IMPLEMENTASI PERGERAKAN ROBOT PENJAGA GAWANG KRSBI (BERODA) DENGAN METODE *FUZZY PID*

SKRIPSI

Oleh

Cahya Mulya Adi

NIM 161910201008

PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2020

IMPLEMENTASI PERGERAKAN ROBOT PENJAGA GAWANG KRSBI (BERODA) DENGAN METODE *FUZZY PID*

SKRIPSI

diajukan guna melengkapi skripsi dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Strata 1 Jurusan Teknik Elektro
dan mencapai gelar sarjana

Oleh

Cahya Mulya Adi

NIM 161910201008

PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2020

PERSEMBAHAN

Alhamdulillah, puji syukur kehadirat Allah SWT atas nikmat dan karuniaNya yang telah melimpahkan rahmat yang tak ternilai sehingga saya bisa menyelesaikan penelitian ini. Banyak doa, semangat dan bantuan yang penulis dapatkan dari berbagai pihak dalam melangsungkan penelitian. Akhirnya, skripsi ini dipersembahkan untuk:

1. Allah SWT, yang Maha Pengasih lagi Maha Penyayang;
2. Nabi besar Muhammad SAW, yang menjadi suri tauladan bagi seluruh umat;
3. Kedua orang tua yang amat saya sayangi, yakni Bapak Buali dan Bunda Titik Hidayatiningsih yang selalu mendidik dan membimbing, serta mendoakan saya. Serta kakak saya Anang Setya Aji yang selalu menghibur dan memberikan semangat;
4. Bapak Sumardi, S.T., M.T selaku Dosen Pembimbing Utama dan Bapak Wahyu Muldayani, S.T., M.T selaku Dosen Pembibing Anggota yang telah membimbing dengan sangat sabar dan memberikan saran-saran demi kesempurnaan skripsi ini;
5. Dulur INDUKTRO 16 yang memberikan cerita indah selama masa perkuliahan;
6. Ex-pengurus Robotika dan anggota UKM robotika, seluruh asisten Laboratorium Elektronika Terapan dan teman-teman Elektronika 2016. Terima kasih Om wawan Oi Cb, Mas Turasno telah memberikan saran-saran dan solusi dalam diskusi.
7. Squad penghuni LPM Tigo, Syaiful, Zaki, Fikri, Rosida dan Tika yang selalu memberi semangat dan terima kasih banyak sudah menjadi tempat belajar dan keluh kesah;
8. Para guru sejak Taman Kanak-kanak hingga Perguruan Tinggi yang terhormat, terima kasih telah memberikan ilmu;

9. Serta rekan-rekan yang penulis kenal dan rekan-rekan yang membaca serta menjadikan skripsi ini sebagai referensi. Terima kasih untuk segalanya.

MOTTO

"Barangsiapa yang menumpuh jalan untuk menuntut ilmu niscaya allah akan memudahkan jalan baginya menuju surga"

(Imam Asy-Syafi'i)

"Hakikat hidup bukanlah apa yang kita ketahui, melainkan apa yang kita kerjakan yang paling mengakar dihati, jiwa dan initi kehidupan."

(Emha Ainun Najib)

PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Cahya Mulya Adi

NIM : 161910201008

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "IMPLEMENTASI PERGERAKAN ROBOT PENJAGA GAWANG KRSBI (BERODA) DENGAN METODE *FUZZY PID* adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada industry manapun dan bukan karya jiplakan. Saya bertanggung jawab penuh atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapatkan sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 2 juni 2020

Yang menyatakan,

Cahya Mulya Adi

NIM 161910201008

SKRIPSI

IMPLEMENTASI PERGERAKAN ROBOT PENJAGA GAWANG KRSBI (BERODA) DENGAN METODE *FUZZY PID*

Oleh

Cahya Mulya Adi

NIM 161910201008

Pembimbing:

Dosen Pembimbing Utama : Sumardi, S.T., M.T.

Dosen Pembimbing Anggota : Wahyu Muldayani, S.T., M.T.

PENGESAHAN

Skripsi berjudul "IMPLEMENTASI PERGERAKAN ROBOT PENJAGA GAWANG KRSBI (BERODA) DENGAN METODE *FUZZY PID*" karya Cahya Mulya Adi telah diuji dan disahkan pada:

Hari : Kamis

Tanggal : 02 Juli 2020

Tempat : Fakultas Teknik Universitas Jember

Tim penguji,

Ketua,	Anggota I,
Ir. Sumardi, S.T., M.T.	Wahyu Muldayani, S.T., M.T.
NIP 196701131998021001	NRP 760016799
Anggota II,	Anggota III,
Dr. Ir. Satryo Budi Utomo, S.T., M.T.	Ali Rizal Chaidir, S.T., M.T.
NIP 98501262008011002	NRP 760015754

Mengesahkan

Dekan,

Dr. Triwahju Hardianto, S.T., M.T.

NIP 197008261997021001

RINGKASAN

IMPLEMENTASI PERGERAKAN ROBOT PENJAGA GAWANG KRSBI (BERODA) DENGAN METODE FUZZY PID; Cahya Mulya Adi; 161910201008; 2020; 65 Halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Kontes Robot Sepakbola Indonesia merupakan salah satu kegiatan bagian dari Kontes Robot Indonesia (KRI). Yang diselenggarakan secara rutin tiap tahunnya oleh Kementrian Riset, Teknologi dan Pendidikan Tinggi. Sebagai ajang kompetisi rancang bangun dan rekayasa dalam bidang robotika. Saat ini hampir semua mengetahui tentang robot. Dunia ilmu dan teknologi bidang robotika mengalami perkembangan yang sangat pesat, banyak para ahli melakukan penelitian beragam inovasi robot agar dapat meringankan pekerjaan manusia. Pada penelitian ini memiliki tujuan bagaimana robot penjaga gawang dapat mengetahui gerakan bola yang menuju gawang. Kemudian bagaimana robot menentukan arah dengan sensor Cmps dan terakhir bagaimana proses pergerakan robot menggunakan Kontrol *Fuzzy* PID.

Penelitian ini dilakukan di Labolatorium Elektronika dan Terapan Jurusan Teknik Elektro Fakultas Teknik Universitas Jember dan sekretariat UKM Robotika Universitas Jember. Hal terpenting pada robot penjaga gawang ini yaitu robot dapat membuat kontrol kestabilan respon motor menggunakan kontrol *Fuzzy* PID sehingga robot penjaga gawang mampu menghalang bola yang melaju kearah gawang. Dan robot penjaga gawang mampu navigasi secara otomatis ketika mendapatkan pembacaan bola disisi samping robot dengan menentukan arah bola menggunakan sensor Cmps. Robot penjaga gawang menggunakan sensor kamera sebagai pembacaan bola yang kemudian diproses oleh komputer. Kemudian menggunakan sensor ultrasonic *ping* untuk mendeteksi dinding gawang dan sensor Cmps12 digunakan navigasi posisi robot. Dengan menggunakan *driver* motor BTS 7960 sebagai penggerak motor PG45.

Pengujian sensor dilakukan sesuai dengan perbandingan standart alat ukur yang sudah tersedia, pengujian keseluruhan dilakukan 3 kali penendangan di 5 titik yang berbeda dan didapatkan hasil akhir. Robot dapat mengetahui gerak bola dengan menggunakan kamera yang menggunakan nilai pixel sebagai acuan pembacaan posisi bola ditunjukan pada gambar 4.8 dimana semua posisi bola 100% terdeteksi oleh robot. Kemudian robot mampu bermanuver sesuai plan yang diinginkan dengan kontrol PID menggunakan tuning PID Ziegler Nichols 2 dengan respon yang baik ditunjukkan pada gambar 4.19 Dengan nilai Kp sebesar 2,4, nilai Ki sebesar 0,53, nilai Kd sebesar 2,7 dan dihasilkan nilai *rise time* (Ts) sebesar 3ms, dan *settling time* sebesar 11ms. Dan setelah dilakukan pengujian keseluruan robot menggunakan medote Fuzzy PID robot mampu menghadang bola dengan sempurna ditunjukan pada table 4.6 dengan presentasi 93,4%.

PRAKATA

Puji syukur kehadirat Allah SWT atas nikmat dan karuniaNya yang telah melimpahkan rahmat yang tak ternilai sehingga saya bisa menyelesaikan penelitian sekaligus penyusunan skripsi yang berjudul " **IMPLEMENTASI PERGERAKAN ROBOT PENJAGA GAWANG KRSBI (BERODA) DENGAN METODE *FUZZY PID*** ". Skripsi ini disusun untuk memenuhi salah satu syarat dalam penyelesaian pendidikan strata satu (S1) pada Jurusan Teknik Elektro Fakultas Teknik Universitas Jember. Selama penyusunan skripsi ini tentunya terdapat banyak pihak yang telah memberikan bantuan berupa saran, motivasi, fasilitas, doa dan dukungan lainnya. Oleh karena itu penulis menyampaikan terima kasih kepada:

1. Bapak Dr. Triwahju Hardianto, S.T., M.T. selaku Dekan Fakultas Teknik Universitas Jember;
2. Bapak Dr. Bambang Sri Kaloko S.T., M.T . selaku Ketua Jurusan Teknik Elektro Universitas Jember;
3. Bapak Ir. Sumardi, S.T., M.T selaku Dosen Pembimbing Utama dan bapak Wahyu Muldayani, S.T., M.T selaku Dosen Pembibing Anggota yang telah membimbing dengan sangat sabar dan memberikan saran-saran demi kesempurnaan skripsi ini;
4. Bapak Dr. Satryo Budi Utomo S.T., M.T. dan Bapak ali Rizal Chaidir, S.T., M.T. selaku dosen penguji yang telah memberikan saran untuk memperbaiki dan menyempurnakan skripsi ini;
5. Keluarga besar Teknik Elektro 2016 (INDUKTRO), terima kasih telah menjadi keluarga yang sangat baik dan luar biasa;
6. Serta seluruh teman-teman seperjuangan yang tidak dapat saya sebutkan satu-satu. Saya sampaikan terima kasih banyak atas semangat, doa dan saran yang telah diberikan.

Penulis menyadari bahwa selama penyusunan skripsi ini masih terdapat banyak kekurangan. Harapan penulis adalah semoga skripsi ini dapat bermanfaat dalam mengembangkan ilmu pengetahuan khususnya dalam bidang teknik elektro. Kritik dan saran diharapkan dapat terus berlanjut sehingga dapat memperbaiki dan menyempurnakan skripsi ini untuk penelitian selanjutnya.

Jember, 3 juni 2020

Penulis

Daftar Isi

Daftar Tabel

Daftar Gambar

BAB 1. PENDAHULUAN

1.1. Latar Belakang

Saat ini hampir semua mengetahui tentang robot. Dunia ilmu dan teknologi bidang robotika mengalami perkembangan yang sangat pesat, banyak para ahli melakukan penelitian beragam inovasi robot agar dapat meringankan pekerjaan manusia. Seperti halnya robot melakukan perpindahan suatu posisi dari satu posisi ke posisi lain dengan mengunakan suatu aktuator.

Kontes Robot Sepakbola Indonesia merupakan salah satu kegiatan bagian dari Kontes Robot Indonesia (KRI). Yang diselenggarakan secara rutin tiap tahunnya oleh Kementrian Riset, Teknologi dan Pendidikan Tinggi. Sebagai ajang kompetisi rancang bangun dan rekayasa dalam bidang robotika.

Pada permainan Robot Sepakbola beroda ini memiliki aturan tiga pemain yaitu dua pemain penyerang dan satu penjaga gawang. Dimana robot penjaga gawang berfungsi sebagai penjagaan gawang daerah sendiri. Peran penjaga gawang dalam permainan sepakbola ini sanagat vital karena merupakan pertahanan terakhir sebelum gol tercetak.oleh karena itu diperlukan penjagaan dan respon yang bagus untuk menghadang bola. Untuk mendapatkan respon yang bagus pada pergerakan robot penjaga gawang maka ini maka digunakan metode *fuzzy*PID sebagai kontrolnya. *Input* yang digunakan pada penelitian ini yaitu pembacaan nilai jarak oleh sensor jarak (ultrasonik) pada dinding gawang. Yang diproses menggunakan metode PID untuk mengontrol aktuator yang berupa motor.

Penelitian ini memberikan penjelasan mengenai sistem kerja dari robot penjaga gawang dari tim JR EVO dengan menggunakan metode *Fuzzy*PID. Mikrokontroler yang digunakan pada robot penjaga gawang adalah Arduino Mega 2560 dengan menggunakan Sensor berupa kamera, sensor kompas, sensor ultrasonic. 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.

Menurut penelitian yang dilakukan oleh Zulkifli, (2017), pada "Implementasi logika *Fuzzy* pada robot beroda penghalang". Dalam jurnalnya membahas tentang robot penghindar halangan ini digunakan untuk mengendalikan kecepatan robot berdasarkan jarak halangan dan sudut yang terbaca oleh sensor, sistem kerja dari robot beroda ini yaitu hanya menggunakan sudut kamera dan sensor ultrasonic Untuk menghalang benda yang ada pada suatu lapangan.

Menurut jurnal M Kantori (2017) *Mobile Robot* Pemain Bola". Dalam jurnalnya ini membahas tentang perancangan mekanik robotyang mempertahankan posisi robot agar tetap pada posisi stabilnya pada saat melewati lintasan yang tidak rata. Robot ini didesain menggunakan empat buah roda omni yang ditambah dengan sempring pada bagian kedudukan motor agar roda tidak kaku atau tetap pada saat ada lintasan yang tidak datar. Dari data pengujian menunjukan bahwa rata-rata kesalahan lateral kurang dari 1 cm dalam satu meter, sehingga robot bisa berjalan pada garis lurus akurat.

Pada penelitian ini, robot ini dirancang dengan menggunakan sensor cmps12, kamera dan sensor ultrasonic *Ping* Untuk penggerak robot kiper ini berupa motor dc pg45. Kelebihan pengguaan sensor Cmps12 ini digunakan saat bola sulit dibaca oleh kamera pada sudut sempit . Penelitian sebelumnya untuk pembacaan objek hanya menggunakan kamera sebagai sensor deteksinya.

1.2. Rumusan Masalah

Dari latar belakang yang telah diuraikan diatas dapat ditarik rumusan masalah dari penelitian antara lain :

1. Bagaimana robot penjaga gawang dapat mengetahui Gerakan bola yang menuju gawang.
2. Bagaimana Robot menentukan arah dengan sensor Cmps.
3. Bagaimana proses pergerakan robot menggunakan Kontrol *Fuzzy* PID.

1.3. Tujuan Penelitian

Tujuan dari penelitian ini antara lain :

1. Dapat membuat kontrol kestabilan respon motor menggunakan kontrol *Fuzzy*PID.
2. Robot penjaga gawang mampu menghalang bola yang melaju kearah gawang.
3. Robot penjaga gawang mampu navigasi secara otomatis untuk menentukan arah dengan sensor Cmps

1.4. Manfaat Penelitian

Manfaat dari penelitian ini yaitu :

1. Untuk meng-implementasikan pergerakan robot yang dapat melakukan proses pergerakan secara real time dengan respon yang bagus
2. Robot penjaga gawang agar mampu menghadang bola

1.5. Batasan Masalah

Untuk memfokuskan tujuan penelitian maka penulis memberikan batasan masalah pada rencana penelitian ini. Adapaun batasan masalahnya antara lain:

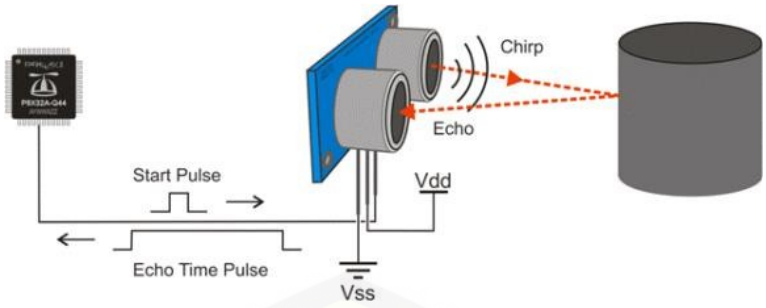
1. Robot beroda berjenis *differensial steering*
2. Menggunakan 4 buah sensor ultrasonic PING 2 di bagian belakang dan 1 bagian samping kanan kiri robot.
3. Sensor kamera untuk pembacaan objek berupa bola.
4. Sensor suhu cmps12 untuk mendeteksi sudut pada arah gawang.
5. Aktuator berupa Motor untuk pergerakan robot
6. Akuisisi data berbasis Arduino Mega 2560.
7. Robot hanya melakukan menghadang bola
8. Sistem kendali menggunakan control *Fuzzy*PID untuk mengatur respon motor pada robot
9. Tidak membahas pengolahan cintra.

BAB 2. TINJAUAN PUSTAKA

2.1. Sensor Ultrasonik (*Ping*)

Sensor *Ping* adalah sensor ultrasonik yang dapat membaca jarak benda dengan cara memantulkan gelombang ultrasonik dengan frekuensi 40 KHz dan sensor ini dapat mendeteksi pantulannya sendiri. Sensor ini dapat membaca jarak antara 3 cm hingga 300 cm. keluaran dari sensor *ping* ini berupa sinyal pulsa yang lebarnya menjelaskan jarak. Lebar pulsanya bervariasi antara 115 uS sampai 18,5 mS. Pada dasanya, *ping* terdiri dari sebuah chip pembangkit sinyal 40KHz, sebuah speaker ultrasonik dan sebuah mikropon ultrasonik. Speaker ultrasonik mengubah sinyal 40 KHz menjadi suara sementara mikropon ultrasonik berfungsi untuk mendeteksi pantulan suaranya.

Pin signal dapat digunakankan secara langsung dengan mikrokontroler tanpa menggunakan komponen tambahan apapun. *Ping* hanya akan mengirimkan data ketika ada pulsa trigger yang masuk dari mikrokontroler (Pulsa *high* selama 5uS). Suara ultrasonik dengan frekuensi sebesar 40KHz akan dipancarkan selama 200uS. Suara ini akan merambat di udara dengan kecepatan 344.424m/detik (atau 1cm setiap 29.034uS), untuk mengenai benda kemudian terpantul kembali ke *Ping* Selama menunggu pantulan, *Ping* akan menerima sebuah pulsa yang akan mengirim perintah berhenti (low) ketika suara pantulan terdeteksi kembali oleh *Ping* Oleh karena itulah lebar pulsa tersebut dapat menjelaskan jarak antara sensor *Ping* dengan benda. Untuk penjelasan sensor *Ping* hanya memakai 3 pin, yaitu pin trigger sama echo dijadikan dalam 1 pin, kemudian ada pin VCC dan Gnd sehingga dengan menggunakan sensor *Ping* dapat menghemat penggunaan I/O mikrokontroler. (Budiarso, 2015)

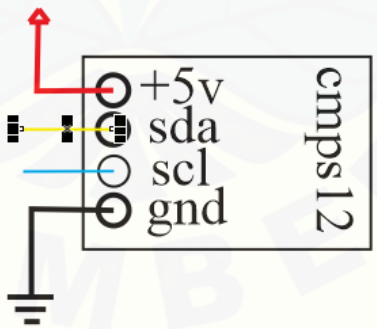


Gambar 2.1 Sensor Ultrasonik (*PING*)

Sumber : (Budiarso Z. , 2015)

2.2. Sensor CMPS12

CMPS12 adalah modul magnetik kompas, modul kompas ini didesain khusus dalam bidang robotik untuk tujuan navigasi robot. Kompas ini menggunakan dua sensor medan magnet KMZ51 buatan Philips yang cukup peka untuk mendeteksi medan magnet bumi. Dua sensor ini dipasang saling bersilangan. Pada modul kompas telah dipasang rangkaian pengkondisi sinyal dan mikrokontroler. Sehingga kita dapat mengakses berapa derajat posisi kompas secara langsung.(Musbikhin, 22 april 2014). CMPS12 merupakan kompas kompensasi generasi ke-4. Kompas ini menggunakan magnetometer 3 sumbu, 3 axis gyro dan accelerometer 3 axis. Isi dari sensor ini adalah menjalankan BNO055 yang luar biasa algoritma untuk menghapus kesalahan yang disebabkan oleh kemiringan medan.



Gambar 2.2 Sensor CMPS12

(Sumber : (Taufikurrahman, 2017)

Spesifikasi:

Catu Daya - 3.3v-5v 18mA Typ.

Resolusi - 0,1 Derajat

Akurasi - Lebih baik dari 1%. setelah kalibrasi

Level sinyal - 3.3v, 5v toleran

Mode I2C - hingga 400khz

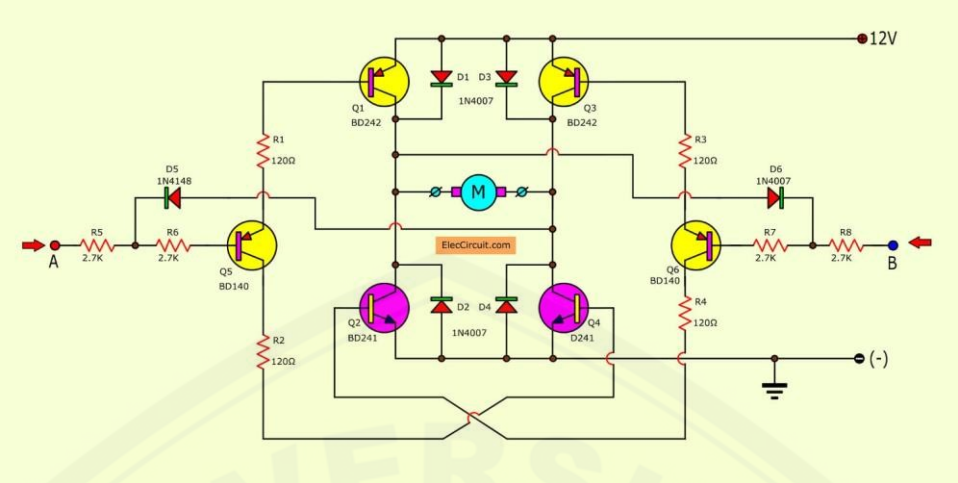
Mode serial - 9600, 19200, 38400 baud

2.3. Driver Motor BTS 7960

Motor DC merupakan komponen elektronika yang memerlukan sumber daya untuk menggerakkannya. Tidak semua motor dc dapat disambung langsung ke mikrokontrol ada yang memerlukan sumber daya dari luar untuk itu dibutuhkan *driver* motor yang dapat membantu mikrokontrol utnuk mengendalikan motor dc. Selain itu *driver* motor menghindari arus balik dari motor ketika motor berhenti yang dapat membuat mikrokontrol itu sendiri rusak. Maka dari itu *driver* motor dapat membatu mencegah rusaknya mikrokontrol dari arus balik pada motor. Beberapa jenis *driver* motor yang sering digunakan diantaranya adalah *H-Bridge* transistor, *H-Bridge* MOSFET, dan IC *driver motor*

Pada *driver motor* DC ini dapat mengeluarkan arus hingga 43A, dengan memiliki fungsi PWM. Tegangan sumber DC yang dapat diberikan antara 5.5V-27VDC, sedangkan tegangan *input* level antara 3.3V-5VDC, *driver* motor ini menggunakan rangkaian *full H-bridge* dengan IC BTS7960 dengan perlindungan saat terjadi panas dan arus berlebihan.

Driver motor juga dapat digunakan untuk mengontrol arah serta kecepata putaran pada motor. Salah satu yang sering digunakan adalah modul IC *driver* motor BTS 7960 dalam modul IC ini terdapat rangkaian *H-Bridge* yang berguna untuk mengontrol putaran motor sesuai data masukan digital. Pada BTS 7960 terdapt pin untuk mengatur PWM (*Pulse Width Modulator*) yang akan mengatur kecepatan motor dc. Konstruksi pin *driver motor* DC BTS 7960 adalah sebagai berikut.



Sumber : (Widiarto, 2018)

Spesifikasi

- Double BTS7960 *high current* (43A) *H-bridge drivers*
- *Input voltage* 6V-27V
- Model: IBT-2
- *Maximum current* 43A
- *Input* level: 3.3-5V
- *Control mode* PWM *or level*

Pin Input

1. RPWM = *Input PWM Forward Level Aktif High*
2. LPWM = *Input PWM Reverse Level Aktif High*
3. R_EN = *Input Enable Forward Drive, Aktif High*
4. L_EN = *Input Enable Reverse Drive, Aktif High*
5. R_IS = *Forward Drive ,Side current alarm output*
6. L_IS = *Reverse Drive ,Side current alarm output*
7. Vcc = +5 V *Power Supply* Mikrokontroler
8. Gnd = *Gnd Power Supply* Mikrokontroler

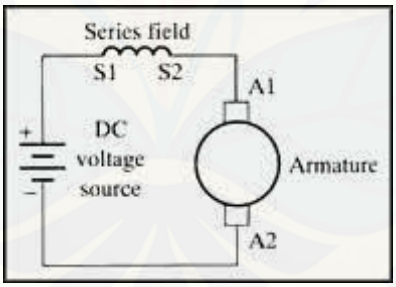
Pin Output

1. W- = Di hubungkan ke Motor DC (V-)
2. W+= Di hubungkan ke Motor DC (V+)
3. B+ = Tegangan *Input* V+ Motor
4. B- = Tegangan *Input* V- Motor

2.4. Motor DC

Motor DC adalah motor listrik yang bergerak dengan suplai tegangan arus searah yang akan dirubah menjadi energi gerak mekanik. Bagian – bagian pada motor dc diantaranya adalah *stator* (bagian yang tidak berputar) dan disebut *rotor* (bagian yang berputar). Motro dc menggunakan arus langsung yang tidak langsung/*direct-unidirectional*. Bagian utama dari motor dc adalah :

1) Kutub medan, motor dc memiliki dua yaitu kutub utara dan kutub selatan.
2) Dinamo dihubungkan dengan as yang berfungsi utnuk menggerakan beban.
3) *Commutator* berfungsi untuk transmisi arus antara dinamo dan sumber daya.



Gambar 2.4 Diagram Motor DC

(Sumber: brontoseno, 2014)

Beberapa keuntungan menggunakan motor dc adalah pengendali kecepatan, yang tidak mempengaruhi kualitas pasokan daya. Motor dc dapat dikendalikan dengan mengatur :

1) Tegangan dinamo dengan meningkatnya tegangan maka kecepatan meningkat.
2) Arus medan dengan menurunkan arus maka kecepatan meningkat.

(Elektronika Dasar, 2012)

2.5. Camera Logitech

Camera Logitech digital adalah alat untuk membuat gambar dari objek untuk selanjutnya dibiaskan melalui lensa pada sensor CCD dan akhir-akhir ini pada sensor BSI-CMOS (*Back Side Illuminated*) sensor yang lebih irit daya untuk kamera yang lebih canggih yang hasilnya kemudian direkam dalam format digital ke dalam media simpan digital. Kemudahan dari kamera digital adalah hasil gambar yang dengan cepat diketahui hasilnya secara instan dan kemudahan memindahkan hasil (*transfer*). Beberapa kamera digital, terutama DSLR dan high-end cameras dilengkapi fasilitas RAW yang dapat ditindaklanjuti di komputer mengunakan perangkat lunak tertentu untuk hasil terbaik

Gambar 2.5 *Camera Logitech*

Sumber: (Shadiq, 2014)

Tabel 2.1 Spesifikasi kamera logitech

Dimensi	29 mm x 94 mm x 24 mm
Berat	1,5 m
Resolusi maksimal	162 g
Jenis fokus	1920 x 1080 - 1280 x 720 pixels
Teknologi lensa	*Full HD Glass*
Mikrofon internal	Stereo
Kompatibel	Windows 10 atau versi terbaru,Windows 8, Windows 7

2.6. Roda Omni Wheel 127mm

Roda Omni atau roda poli adalah roda dengan cakram kecil di sekelilingnya yang tegak lurus dengan arah belok. Efeknya adalah roda dapat digerakkan dengan kekuatan penuh, tetapi juga akan meluncur ke samping dengan sangat mudah. Roda ini sering digunakan dalam sistem penggerak holonomis. Sebuah platform yang menggunakan tiga roda omni dalam konfigurasi segitiga umumnya disebut Kiwi Drive. Platform Killough serupa; dinamakan sesuai dengan karya *Stephen Killough* dengan platform omnidirectional di *Oak Ridge National Laboratory*. Desain Killough 1994 menggunakan sepasang roda yang dipasang di kurungan dengan sudut yang tepat satu sama lain dan dengan demikian mencapai gerakan holonomis tanpa menggunakan roda omni sejati. Kelebihan *omnidirectional* adalah dapat bergerak ke segala arah tanpa harus mengubah arah hadapnya. Arah gerak pada robot ini bergantung pada perbandingan kecepatan pada tiap roda yang didapat dari perhitungan kinematika robot.

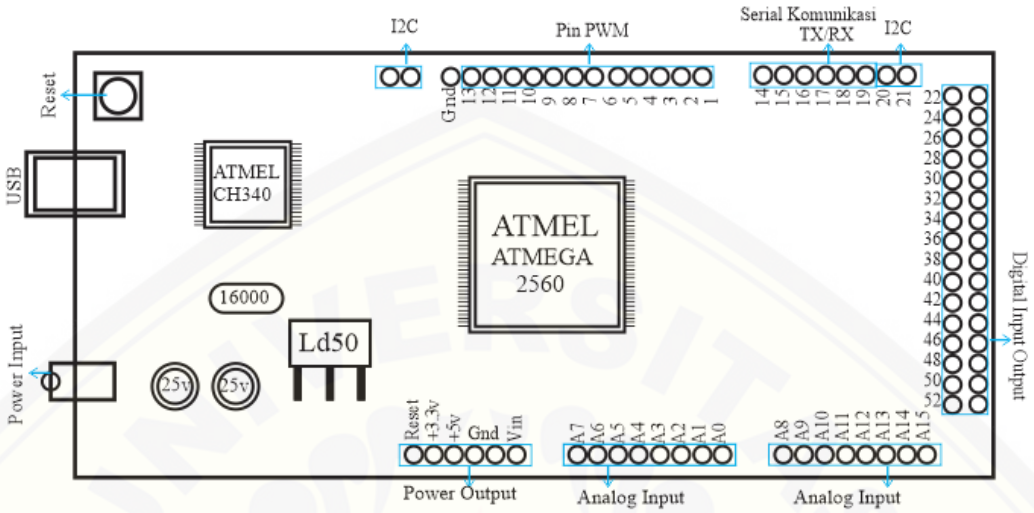
Gambar 2.6 Roda Omni 127mm

(Sumber: Wahyu Setyo Pambudi, 2011)

2.7. Arduino Mega

Arduino Mega 2560 merupakan jenis – jenis dari arduino, sedangkan arduino adalah sebuah mikrokontroler yang memiliki sifat *open source* sehingga memudahkan dalam memakainya. Arduino memiliki prosesor Atmel AVR dan memiliki bahasa pemrograman sendiri serta banyak bantuan *library* pada arduino.

Arduino mega 2560 sendiri memiliki chip ATmega 2560 serta banyak pin pada *board* arduino. Adapun spesifikasi dari arduino mega 2560 sendiri adalah :



Gambar 2.7 Bentuk Fisik Arduino Mega 2560

sumber : (JauhariArifin, 2016)

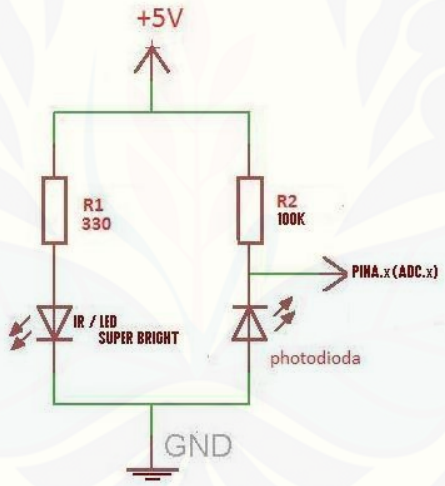
Tabel 2.2 spesifikasi arduino mega 2560

Chip mikrokontroller	**ATmega2560**
Tegangan operasi	5V
Tegangan *input* (yang direkomendasikan, via jack DC)	7V - 12V
Tegangan *input* (limit, via jack DC)	6V - 20V
Digital I/O pin	54 buah, 6 diantaranya menyediakan PWM *output*
Analog *Input* pin	16 buah
Arus DC per pin I/O	20 mA
Arus DC pin 3.3V	50 mA
Memori *Flash*	256 KB, 8 KB telah digunakan untuk *bootloader*
SRAM	8 KB
EEPROM	4 KB

Clock speed	16 Mhz
Dimensi	101.5 mm x 53.4 mm
Berat	37 g

2.8. Proximity sensor

Proximity sensor merupakan sensor yang digunakan untuk mendeteksi suatu obyek benda berdasarkan jarak benda tersebut terhadap sensor. *Proximity* sensor ini akan mendeteksi obyek benda dengan jarak yang cukup dekat berkisar 1 m sampai beberapa centimeter dari sensor. Sensor ini sering diimplementasikan pada industri pabrik, perkantoran, dunia robot, dan lain-lain. sensor *proximity* merupakan sensor yang mampu mendeteksi keberadaan suatu obyek logam maupun non logam tanpa menggunakan kontak fisik.



Gambar 2.8 Diagram *Proximity* sensor

Sumber: (Sigit, 2007)

2.9. Silinder Pneumatic

Pengalihan pneumatik di sini. Untuk urutan tertinggi manusia dalam Gnostisisme, lihat Pneumatik (Gnostisisme). Lokomotif pneumatik (udara terkompresi) seperti ini sering digunakan untuk mengangkut kereta api di tambang. Pneumatik (Dari Bahasa Yunani: πνεύμα pneuma) adalah cabang teknik yang memanfaatkan gas atau udara bertekanan. Sistem pneumatik yang digunakan dalam

industri umumnya didukung oleh udara terkompresi atau gas inert terkompresi. Kompresor yang terletak di pusat dan bertenaga listrik menggerakkan silinder, motor udara, dan perangkat pneumatik lainnya. Sistem pneumatik yang dikendalikan melalui katup solenoida manual atau otomatis dipilih ketika memberikan alternatif yang lebih murah, lebih fleksibel, atau lebih aman untuk motor listrik dan aktuator.

Gambar 2.9 *Silinder Pneumatic*

(Sumber: Muhammad Subhan, 2016)

2.10 Solenoid *valve*

Solenoid *valve* adalah elemen kontrol yang paling sering digunakan dalam *fluidics* Tugas dari solenoid *valve* adalah untuk mematikan, *release, dose, distribute atau mix fluids*. Solenoid *Valve* banyak sekali jenis dan macamnya tergantung type dan penggunaannya, namun berdasarkan modelnya solenoid *valve* dapat dibedakan menjadi dua bagian yaitu solenoid valve single coil dan solenoid *valve double coil* keduanya mempunyai cara kerja yang sama.

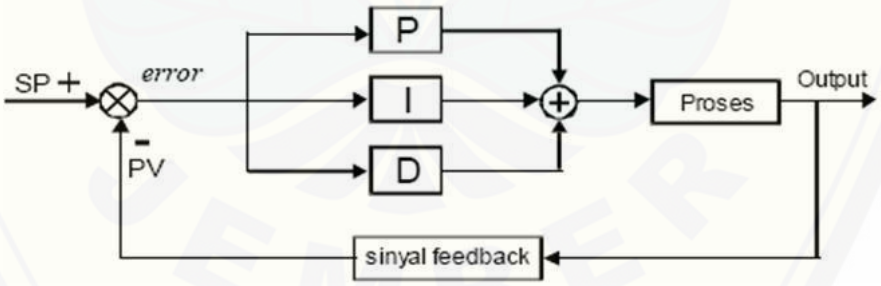
Solenoid *valve* banyak digunakan pada banyak aplikasi. Solenoid *valve* menawarkan *switching* cepat dan aman, keandalan yang tinggi, awet/masa service yang cukup lama, kompatibilitas media yang baik dari bahan yang digunakan, daya kontrol yang rendah dan desain yang kompak. Solenoid valve mempunyai banyak variasi dalam hal kegunaan atau kebutuhan dari mesin tersebut

Gambar 2.10 Solenoid *valve*

(Sumber : Hasan Mahmud, 2014)

2.11. Sistem Kendali PID (Proportional Integral Derivative)

Kontrol PID *(ProportionalIntegralDerivative)* adalah sebuah sistem kontrol yang digunakan untuk menambah tingkat kepresisian suatu sistem dengan menggunkan sistem umpan balik *(feedback)* yang terdapat dalam sistem tesebut. Kontrol PID terbentuk dari tiga buah kontrol yaitu kontrol P *(Proportional)*, D (*Derivative*) dan I (*Integral*). Dimana setiap kontrol memiliki karakteristik serta kelebihan dan kekurangan. Blok diagram sistem kendali PID ditunjukan pada Gambar 2.10



Gambar 2.11 Blok diagram pengendali PID

(Sumber : fahmizal, 2010)

Adapun persamaan sistem kendali PID adalah :

$$PID = Kp \,.\, e\,(t) + \frac{Kp}{Ti}\int_0^t e\,(t)\,dt + Kp.\,Td\,\frac{de(t)}{dt} \qquad (6.1)$$

Dengan :

PID = *output* dari kendali PID

Kp = konstanta *proporsional*

Ki = Kp/Ti

Ti = konstanta *integral*

Kd = Kp.Td

Td = konstanta *derivatif*

e(t) = *error*

Untuk mendapatkan respon yang baik dari kontrol PID parameter harus mengatur masing-masing parameter P, I dan D.

2.11.1. Cara Mentuning Manual Parameter PID

Ada beberapa cara mentuning parameter dari PID salah satunya yaitu cara dengan coba-coba (*tranning and error*) dengan cara parameter-parameter PID diubah- ubah dengan melihat tabel respon PID untuk menentukan respon yang diinginkan. Dengan menganalisa respon PID, maka nilai-nilai Kp, Ki, dan Kd bisa diubah-ubah sesuai dengan Tabel 10.1 Parameter PID ditunjukan pada Tabel 610.1

Tabel 2.3 Parameter PID

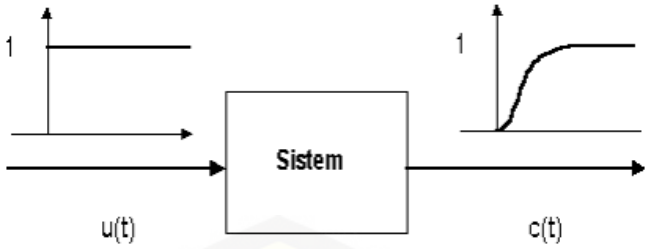
Parameter	***Rise Timer***	***Overshoot***	***Setting Time***	***S-S Error***
Kp	Berkurang	Bertambah	0	Berkurang
Ki	Berkurang	Bertambah	Bertambah	Menghilangkan
Kd	Minor *Change*	Berkurang	Berkurang	Minor *Change*

2.11.2. Metode *Ziegler-Nichols*

Kemudian ada juga dengan menggunakan metode *Ziegler-Nichols* metode ini adalah mencari parameter-parameter PID dengan rumus yang telah ditentukan. Metode ini memiliki 2 tipe, yaitu tipe 1 (*open loop*) dan tipe 2 (*closed loop*).

A. *Ziegler-Nichols tipe 1 (open loop)*

Ziegler-*Nichol open loop* Sistem diberi *input step* sehingga respon *open loop* terbentuk.



Gambar 2.12 Sistem Diberi *Input* Step

Sumber : (Setiawan, 2008)

Kemudian dari respon *open loop* diperoleh parameter-paramter ZN tipe 1 (L dan T). Proses desain menentukan parameter L dan T ditunjukan pada Gambar 2.11.2

Gambar 2.13 Proses Desain Penentuan Parameter L dan T

(Sumber : Bachri, 2004)

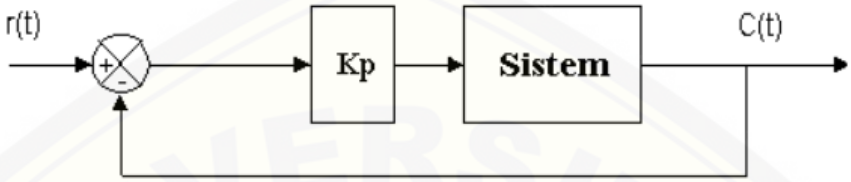
Kemudian ketika parameter L dan T diperoleh, nilai-nilai Kp, Ti, dan Td bisa dicari dengan menggunakan rumus PID untuk metode ZN tipe 1. Tabel 6.4 menunjukan tabel parameter PID untuk ZN tipe 1.

Type Of Controller	Kp	Ti	Td
P	T/L	∞	0
PI	0.9T/L	L/0.3	0
PID	1.2T/L	2L	0.5l

Tabel 2.4 Parameter PID untuk ZN tipe 1

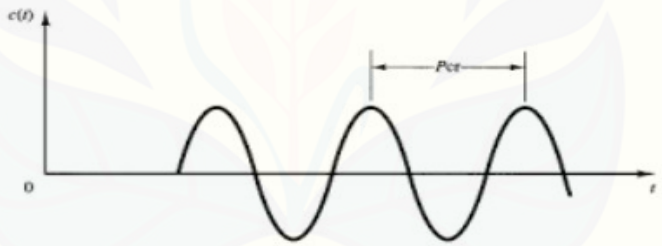
B. *Ziegler-Nichols tipe 2 (closed loop)*

Dalam metode ZN tipe 2, menggunakan sistem *closed loop* sebagai *feedback* respon. Pada metode ini yang digunakan Kp saja. Sistem dibuat hingga berosilasi terus menerus dengan mengatur besarnya nilai Kp.



Gambar 2.14 Sistem *closed loop* dengan Menggunakan Kp

Besarnya nilai Kp saat respon sistem berosilasi terus menerus merupakan nilai Kcr. Dari respon yang dihasilkan, parameter lain ZN tipe 2 selain Kcr, yaitu Pcr dapat dicari. Proses desain menentukan parameter.



Gambar 2.15 Proses Mendesain Menentukan Parameter Motor

(Sumber : Bachri, 2004)

Setelah mendapatka nilai parameter Kcr dan Pcr, nilai-nilai Kp, Ti, dan Td dapat dihitung dengan rumus mencari parameter PID untuk ZN tipe 2. Tabel 6.5 menunjukan tabel parameter PID untuk ZN tipe 2.(Fauziansyah, 2015)

Tabel 2.5 Parameter PID Untuk ZN Tipe 2

Type Of Controller	Kp	Ti	Td
P	0.5Kcr	∞	0
PI	0.45Kcr	(1/1.2)Pcr	0
PID	0.6Kcr	0.5Pcr	0.125Pcr

2.12 Logika *Fuzzy*

Fuzzy logic pertama kali dikembangkan pada tahun 1960-an oleh Prof. Lotfi A. Zadeh, seorang peneliti dari Universitas California. *Fuzzy* dalam Bahasa Inggris mempunyai arti samar atau tidak jelas. Sehingga, logika *fuzzy* mengandung unsur ketidakpastian yang berbeda dengan logika tegas (crisp). Pada logika tegas, hanya dikenal dua nilai yaitu salah atau benar (nol atau satu), yang dikenal dengan logika *Boolean*. Sedangkan *fuzzy logic* mengenal nilai antara benar dan salah, yang dinyatakan dalam derajat kebenaran yang nilainya antara nol sampai satu.

Himpunan *fuzzy* merupakan pengelompokan elemen-elemen dengan dua atribut berupa variabel bahasa (*linguistic variable*) yang mewakili suatu kondisi dengan bahasa alami seperti panas, dingin, cepat, lambat, dan sebagainya, serta variabel numerik (*numeric variable*) mewakili ukuran dari suatu variabel yang dinyatakan dalam angka. Himpunan *fuzzy* didasarkan pada gagasan untuk memperluas jangkauan fungsi karakteristik sehingga fungsi tersebut akan mencakup bilangan riil pada interval nol sampai satu [0,1], yang dinyatakan dengan fungsi keanggotaan dalam semesta pembicaraan (bilangan riil).

Fuzzy logic umumnya diterapkan pada masalah - masalah yang mengandung unsur ketidakpastian (*uncertainty*), ketidaktepatan (*imprecise*), *noisy* dan sebagainya. *Fuzzy logic* menjembatani bahasa mesin yang presisi dengan bahasa manusia yang menekankan pada makna atau arti (*significance*). *Fuzzy logic* dikembangkan berdasarkan cara berpikir manusia.

Sistem yang dapat melakukan penalaran dengan prinsip serupa seperti manusia melakukan penalaran dengan nalurinya. Terdapat beberapa jenis FIS berdasarkan metode yang membangunnya yaitu Mamdani, Sugeno dan Tsukamoto. Metode pada FIS merupakan cara penarikan kesimpulan dari sekumpulan kaidah *fuzzy*. Proses dalam kendali *fuzzy logic* input yang diberikan dan output yang dihasilkan berupa bilangan tegas tertentu. Melalui fuzifikasi, input nilai tegas (*non-fuzzy*) diproses menjadi nilai-nilai *fuzzy* sesuai dengan himpunan *fuzzy* terkait agar data dapat diolah oleh sistem *fuzzy* (Zulkifli, 2017)

BAB 3. Metodologi Penelitian

3.1. Tempat Penelitian

Penelitian ini dilakukan di Labolatorium Elektronika dan Terapan dan sekretariat UKM Robotika Teknik, Universitas Jember.

3.2. Waktu Penelitian

Waktu penelitian ini dimulai pada Oktober 2019 – April 2020, dengan rincian sebagai berikut :

Tabel 3.1 Rencana Kegiatan Tugas Akhir

No	Kegiatan	Bulan					
		1	2	3	4	5	6
1	Studi Literatur						
2	Perancangan alat						
3	Penelitian dan pengujian						
4	Pembahasan						
5	Kesimpulan						

Tahapan Penelitian

Langka-langkah yang akan digunakan dalam penelitian ini adalah sebagai berikut :

a. Studi Literatur. (mempelajari dan memahami teori-teori yang di peroleh dari beberapa buku, internet serta jurnal-jurnal dengan topik yang sama sehingga penulis dapat memahami konsep dan mendapatkan referensi dalam menyelesaikan masalah yang diteliti).
b. Melakukan perancangan perangkat keras. (melakukan perancangan robot)
c. Melakukan perancangan perangkat lunak.(melakukan perancangan komponen-komponen yang dibutuhkan pada robot penjaga gawang.)

d. Melakukan pengujian pengintegrasian perangkat keras dan perangkat lunak. Pertama pengujian ini dilakukan secara terpisah dan selanjutnya akan dilakukan pengujian secara keseluruhan.
e. Menganalisa data yang telah diperoleh saat pengujian.

3.3. Alat dan Bahan Penelitian

Alat dan bahan yang digunakan dalam penelitian ini adalah:

1. Arduino Due
2. Arduino Mega 2560
3. Motor DC PG45
4. *Driver* Motor BTS7960
5. Sensor Kamera *Logitech*
6. Sensor Ultrasonik *ping*
7. Sensor CMPS12
8. Catu Daya

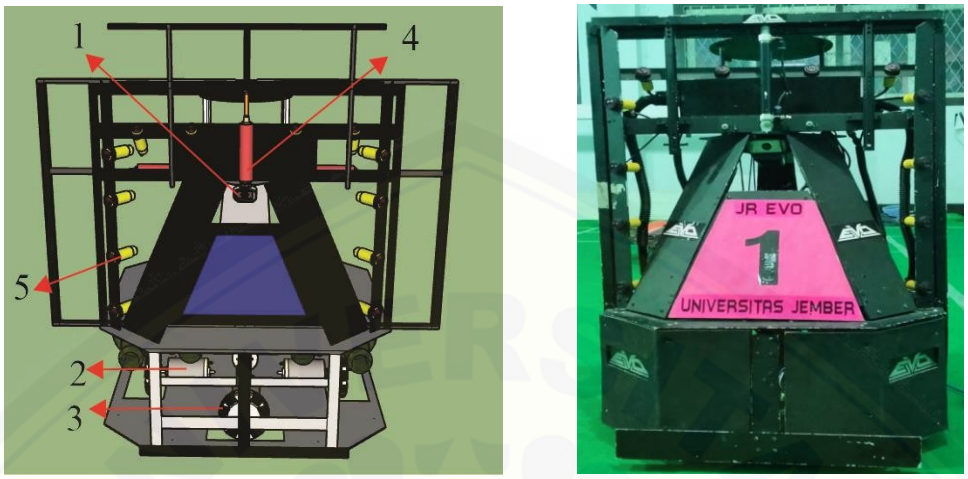
3.4. Perancangan Alat

Alat ini menggunakan sensor ultrasonik yang ditempatkan bagian *body* pada robot. Terdapat empat sensor ultrasonik yang dipasang pada bagian belakang dua sensor, kanan satu sensor, dan kiri satu sensor pada *body* robot, sensor ultrasonik ini berfungsi sebagai pendeteksi dinding gawang yang ada disekitar robot. Robot ini juga menggunakan satu buah sensor kamera yang diapasang pada bagian atas depan robot yang digunakan untuk membaca objek bola sehingga bola dapat dibaca oleh robot. Kemudian digunakan juga sebuah sensor CMPS12 yang digunakan untuk pembacaan arah hadap robot. Bahan pembuatan robot menggunakan plat alumuniaum tebal 5mm.

Keterangan:

1. Camera *logitech*
2. Motor PG 45
3. Roda *omni direction*
4. *Pneumatic*
5. Sensor *proximity*

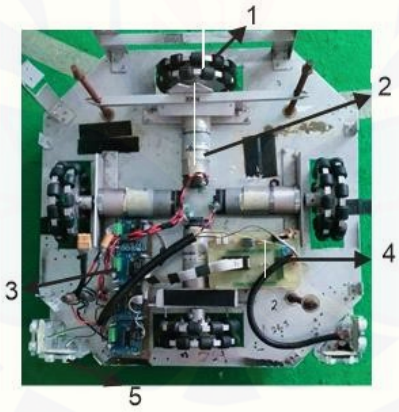
6. Sensor jarak
7. Sensor cmps12



Gambar 3.1 (a) tampak depan dan (b) Robot asli

3.5. Perancangan Elektronik

Pada bagian ini akan dijelaskan berupa perencanaan perangkat keras yang akan digunakan.



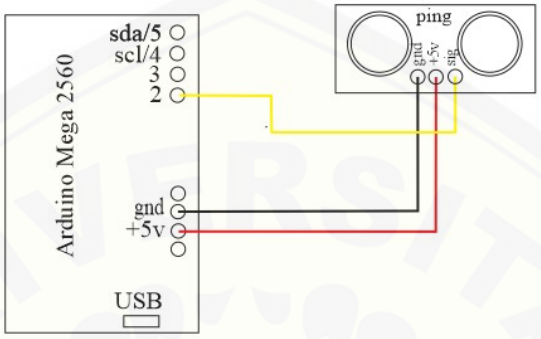
Gambar 3.2 Rangkaian Perangkat Keras Robot Penjaga Gawang

Keterangan :

1. Roda Omni
2. Motor PG45
3. Driver motor BTS7960
4. Mikrokontroler Arduino Due
5. Sensor ultrasonik *ping*

a. Rangkaian Sensor ultrasonik *ping* dengan Arduino

Sensor ultrasonik adalah alat yang digunakan untuk mendeteksi objek-objek sekitar robot. Pada gambar 3.3 menunjukkan bahwa sensor menggunakan 3 pin yaitu *sig* 5volt dan (G) *groud*



Gambar 3.3 Rangakian Sensor Ultrasonik *ping*

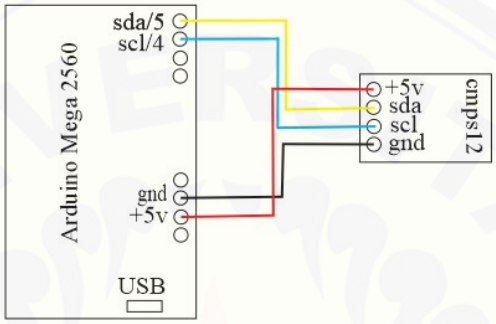
Listing Program ping Sensor *ping* sendiri befungsi sebagai *input* yang digunakan sebagai deteksi dinding pada gawang. Sensor ini bekerja mentedeksi dinding selama dua sekon ketika *low* dan mendeteksi dinding *high* selama lima sekon.

```
pinMode(sKanan, OUTPUT);
digitalWrite(sKanan, LOW);
delay(2);
digitalWrite(sKanan, HIGH);
delay(5);
digitalWrite(sKanan, LOW);
pinMode(sKanan, INPUT);
duration = pulseIn(sKanan, HIGH);
nsKanan = microsecondsToCentimeters(duration);
```

b. Rangkaian modul cmps12 dengan Arduino

Modul sensor cmps12 menggunakan sensor medan magnet yang dibuat dari Philips KMZ51. Sensor ini cukup sensitif untuk mendeteksi sebuah medan magnet bumi. Modul ini bekerja dengan mendeteksi magnetik bumi. Data yang dihasilkan dari kompas elektronik ini berupa data biner. Koneksi dari modul ke

mikrokontroller dapat dilakukan dengan 2 cara yaitu dengan mengunakan data PWM *(Pulse Width Modulation)* atau dengan I2C (*Inter Intergrated Circuit*). Jika menggunakan *interface* PWM, pulsa keluaran memiliki rentang 1mS untuk 0° atau arah utara sampai dengan 36.99 mS untuk 359.90°. Cara yang kedua mengunakan I2C, metode ini dapat digunakan langsung sehingga data yang dibaca tepat 0° – 360° sama dengan 0 – 255.



Gambar 3.4 Rangkaian modul cmps12 dengan arduino

Listing Program sensor Cmps 12, fungsi sensor cmps sendiri digunakan sebagai deteksi arah, pada robot ini kompas digunakan sebagai arah hadap robot.

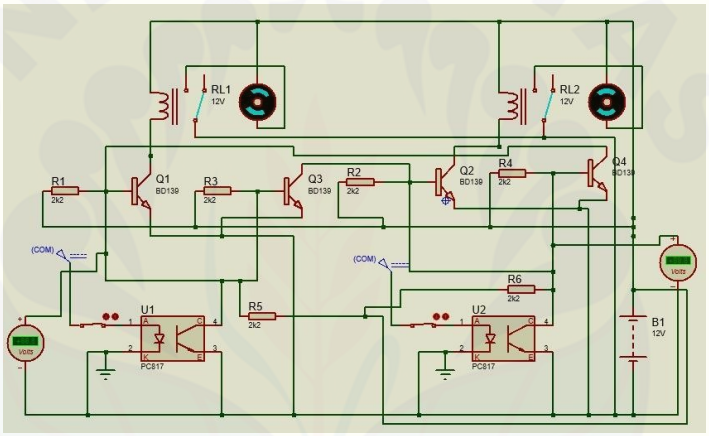
```
kompass = ((cmps03.read()/10)-kal);
 if (kompass<180)
  kompas = kompass;
 else
  kompas = (kompass - 360);
```

c. Rangkaian *Driver Pneumatic*

Driver Pneumatic pada robot ini digunakan sebagai penggerak lengan robot dimana lengan robot menggunakan silinder pneumatic sebagai aktuator gerak serta menggunakan sensor *proximity* dimana sensor *proximity* berfungsi sebagai pembaca arah datang bola. Pada *driver* ini terdiri dari tiga bagian yaitu kanan, kiri dan atas. Cara kerja dari driver ini yaitu ketika salah satu sensor pada satu bagian atas mendeteksi adanya bola maka semua sensor pada bagian atas berlogika *high* atau bias disebut dengan rangkaian pararel. Sedangkan pada lengan bagian kanan dan kiri memiliki aturan yang berbeda, ketika lengan kanan dalam kondisi *high* maka lengan kiri tidak boleh dalam kondisi yang sama, maka

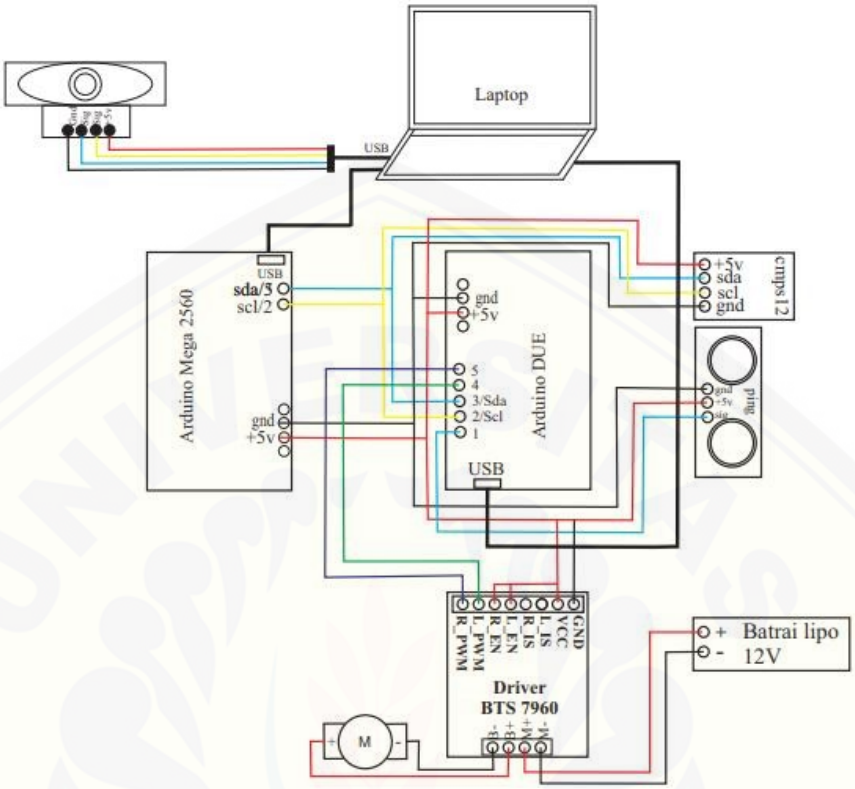
dalam rangkaian ini menggunakan rangkaian pembanding agar lengan kanan dan lengan kiri tidak moleh bergerak bersamaan.

Rangkaian *Driver Pneumatic* terbuat dari beberapa komponen yaitu resistor, BD139, optocoupler, realay. Resistor sendiri digunakan sebagai pembatas arus yang melintas pada suatu rangkaian, yang kemudian dilanjutkan terhubung oleh optocoupler yang berfungsi untuk mengirim sinyal listrik secara terpisah menjadi 2 bagian. BD139 sebagai saklar dengan memanfaatkan titik saturasi dan *cut off* Relay 24v dikarenakan solenoid dari pneumatic memerlukan catu daya 24v maka menggukanan relay 24v, fungsi dari relay digunakan uneuk memutusakan arus yang melintas pada rangkaian.



Gambar 3.5 Skematik Rangkaian *Driver Pneumatic*

3.6. Perangcangan Sistem



Gambar 3.6 Perancangan Sistem

Gambar 3.6 adalah desain elektronik tentang blok diagram robot penjaga gawang KRSBI beroda dimana setiap blok disusun menjadi satu sistem. Sensor ultrasonic PING digunakan sebagai pembacaan dinding gawang yang akan mengirimkan data jarak ke microkontroler. Cmps12 sensor ini digunakan sebagai menentukan arah robot ketika bola berada pada sisi lapangan. Kemudian kamera berfungsi sebagai pembacaan objek bola dimana kamera akan mengirimkan data citra yang akan diproses oleh laptop kemudian dikirimkan ke microkontroler. Dari microkontroler akan mengirimkan data berupa pwm pada driver motor, dari driver motor mengirimkan data berupa tegangan sehingga motor bergerak sesuai perintah. Dalam proses pembuatan robot penjaga gawang KRSBI beroda dengan metode *fuzzy* PID memiliki 3 bagian yaitu *Input, Process* dan *Output*. Kemudian penjelasan masing – masing bagian :

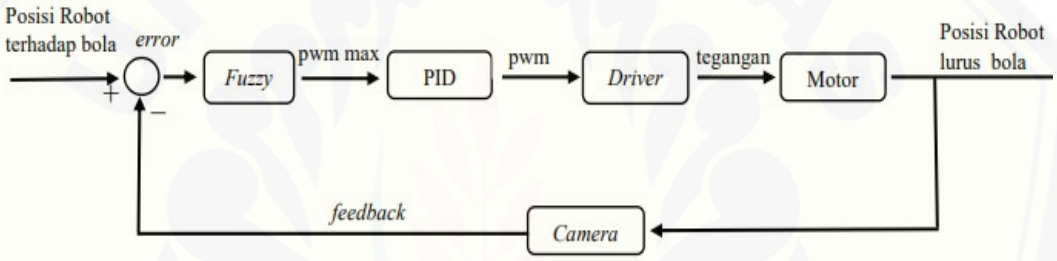
1. *Input*, Bagian *Input* terdiri dari sensor *camera* sensor ini berguna untuk mendeteksi bola dan sensor *ping* sebagai deteksi dinding gawang.

2. *Process*, Bagian *Process* terdapat arduino, arduino disini sebagai pengendali untuk memproses informasi *input* menuju *output* *Process* disini menggunakan metode *fuzzy* PID.

3. *Output*, Bagian *Output* terdapat 4 motor DC. Motor DC digunakan sebagai penggerak robot.

3.7. Perancangan Sistem Kontrol

ada bagian ini akan dijelaskan desain sistem kontrol pada robot untuk mengetahui respon pada robot.

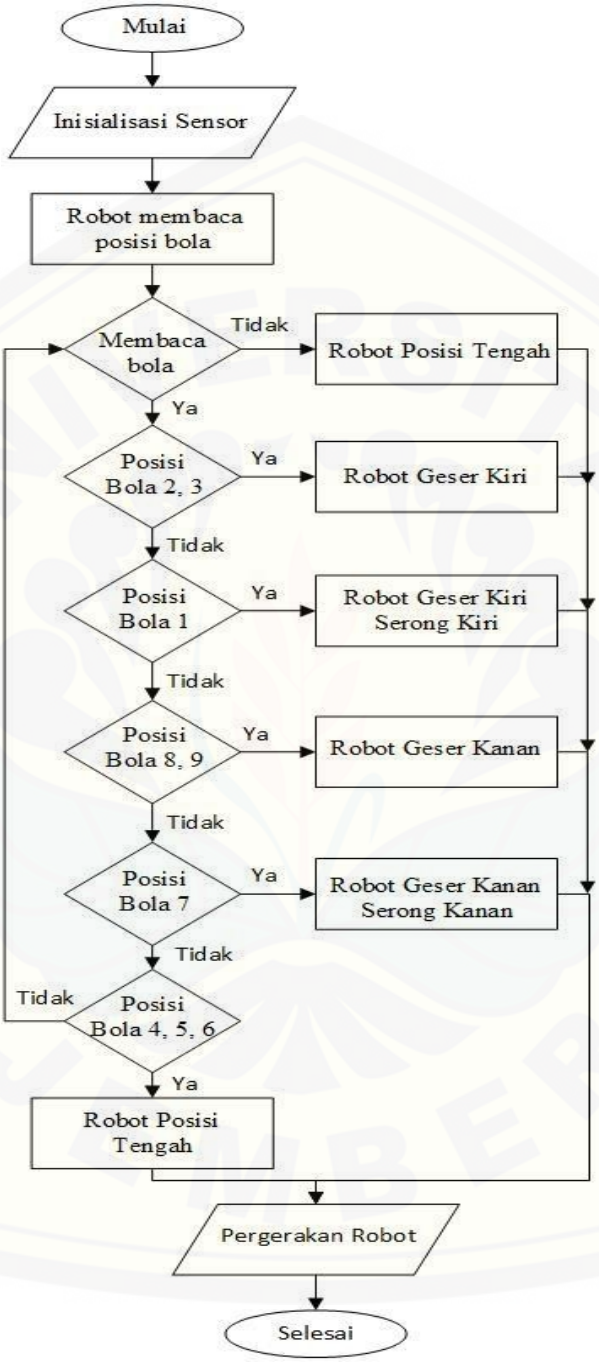


Gambar 3.7 Diagram Blok Sistem kontrol

Desain sistem kontrol diatas merupakan respon yang akan diprogramkan oleh robot, respon pada robot dipengaruhi oleh sensor khususnya sensor ultrasonik PING dimana prilaku robot yang paling utama yaitu robot dapat menyisir pada dinding gawang dengan baik, untuk dapat menyisir maka robot harus bisa membaca objek bola yang ada pada sekitarnya. Kemudian ketika robot sudah dapat membaca bola dan menyisir robot akan mengirimkan data jarak yang telah dibaca secara *real time* untuk memantau kondisi bola.

3.8. Flowchar Robot

a. Flowchat pergerakan robot penjaga gawang KRSBI beroda



Gambar 3.8 Flowchart Robot Kiper

b. Algoritma Robot Kiper

Algoritma Robot penjaga gawang KRSBI beroda pertama inisialisasi sensor yang ada pada robot yaitu sensor cmps12, ping, dan kamera. Kemudian dilakukan kalibrasi pembacaan bola oleh kamera jika robot dapat membaca bola akan dilakukan pembacaan bola sesuai target yang sudah ditentukan. Robot dapat membaca posisi bola dimanapun bola berada sesuai dengan pemetaan pembaacan kamera. Jika bola berada disebelah kiri robot maka robot akan bergeser ke kiri sesaui kecepatan yang ditentukan. Jika robot membaca bola disebelah kanan maka robot akan geser ke kanan, apabila robot tidak membaca bola maka robot akan tetap di tengah.

3.9. Perancangan Software

Graphic User Interface (GUI) adalah antarmuka pada suatu sistem yang menggunakan menu grafis agar mempermudah pengguna atau operator untuk berinteraksi langsung dengan sistem tersebut. GUI memiliki fungsi dua arah yaitu untuk menampilkan informasi kepada pengguna dan juga dapat diberi sebuah perintah oleh pengguna. Dalam penelitian tugas akhir ini, GUI yang digunakan yaitu menggunakan aplikasi Microsoft Visual Basic 2010. Fungsi GUI pada sistem ini juga untuk mengetahui pembacaan posisi bola di lapangan. Pada tampilan ini menampilkan hasil pembacaan bola seperti sudut pembacaan bola, jarak bola, koordinat bola dan mengetahui posisi bola. Semua pembacaan tersebut menggunakan acuan perbandingan nilai pixel pada kamera.

Gambar 3.9 Perancangan *Software Visual Basic*

3.10 Sistem Kontrol *FUZZY*

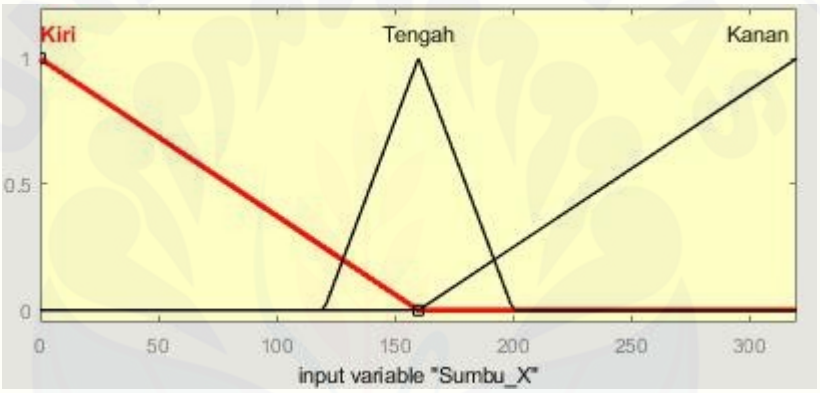
Logika *fuzzy* adalah konseptual mudah dipahami dan memiliki pendekatan alami. Logika *fuzzy* fleksibel dan dapat dengan mudah ditambah dan disesuaikan. Hal ini sangat toleran terhadap data yang tidak tepat dan terhadap model yang nonlinier/ kompleksitas sedikit. Hal ini juga bisa dicampur dengan teknik kontrol konvensional. Ada tiga komponen utama dari sistem fuzzy: set fuzzy, aturan fuzzy, dan bilangan fuzzy.

Logika fuzzy merupakan sistem yang dapat melakukan penalaran dengan prinsip serupa seperti manusia melakukan penalaran dengan nalurinya. Terdapat beberapa jenis FIS berdasarkan metode yang membangunnya yaitu Mamdani, Sugeno dan Tsukamoto. Metode pada FIS merupakan cara penarikan kesimpulan dari sekumpulan kaidah *fuzzy* Proses dalam kendali *fuzzylogic* input yang diberikan dan output yang dihasilkan berupa bilangan tegas tertentu. Melalui fuzifikasi, input nilai tegas (*non-fuzzy*) diproses menjadi nilai-nilai fuzzy sesuai dengan himpunan fuzzy terkait agar data dapat diolah oleh sistem fuzzy. (Cviklovic dkk, 2016).

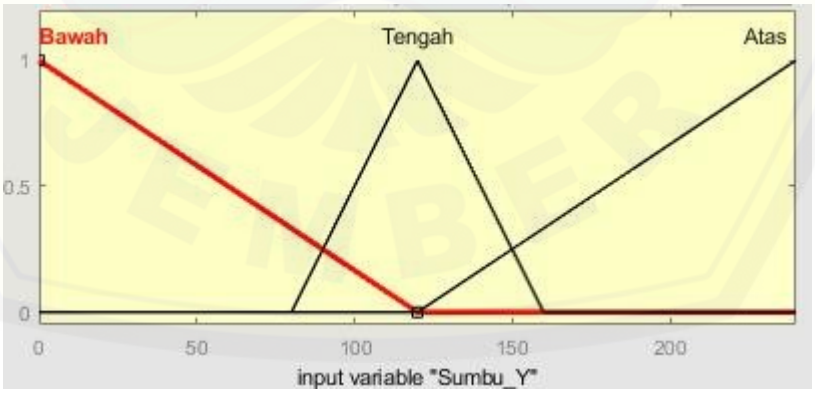
Berikut Arsitektur *Fuzzy Logic*:

1. Fuzzifikasi, Proses konversi input (masukan) yang bersifat tegas (*crips*) kedalam bentuk (*fuzzy*) variabel linguistik menggunakan fungsi keanggotaan.
2. Sistem Inferensi, Proses pengkonversian *input-fuzzy* menggunakan aturan-aturan "*If-Then*" menjadi *Output-Fuzzy*
3. Defuzzifikasi, Proses konversi *Output-Fuzzy* dari sistem inferensi ke dalam bentuk tegas (*crips*) menggunakan fungsi keanggotaan serupa (sebelumnya) menjadi sebuah nilai.

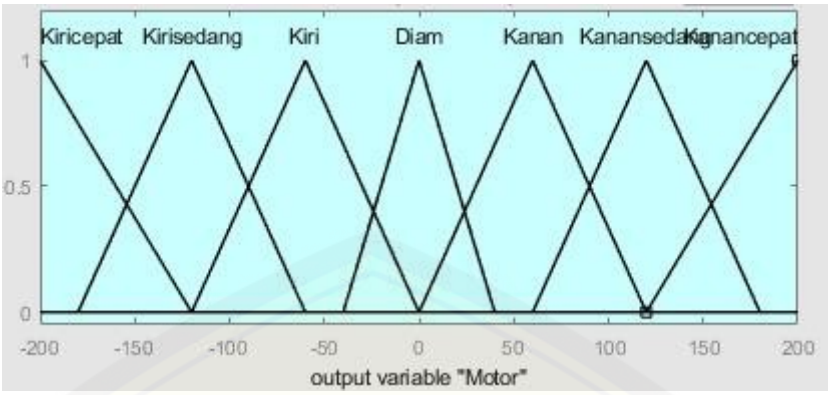
3.9.1. Fungsi Keanggotaan *Fuzzy*



Gambar 3.10 Fungsi Keanggotaan Variabel Kamera sumbu X



Gambar 3.11 Fungsi Keanggotaan Variabel Kamera sumbu Y



Gambar 3. 12 Fungsi Keanggotaan Motor

3.9.2. *Fuzzy Rule* Robot Kiper JR EVO

Tabel 3.2 *Fuzzy Rule*

Y \ X	X ≤126	126 <X<192	X≥192
Y≤80	Kiri cepat	Diam	Kanan cepat
80<Y<160	Kiri sedang	Diam	Kanan sedang
Y≥160	Kiri pelan	Diam	Kanan pelan

Pengambilan keputusan keanggotaan tiap-tiap fungsi keanggotaan himpunan *fuzzy* masukan ke dalam basis aturan yang telah ditetapkan. Serta *fuuzy rule* yang dibuat untuk mengendalikan sebuah robot penjaga gawang untuk basis aturan output dari nilai sumbu dari kamera yaitu sumbu X dan sumbu Y dimana nilai sumbu X dan Y didapatkan oleh nilai *Pixsel.*

Listing program *Fuzzy* Pengambilan keputusan keanggotaan

```
Fuzzy_Matrix[0][0]=min(CXL,CYD);
Fuzzy_Matrix[1][0]=min(CXM,CYD);
Fuzzy_Matrix[2][0]=min(CXR,CYD);
Fuzzy_Matrix[0][1]=min(CXL,CYM);
Fuzzy_Matrix[1][1]=min(CXM,CYM);
Fuzzy_Matrix[2][1]=min(CXR,CYM);
Fuzzy_Matrix[0][2]=min(CXL,CYU);
```

```
Fuzzy_Matrix[1][2]=min(CXM,CYU);
Fuzzy_Matrix[2][2]=min(CXR,CYU);

CXL=0; CXM=0; CXR=0;
CYD=0; CYM=0; CYU=0;
//Camera X
        if(Camera_X<=126) CXL=1;
        else if(Camera_X>=192) CXR=1;
      else
        if(Camera_X>=160){
              CXR=(Camera_X-160)/(192-160);
              CXM=1-CXR;
      else
              CXL=(Camera_X-126)/(160-126);
              CXM=1-CXL;
//Camera Y
          if(Camera_Y<=80) CYD=1;
          else if(Camera_Y>=160) CYU=1;
        else{
          if(Camera_Y>=120){
              CYU=(Camera_Y-120)/(160-120);
              CYM=1-CYU;
         else{
               CYD=(Camera_Y-80)/(120-80);
              CYM=1-CYD;
```

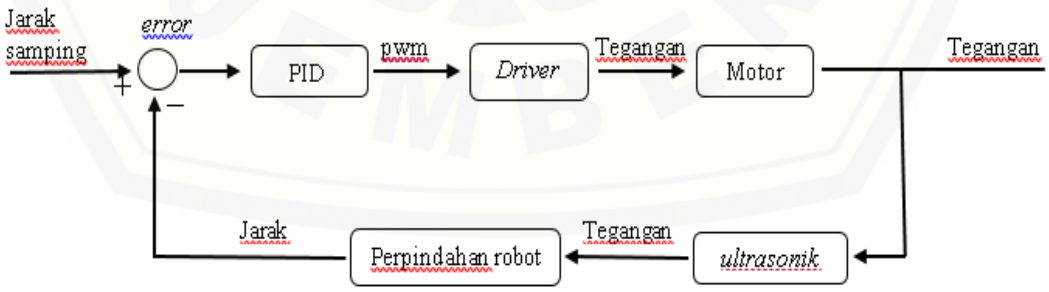
Keterangan :

- Kiri pelan: ketika sumbu X dan sumbu Y pada kamera membaca nilai sudut X ≤126 dan Y≤80
- Kiri sedang : ketika sumbu X dan sumbu Y pada kamera membaca nilai sudut X ≤126 dan 80<Y<160

- Kanan cepat : ketika sumbu X dan sumbu Y pada kamera membaca nilai sudut X ≤126 dan Y≥160
- Tengah : ketika sumbu X dan sumbu Y pada kamera membaca nilai sudut 126 <X<192 dan Y≤80
- Tengah : ketika sumbu X dan sumbu Y pada kamera membaca nilai sudut 126 <X<192 dan 80<Y<160
- Tengah : ketika sumbu X dan sumbu Y pada kamera membaca nilai sudut 126 <X<192 dan Y≥160
- Kanan pelan : ketika sumbu X dan sumbu Y pada kamera membaca nilai sudut X≥192 dan Y≤80
- Kanan sedang : ketika sumbu X dan sumbu Y pada kamera membaca nilai sudut X≥192 dan 80<Y<160
- Kanan cepat : ketika sumbu X dan sumbu Y pada kamera membaca nilai sudut X≥192 dan Y≥160

3.11 Sistem Kontrol Robot menggunakan PID

Pada rancangan robot beroda menggunakan metode PID untuk proses pergerakan robot menggunakan kontrol PID sebagai kontrol dari aktuator pada robot dalam hal ini yaitu motor Dc sebagai penggerak dari robot. Kontrol PID sendiri berfungsi sebagai penstabil kecepatan motor untuk proses dari pergeseran robot.



Gambar 3.13 Diagram blok kontrol PID

Pada gambar 3.7 diagram blok kontrol PID dapat dijelaskan bahwa kontrol PID di set didalam miktokontroler dimana *output* dari mikrokontroler yang berupa

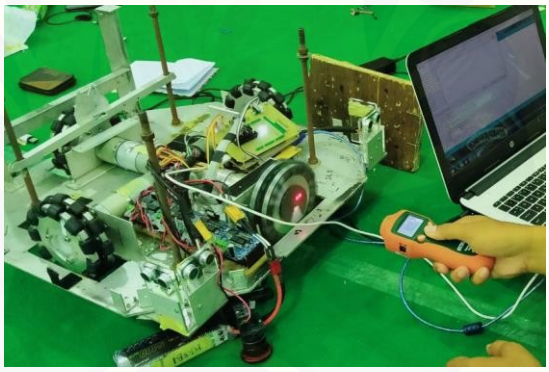
kecepatan pada motor DC itu sendiri digunakan sebagai *feedback* menggunakan sensor untuk memenuhi *input* berupa *set point* atau target yang telah ditentukan.

3.12 Metode Pengujian

Pengujian robot penjaga gawang dibagi menjadi dua yaitu pengujian metode *fuzzy* PID, dan pengujian keseluruhan.

3.11.1. Pengujian Motor Dc

Pada pengujian tahap ini yaitu pengujian terhadap aktuator dimana kita dapat mengetahui kecepatan motor yang digunakan sebagai penggerak dari robot. Pengujian motor Dc ini menggunakan rotary encoder sebagai sensor kecepatan yang akan dibandingkan dengan pembacaan nilai tacho meter dengan input PWM yang berbeda-beda.



Gambar 3.14 Pengujian motor

3.11.2. Pengujian Kontrol PID

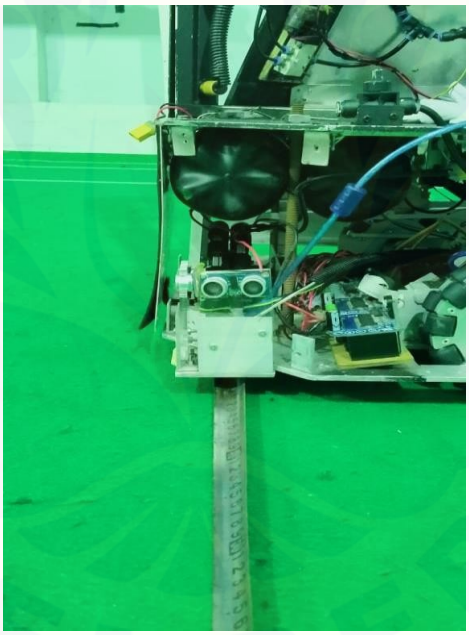
Pada pengujian kontrol PID diharapkan motor mampu bergerak sesuai dengan respon yang diharapkan. Untuk itu dibutuhkan proses *tunning* pada parameter PID. Pada pengujian ini dilakukan beberapa percobaan untuk mencari nilai Kp, Ki, dan Kd yang terbaik. Metode yang digunakan untuk mendapatkan parameter PID tersebut adalah metode Ziegler Nichols 2. Dalam metode tersebut, terdapat beberapa tahap yang harus dilakukan untuk mendapatkan nili Kp, Ki, dan Kd. Dalam pengujian ini, set poin yang ditetapkan adalah berupa target jarak yang akan ditempuh oleh robot. Setelah mendapatkan parameter PID, diharapkan dengan kontrol PID tersebut, robot mampu bergerak menuju target yang telah ditetapkan.

3.11.3. Pengujian sensor

Pada pengujian sensor diharapkan semua sensor mampu membaca bidang dinding gawang sesuai dengan yang diharapkan. Untuk itu dibutuhkan proses *tunning* agar sesusai dengan parameter. Pada pengujian ini dilakukan beberapa pengujian yaitu pengujian sensor dengan jarak dekat, pengujian sensor jarak jauh dan pengujian sensor CMPS12.

3.11.4. Pengujan sensor *ping*

Pada pengujian sensor ini dilakukan untuk mengetahui tingkat akurasi dari pembacaan sensor *ping.* Dimana pembacaan sensor *ping* digunakan untuk mengetahui jarak robot dengan gawang, dan digunkan untuk nilai *input* pada *set point* PID.



Gambar 3.15 Pengujian sensor

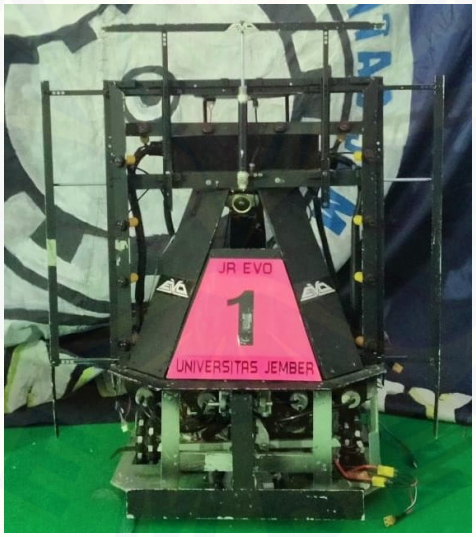
3.11.5. Pengujuan sensor CMPS 12

Pengujian pada modul sensor CMPS 12 digunakan untuk mengukur tingkat presisi pada pembacaan arah sudut pada robot. Dimana sensor ini nantinya digunakan untuk nilai *set point* pada robot saat robot bermanuver mengikuti arah bola.

Gambar 3.16 Pengujian sensor Cmps12

3.11.6. Pengujian Lengan Robot

Pengujian lengan robot ini menggunakan sensor proximity dengan menggunakan aktuator berupa pneumatic. Pada rangkaian ini menggunakan sistem *high low*, jadi ketika sensor mendapat gerakan maka pneumatic akan bergerak.



Gambar 3. 17 Lengan Robot

3.11.7. Pengujian Keseluruhan

Pengujian keseluruhan meliputi pengujian tahapan-tahapan dan pengujian control *Fuzzy*PID. Pada pengujian ini meliputi cara kerja robot secara keseluruhan. Dimana pengujian ini mengacu pada parameter metode *Fuzzy*PID yang telah dibuat. Untuk pengujian keseluruan dilakukan penendangan di setiap sisi yang berbeda, pengujian ini dilakukan penendangan di 5 titik sebanyak 3 kali dalam 1

posisi. Dalam pengujian ini robot harus dapat membaca arah datang bola dan membaca dinding gawang pembatas. Kemudian robot dapat bermanuver dengan sensor *cmps12* ketika bola berada disisi sebelah kanan dan kiri. setelah dilakukan pengujian keseleruan robot diharapkan menghasilkan respon yang bagus menggunakan control *Fuzzy* PID.

BAB 5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil yang telah didapatkan dalam penelitian, dapat diambil beberapa kesimpulan yaitu sebagai berikut :

1. Robot dapat mengetahui gerak bola dengan menggunakan kamera yang menggunakan nilai pixel sebagai acuan pembacaan posisi bola dengan semua posisi bola 100% terdeteksi oleh robot.
2. Robot mampu bermanuver sesuai plan yang diinginkan dengan kontrol PID menggunakan tuning PID *Ziegler Nichols* dengan respon yang baik gambar 4.19 Dengan nilai Kp sebesar 2,4, nilai Ki sebesar 0,53, nilai Kd sebesar 2,7 dan dihasilkan nilai *rise time* (Ts) sebesar 3ms, dan *settling time* sebesar 11ms.
3. Setelah dilakukan pengujian keseluruan robot menggunakan medote *Fuzzy* PID robot mampu menghadang bola dengan sempurna dengan presentasi 93,4%.

5.2. Saran

Pada penelitian tentang implementasi pergerakan robor KRSBI beroda dengan *Fuzzy* PID pada robot penjaga gawang sepak bola beroda JR Evo Universitas Jember, terdapat beberapa saran dari penulis dengan harapan penelitian selanjutnya dapat membuahkan hasil yang lebih maksimal, yaitu :

1. Pada penelitian selanjutnya untuk peletakan kamera harus lebih diperhatikan lagi diamana pada saat bola berada diposisi lebih dekat dengan robot harus kelihatan. Karena pada saat bola terlalu mendekat bola tidak terbaca
2. Pada penelitian selanjutnya diharapkan peneliti selanjutnya sudah tidak menggunakan sensor jarak lagi untuk membaca dinding gawang dimana ketika menggunakan sensor jarak memiliki respon yang lama ketika robot bergerak dengan cepat, maka disarankan menggunakan Rotary encoder untuk mendeteksi posisi robot.

DAFTAR PUSTAKA

Al Hasmy, C. A. R., Ardila, F., & Setiwardhana. (2011). Penentuan Peran Dalam Robot Sepak Bola Dengan. *EEPIS Final Project*, *1*.

Arief, K., Purwanto, D., & Kusuma, H. (2019). Algoritma Menghadang Bola dengan Metode Fuzzy Logic untuk Robot Penjaga Gawang Sepak Bola Beroda. *Jurnal Teknik ITS*, *7*(2). https://doi.org/10.12962/j23373539.v7i2.30970

Arifin, J., Zulita, L. N., & Hermawansyah. (2016). Perancangan Murottal Otomatis Menggunakan Mikrokontroller Arduino Mega 2560. *Jurnal Media Infotama*, *12*(1), 89–98. https://jurnal.unived.ac.id/index.php/jmi/article/view/276/257

Asih, M. S. (2018). Sistem Pendukung Keputusan Fuzzy Mamdani pada Alat Penyiraman Tanaman Otomatis. *Jurnal Sistem Informasi*, *5*(1), 41–52. http://jurnal.uinsu.ac.id/index.php/query/article/view/1566/1271

Budiarso, Z., & Prihandono, A. (2015). Implementasi Sensor Ultrasonik Untuk Mengukur Panjang Gelombang Suara Berbasis Mikrokontroler. *Jurnal Teknologi Informasi DINAMIK*, *20*(2), 171–177.

Fithrony, M. H., Ningrum, E. S., & Sumantri, B. (2017). *Implementasi Metode Fuzzy Logic untuk Kontrol Pergerakan Autonomous Mobile Robot pada Aplikasi Soccer Robot*. *May 2014*, 1–5.

Kusuma, A. T., Agustian, I., Hadi, F., Suandi, A., Elektro, T., Bengkulu, U., Mesin, T., & Bengkulu, U. (2017). Sistem Kendali Fuzzy-Pid Pada Robot Wall Follower. *Jurnal Amplifier*, *7*(1), 1–5.

Naufal, M. R., Belga, R. I., Hartatik, O. T., Wicaksono, S. E., Wibowo, A. W. N., Pradana, A. W., & Yatmono, S. (2018). Rancang Bangun Mekanik Ekspansi Robot Goalkeeper Sepak Bola Beroda Menggunakan Pneumatik. *The 6th Indonesian Symposium on Robotic Systems and Control (ISRSC)*, *1*(1), 98–103. https://doi.org/10.17605/OSF.IO/8GYAT

Pratama, R. W., Ananda, G. R., Putra, Y. D., Mahadika, P., Maulana, Y., & Sistem, A. D. (2017). Rancang Bangun Robot Sepak Bola Beroda. *5th Indonesian Symposium on Robotic System and Control*, 149–151.

Rahman, M., & Aprilianto, H. (2017). Penerapan Metode Fuzzy Pada Robot Beroda Menggunakan Omni-Directional Wheels. *Jutisi*, *5*(2), 1075–1082.

Ramadhani, W. J. A. (2019). Jurnal Teknologi Informasi dan Pendidikan. *PERANCANGAN SISTEM PELAYANAN RESTORAN BERBASIS WEB MOBILE MENGGUNAKAN FRAMEWORK YII2 Maharani*, *12*(1), 4.

Rochamnto. (2014). *Implementasi Robot Three Omni-Directional Menggunakan Kontroler Pid Pada Robot Kontes Robot Abu Indonesia (Krai)*

SAPUTRA, C., & SETIAWAN, R. (2019). Penerapan Sistem Navigasi Sensor Kompas Pada Robot Sepak Bola Beroda. *Jurnal Processor*, *14*(1), 35. https://doi.org/10.33998/processor.2019.14.1.556

Shadiq, H. M., Sudjadi, S., & Darjat, D. (2015). Perancangan Kamera Pemantau Nirkabel Menggunakan Raspberry Pi Model B. *Transient: Jurnal Ilmiah Teknik Elektro*, *3*(4), 546–551. https://doi.org/10.14710/TRANSIENT.3.4.546-551

Stone, A. A. R., Suciati, N., & Navastara, D. A. (2018). Segmentasi Citra pada Robot Sepak Bola Beroda Menggunakan Multilayer Neural Network dan Fitur Warna HSV. *Jurnal Teknik ITS*, *7*(2), 276–281. https://doi.org/10.12962/j23373539.v7i2.33741

Wibowo, B. C., & Iqbal, M. (2013). Implementasi Metode Logika Fuzzy Pada Kontrol Keseimbangan Robot Mobil Beroda Dua. *Simetris : Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, *3*(1), 41. https://doi.org/10.24176/simet.v3i1.87

Widiarto, Y. D., Najoan, M. E. I., Putro, M. D., & Elektro-ft, J. T. (2018). Sistem Penggerak Robot Beroda Vacuum Cleaner Berbasis Mini Computer Raspberry Pi. *E-Journal Teknik Elektro Dan Komputer*, *7*(1), 25–32. https://doi.org/10.35793/jtek.7.1.2018.19140

Zulkifli, Sanjaya, B. W., & Priyatman, H. (2017). Implementasi Logika Fuzzy pada robot beroda penghindar halangan berbasis Arduino Uno R3. *Jurnal Teknik Elektro Universitas Tanjungpura*, *1*(1), 1–9.

LAMPIRAN

1. *Listing* Program Arduino

```
#include <Wire.h>
#define CMPS12_ADDRESS 0x60
#define ANGLE_8  1
#include "CMPS03.h"
CMPS03 cmps03;

unsigned char high_byte, low_byte, angle8;
char pitch, roll, anglec, angles, angle;
unsigned int angle16;
long duration, duration1, duration2, duration3, inches, nsKanan,
nsKiri, nbKanan, nbKiri, ncKanan, ncKiri, nxKanan, nxKiri;
const int sKanan = 23, bKanan = 25, sKiri = 27, bKiri = 29;
int sska, sski;
int sudut, jarak, fsudut, fjarak, csudut, cjarak;
int a, b, c, d;
int x, y, z, X, Y, ry, rx, nx, ny, L;
char kode;
int kaAtas, kiAtas;
int kaBawah, kiBawah;
int nilai;

int button1, button2;
int nC = 0;
int kompass, kompas, kal, pwmB1, pwmB2;

volatile float Fuzzy_Matrix[3][3];
float error_kompas, Camera_X, Camera_Y;
int hasil;
float OVN, ON, Ns, OZ, Ps, OP, OVP;
volatile int CXL, CXM, CXR, CYD, CYM, CYU,  SP, SZ, SN;
float VN = -200, N = -70, NL = -120, Z = 0, PL = 120, P = 70, VP =
200;

#define pwm1 8
#define pwm2 9
```

```
#define pwm3 6
#define pwm4 7
#define pwm5 10
#define pwm6 11
#define pwm7 12
#define pwm8 13

float pid_output, pid_i_mem, pid_setpoint = 0, pid_error,
pid_last_error, pid_p = 3.5, pid_i = 0, pid_d = 0.25, pid_max =
200;
float pid_output1, pid_i_mem1, pid_setpoint1 = 0, pid_error1,
pid_last_error1, pid_p1 = 42, pid_i1 = 15.75, pid_d1 = 39.375,
pid_max1 = 255;
float pid_output2, pid_i_mem2, pid_setpoint2 = 0, pid_error2,
pid_last_error2, pid_p2 = 42, pid_i2 = 15.75, pid_d2 = 39.375,
pid_max2 = 255;
float pid_output3, pid_i_mem3, pid_setpoint3 = 0, pid_error3,
pid_last_error3, pid_p3 = 3.5, pid_i3 = 0, pid_d3 = 0.25, pid_max3
= 230;
float pid_output4, pid_i_mem4, pid_setpoint4 = 0, pid_error4,
pid_last_error4, pid_p4 = 3.5, pid_i4 = 0, pid_d4 = 0.25, pid_max4
= 230;
float pid_output5, pid_i_mem5, pid_error5, pid_last_error5, pid_p5
= 1.5, pid_i5 = 0, pid_d5 = 0.015, pid_max5 = 100;
float pid_output6, pid_i_mem6, pid_error6, pid_last_error6,
pid_setpoin6, pid_p6 = 2.4, pid_i6 = 0, pid_d6 = 2.7, pid_max6 =
150;
volatile long encoder0Pos = 0, encoder1Pos = 0;

long i, rpm = 0, scan_time = 1350, rpmmotor;

void setup() {

  Wire.begin();
  Serial.begin(115200);
  pinMode(pwm1, OUTPUT);
  pinMode(pwm2, OUTPUT);
  pinMode(pwm3, OUTPUT);
```

```
  pinMode(pwm4, OUTPUT);
  pinMode(pwm5, OUTPUT);
  pinMode(pwm6, OUTPUT);
  pinMode(pwm7, OUTPUT);
  pinMode(pwm8, OUTPUT);
  pinMode(31, OUTPUT);
  pinMode(33, OUTPUT);
  pinMode(35, OUTPUT);
  kal = cmps03.read() / 10;
  kal = kal;

  attachInterrupt(digitalPinToInterrupt(4), hitung_rpm, RISING);
  attachInterrupt(digitalPinToInterrupt(5), hitung_rpm, RISING);
}

void loop() {
  Wire.requestFrom(8, 7);    // request 6 bytes from slave device
#2
  fsudut = Wire.read();
  fjarak = Wire.read();
  csudut = Wire.read();
  cjarak = Wire.read();
  nx = Wire.read();
  ny = Wire.read();
  L = Wire.read();
  //---------------------------------------------//
  }
void mulaimanual() {
  {
    if (csudut == 0) sudut = fsudut * -1;
    else sudut = fsudut;
  }
  {
    if (cjarak == 0) jarak = fjarak * -1;
    else jarak = fjarak;
  }
 }
void start() {
```

```
  button1 = digitalRead(23);
  Serial.println(button1);
  if (button1 == 1) {
    motor2(0, 0);
    motor1(0, 0);
    delay(1000);
    motor2(-150, 150);
    delay(3000);
    motor2(0, 0);
    delay(200);
    motor1(-150, 150);
    delay(2000);
    motor1(0, 0);
    delay(200);
    motor2(150, -150);
    delay(2000);
    halang();
  }
  else if (button1 == 0) {
    halang();
  }
}
void halang() {
  nilai_belakang(38, 40);

  if (nsKanan <= 40) {
    if (sudut > 15 || sudut == -34) {
      nilai = 0;
    }
    else {
      nilai = 1;
    }
  }
  if (nsKiri <= 40) {
    if (sudut < -15 || sudut == -34) {
      nilai = 0;
    }
    else {
```

```
    nilai = 2;
  }
}
if (nsKanan > 40 && nsKiri > 40) {
  nilai = 0;
}
if (nilai == 0) {
  if (jarak == 288 || sudut == -34) {
    nilai_samping(75, 75);
  }
  else if (jarak <= 70) {
    X = map(sudut, -10, -180, 235, 240);
    Y = map(sudut, -10, 180, 235, 240);

    if (sudut >= -40 && sudut <= 40) {
      digitalWrite(31, HIGH);
      motor1(0, 0);
    }
    else if (sudut > 40) {
      digitalWrite(31, LOW);
      motor1( -X, X);
    }
    else if (sudut < -40) {
      digitalWrite(31, LOW);
      motor1(Y, -Y);
    }
  }
  else if (jarak <= 110 && jarak > 70) {
    X = map(sudut, -10, -180, 235, 240);
    Y = map(sudut, -10, 180, 235, 240);

    if (sudut >= -25 && sudut <= 25) {
      digitalWrite(31, HIGH);
      motor1(0, 0);
    }
    else if (sudut > 25) {
      digitalWrite(31, LOW);
      motor1( -X, X);
```

```
  }
  else if (sudut < -25) {
    digitalWrite(31, LOW);
    motor1(Y, -Y);
  }
}
else if (jarak < 160 && jarak > 110) {
  X = map(sudut, -10, -180, 235, 240);
  Y = map(sudut, 9, 180, 235, 240);
  if (sudut >= -15 && sudut <= 18) {
    digitalWrite(31, LOW);
    motor1(0, 0);
  }
  else if (sudut > 18) {
    digitalWrite(31, LOW);
    motor1(-X, X);
  }
  else if (sudut < -15) {
    digitalWrite(31, LOW);
    motor1(Y, -(Y));
  }
}
else if (jarak >= 160 && jarak < 210) {
  X = map(sudut, -15, -180, 235, 240);
  Y = map(sudut, 15, 180, 235, 240);
  if (sudut >= -6 && sudut <= 6) motor1(0, 0);
  else if (sudut > 15) {
    motor1(-X, (X));
  }
  else if (sudut < -15) {
    motor1(Y, -(Y));
  }
}
else {
  if (sudut >= -20 && sudut <= 20) motor1(0, 0);
  else if (sudut > 20) {
    ngiri();
  }
```

```
      else if (sudut < -20) {
        nganan();
      }
    }
  }

  else if (nilai == 1) {
    nilai_sampingkanan(25);
  }
  else if (nilai == 2) {
    nilai_sampingkiri(25);
  }
}
void motor1(int tri, int one) {
{ if (one > 0) {
      analogWrite(pwm1, one);
      analogWrite(pwm2, 0);
    }
    else if (one < 0) {
      analogWrite(pwm1, 0);
      analogWrite(pwm2, abs(one));
    }
    else {
      analogWrite(pwm1, 0);
      analogWrite(pwm2, 0);
    }
  }
{ if (tri > 0) {
      analogWrite(pwm5, tri);
      analogWrite(pwm6, 0);
    }
    else if (tri < 0) {
      analogWrite(pwm5, 0);
      analogWrite(pwm6, abs(tri));
    }
    else {
      analogWrite(pwm5, 0);
      analogWrite(pwm6, 0);
```

```
    }
  }
}
void motor2(int four, int two) {
{ if (two > 0) {
      analogWrite(pwm4, two);
      analogWrite(pwm3, 0);
    }
    else if (two < 0) {
      analogWrite(pwm4, 0);
      analogWrite(pwm3, abs(two));
    }
    else {
      analogWrite(pwm4, 0);
      analogWrite(pwm3, 0);
    }
  }
{ if (four > 0) {
      analogWrite(pwm8, four);
      analogWrite(pwm7, 0);
    }
    else if (four < 0) {
      analogWrite(pwm8, 0);
      analogWrite(pwm7, abs(four));
    }
    else {
      analogWrite(pwm8, 0);
      analogWrite(pwm7, 0);
    }
  }
}
void ngiri() {
  motor1(-240, 240);
}
void nganan() {
  motor1(240, -240);
}
void maju () {
```

```
  motor2(-235, 235);
}
void mundur() {
  motor2(235, -235);
}
void taka() {
  int stateka = digitalRead(37);
  if (stateka == 1)
  {
    digitalWrite(33, HIGH);
  }
  else
  {
    digitalWrite(33, LOW);
  }
}
void taki() {
  int stateki = digitalRead(39);
  if (stateki == 1)
  {
    digitalWrite(35, HIGH);
  }
  else
  {
    digitalWrite(35, LOW);
  }
}
void takaka() {
  int nilaika = digitalRead(37);
  int nka = 0;
  nka++;
  if (nka >= 200)nka = 0;
  if (nka < 100)digitalWrite(33, HIGH);
  else digitalWrite(33, LOW);
}
void takiki() {
  int nilaiki = digitalRead(39);
  int nki = 0;
```

```
  nki++;
  if (nki >= 200)nki = 0;
  if (nki < 100)digitalWrite(35, HIGH);
  else digitalWrite(35, LOW);
}
void nilai_samping(int kiri, int kanan) {
  int nsa = -kanan + nsKanan;
  if (nsKanan >= kanan) nsa = 0;
  int nsb = kiri - nsKiri;
  if (nsKiri >= kiri) nsb = 0;
  int nilai1 = nsa + nsb;

  pid_error = (nilai1 - pid_setpoint);
  pid_i_mem += pid_i * pid_error;
  if (pid_i_mem > pid_max)pid_i_mem = pid_max;
  else if (pid_i_mem < pid_max * -1)pid_i_mem = pid_max * -1;
  pid_output = pid_p * pid_error + pid_i_mem + pid_d * (pid_error
- pid_last_error);
  if (pid_output > pid_max)pid_output = pid_max;
  else if (pid_output < pid_max * -1)pid_output = pid_max * -1;
  pid_last_error = pid_error;
  motor1(-pid_output, pid_output);
}

void nilai_belakang(int bkiri, int bkanan) {
  pid_calculate5();
  int nba = -bkanan + nbKanan;
  if (nbKanan >= bkanan - 2 && nbKanan <= bkanan + 2) nba = 0;
  int nbb = -bkiri + nbKiri;
  if (nbKiri >= bkiri - 2 && nbKiri <= bkiri + 2) nbb = 0;

  pid_error1 = (nba - pid_setpoint1) ;
  pid_i_mem1 += pid_i1 * pid_error1;
  if (pid_i_mem1 > pid_max1)pid_i_mem1 = pid_max1;
  else if (pid_i_mem1 < pid_max1 * -1)pid_i_mem1 = pid_max1 * -1;

  pid_output1 = pid_p1 * pid_error1 + pid_i_mem1 + pid_d1 *
(pid_error1 - pid_last_error1);
```

```
  if (pid_output1 > pid_max1)pid_output1 = pid_max1;
  else if (pid_output1 < pid_max1 * -1)pid_output1 = pid_max1 * -
1;
  pid_last_error1 = pid_error1;

  pid_error2 = (nbb - pid_setpoint2) ;
  pid_i_mem2 += pid_i2 * pid_error2;
  if (pid_i_mem2 > pid_max2)pid_i_mem2 = pid_max2;
  else if (pid_i_mem2 < pid_max2 * -1)pid_i_mem2 = pid_max2 * -1;
  pid_output2 = pid_p2 * pid_error2 + pid_i_mem2 + pid_d2 *
(pid_error2 - pid_last_error2);
  if (pid_output2 > pid_max2)pid_output2 = pid_max2;
  else if (pid_output2 < pid_max2 * -1)pid_output2 = pid_max2 * -
1;
  pid_last_error2 = pid_error2;
  pwmB1 = -pid_output1 + pid_output5;
  if (pwmB1 < 0) pwmB2 = 0;
  else if (pwmB1 > 255) pwmB1 = 255;
  pwmB2 = pid_output1 + pid_output5;
  if (pwmB2 < 0) pwmB2 = 0;
  else if (pwmB2 > 255) pwmB2 = 255;
  motor2(-pid_output1, pid_output2);
}
void nilai_sampingkanan(int kanan) {
  int nkanan = -kanan + nsKanan;
  if (nsKanan >= kanan - 2 && nsKanan <= kanan + 2) nkanan = 0;
  if (nsKanan <= kanan) {
    pid_error3 = (nkanan - pid_setpoint3) * 30;
  }
  else if (nsKanan > kanan) {
    pid_error3 = (nkanan - pid_setpoint3) * 2;
  }
  pid_i_mem3 += pid_i3 * pid_error3;
  if (pid_i_mem3 > pid_max3)pid_i_mem3 = pid_max3;
  else if (pid_i_mem3 < pid_max3 * -1)pid_i_mem3 = pid_max3 * -1;
  pid_output3 = pid_p3 * pid_error3 + pid_i_mem3 + pid_d3 *
(pid_error3 - pid_last_error3);
  if (pid_output3 > pid_max3)pid_output3 = pid_max3;
```

```
  else if (pid_output3 < pid_max3 * -1)pid_output3 = pid_max3 * -
1;
  pid_last_error3 = pid_error3;
  motor1(-pid_output3, pid_output3);
}
void nilai_sampingkiri(int kiri) {
  int nkiri = -kiri + nsKiri;
  if (nsKiri >= kiri - 2 && nsKiri <= kiri + 2) nkiri = 0;

  if (nsKiri <= kiri) {
    pid_error4 = (nkiri - pid_setpoint4) * 30;
  }
  else if (nsKiri > kiri) {
    pid_error4 = (nkiri - pid_setpoint4) * 2;
  }

  pid_i_mem4 += pid_i4 * pid_error4;
  if (pid_i_mem4 > pid_max4)pid_i_mem4 = pid_max4;
  else if (pid_i_mem4 < pid_max4 * -1)pid_i_mem4 = pid_max4 * -1;
  pid_output4 = pid_p4 * pid_error4 + pid_i_mem4 + pid_d4 *
(pid_error4 - pid_last_error4);
  if (pid_output4 > pid_max4)pid_output4 = pid_max4;
  else if (pid_output4 < pid_max4 * -1)pid_output4 = pid_max4 * -
1;
  pid_last_error4 = pid_error4;
  motor1(pid_output4, -pid_output4);
}
void hitung_rpm() {
  rpm++; // rutin interrupt 0 (INT 0)
}
void cariBola() {
  if (nsKiri > 20 || nsKanan > 20) {
    if (L == 0) {
      nilai_samping(50, 50);
    }
    else if (L == 33) {
      motor1(-100, 100);
    }
```

```
    else if (L == 13) {
      motor1(100, -100);
    }
    else if (L == 12) {
      motor1(150, -150);
    }
    else if (L == 32) {
      motor1(-150, 150);
    }
    else if (L == 31) {
      motor1(-200, 200);
    }
    else if (L == 11) {
      motor1 (200, -200);
    }
    else if (L == 12 || L == 22 || L == 32) {
      motor1(0, 0);
    }
  }
  else {
    motor1(0, 0);
  }
}
void cmps(){
  kompass = (cmps03.read()/10);
  if (kompass<180){
    kompas = kompass;
  }
  else {
    kompas = (kompass - 360);
  }
  Serial.println(kompas);
}

void pid_calculate5() { //arah tendangan
  pid_error5 = kompas - 0;
  pid_i_mem5 += pid_i5 * pid_error5;
  if (pid_i_mem5 > pid_max5)pid_i_mem5 = pid_max5;
```

```
  else if (pid_i_mem5 < pid_max5 * -1)pid_i_mem5 = pid_max5 * -1;
  pid_output5 = pid_p5 * pid_error5 + pid_i_mem5 + pid_d5 *
(pid_error5 - pid_last_error5);
  if (pid_output5 > pid_max5)pid_output5 = pid_max5;
  else if (pid_output5 < pid_max5 * -1)pid_output5 = pid_max5 * -
1;
  pid_last_error5= pid_error5;
}
void pid_calculate6() { //arah tendangan
  pid_setpoin6 = 45;
  pid_error6 = kompas - pid_setpoin6;
  pid_i_mem6 += pid_i6 * pid_error6;
  if (pid_i_mem6 > pid_max6)pid_i_mem6 = pid_max6;
  else if (pid_i_mem6 < pid_max6 * -1)pid_i_mem6 = pid_max6 * -1;
  pid_output6 = pid_p6 * pid_error6 + pid_i_mem6 + pid_d6 *
(pid_error6 - pid_last_error6);
  if (pid_output6 > pid_max6)pid_output6 = pid_max6;
  else if (pid_output6 < pid_max6 * -1)pid_output6 = pid_max6 * -
1;
  pid_last_error6= pid_error6;
  motor1(pid_output6, pid_output6);
  motor2(-pid_output6, -pid_output6);
}
void Create_Fuzzy_Matrix(){
  Fuzzy_Matrix[0][0]=min(CXL,CYD);
  Fuzzy_Matrix[1][0]=min(CXM,CYD);
  Fuzzy_Matrix[2][0]=min(CXR,CYD);

  Fuzzy_Matrix[0][1]=min(CXL,CYM);
  Fuzzy_Matrix[1][1]=min(CXM,CYM);
  Fuzzy_Matrix[2][1]=min(CXR,CYM);

  Fuzzy_Matrix[0][2]=min(CXL,CYU);
  Fuzzy_Matrix[1][2]=min(CXM,CYU);
  Fuzzy_Matrix[2][2]=min(CXR,CYU);
}
void Fuzzification(){
  CXL=0; CXM=0; CXR=0;
```

```
  CYD=0; CYM=0; CYU=0;
  if(Camera_X<=106) CXL=1;
  else if(Camera_X>=212) CXR=1;
  else{
    if(Camera_X>=160){
      CXR=(Camera_X-160)/(212-160);
      CXM=1-CXR;
    }
    else{
      CXL=(Camera_X-106)/(160-106);
      CXM=1-CXL;
    }
  }
  if(Camera_Y<=80) CYD=1;
  else if(Camera_Y>=160) CYU=1;
  else{
    if(Camera_Y>=120){
      CYU=(Camera_Y-120)/(160-120);
      CYM=1-CYU;
    }
    else{
      CYD=(Camera_Y-80)/(120-80);
      CYM=1-CYD;
    }
  }
}
void Defuzzification(){
  OVN=Fuzzy_Matrix[0][0];
  ON=Fuzzy_Matrix[0][2];
  Ns=Fuzzy_Matrix[0][1];

OZ=max(Fuzzy_Matrix[1][0],max(Fuzzy_Matrix[1][1],Fuzzy_Matrix[1][2
]));
  OP=Fuzzy_Matrix[2][2];
  Ps=Fuzzy_Matrix[2][1];
  OVP=Fuzzy_Matrix[2][0];
  if(!(OVN==0 && ON==0 && Ns==0 && OZ==0 && Ps==0 && OP==0 &&
OVP==0)){
```

```
hasil=(OVN*VN+ON*N+Ns*NL+OZ*Z+Ps*PL+OP*VP+OVP*P)/(OVN+ON+Ns+OZ+Ps+
OP+OVP);
  }
 }
void cekSensor() {
nsKanan,nsKiri, nbKanan,nbKiri;

  //----sKanan------
  pinMode(sKanan, OUTPUT);
  digitalWrite(sKanan, LOW);
  delay(2);
  digitalWrite(sKanan, HIGH);
  delay(5);
  digitalWrite(sKanan, LOW);
  pinMode(sKanan, INPUT);
  duration = pulseIn(sKanan, HIGH);
  nsKanan = microsecondsToCentimeters(duration);
  //----sKiri-----
  pinMode(sKiri, OUTPUT);
  digitalWrite(sKiri, LOW);
  delay(2);
  digitalWrite(sKiri, HIGH);
  delay(5);
  digitalWrite(sKiri, LOW);
  pinMode(sKiri, INPUT);
  duration1 = pulseIn(sKiri, HIGH);
  nsKiri = microsecondsToCentimeters(duration1);
  //-----bKanan----
  pinMode(bKanan, OUTPUT);
  digitalWrite(bKanan, LOW);
  delay(2);
  digitalWrite(bKanan, HIGH);
  delay(5);
  digitalWrite(bKanan, LOW);
  pinMode(bKanan, INPUT);
  duration2 = pulseIn(bKanan, HIGH);
  nbKanan = microsecondsToCentimeters(duration2);
  //-----bKiri-----
```

```
  pinMode(bKiri, OUTPUT);
  digitalWrite(bKiri, LOW);
  delay(2);
  digitalWrite(bKiri, HIGH);
  delay(5);
  digitalWrite(bKiri, LOW);
  pinMode(bKiri, INPUT);
  duration3 = pulseIn(bKiri, HIGH);
  nbKiri = microsecondsToCentimeters(duration3);
}
long microsecondsToInches(long microseconds) {
  return microseconds / 74 / 2;
}
long microsecondsToCentimeters(long microseconds) {
  return microseconds / 29 / 2;
 }
}
```