ke gawang lawan secara *autonomous* di dalam lapangan berukuran 9m x 6m. Sehingga robot perlu dilengkapi dengan berbagai sensor, terutama kamera agar dapat mendeteksi dan bermain sepak bola. Kamera umunya dipasang di

depan robot untuk mendeteksi bola, gawang, robot lainnya, bahkan lapangan.

Masalah yang dihadapi adalah kamera tidak dapat mendeteksi bola yang berada di belakang robot sehingga robot harus bergerak ketika mencari bola dengan algoritma tertentu, yang tidak menjamin keberhasilannya^[1]. Salah satu solusinya dengan digunakan *omnidirectional vision* untuk mendeteksi dan mengetahui posisi bola di lapangan.

Kelebihan dari sistem ini adalah robot dapat melihat lapangan sekitarnya tanpa membutuhkan komponen tambahan seperti servo atau motor, sehingga robot tidak perlu bergerak dalam mencari bola. Dengan menggunakan sistem ini, robot menjadi lebih efektif dalam mencari bola saat bermain sepak bola, namun dalam implementasinya sensor kamera *omnidirectional* harganya mahal. Pada penelitian ini dibahas perancangan dan realisasi sistem *omnidirectional vision* sederhana untuk robot KRSBI Beroda.

II. PERANCANGAN DAN REALISASI

Omnidirectional vision secara sederhana dapat didefinisikan dengan kemampuan melihat dengan sudut pandang 360° pada bidang horisontal. *Omnidirectional vision* sangat penting untuk kebutuhan melihat area yang cukup luas seperti pada *panoramic photography* dan robotika^[2]. Pada normalnya, sebuah kamera memiliki ruang pandang dari beberapa derajat sampai maksimal 180°. Salah satu cara yang paling efektif dalam membuat *omnidirectional vision* adalah dengan mengkombinasikan cermin dan kamera yang menghasilkan sebuah sistem yang tidak membutuhkan pergerakan kamera untuk mendeteksi objek^[3]. Dengan menggunakan cermin *spherical/conical* yang diletakkan di depan kamera, cermin dapat memantulkan 360° lingkungan di sekitar kamera. Sistem *omnidirectional vision* ini digunakan untuk mendeteksi objek seperti bola, gawang, maupun robot-robot lain yang berada di sekitarnya dengan menggunakan berbagai algoritma pengolahan citra digital. Keuntungan menggunakan *omnidirectional vision* pada robot sepak bola beroda ini adalah robot dapat mendeteksi objek meskipun objek berada di belakang robot, sehingga robot tidak harus bergerak ataupun berputar ketika mencari objek.

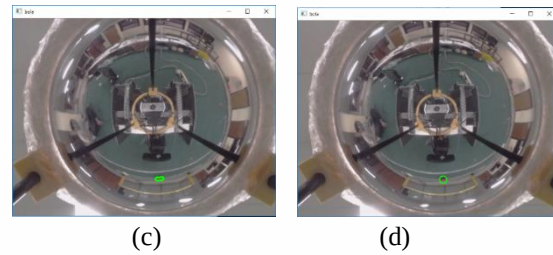
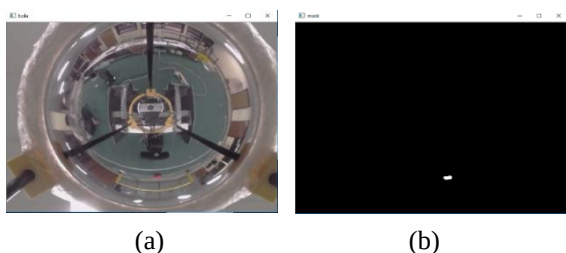
A. Sistem Omnidirectional Vision

Kondisi penerangan cahaya pada lapangan yang dipakai adalah 100-200 lux SNI. Tinggi struktur robot tanpa sistem *omnidirectional vision* adalah 50cm, dan tinggi maksimum robot 80cm. Sehingga perancangan sistem *omnidirectional vision* mengacu pada kondisi intensitas cahaya ini, dengan jarak maksimum cermin spherical dengan webcam adalah 30cm. Sistem yang dibangun terdiri atas 3 bagian, yaitu cermin *spherical*, kamera webcam, dan *single board computer* yang berisi algoritma pengolahan citra.

Citra didapat dari webcam Logitech C930e, lalu diproses oleh *single board computer* LattePanda dengan prosesor Intel Atom Quad Core 1.8GHz RAM4GB DDRL3L, 64GB EMMC, OS Windows 10. Untuk algoritma pengolahan citra digunakan bahasa Python V3.6.5.

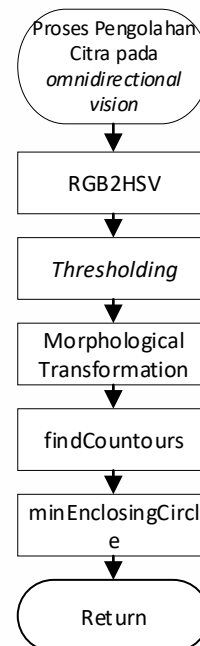
B. Algoritma Pengolahan Citra

Sistem pengolahan citra ini bertujuan agar *omnidirectional vision* dapat mendeteksi bola. Citra yang diterima oleh LattePanda melalui webcam Logitech c930e memiliki model warna RGB, diubah menjadi model warna HSV dengan fungsi *cvtColor* karena model warna HSV tidak rentan terhadap perubahan intensitas penerangan dibandingkan model warna RGB. Setelah itu, citra di-*threshold* dengan menggunakan fungsi *inRange* sehingga warna dapat di-*filter* berdasarkan batas-batas nilai HSV yang diinginkan. Pada penelitian ini, nilai HSV ditentukan agar dapat mendeteksi warna oranye bola. Nilai ini sedikit bervariasi untuk kondisi intensitas penerangan yang berbeda. Citra hasil *threshold* biasanya memiliki *noise* dari objek yang memiliki nilai HSV hampir sama dengan nilai HSV bola. Untuk itu, transformasi *morphology* digunakan untuk menghilangkan *noise* tersebut. Fungsi *morphology* yang dipakai adalah *erode* dan *dilation*. *Erode* digunakan untuk menghilangkan *noise* dan *dilation* digunakan untuk memperkaya pixel dari gambar setelah menggunakan fungsi *erode*. Selanjutnya menggunakan fungsi *findContours* untuk menghubungkan titik-titik yang saling berhubungan (pada batas), sehingga terbentuk sebuah bidang. Fungsi *minEnclosingCircle* digunakan untuk mencari bentuk yang hampir mirip dengan lingkaran dari bidang yang sudah didapat dengan fungsi *findContours*. Hasil proses pengolahan citra dapat dilihat pada Gambar 1.



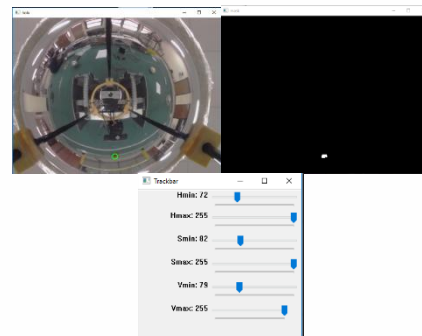
Gambar 1. Proses pengolahan citra, original (a), setelah *thresholding* (b), setelah mendeteksi *contours* (c), dan hasil akhir (d)

Gambar 2 memperlihatkan urutan proses pengolahan citra yang dilakukan dalam bentuk diagram alir.



Gambar 2. Diagram alir sistem pengolahan citra pada *omnidirectional vision*

Dengan kondisi intensitas cahaya ruang pada 150 lux, nilai *thresholding* yang diperoleh nilai H_{min} 72, H_{max} 255, S_{min} 82, S_{max} 255, V_{min} 79, dan V_{max} 255. Gambar 3 menunjukkan hasil *thresholding* yang dilakukan.



Gambar 3. Nilai *thresholding* untuk intensitas cahaya ruang 150 lux

C. Konstruksi Sistem Omnidirectional Vision

Salah satu faktor yang mempengaruhi sistem pengolahan citra pada *omnidirectional vision* adalah ukuran diameter cermin spherical yang digunakan. Cermin *spherical* diperoleh dari potongan bola hiasan berbahan *stainless*. Gambar 4 menunjukkan perbandingan bola *stainless* yang terdapat di pasaran, yaitu diameter 13,8cm, 14,32cm, dan 17,2 cm.



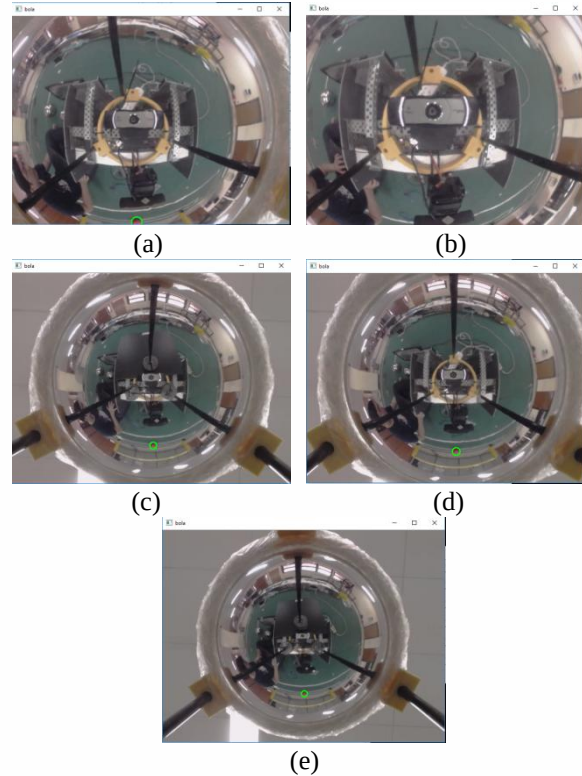
Gambar 4. Tiga ukuran cermin yang dicoba (dari kiri 13,8cm, 14,32cm, 17,2cm)

Webcam pada posisi teratas dari robot seperti pada Gambar 5, kemudian diinginkan webcam mampu menangkap citra seluruh cermin *spherical*. Dari ketiga ukuran diameter diperoleh yang terbaik adalah bola *stainless* dengan diameter 14,32cm. Karena pada citra yang ditangkap oleh webcam dengan menggunakan bola *stainless* diameter 14,32cm memiliki pantulan objek lebih besar dan jangkauan pandang lebih luas daripada dengan menggunakan bola *stainless* diameter 13,8cm. Sedangkan pada bola *stainless* diameter 17,2cm terjadi jarak cermin dengan webcam 32cm sehingga akan melebihi batas maksimum dimensi robot.



Gambar 5. Posisi webcam dengan cermin *spherical*

Setelah ditentukan ukuran diameter cermin *spherical*, berikutnya adalah menentukan jarak cermin terhadap webcam. Gambar 6 menunjukkan hasil percobaan citra yang ditangkap webcam dengan jarak cermin 20cm, 22,5cm, 25cm, 27,5cm, dan 30cm.



Gambar 6. Citra dan deteksi bola saat jarak webcam dengan cermin 20cm (a), 22,5cm (b), 25cm (c), 27,5cm (d), dan 30cm (e)

Berdasarkan data citra pada Gambar 6(a), jangkauan pandang maksimum hanya 75cm dari robot. Gambar 6(b) jangkauan pandang yang lebih luas tetapi masih ada bagian cermin yang tidak terlihat. Gambar 6(c) seluruh bagian cermin dapat terlihat. Sedangkan Gambar 6(d) dan Gambar 6(e) citra bola yang tertangkap webcam lebih kecil dibanding Gambar 6(c) dan banyak daerah yang bukan cermin. Dari percobaan ini, jarak antara cermin dan kamera dipilih sejauh 25 cm, karena memiliki jangkauan pandang mencakupi seluruh lapangan (9m x 6m) dan semua cermin tepat ditangkap oleh webcam.

D. Persamaan Relasi Antara Pixel pada Frame Webcam dengan Jarak Bola

Saat robot menggunakan *omnidirectional vision* ini, ternyata posisi bola sebenarnya dengan yang terbaca pada pixel frame webcam tidak berbanding lurus. Sehingga perlu dicari persamaannya. Untuk mendapatkan relasi tersebut, titik pusat dari frame webcam harus diketahui terlebih dahulu. Titik pusat frame webcam dapat ditentukan dengan menempatkan bola sejauh 50cm dari robot pada posisi depan, belakang, kanan, dan kiri robot, serta membaca pixel pada webcam (x,y). Masing-masing posisi dilakukan 5 kali pengambilan data yang dirangkum pada Tabel 1.

Tabel 1. Pixel posisi bola pada frame webcam (x,y)

Depan	Belakang	Kanan	Kiri
(338, 337)	(306, 98)	(456, 223)	(216, 233)
(335, 338)	(306, 99)	(457, 220)	(216, 234)
(337, 339)	(304, 98)	(456, 224)	(217, 246)
(338, 341)	(304, 99)	(457, 220)	(216, 235)
(336, 340)	(306, 97)	(456, 223)	(216, 233)

Berdasarkan data yang didapat pada Tabel 1, titik pusat cermin *spherical* pada frame webcam dapat ditentukan dengan Persamaan 1 dan 2.

$$\text{TitikPusat}_{(x)} = \frac{\bar{x}_{kanan} + \bar{x}_{kiri}}{2} \quad (1)$$

$$\text{TitikPusat}_{(y)} = \frac{\bar{y}_{depan} + \bar{y}_{belakang}}{2} \quad (2)$$

dengan parameter :

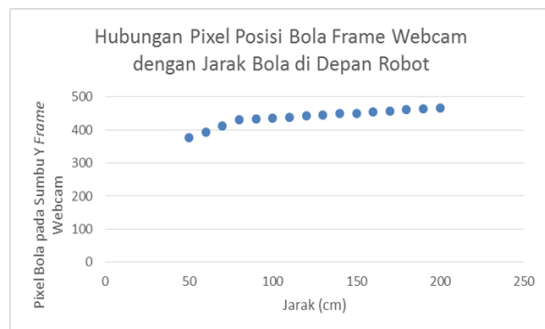
$\bar{x}_{kanan}$: rerata pixel x ketika bola di kanan robot

$\bar{x}_{kiri}$: rerata pixel x ketika bola di kiri robot

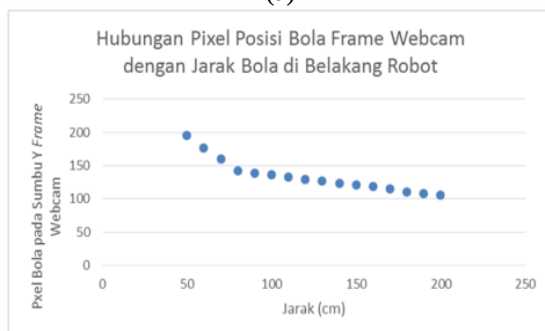
$\bar{y}_{depan}$: rerata pixel y ketika bola di depan robot

$\bar{y}_{belakang}$: rerata pixel y ketika bola di belakang robot

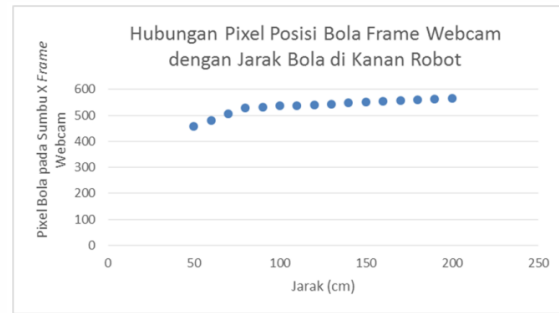
Titik pusat diperoleh pada pixel (285,336), dengan pembulatan terdekat. Setelah menentukan titik pusat dari frame webcam, dapat dicari posisi dan jarak bola dari robot dengan kajian empiris untuk menentukan persamaannya. Hubungan antara jarak dan pixel frame pada webcam sumbu Y dapat dilihat pada grafik Gambar 7 dan sumbu X pada grafik Gambar 8.



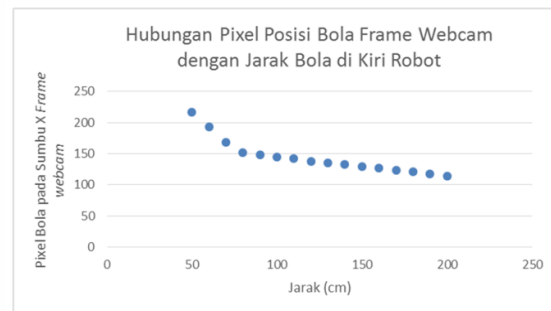
(a)



(b)

Gambar 7. Hubungan Jarak dan Pixel Sumbu Y Bola di Depan Robot (a), Bola di Belakang Robot (b)


(a)



(b)

Gambar 8. Hubungan Jarak dan Pixel Sumbu X Bola di Kanan Robot (a), Bola di Kiri Robot (b)

Pada Grafik Gambar 7 dan Gambar 8, terlihat bahwa terdapat diskontinu pada saat jarak 80cm untuk semua keadaan. Sedangkan saat jarak lebih kecil atau lebih besar daripada 80cm dapat didekati dengan persamaan garis lurus.

1. Untuk jarak bola < 80cm

Posisi bola di kiri atau kanan robot:

$$r_x = \frac{x-336}{2.402}$$

Posisi bola di depan atau belakang robot:

$$r_y = \frac{y-285}{1.8}$$

2. Untuk jarak bola ≥ 80cm

Posisi bola di depan robot:

$$r_y = \frac{10(y-429)}{3} + 80$$

Posisi bola di belakang robot:

$$r_y = -\frac{10(y-429)}{3} + 80$$

Posisi bola di kanan robot:

$$r_x = \frac{10(x-528)}{3} + 80$$

Posisi bola di kiri robot:

$$r_x = -\frac{10(x-528)}{3} + 80$$

dengan parameter :

r_x = Jarak bola pada lapangan terhadap sumbu x

r_y = Jarak bola pada lapangan terhadap sumbu y

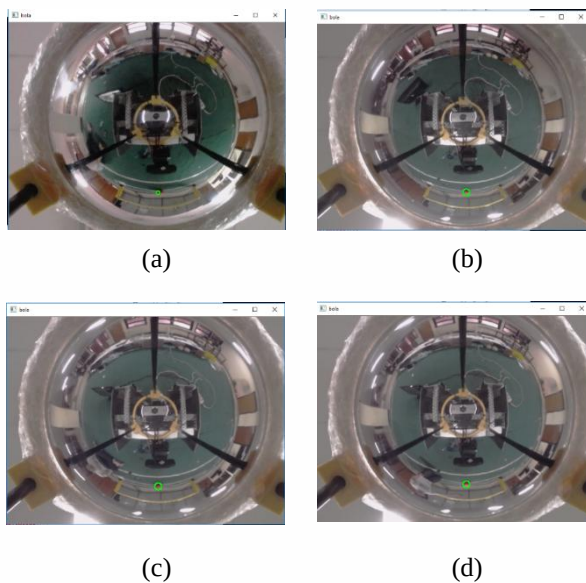
x,y = posisi bola pada frame kamera

III. UJI COBA DAN ANALISIS

Uji coba yang dilakukan pada sistem dibagi menjadi dua bagian, yaitu uji coba pengaruh intensitas cahaya ruang dan posisi bola.

A. Ujicoba Pengaruh Intensitas Cahaya Ruang

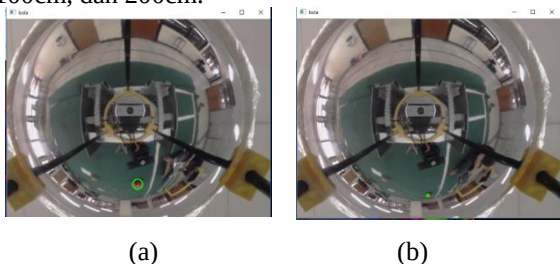
Pada ujicoba ini dilakukan dengan bola berjarak 150cm di depan robot, kondisi intensitas pencahayaan ruang dari 5 lux sampai 191 lux (sesuai kemampuan lampu penerangan yang ada). Bola mulai terdeteksi saat intensitas cahaya 23 lux, walaupun ada bagian bola yang tidak terdeteksi dengan baik, karena kontur yang diperoleh tidak berbentuk lingkaran. Pada saat intensitas cahaya 55 lux sampai 191 lux, bola terdeteksi dengan sempurna. Gambar 9 memperlihatkan hasil deteksi sistem *omnidirectional vision* pada beberapa kondisi penerangan.



Gambar 9. Hasil Deteksi Bola dengan Intensitas Penerangan Ruang 23lux (a), 55lux (b), 151lux (c), dan 191 lux (d)

B. Ujicoba Deteksi Posisi dan Jarak Bola

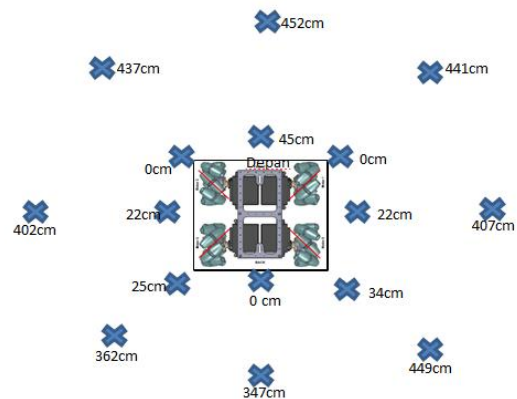
Percobaan ini dilakukan untuk dapat mengetahui pengaruh posisi bola terhadap pendeteksian bola agar dapat mengetahui jarak terdekat dan jarak terjauh bola yang dapat dideteksi. Hasil ujicoba ditampilkan pada Gambar 10 saat posisi bola di depan robot pada jarak 100cm, dan 200cm.



Gambar 10. Hasil Deteksi Bola dengan Jarak 100cm (a), dan 200cm (b)

Untuk mengetahui performansi sistem *omnidirectional vision* maka dilakukan pengujian

yang sama untuk posisi bola pada delapan penjuru angin seperti yang ditunjukkan pada Gambar 11. Uji coba deteksi bola dari jarak 0 (mengenai robot) sampai tidak terdeteksi, untuk delapan posisi bola.



Gambar 11. Performansi *Omnidirectional Vision*

Pada data pengamatan Gambar 11 dapat dilihat jarak maksimum dan minimum bola yang dapat dideteksi oleh *omnidirectional vision*. Jarak minimum bola terdeteksi ketika di depan robot karena terhalang oleh kamera depan, sedangkan pada sisi lain terhalang oleh badan robot. Bola dapat terdeteksi tepat di samping robot pada posisi kanan, kiri, depan, dan belakang robot.

IV. KESIMPULAN

1. Sistem *omnidirectional vision* yang dibangun untuk robot KRSBI Beroda berhasil mendeteksi bola di lapangan dengan radius terjauh antara 347cm sampai 452cm, tergantung pada posisi bola terhadap robot.
2. Algoritma pengolahan citra dengan fungsi *thresholding*, *morphological transformation*, *findContours*, dan *minEnclosingCircle* berhasil digunakan pada *omnidirectional vision* untuk mendeteksi bola dengan sudut pandang 360° sehingga mampu mendeteksi bola di belakang robot.
3. Intensitas cahaya masih berpengaruh pada proses pengolahan citra untuk mendeteksi bola. Pada sistem yang dibuat mampu mendeteksi bola pada penerangan 55lux sampai 191lux.
4. Posisi dan jarak bola yang terdeteksi memiliki hubungan yang khusus terhadap pixel frame webcam, diskontinu pada jarak 80cm untuk posisi bola di depan, belakang, kiri, dan kanan robot.

V. DAFTAR PUSTAKA

- [1] Muliady dan G. Arisandy, "Implementasi Sistem Gerak *Holonomic* pada Robot KRSBI Beroda 2017", Jurnal Teknik dan Ilmu Komputer UKRIDA, Vol.7 No.25, Jan-Mar 2018
- [2] T.Parry, "Extensive Guide to 360 Cameras", 2016
- [3] V. G. Junior and J. O. Junior, "Development of An Omnidirectional Vision System". Journal of the Brazilian Society of Mechanical Sciences and Engineering. 2006
- [4] D.J. Bora, A.K. Gupta, and F.A. Khan. "Comparing the Performance of L*A*B and HSV Color Spaces with Respect to Color Image Segmentation", IJETAE Vol.5, 2015
- [5] K.Shreedhar and B.Panlal. "Enhancement of Images Using Morphological Transformations,. IJCSIT Vol 4,No 1, 2012
- [6] OpenCV.findContours.https://docs.opencv.org/3.1.0/d4/d73/tutorial_py_contours_begin.html
- [7] Hadiano, "Implementasi *Omnidirectional Vision* untuk Mendeteksi Objek pada Robot Sepak Bola Beroda", Laporan Tugas Akhir Prodi Teknik Elektro, Universitas Kristen Maranatha, 2018



UMY

UNIVERSITAS
MUHAMMADIYAH
YOGYAKARTA

Unggul & Islami



KONTES ROBOT INDONESIA

10 - 13 JULI 2018

UNIVERSITAS MUHAMMADIYAH YOGYAKARTA

ISBN 978-602-5450-43-3



9 786025 450433