TUGAS AKHIR – EE 184801

SISTEM KONTROL JARAK TENDANG BOLA PADA ROBOT SEPAK BOLA BERODA MENGGUNAKAN *FUZZY LOGIC*

Muhammad Revo Khairullah
NRP 07111640000006

Dosen Pembimbing
Dr. Ir. Djoko Purwanto, M.Eng.
Dr. Ir. Hendra Kusuma, M.Eng.Sc.

DEPATERMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

TUGAS AKHIR – EE 184801

SISTEM KONTROL JARAK TENDANG BOLA PADA ROBOT SEPAK BOLA BERODA MENGGUNAKAN *FUZZY LOGIC*

Muhammad Revo Khairullah
NRP 07111640000006

Dosen Pembimbing
Dr. Ir. Djoko Purwanto, M.Eng.
Dr. Ir. Hendra Kusuma, M.Eng.Sc

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

FINAL PROJECT – EE 184801

WHEELED SOCCER ROBOTS DISTANCE KICKING CONTROL SYSTEM USING FUZZY LOGIC

Muhammad Revo Khairullah
NRP 07111540000059

Supervisor(s)
Dr. Ir. Djoko Purwanto, M.Eng.
Dr. Ir. Hendra Kusuma, M.Eng.Sc.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Electrical and Intelligent Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2020

PERNYATAAN KEASLIAN
TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Sistem Kontrol Jarak Tendang Bola pada Robot Sepak Bola Beroda Menggunakan *Fuzzy Logic***" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juni 2020

Muhammad Revo Khairullah
NRP 0711 16 4000 0006

SISTEM KONTROL JARAK TENDANG BOLA PADA ROBOT SEPAK BOLA BERODA MENGGUNAKAN *FUZZY LOGIC*

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik

Pada

Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I

Dr. Ir. Djoko Purwanto, M.Eng.
NIP. 19651211 199002 1 002

SURABAYA
JULI, 2020

SISTEM KONTROL JARAK TENDANG BOLA PADA ROBOT SEPAK BOLA BERODA MENGGUNAKAN *FUZZY LOGIC*

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik

Pada

Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing II

Dr. Ir. Hendra Kusuma, M.Eng.Sc.
NIP. 19640902 198903 1 003

SURABAYA
JULI, 2020

Sistem Kontrol Jarak Tendang Bola pada Robot Sepak Bola Beroda Menggunakan *Fuzzy Logic*

Nama	: Muhammad Revo Khairullah
Pembimbing I	: Dr. Ir. Djoko Purwanto, M.Eng.
Pembimbing II	: Dr. Ir. Hendra Kusuma, M.Eng.Sc.

ABSTRAK

Mekanisme penendang merupakan salah satu mekanisme yang berperan penting dalam robot sepak bola. Ada tiga jenis mekanisme penendang yang digunakan di Indonesia, yaitu *pneumatic*, motor dan solenoid. Mekanisme yang pernah dikembangkan oleh tim robot ITS adalah menggunakan motor *brushless* yang dihubungkan dengan *gearbox*. Namun, mekanisme ini memerlukan ruang gerak yang besar dan mudah rusak apabila tendangannya terhalang oleh suatu benda. Maka dari itu, sebagai solusi dari permasalahan tersebut dirancang mekanisme penendang dengan menggunakan solenoid yang dilengkapi dengan sistem penggerak naik-turun untuk mengatur tendangan yang dihasilkan. Tendangan robot dikontrol menggunakan metode *fuzzy logic* dengan masukan berupa nilai ketinggian dan jarak target. Sementara keluarannya berupa posisi ketinggian mekanisme penendang yang hasil tendangannya akan sesuai dengan jarak dan tinggi yang di-*input*-kan. Nilai jarak target didapatkan dari sistem *odometry*, sementara untuk ketinggiannya dimasukkan secara manual oleh *user*. Dari hasil pengujian didapatkan bahwa metode *fuzzy logic* mampu mengikuti karakteristik mekanisme penendang. Hal ini dibuktikan dengan persentase keberhasilan tendangan mencapai 100% dan rata-rata akurasinya 70-90%. Jarak maksimum yang dapat dicapai oleh sistem adalah 9 meter dan rata-rata nilai RMSE dari sistem *odometry* adalah 10-20 satuan. Namun sistem ini masih kurang efektif karena *error* pembacaan *odometry* akan semakin besar seiring dengan robot yang terus aktif bergerak. Sehingga, perlu dilakukan kalibrasi secara berkala atau mengembangkan sistem pembacaan posisi dan jarak target yang lebih akurat.

Kata Kunci : Mekanisme Penendang, Jarak Tendang, *Fuzzy Logic*.

(..... Halaman ini sengaja dikosongkan .....)

Wheeled Soccer Robots Distance Kicking Control System Using Fuzzy Logic

Name	: Muhammad Revo Khairullah
Supervisor I	: Dr. Ir. Djoko Purwanto, M.Eng.
Supervisor II	: Dr. Ir. Hendra Kusuma, M.Eng.Sc.

ABSTRACT

Kicking mechanism is a such most important system in soccer robots. There are three types of kicking mechanisms used in Indonesia, pneumatic, motor and solenoid. The mechanism that was developed by ITS Robotics Team is using brushless DC motor with a gearbox. But this mechanism requires a lot of space to work and the gearbox is so fragile. Therefore, the kicking mechanism is created using solenoid. This mechanism can move up and down to control the distance and high of the ball kicked. The kicking is controlled by fuzzy logic method. The fuzzy logic uses the height of target and the robot's distance towards the target as an input system and the output is a kicking mechanism position. The distance between robot and target is obtained from the odometry system and the targets height is manually inputted by user. From the test results obtained that the fuzzy logic method can follow the characteristics of the kicking mechanism with the kicking success percentage reaching 100% and the accuracy rates is 70-90%. The maximum distance that can be reached by system is 9 meters and the average RMSE of the odometry system is 10-20 units. However, this system is still ineffective because the error of odometry system is getting bigger along with the robot's movement. Therefore, it is necessary to develop a new system that have a more accurate position reading and target distance.

Keywords: Kicking Mechanism, Kicking Distance, Fuzzy Logic.

(..... Halaman ini sengaja dikosongkan .....)

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT yang telah melimpahkan segala rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan penelitian tugas akhir dengan judul "Sistem Kontrol Jarak Tendang Bola pada Robot Sepak Bola Beroda Menggunakan *Fuzzy Logic*" dengan lancar dan tepat waktu.

Tugas akhir ini dikerjakan sebagai prasyarat untuk mendapatkan gelar Sarjana Teknik dari Departemen Teknik Elektro ITS, serta sekaligus sebagai bahan riset penelitian tim robot sepak bola beroda IRIS ITS. Penulis ingin menyampaikan terima kasih kepada :

1. Bapak, Almh. Ibu, Kakak, serta seluruh keluarga yang senantiasa selalu memberikan dukungan baik berupa moral maupun material kepada penulis selama menempuh perkuliahan.
2. Bapak Dr. Ir. Djoko Purwanto, M.Eng. dan Bapak Dr. Ir. Hendra Kusuma, M.Eng.Sc. selaku dosen pembimbing yang selalu memberikan arahan berupa saran dan bantuan selama pengerjaan tugas akhir ini.
3. Bapak dan ibu dosen Departemen Teknik Elektro, khususnya Bidang Studi Elektronika yang telah memberikan pembelajaran kepada saya.
4. Seluruh rekan serta dosen pembimbing Tim Robotika ITS, khususnya Bapak Rudy Dikairono, ST. MT. selaku pembina Tim IRIS ITS yang selalu meluangkan waktunya untuk kami.
5. Seluruh anggota Tim IRIS ITS yang telah membantu dalam pengerjaan robot dan penyelesaian tugas akhir ini.

Penyusunan tugas akhir ini tentu masih sangat jauh dari kata sempurna dan masih banyak kekurangan. Untuk itu penulis sangat mengharapkan atas kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi tim, departemen, kampus dan khususnya perkembangan dunia robot sepak bola di Indonesia.

Surabaya, 5 Juni 2020

Muhammad Revo Khairullah

(..... Halaman ini sengaja dikosongkan .....)

DAFTAR ISI

ABSTRAK ..... **i**
ABSTRACT ..... **iii**
KATA PENGANTAR ..... **v**
DAFTAR ISI ..... **vii**
DAFTAR GAMBAR ..... **ix**
DAFTAR TABEL ..... **xi**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang **1**
1.2 Perumusan Masalah **2**
1.3 Batasan Masalah **2**
1.4 Tujuan **2**
1.5 Metodologi **3**
1.6 Sistematika Penulisan **3**
1.7 Relevansi dan Manfaat **4**
BAB II TINJAUAN PUSTAKA **5**
2.1 Gerak Parabola **5**
2.2 Solenoid (Lilitan) **8**
2.3 *Microcontroller* STM32F407 *Discovery* **9**
2.4 *Rotary Encoder* **10**
2.5 Mini PC **12**
2.6 *Fuzzy* Logic **12**
2.6.1 Himpunan *Fuzzy* 13
2.6.2 Fungsi Keanggotaan 14
2.6.3 Operasi Himpunan *Fuzzy* 16
2.6.4 Penalaran Monoton 17
2.6.5 Fungsi Implikasi 18
2.6.6 Sistem Inferensi *Fuzzy* 19
BAB III PERANCANGAN SISTEM **23**
3.1 Diagram Blok Sistem **24**
3.2 Perancangan Mekanik Robot **25**
3.2.1 Desain Keseluruhan Robot 26
3.2.2 Desain dan Penempatan *Odometry* 28
3.2.3 Desain Mekanisme Penendang 29
3.3 Perancangan Elektronik Robot **32**
3.3.1 Power Suplai 33
3.3.2 Sistem Penendang 34
3.3.3 Human Machine Interface (HMI) pada Robot 41

3.4 Algoritma dan Sistem Perangkat Lunak.....42
3.4.1 Sistem Pembacaan *Odometry* 42
3.4.2 Sistem Kontrol Penendang 43
3.4.3 Algoritma Logika *Fuzzy* 44
BAB IV HASIL DAN ANALISA DATA 49
4.1 Pengujian Sistem *Odometry* 49
4.2 Pengujian Sistem Penendang 51
4.3 Pendefinisian dan Pengujian Sistem *Fuzzy Logic* 53
4.3.1 Pendefinisian Parameter Sistem 53
4.3.2 Pengujian Sistem Logika *Fuzzy*.....55
4.4 Pengujian Keseluruhan Sistem 56
4.4.1 Pengujian Awal 56
4.4.2 Pengujian Akhir.....57
BAB V PENUTUP 63
5.1 Kesimpulan 63
5.2 Saran 64
DAFTAR PUSTAKA.....65
LAMPIRAN 67

DAFTAR GAMBAR

Gambar 2. 1 Gerak Parabola diuraikan ke Vektor X dan Y 6
Gambar 2. 2 Prinsip Kerja Solenoid 9
Gambar 2. 3 STM32F407 Discovery..... 10
Gambar 2. 4 Prinsip Kerja *Rotary encoder*..... 11
Gambar 2. 5 Mini PC Intel NUC6i7KYK 12
Gambar 2. 6 Representasi Linear Naik..... 14
Gambar 2. 7 Representasi Linear Turun..... 15
Gambar 2. 8 Kurva Segitiga 15
Gambar 2. 9 Kurva Trapesium 16
Gambar 2. 10 Fungsi Implikasi : MIN..... 18
Gambar 2. 11 Fungsi Implikasi : DOT 18
Gambar 2. 12 Inferensi Menggunakan Metode Tsukamoto 20
Gambar 2. 13 Proses Defuzzifikasi 21

Gambar 3. 1 Skema Sistem Keseluruhan 23
Gambar 3. 2 Diagram Blok Sistem..... 24
Gambar 3. 3 Desain 3D Robot..... 26
Gambar 3. 4 Desain 3D Base 1 Robot..... 27
Gambar 3. 5 Omniwheel Rangkap Tiga 27
Gambar 3. 6 Sistem Kendali Bola 28
Gambar 3. 7 Desain Penempatan *Odometry*..... 29
Gambar 3. 8 Ilustrasi Solenoid dan Proyektil..... 30
Gambar 3. 9 Mekanisme Penendang 31
Gambar 3. 10 Ilustrasi Robot Menendang..... 31
Gambar 3. 11 Mekanisme Naik-Turun Penendang 32
Gambar 3. 12 Diagram Blok Sistem Elektronik..... 32
Gambar 3. 13 Baterai LiIon..... 33
Gambar 3. 14 Modul Booster 450V 34
Gambar 3. 15 Rangkaian Switching IGBT..... 35
Gambar 3. 16 Plunger Material [15, p21]..... 37
Gambar 3. 17 Grafik Plunger Length [15, p19]..... 37
Gambar 3. 18 Grafik Diameter Proyektil [15, p20]..... 38
Gambar 3. 19 Grafik Shielding Material [15, p23]..... 39
Gambar 3. 20 Tampilan HMI Pada Robot..... 41

Gambar 3. 21 Sensor IMU MP-6050 43
Gambar 3. 22 Giroskop 3 Sumbu dan Akselerometer 3 Sumbu.....43
Gambar 3. 23 Diagram Blok Sistem Logika *Fuzzy* 45
Gambar 3. 24 Fungsi Keanggotaan Variabel Jarak 45
Gambar 3. 25 Fungsi Keanggotaan Variabel Tinggi 46
Gambar 3. 26 Fungsi Keanggotaan Keluaran *Fuzzy* 48

Gambar 4. 1 Robot dan Lokasi Pengujian.....49
Gambar 4. 2 Persebaran Titik Pengujian *Odometry* 50
Gambar 4. 3 Grafik Jarak Tendang Bola.....52
Gambar 4. 4 Grafik Ketinggian Maksimum Tendangan 52
Gambar 4. 5 Peta Persebaran Titik Pengujian 58
Gambar 4. 6 Hasil Pengujian Tinggi 0 60
Gambar 4. 7 Hasil Pengujian Tinggi 35 61
Gambar 4. 8 Hasil Pengujian Tinggi 65 61
Gambar 4. 9 Hasil Pengujian Tinggi 80 61

DAFTAR TABEL

Tabel 3. 1 Aturan Logika *Fuzzy* 47

Tabel 4. 1 Data RMSE Sistem *Odometry* 51
Tabel 4. 2 Pengujian Nilai Batas Himpunan *Fuzzy* 54
Tabel 4. 3 Penyusunan Aturan *Fuzzy* 55
Tabel 4. 4 Pengujian Sistem Logika *Fuzzy* 56
Tabel 4. 5 Uji Coba Awal Sistem 57
Tabel 4. 6 Hasil Pengujian Keseluruhan Sistem-1 58
Tabel 4. 7 Hasil Pengujian Keseluruhan Sistem-2 60
Tabel 4. 8 Definisi Warna Titik 62
Tabel 4. 9 Hasil Pengujian Akhir 62

(..... Halaman ini sengaja dikosongkan .....)

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pada Kontes Robot Indonesia kategori Sepak Bola Beroda, robot dari setiap tim diharuskan memiliki kemampuan untuk mendeteksi bola dan menendang bola ke gawang lawan agar dapat mencetak gol. Tim yang berhasil mencetak gol ke gawang lawan dengan jumlah terbanyak yang akan menang. Hal ini mengacu pada peraturan kompetisi internasional yaitu *RoboCup* pada kategori *Middle Size League* (MSL) [1]. Pada kompetisi tersebut setiap tim membutuhkan mekanisme penendang dengan akurasi yang sangat tinggi.

Mekanisme penendang berdasarkan jenis aktuatornya pada umumnya ada 3 yaitu, *pneumatic*, motor dan solenoid [2]. Berdasarkan hasil riset sebelumnya, mekanisme penendang menggunakan motor kurang efektif karena memerlukan ruang gerak yang cukup besar dan ketahanan dari motor dan *gearbox*-nya yang lemah. Sementara untuk sistem kontrol tendangnya masih kurang efektif dikarenakan nilai jarak tendangnya masih tidak sesuai dengan sasaran yang diinginkan. Oleh karena itu, pada penelitian kali ini akan dikembangkan sistem kontrol jarak tendang bola menggunakan metode *fuzzy logic*.

Mekanisme penendang yang akan digunakan adalah berjenis solenoid. Dengan menggunakan solenoid sebagai aktuator tendangnya tentu tidak membutuhkan ruang yang besar karena tidak perlu berotasi. Ujung dari proyektil pada solenoid akan dihubungkan pada suatu mekanisme yang dapat bergerak naik-turun. Kemudian, jarak dan ketinggian tendangan yang dihasilkan dapat diatur melalui mekanisme naik-turun tersebut. Posisi mekanisme naik-turun tersebut dikontrol menggunakan sistem *fuzzy logic* dengan masukannya berupa nilai jarak robot terhadap target tendangan dan ketinggian target. Sementara keluarannya berupa posisi mekanisme naik-turun penendang.

Data jarak dan ketinggian dari target diterima oleh *microcontroller* yang selanjutnya digunakan sebagai masukan *fuzzy.* Kemudian, kedua data tersebut akan masuk ke proses pengklasifikasian *membership function* dalam prosedur *Fuzzification.* Data yang sudah dikelompokkan menjadi variabel *fuzzy* dan himpunan *fuzzy* akan masuk ke *Rule Set* dan akan mengeluarkan suatu *statement* sesuai dengan *rulebase* yang telah ditentukan. Tahap selanjutnya adalah proses *Defuzzification* yang

menghasilkan keluaran data yang akan diolah pada proses selanjutnya, yaitu melakukan eksekusi tendangan. Keluaran dari proses *fuzzy logic* ini berupa posisi dari mekanisme naik-turun penendang yang harus dicapai. Selanjutnya, *microcontroller* akan menggerakkan mekanisme penendang hingga mencapai posisi yang telah ditentukan oleh sistem *fuzzy logic*. Dengan mengontrol posisi mekanisme penendang ini diharapkan hasil tendangannya dapat mencapai jarak dan tinggi sesuai dengan masukannya.

Pengimplementasian dari penelitian ini nantinya dapat diterapkan pada robot sepak bola beroda untuk melakukan tendangan langsung ke gawang maupun tendangan mengoper pada kawan. Algoritma dan strategi yang digunakan nantinya menyesuaikan dengan situasi dan kondisi saat robot sedang bermain di lapangan.

1.2 Perumusan Masalah

Permasalahan pada penelitian ini adalah sebagai berikut :

1. Cara mengimplementasikan solenoid sebagai aktuator penendang bola.
2. Cara mengontrol jarak tendang bola.

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. Ukuran lapangan dan gawang disesuaikan dengan aturan Kontes Robot Indonesia.
2. Ukuran dan jenis bola yang digunakan disesuaikan dengan aturan Kontes Robot Indonesia.
3. Batas ketinggian target disesuaikan dengan tinggi gawang pada aturan Kontes Robot Indonesia.
4. Nilai ketinggian target diperoleh dengan memasukkan nilai secara manual oleh penulis.

1.4 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut :

1. Mengimplementasikan mekanisme penendang naik-turun yang dihubungkan pada proyektil solenoid.
2. Mengimplementasikan metode *fuzzy logic* sebagai *controller* tendangannya.

1.5 Metodologi

Metodologi yang digunakan pada penelitian tugas akhir ini adalah sebagai berikut :

1. **Studi Literatur**

   Di tahap studi literatur akan ada pengumpulan dasar teori dari metode yang digunakan untuk menunjang dalam pengerjaan tugas akhir. Dasar teori tersebut bersumber dari buku, jurnal ilmiah dan forum diskusi yang relevan terkait penelitian ini.

2. **Perancangan dan Desain Mekanik Robot**

   Pada tahap ini akan dilakukan perancangan bentuk robot dan penempatan solenoid beserta divais-divais lain yang turut menunjang dalam penelitian ini.

3. **Perancangan dan Pengujian Penendang**

   Pada tahap ini akan dilakukan perancangan dan pengujian sistem penendang guna mendapatkan persamaan tendangnya.

4. **Pengujian Keseluruhan Sistem**

   Pada tahap ini robot akan diuji keseluruhan sistemnya baik mulai dari mendapatkan bola sampai melakukan eksekusi tendangan bola.

5. **Penyempurnaan Sistem**

   Pada tahap ini akan dilakukan penyempurnaan sistem berdasarkan dari hasil pengujian yang telah dibuat.

6. **Penulisan Laporan Tugas Akhir**

   Tahap penulisan laporan tugas akhir adalah membuat dokumentasi pelaksanaan tugas akhir meliputi dasar teori, proses perancangan hingga pembuatan dan pengujian alat.

1.6 Sistematika Penulisan

Sistematika penelitian dari penelitian tugas akhir ini direncanakan sebagai berikut :

- **BAB 1 : PENDAHULUAN**

  Pada Bab ini berisikan hal-hal yang meliputi latar belakang, perumusan masalah, batasan masalah, tujuan, metodologi, sistematika penulisan serta relevansi dan manfaat.

- **BAB 2 : TINJAUAN PUSTAKA DAN TEORI PENUNJANG**

  Bab ini berisikan tentang teori penunjang serta literatur yang berkaitan dengan penelitian ini. Sumber yang digunakan dapat berupa teori yang tertulis pada buku, jurnal ilmiah, konferensi

internasional dan segala hal yang dapat menunjang dalam mengerjakan penelitian ini.

- **BAB 3 : PERANCANGAN SISTEM**
  Bab ini membahas tentang perancangan sistem secara menyeluruh mulai dari desain robot hingga sistem kerja robot.
- **BAB 4 : PENGUJIAN**
  Bab ini berisikan tentang pengujian sistem setelah selesai dirancang dan analisa hasil pengujiannya.
- **BAB 5 : PENUTUP**
  Bab ini berisikan tentang kesimpulan yang didapatkan dari pengerjaan penelitian ini serta saran-saran untuk pengembangan penelitian berikutnya.

1.7 Relevansi dan Manfaat

Manfaat yang didapatkan dari penelitian ini adalah meningkatkan akurasi tendangan pada robot sepak bola beroda agar didapatkan tendangan yang efektif. Selain itu, juga diharapkan dapat diterapkan pada kompetisi Kontes Robot Indonesia kategori KRSBI-Beroda dan kompetisi internasional *RoboCup* kategori *Middle Size League* (MSL).

BAB II
TINJAUAN PUSTAKA

2.1 Gerak Parabola

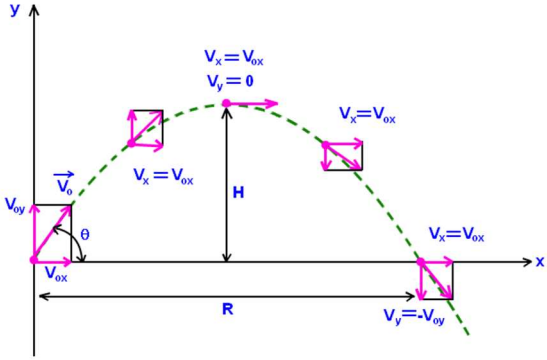
Gerak parabola atau gerak peluru merupakan paduan dari gerak pada bidang horizontal yang tidak memiliki percepatan dan gerak pada bidang vertikal dengan percepatan konstan sebesar g (Nilai satuan percepatan pada gravitasi $g = 9{,}8\ m/s^2$) [3]. Menurut salah seorang ilmuwan yang bernama **Galileo**, mengemukakan dalam salah satu bukunya yang berjudul *Discources on Two New Sciences*, gerak parabola dapat didefinisikan menjadi suatu gerak lurus beraturan pada bidang horizontal (sumbu-x) dan gerak lurus berubah beraturan pada bidang vertikal (sumbu-y) secara terpisah.

Pada dasarnya gerak parabola merupakan gerakan yang terjadi pada suatu benda yang mana gerakan tersebut membentuk sudut terhadap bidang horizontal dengan kecepatan awal tertentu. Kecepatan awal yang terjadi pada gerak parabola kemudian dapat diuraikan menjadi dua bentuk kecepatan V_x (kecepatan terhadap sumbu-x) dan V_y (kecepatan terhadap sumbu-y). Dua bentuk kecepatan inilah yang disebutkan oleh **Galileo**, yaitu V_x sebagai gerak lurus beraturan dan V_y sebagai gerak lurus berubah beraturan. Pada sumbu-x, disebutkan sebagai gerak lurus beraturan dikarenakan kecepatan V_x tidak dipengaruhi oleh gaya lain yang bekerja di sekitarnya, sehingga dapat didefinisikan sebagai gerak lurus beraturan (GLB). Sementara pada sumbu-y, kecepatan V_y dipengaruhi oleh gaya gravitasi yang akan selalu menarik semua benda jatuh ke bawah. Sehingga, nilai kecepatan V_y akan terus berkurang dari waktu ke waktu sampai batas waktu tertentu. Kemudian setelah nilai kecepatan V_y telah bernilai nol (0), maka benda akan mengalami gerak jatuh bebas. Hal ini yang menyebabkan sumbu-y didefinisikan sebagai gerak lurus berubah beraturan (GLBB).

Gerak parabola ini merupakan gerak yang spesial pada gerak dua dimensi. Hal ini dikarenakan gaya gesek udara yang terjadi dapat diabaikan sehingga gaya yang bekerja hanya gaya berat yang dihasilkan oleh percepatan gravitasi [4].

Gerak parabola dipelajari pada bahasan kinematika, ilmu yang membahas tentang gerak benda tanpa memperhatikan penyebabnya, sehingga gerakan benda hanya ditinjau setelah diberi kecepatan awal dan benda bergerak dalam lintasan melengkung di mana hanya ada pengaruh

gravitasi [4]. Hal ini menyebabkan pengaruh lain seperti gesekan benda dengan udara dapat diabaikan [5].



Gambar 2. 1 Gerak Parabola diuraikan ke Vektor X dan Y

Pada sumbu x berlaku persamaan gerak lurus beraturan. Jika pada sumbu x, kecepatan awal adalah v_{0x}, kecepatan pada saat t adalah v_x, dan posisi adalah x, maka persamaannya menjadi :

$$v_x = v_{0x} \tag{2.1}$$

$$x = v_{0x} \, . t \tag{2.2}$$

Pada sumbu y berlaku persamaan umum gerak lurus berubah beraturan, yaitu :

$$v_x = v_0 + at \text{ dan } x = v_0 t + \frac{1}{2} a t^2 \tag{2.3}$$

Jika pada sumbu y kecepatan awal adalah v_{0y}, kecepatan pada saat t adalah v_y, persamaan $a = -g$ (mengarah ke bawah), dan posisi adalah y, maka persamaannya menjadi :

$$v_y = v_{0y} - gt \tag{2.4}$$

$$y = v_{0y} t - \frac{1}{2} g t^2 \tag{2.5}$$

Untuk kecepatan awal v_{0x} dan v_{0y} dengan besarnya v_0 (kecepatan awal) dan sudut α terhadap sumbu x positif, maka diperoleh :

$$v_{0x} = v_0 \,.\cos\alpha \tag{2.6}$$

$$v_{0y} = v_0 \,.\sin\alpha \tag{2.7}$$

Misalkan pada saat t sekon, benda berada di P, kecepatan benda pada saat itu adalah v. Komponen kecepatan v pada sumbu x adalah v_x dan pada sumbu y adalah v_y, sehingga berlaku

Besar Kecepatan :

$$v = \sqrt{{v_x}^2 + {v_y}^2} \tag{2.8}$$

Arah Kecepatan :

$$\tan\alpha = \frac{v_y}{v_x} \tag{2.9}$$

Perhatikan gambar gerak parabola di atas. Pada saat benda bergerak naik dari titik awal hingga sampai titik tertinggi H, maka kecepatan (besar dan arah) pada sumbu x tetap, tetapi besar kecepatan pada sumbu y berkurang secara beraturan dengan percepatan sama dengan percepatan gravitasi bumi (g).

Pada titik tertinggi H, kecepatan pada sumbu y sama dengan nol, sehingga kecepatan pada titik tertinggi sama dengan kecepatan pada sumbu x. Waktu di titik tertinggi t_H dapat dirumuskan sebagai berikut :
di mana,

t_H = waktu pada titik tertinggi (s)
v_0 = kecepatan awal (m/s)
g = percepatan gravitasi bumi (m/s^2)

$$t_H = \frac{v_0 \,.\sin\alpha}{g} \tag{2.10}$$

Tinggi maksimum (y_H) yang dicapai adalah :

$$y_H = \frac{{v_0}^2 \,.\sin^2\alpha}{2g} \tag{2.11}$$

Maka koordinat x dari titik tertinggi H adalah :

$$x_H = \frac{{v_0}^2 . \sin 2\,\alpha}{2g} \tag{2.12}$$

2.2 Solenoid (Lilitan)

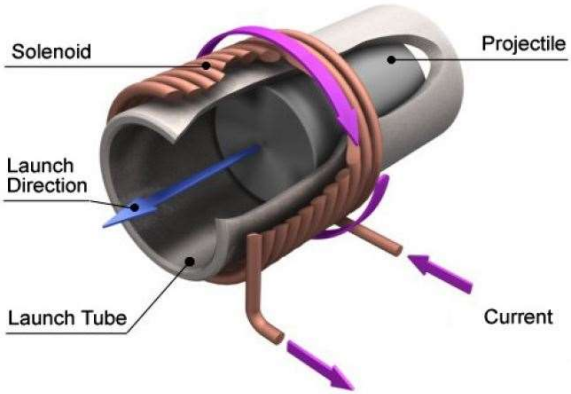
Solenoid merupakan sebuah gulungan/lilitan yang dapat menghasilkan medan magnet. Hal ini berdasarkan prinsip medan magnet dan arus listrik, yaitu saat arus mengalir melewati sebuah kawat yang terhubung melingkar, maka akan timbul medan magnet di sekitarnya. Solenoid terdiri dari sejumlah kawat yang melingkar sehingga membentuk sebuah gulungan/lilitan, sehingga saat dialiri sebuah arus listrik maka akan timbul medan magnet yang cukup besar [6].

Berdasarkan prinsip kerja yang telah disebutkan di atas, maka solenoid terdiri dari tiga komponen utama sebagai penyusunnya yaitu, proyektil (*plunger*), pelindung (*shield*) dan lilitan kawat [7].

Lilitan kawat berfungsi sebagai penghantar listrik yang dapat menghasilkan medan magnet. Medan magnet yang dihasilkan bergantung dari beberapa faktor yaitu, panjang kawat, jumlah lilitan dan ketebalan (diameter) kawat yang digunakan.

Proyektil merupakan silinder pejal yang terdapat di dalam lilitan dan bersifat feromagnetik. Proyektil inilah yang akan berperan sebagai aktuator, menerima gaya yang dihasilkan medan magnet sehingga menyebabkan proyektil dapat bergerak terdorong keluar. Gaya dorong yang diterima bergantung pada besarnya energi yang dihasilkan oleh medan magnet.

Pelindung atau *shielding* merupakan komponen yang juga berperan cukup penting. Pelindung berfungsi untuk memastikan tidak adanya kebocoran medan magnet yang dihasilkan oleh lilitan sehingga mempengaruhi komponen lain yang ada di luar solenoid. Selain itu pelindung juga berperan sebaliknya, yaitu memastikan tidak ada komponen lain yang ada di luar solenoid mempengaruhi medan magnet yang dihasilkan oleh lilitan kawat. Pelindung biasanya dipasang pada bagian terluar solenoid sebagai penutup dari solenoid itu sendiri. Pelindung terbuat dari bahan dengan nilai reluktansi yang tinggi agar energi yang diserap juga semakin tinggi, sehingga tidak ada energi yang bocor keluar.



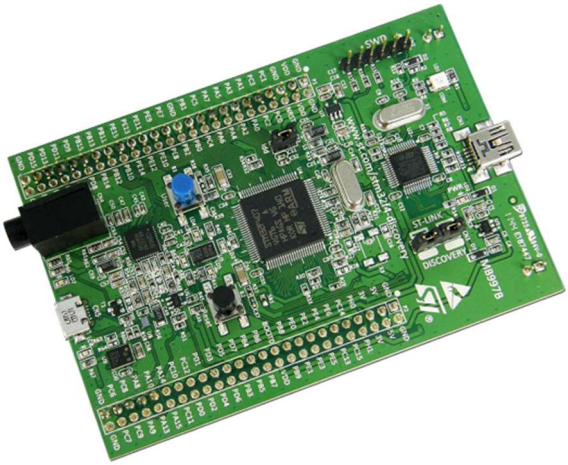
Gambar 2.2 Prinsip Kerja Solenoid

Solenoid ini banyak diimplementasikan dalam kehidupan sehari-hari. Hal ini dikarenakan bahannya yang mudah dicari dan cara pembuatannya yang juga mudah. Selain itu, prinsip kerjanya juga sederhana dan sangat efektif. Solenoid ini dapat menjadi sebuah solusi dalam mencari aktuator yang posisi benda dan arah vektor geraknya searah, tidak seperti motor DC yang arah vektor geraknya berbeda dengan posisi bendanya. Pergerakan dari solenoid ini mirip dengan *pneumatic*, yang menjadi keduanya berbeda adalah solenoid hanya membutuhkan suplai tegangan agar dapat bekerja, sementara *pneumatic* selain membutuhkan suplai tegangan juga membutuhkan suplai udara sebagai gaya dorongnya. Contoh penggunaan solenoid dalam kehidupan sehari-hari adalah solenoid *lockdoor*. Kemudian dalam bidang industri juga menggunakan solenoid *valve*, yang berguna sebagai saklar on/off. Dalam bidang riset dan teknologi, khususnya bidang robotika, solenoid banyak sekali digunakan dalam robot sepak bola beroda sebagai mekanisme penendangnya juga sekaligus lengan *extend* untuk robot penjaga gawang.

2.3 *Microcontroller* STM32F407 *Discovery*

STM32F407 *Discovery* merupakan sebuah modul *microcontroller* berbasis ARM 32bit dengan berbagai macam fitur yang sangat lengkap. Jika dibandingkan dengan modul *microcontroller* lainnya seperti *nucleo* maupun Arduino, STM32F407 *Discovery* ini memiliki fitur yang lebih lengkap dan cara menggunakannya yang mudah.

Modul *microcontroller* ini berbasis prosesor ARM®Cortex®-M4 32-bit yang beroperasi pada frekuensi hingga 168 MHz. Beberapa kelebihan yang dimiliki oleh modul ini adalah sudah memiliki kristal osilator yang tersedia pada *board* modul sehingga tidak perlu menambahkan eksternal osilator sebagaimana pada *board nucleo*. Selain itu modul ini juga sudah dilengkapi dengan STLINK/V2A sehingga untuk mengunduh *binary program*, cukup dihubungkan dengan komputer melalui koneksi USB tanpa memerlukan periferal tambahan.



Gambar 2. 3 STM32F407 *Discovery*

2.4 *Rotary Encoder*

Rotary encoder merupakan sensor posisi yang biasa digunakan untuk mengubah posisi sudut atau sudut gerakan dari *shaft* dengan keluarannya berupa sinyal analog atau digital berdasarkan dari jenisnya. Jenis dari *encoder* ada bermacam-macam, yang mana dapat diklasifikasikan menurut sinyal keluarannya dan juga teknologi *sensing*-nya. Ada dua tipe dasar pada *encoder*, yaitu tipe absolut dan inkremental.

a. Absolut *Encoder*

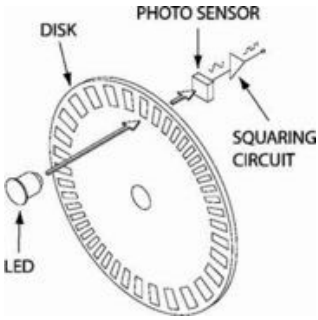
Encoder jenis absolut merupakan *encoder* yang sangat memperhatikan sudut atau posisi tertentu dari perputaran *shaft*. Hal ini membuat *encoder* jenis absolut banyak digunakan untuk membaca pergerakan putaran yang sangat kecil seperti pada mesin CNC, mesin 3D printer dan lengan robot. *Encoder* jenis ini

mempunyai keunggulan yaitu mempertahankan informasi posisinya saat suplai dilepas dari *encoder* dan informasi posisinya akan segera muncul seketika saat suplai kembali disambungkan.

b. Inkremental *Encoder*

Inkremental *encoder* merupakan jenis *encoder* yang paling banyak digunakan daripada jenis yang lain. Hal ini dikarenakan kemampuannya dalam memberikan informasi yang *real-time*. *Encoder* jenis ini mengirimkan informasi perubahan posisi lebih cepat dibandingkan *encoder* jenis lainnya. Maka dari itu inkremental *encoder* biasanya digunakan dalam aplikasi yang membutuhkan pengukuran posisi dan kecepatan yang tepat. Contoh penggunaannya adalah *rotary encoder* pada robot pelayan dan internal *encoder* pada motor DC.

Teknologi *sensing* yang digunakan *encoder* ada bermacam-macam, salah satunya adalah menggunakan optik. Untuk teknologi yang menggunakan optik, prinsip kerjanya cukup sederhana yaitu menggunakan LED dan photosensor (sensor cahaya). Di dalam *encoder* terdapat sebuah piringan yang memiliki bagian berlubang dan tidak berlubang mengelilingi piringan tersebut dengan jumlah tertentu. Lihat **Gambar 2.4**. Ketika piringan berputar dan cahaya dari LED mengenai daerah berlubang maka cahaya tersebut akan diterima oleh *photosensor*, jika cahaya LED tertutup daerah yang tidak berlubang maka *photosensor* tidak akan menerima cahaya dari LED. Hal ini akan terjadi secara terus-menerus selama shaft *encoder* terus berputar. Diterima dan tidaknya cahaya LED oleh *photosensor* akan membentuk sebuah sinyal atau pulsa, sinyal atau pulsa inilah yang kemudian dibaca oleh *microcontroller* untuk kemudian dilakukan perhitungan untuk proses selanjutnya.



Gambar 2. 4 Prinsip Kerja *Rotary Encoder*

2.5 Mini PC

Mini PC merupakan PC portabel yang berukuran kecil dan memiliki fitur yang kurang lebih sama dengan PC pada umumnya, seperti laptop dan komputer *desktop*. Mini PC ini hadir untuk menggantikan peran CPU agar menjadi berukuran lebih kecil dan lebih efektif. Fitur dan spesifikasi yang disajikan kini juga cukup beragam menyesuaikan pasar dan kebutuhan dari konsumen. Bahkan kini juga telah hadir mini PC dengan tipe *high-end*, tipe ini menyajikan fitur dan spesifikasi yang komplit dan dikhususkan untuk *gaming* sehingga dapat disetarakan dengan PC *desktop* dan laptop *gaming*. Dari sekian banyak merek yang memproduksi mini PC, hanya ada beberapa yang namanya cukup terkenal dan paling banyak diminati oleh konsumen yaitu, Asus, Intel NUC dan NVDIA.

Gambar 2. 5 Mini PC Intel NUC6i7KYK

2.6 *Fuzzy* Logic

Logika *fuzzy* merupakan salah satu metode yang digunakan dalam kecerdasan buatan. Metode ini pertama kali diperkenalkan oleh seorang ilmuwan Amerika Serikat berkebangsaan Iran dari Universitas California di Barkeley, yaitu Lotfi A. Zadeh melalui tulisannya pada tahun 1965 tentang teori himpunan *fuzzy*. Metode ini muncul dikarenakan adanya "ketidakpastian" atau warna "abu-abu" dalam menentukan suatu kondisi. Hal ini disebabkan karena ada beberapa kondisi yang tidak bisa disetarakan dengan logika biner Aristoteles [8]. Dalam logika yang sederhana, setiap fakta dapat bernilai benar atau salah. Namun banyak juga informasi yang disampaikan saat ini melibatkan "ketidakpastian", seperti seseorang yang memiliki tinggi badan tanggung (tidak tinggi juga tidak pendek), atau suatu minuman yang dikatakan tidak manis juga tidak pahit. Hal ini menyebabkan informasi menjadi bernilai "abu-abu". Dengan metode logika *fuzzy* memungkinkan seseorang untuk

melampirkan nilai antara 0 sampai dengan 1 pada setiap informasi untuk mewakili ketidakpastian tersebut. Kemudian dengan teori probabilitas digunakan untuk mengukur derajat kebenarannya [9].

Logika *fuzzy* bekerja dengan cara memetakan sejumlah kondisi yang menjadi masukkannya dan mengubahnya menjadi suatu keluaran berupa langkah eksekusi yang harus dilakukan. Proses pemetaan ini menggunakan pernyataan *if-then* yang disebut dengan aturan. Aturan tersebut merujuk pada nilai kondisi masukkannya, dalam hal ini ialah suatu variabel yang mewakili sebagai label atau kategori tertentu. Kemudian setiap aturan akan dievaluasi secara paralel dan tidak memperhatikan urutannya.

Sistem logika *fuzzy* (FLS) atau yang biasa disebut dengan sistem *fuzzy* atau *fuzzy logic controller* ini umumnya terdiri dari sejumlah proses, yaitu fuzzifikasi, aturan atau *rulebase*, dan defuzzifikasi [10].

Metode *fuzzy* ini kemudian menjadi populer dan banyak diimplementasikan di negara Jepang dengan salah satu produk pertama mereka yang menggunakan metode *fuzzy* yaitu mesin cuci otomatis pada tahun 1990.

2.6.1 Himpunan *Fuzzy*

Himpunan *fuzzy* atau dikenal dengan nama himpunan tegas (*crisp*) dapat diartikan sebagai suatu pelabelan dari suatu kondisi dengan nilai batasan yang samar. Pada himpunan tegas, nilai keanggotaan suatu item x dalam suatu himpunan A, yang sering ditulis dengan $\mu_A[x]$, memiliki 2 kemungkinan, yaitu :

- Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
- Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan tersebut.

Pada himpunan *fuzzy* setiap keanggotaan diberi nilai antara 0 sampai dengan 1 untuk mengukur derajat kebenarannya. Himpunan *fuzzy* memiliki 2 atribut, yaitu :

a. *Linguistik*
   Penamaan dari suatu label yang mewakili suatu kondisi tertentu dengan menggunakan bahasa yang mudah dipahami, seperti : Panas-Sejuk-Dingin dan Muda-Parobaya-Tua.
b. *Numeris*
   Penamaan dari suatu himpunan yang mewakili dari sekumpulan angka dengan nilai tertentu, seperti : Himpunan

S = {0, 1, 2, 3, 4} dan himpunan bilangan prima P = {2, 3, 5, 7}.

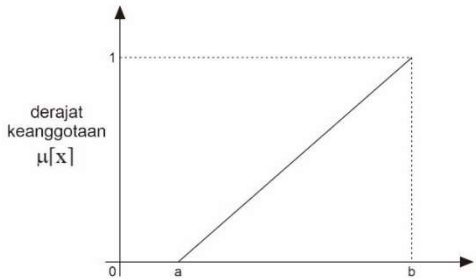
2.6.2 Fungsi Keanggotaan

Fungsi Keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai dengan 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi [11]. Ada beberapa fungsi yang bisa digunakan.

a. Representasi Linear

Pada representasi linear, pemetaan *input* ke derajat keanggotaannya digambarkan sebagai sebuah garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas.

Ada 2 keadaan himpunan *fuzzy* yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan yang lebih tinggi.

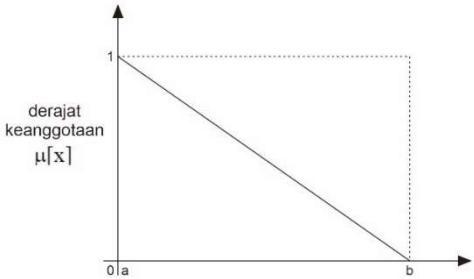


Gambar 2. 6 Representasi Linear Naik

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \\ (x-a)/(b-a)\,; & a \leq x \leq b \\ 1; & x \geq b \end{cases} \qquad (2.13)$$

Kedua, merupakan kebalikan yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah



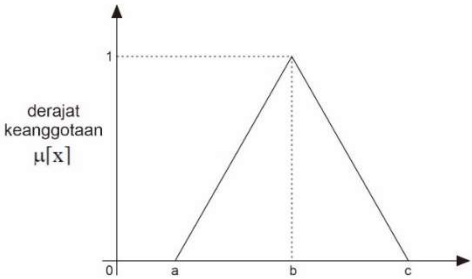
Gambar 2. 7 Representasi Linear Turun

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} (b-x)/(b-a); & a \leq x \leq b \\ 0; & x \geq b \end{cases} \qquad (2.14)$$

b. Representasi Kurva Segitiga

Kurva Segitiga pada dasarnya merupakan gabungan antara 2 garis (linear) seperti terlihat pada gambar di bawah ini.



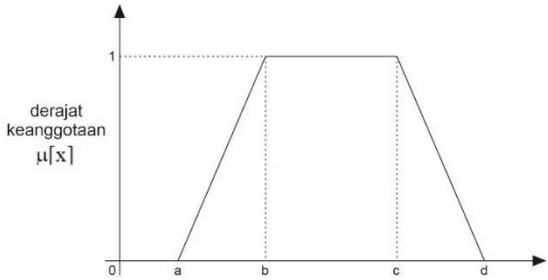
Gambar 2. 8 Kurva Segitiga

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \ atau \ x \geq c \\ (x-a)/(b-a); & a \leq x \leq b \\ (b-x)/(c-b); & b \leq x \leq c \end{cases} \quad (2.15)$$

c. Representasi Kurva Trapesium

Kurva Trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1.



Gambar 2. 9 Kurva Trapesium

Fungsi Keanggotaan :

$$\mu[z] = \begin{cases} 0; & x \leq a \ atau \ x \geq d \\ (x-a)/(b-a); & a \leq x \leq b \\ 1; & b \leq x \leq c \\ (d-x)/(d-c); & c \leq x \leq d \end{cases} \quad (2.16)$$

2.6.3 Operasi Himpunan *Fuzzy*

Ada beberapa operasi yang didefinisikan secara khusus untuk mengombinasi dan memodifikasi himpunan *fuzzy*. Nilai keanggotaan sebagai hasil dan operasi 2 himpunan sering dikenal dengan nama *fire strength* atau α-predikat. Ada 3 operator dasar yang diciptakan oleh Zadeh, yaitu :

a. Operator AND

Operator ini berhubungan dengan operasi interseksi pada himpunan. α-predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan

terkecil antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cap B} = \min(\mu_A[x], \mu_B[y]) \tag{2.16}$$

b. Operator OR

Operator ini berhubungan dengan operasi union pada himpunan. α-predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cap B} = \max(\mu_A[x], \mu_B[y]) \tag{2.17}$$

c. Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan. α-predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangkan nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

$$\mu_A = 1 - \mu_A[x] \tag{2.18}$$

2.6.4 Penalaran Monoton

Metode penalaran secara monoton digunakan sebagai dasar untuk teknik implikasi *fuzzy*. Meskipun penalaran ini sudah jarang sekali digunakan, namun terkadang masih digunakan untuk penskalaan *fuzzy*. Jika 2 daerah *fuzzy* direlasikan dengan implikasi sederhana sebagai berikut :

IF x is A THEN y is B

Transfer fungsi :

$$y = f((x, A), B) \tag{2.19}$$

maka, sistem *fuzzy* dapat berjalan tanpa harus melakukan komposisi dan dekomposisi *fuzzy*. Nilai *output* dapat diestimasi secara langsung dari nilai keanggotaan yang berhubungan dengan antesedennya.

2.6.5 Fungsi Implikasi

Tiap-tiap aturan (proposisi) pada basis pengetahuan *fuzzy* akan berhubungan dengan suatu relasi *fuzzy*. Bentuk umum dari aturan yang digunakan dalam fungsi implikasi adalah :

```
IF x is A THEN y is B
```

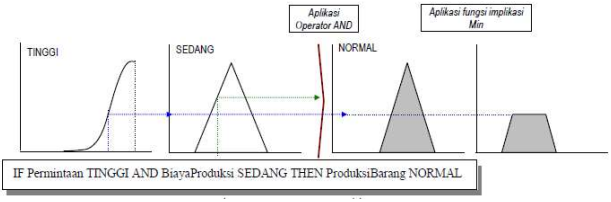
dengan x dan y adalah skalar, dan A dan B adalah himpunan *fuzzy*. Proposisi yang mengikuti IF disebut sebagai anteseden, sedangkan proposisi yang mengikuti THEN disebut sebagai konsekuen. Proposisi ini dapat diperluas dengan menggunakan operator *fuzzy*, seperti :

$$\text{IF } (x_1 \text{ is } A_1) \bullet (x_2 \text{ is } A_2) \bullet (x_3 \text{ is } A_3) \bullet \ldots\ldots \bullet (x_N \text{ is } A_N)$$
$$\text{THEN } y \text{ is } B$$

dengan • adalah operator (misal : OR atau AND).

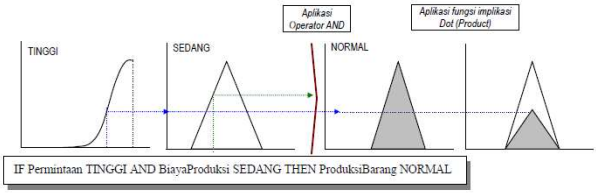
Secara umum, ada 2 fungsi implikasi yang dapat digunakan, yaitu :

a. Min (*minimum*). Fungsi ini akan memotong *output* himpunan *fuzzy*. **Gambar 2.10** menunjukkan salah satu contoh penggunaan fungsi min.



Gambar 2. 10 Fungsi Implikasi : MIN

b. Dot (*product*). Fungsi ini akan menskala *output* himpunan *fuzzy*. **Gambar 2.11** menunjukkan salah satu contoh penggunaan fungsi dot.



Gambar 2. 11 Fungsi Implikasi : DOT

2.6.6 Sistem Inferensi *Fuzzy*

Sistem inferensi *fuzzy* merupakan proses pengolahan data *crisp* yang masuk dan mengubahnya menjadi keluaran data dalam bentuk *crisp* juga. Dalam proses inferensi *fuzzy*, terdapat tiga tahapan utama yang harus dilalui, yaitu : fuzzifikasi, *rulebase* (aturan) dan defuzzifikasi.

a. *Fuzzifikasi*
   Tahapan fuzzifikasi merupakan sebuah proses mengubah data *crisp* pada masukkan menjadi sebuah nilai derajat keanggotaan pada himpunan *fuzzy* tertentu.
b. *Rulebase*
   Rulebase merupakan sebuah aturan-aturan yang dibuat sebagai dasar dalam pengolahan logika *fuzzy*. Nilai data *crisp* pada keluaran juga ditentukan dari aturan-aturan yang telah dibuat.
c. *Defuzzifikasi*
   Tahapan defuzzifikasi merupakan sebuah proses yang berbalikan dengan proses fuzzifikasi. Pada proses ini, nilai keluaran dari pengolahan *fuzzy* masih berupa derajat keanggotaan sehingga masih perlu diubah kembali menjadi berupa data *crisp*.

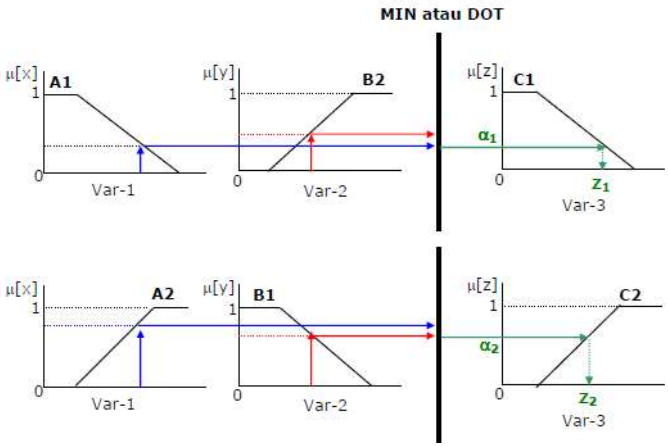
Terdapat tiga metode dalam sistem inferensi *fuzzy*, yaitu : Metode Tsukamoto, Mamdani dan Sugeno.

- Metode Tsukamoto

  Pada Metode Tsukamoto, setiap konsekuen pada aturan yang berbentuk *if-then* harus direpresentasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan monoton (**Gambar 2.12**). Sebagai hasilnya, hasil keluaran inferensi dari tiap-tiap aturan diberikan secara tegas (*crisp*) berdasarkan α-predikat (*fire strength*). Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot.

  Nilai rata-rata terbobot :

$$Z = \frac{\alpha_1 . z_1 + \alpha_2 . z_2}{\alpha_1 + \alpha_2} \qquad (2.20)$$



Gambar 2. 12 Inferensi Menggunakan Metode Tsukamoto

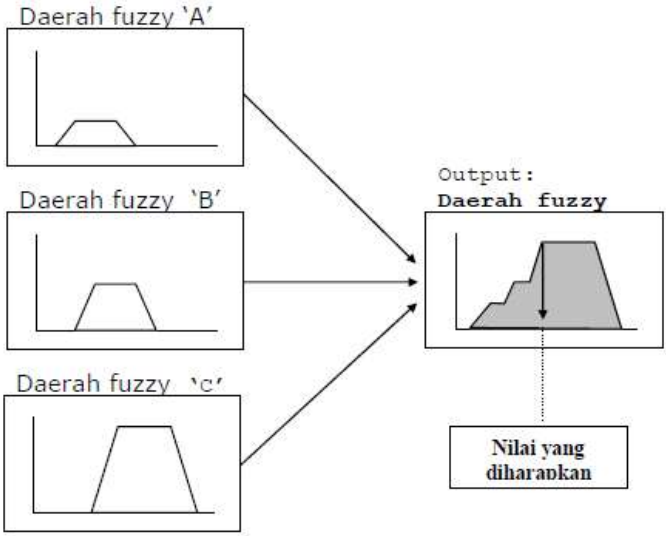
- Metode Mamdani

Metode Mamdani sering juga dikenal dengan nama Metode Max-Min. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Untuk mendapatkan *output*, diperlukan 4 tahapan :

1. Pembentukan himpunan *fuzzy*
   Pada metode ini, baik variabel *input* maupun *output* dibagi menjadi satu atau lebih himpunan *fuzzy*.
2. Aplikasi fungsi implikasi
   Pada metode ini, fungsi implikasi yang digunakan adalah Min.
3. Komposisi aturan
   Tidak seperti penalaran monoton, apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan korelasi antar aturan. Ada 3 metode yang digunakan dalam melakukan inferensi sistem *fuzzy*, yaitu : max, additive dan probabilistik OR (probor).
4. Penegasan (*defuzzy*)
   Input dari proses defuzzifikasi adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan output yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy* tersebut. Sehingga jika diberikan suatu himpunan *fuzzy* dalam *range* tertentu, maka harus dapat

diambil suatu nilai *crisp* tertentu sebagai *output* seperti terlihat pada **gambar 2.13**.

Ada beberapa metode defuzzifikasi pada komposisi aturan Mamdani, antara lain yaitu Metode *Centroid*, Metode *Bisektor*, Metode *Mean of Maximum* (MOM), Metode *Largest of Maximum* (LOM) dan Metode *Smallest of Maximum* (SOM).



Gambar 2. 13 Proses Defuzzifikasi

- Metode Sugeno

Penalaran dengan metode Sugeno hampir sama dengan penalaran Mamdani, hanya saja *output* (konsekuen) sistem tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan linear. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985.

1. Model *Fuzzy* Sugeno Orde-Nol

   Secara umum bentuk model *fuzzy* Sugeno Orde-Nol adalah :

   IF (x_1 is A_1)•(x_2 is A_2)•(x_3 is A_3)• ..... •(x_N is A_N)
   THEN z=k

   Dengan A_i adalah himpunan *fuzzy* ke-i sebagai anteseden, dan k adalah suatu konstanta (tegas) sebagai konsekuen.

2. Model *Fuzzy* Sugeno Orde-Satu

   Secara umum bentuk model *fuzzy* Sugeno Orde-Satu adalah :

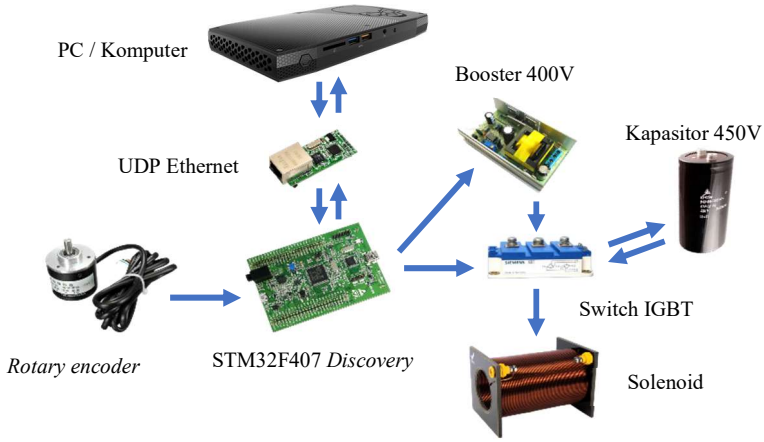
   IF (x_1 is A_1)• ..... •(x_N is A_N) THEN z= $\mathbf{p_1 * x_1 + \ldots + p_N * x_N + q}$

   Dengan A_i adalah himpunan *fuzzy* ke-i sebagai anteseden, dan p_i adalah suatu konstanta (tegas) ke-i dan q juga merupakan konstanta dalam konsekuen.

BAB III
PERANCANGAN SISTEM

Pada bab ini akan dijelaskan tentang perancangan sistem secara menyeluruh. Tujuan dari perancangan sistem ini adalah agar robot, khususnya robot sepak bola beroda, dapat melakukan eksekusi tendangan bola dengan optimal baik secara mekanik, elektronik maupun algoritma sistem. Eksekusi tendangan yang dilakukan oleh robot dapat dikatakan optimal apabila hasil tendangannya telah sesuai dengan target yang dituju, baik itu jarak target maupun ketinggian target.

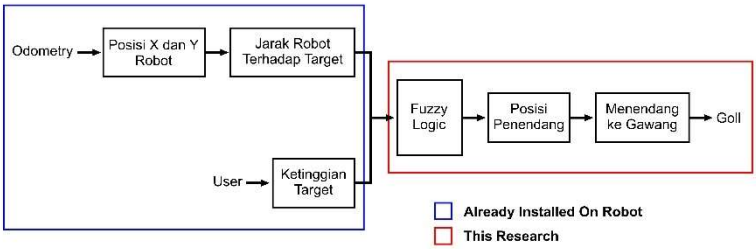
Sistem ini dirancang menggunakan *rotary encoder* sebagai pembacaan nilai posisi robot, solenoid sebagai aktuator tendangannya, STM32F407 *Discovery* sebagai *microcontroller*-nya, komputer/PC sebagai basis pemrogramannya dan Logika *Fuzzy* sebagai *controller* tendangannya. Seluruh komponen pada sistem diatur melalui *microcontroller* yang kemudian datanya dikirim melalui komunikasi serial kepada PC untuk dilakukan komputasi logika *fuzzy*. Skema sistem dalam penelitian ini dapat dilihat pada **Gambar 3.1**.



Gambar 3. 1 Skema Sistem Keseluruhan

3.1 Diagram Blok Sistem

Perancangan sistem ini terdiri dari dua jenis perangkat yang digunakan, yaitu perangkat keras dan perangkat lunak. Perangkat keras yang digunakan merupakan seluruh komponen yang dibutuhkan untuk menyusun sistem ini, seperti PC Komputer, *Microcontroller* STM32F407 *Discovery*, *Rotary encoder* dan Solenoid. Sementara perangkat lunak yang digunakan adalah algoritma dari sistem kontrol tendangan, dalam hal ini adalah kontroller logika *fuzzy*. Berikut merupakan blok diagram dari sistem.



Gambar 3. 2 Diagram Blok Sistem

Berdasarkan diagram blok pada **Gambar 3.2** di atas, dapat dilihat bahwa penelitian ini berfokus pada perancangan mekanisme penendang dan *controller* tendangannya yang menggunakan metode *fuzzy logic* (kotak warna merah). Sementara untuk pembacaan posisi yang menggunakan sistem *odometry* sudah ada pada robot (kotak warna biru). Berikut merupakan penjelasan dari blok diagram sistem :

- *Odometry*
  Odometry berperan untuk membaca posisi robot yang bergerak secara terus menerus. Dengan digunakannya *odometry* diharapkan dapat melacak dan mengetahui posisi koordinat robot secara *real-time*.
- Jarak Target.
  Nilai jarak target berperan sebagai nilai *input* untuk sistem kontroler logika *fuzzy*. Nilai jarak target didapatkan dengan melakukan perhitungan jarak antara koordinat XY posisi robot dengan koordinat XY posisi target. Pada penelitian ini target telah ditentukan yaitu sebagai titik tengah gawang lawan dengan koordinat XY (400, 1200). Perhitungan jarak menggunakan rumus

persamaan jarak antara dua titik, sehingga didapatkan nilai jarak terdekatnya.

$$r = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} \tag{3.1}$$

di mana,

r = Jarak terdekat antara robot dan target
x_1 = Koordinat X target
y_1 = Koordinat Y target
x_0 = Koordinat X robot
y_0 = Koordinat Y robot

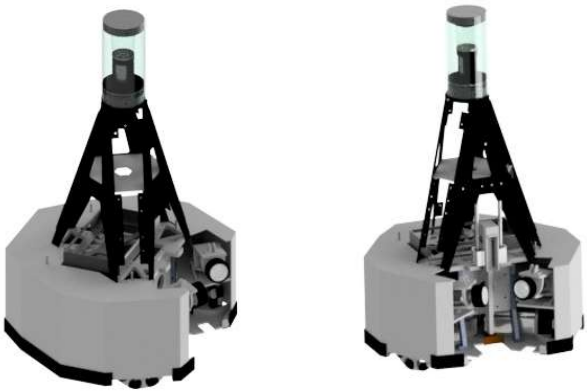
- Ketinggian Target
  Ketinggian target juga berperan sebagai nilai *input* untuk sistem kontroler logika *fuzzy*. Namun pada penelitian ini telah diberi batasan bahwa untuk nilai ketinggian target didapatkan dari *user* yang memasukkan nilainya secara manual. Nilai ketinggian target juga dibatasi dari 0 – 100 cm, hal ini disesuaikan dengan ketinggian gawang yang tertera pada aturan Kontes Robot Sepak Bola Beroda Indonesia.
- Sistem Logika *Fuzzy*
  Sistem logika *fuzzy* pada sistem ini berperan sebagai kontroler utama dari sistem. Logika *fuzzy* bekerja dengan menerima *input* berupa nilai jarak target dan ketinggian target, kemudian keluarannya berupa nilai pulsa yang akan dibangkitkan oleh *microcontroller* sebagai pemicu untuk melakukan tendangan.

3.2 Perancangan Mekanik Robot

Pada Subbab ini akan dijelaskan tentang perancangan mekanik robot secara menyeluruh mulai dari desain dan perancangannya sampai dengan pengimplementasiannya pada robot. Perancangan mekanik robot yang akan dijelaskan merupakan hal-hal yang berhubungan dengan penelitian ini, yaitu mulai dari desain keseluruhan robot, desain dan penempatan *odometry* serta perancangan dan penempatan mekanisme penendang menggunakan solenoid. Selain ketiga hal tersebut, juga akan sedikit dibahas tentang bentuk robot, sistem kendali bola pada robot dan komponen-komponen lainnya yang mendukung berjalannya sistem ini.

3.2.1 Desain Keseluruhan Robot

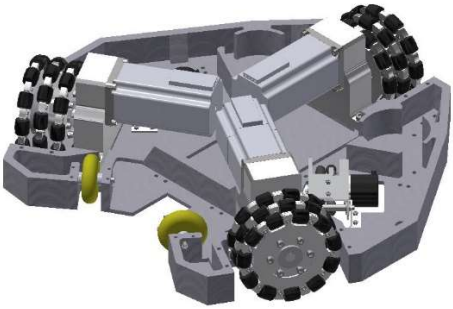
Badan robot didesain menggunakan bahan dasar aluminium lempengan yang dibentuk dan disusun sedemikian rupa sehingga robot dapat bekerja dengan efektif. Ukuran dan bentuk robot disesuaikan dengan peraturan yang tertera pada Kontes Robot Indonesia divisi Sepak Bola Beroda dan kompetisi Internasional RoboCup *Middle Size League* (MSL), yang mana dimensi maksimal untuk robot penyerang adalah 52 x 52 x 80 cm dengan bobot maksimal adalah 40 Kg.



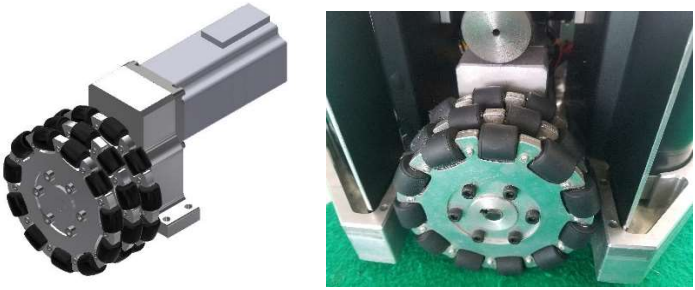
Gambar 3. 3 Desain 3D Robot

Robot yang digunakan dalam penelitian ini merupakan robot generasi ketiga dari tim robot Iris ITS. Robot ini memiliki bentuk segitiga pada bagian bawah dan mengerucut ke atas, sehingga memiliki bentuk yang ramping pada bagian atas robot. Robot ini bergerak dengan menggunakan motor 3WD yang dipasang pada *body* bagian bawah robot. Ketiga motor yang dipasang pada robot memiliki jarak 120 derajat antar motornya, seperti yang terlihat pada **Gambar 3.4**. Roda penggerak yang digunakan adalah jenis *swedish wheel* dengan sudut 90 derajat, atau lebih dikenal dengan nama *omniwheel* yang dirangkap tiga. Lihat **Gambar 3.5**. Hal ini bertujuan agar robot dapat bergerak bebas ke segala arah tanpa adanya selip. Sistem penggerak robot menjadi komponen

pendukung yang cukup penting karena akan menentukan kemampuan robot dalam mengejar dan menggiring bola.



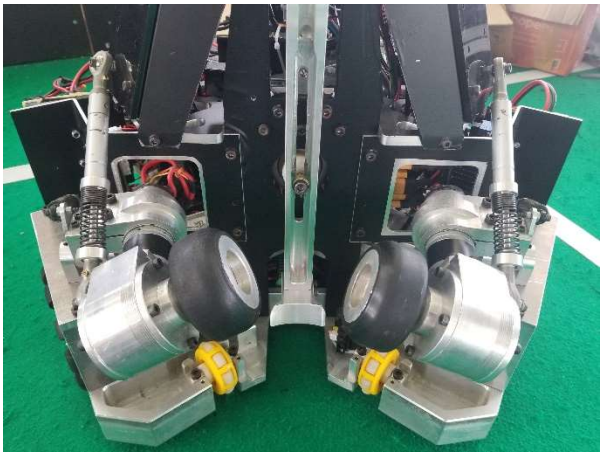
Gambar 3. 4 Desain 3D Base 1 Robot



Gambar 3. 5 Omniwheel Rangkap Tiga

Robot ini juga dilengkapi dengan sistem kendali bola (*ball handling system*) yang dipasang pada bagian depan robot sebelah kanan dan kiri. Sistem kendali bola pada robot terdiri dari 2 motor DC yang dilengkapi dengan roda karet. Fungsi dari sistem ini adalah agar robot dapat menarik, mengontrol dan menggiring bola seiring dengan pergerakan robot ke segala penjuru lapangan. Sistem kendali bola juga menjadi cukup penting dalam keseluruhan permainan, hal ini dikarenakan sistem ini yang akan menentukan dalam perolehan bola selama permainan berlangsung. Pengimplementasian sistem kendali bola pada robot dapat dilihat pada **Gambar 3.6**.

Selain sistem kendali bola, robot juga dilengkapi dengan mekanisme penendang yang diletakkan pada bagian depan-tengah robot (di antara kedua sistem kendali bola). Selanjutnya mekanisme ini akan dibahas lebih lanjut pada Subsubbab 3.2.3.



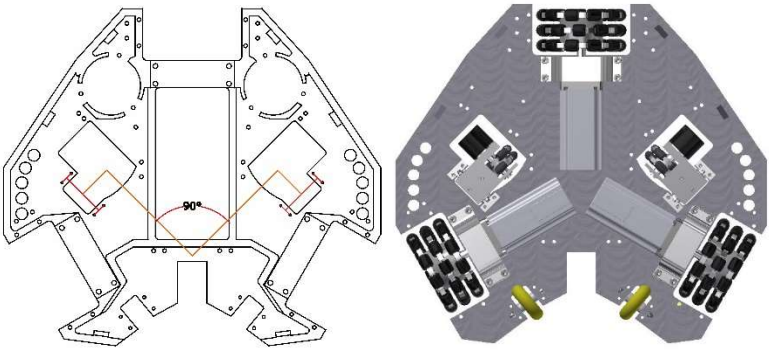
Gambar 3. 6 Sistem Kendali Bola

3.2.2 Desain dan Penempatan *Odometry*

Sistem *odometry* merupakan sistem yang biasa digunakan untuk mendeteksi atau melakukan pembacaan terhdap adanya perubahan sudut atau perputaran posisi shaft pada *encoder*. Umumnya sistem *odometry* menggunakan gabungan antara 2 sampai 3 *encoder* yang diposisikan dengan jarak tertentu antar *encoder*-nya. Hal ini bertujuan agar mendapatkan koordinat posisi yang optimal ketika robot bergerak, baik secara translasi maupun rotasi.

Sistem *odometry* digunakan pada robot agar dapat memetakan lapangan dalam koordinat XY, sehingga setiap pergerakan robot di dalam lapangan dapat terbaca dalam suatu nilai koordinat tertentu. Sistem yang diimplementasikan pada robot menggunakan gabungan 2 *rotary encoder* yang dipasang secara terpisah dan membentuk sudut 90 derajat. Lihat **Gambar 3.7**. Kedua *rotary encoder* tersebut dihubungkan dengan sebuah *omniwheel* kecil dengan ukuran diameter 58mm yang diletakkan pada bagian

bawah *body* robot. Setiap *rotary encoder* juga dipasang pegas dengan tujuan memastikan *omniwheel* selalu menyentuh permukaan lapangan agar meminimalkan terjadinya selip dan menghasilkan pembacaan koordinat posisi seakurat mungkin. Selain itu sistem *odometry* juga berguna sebagai *feedback* untuk kontrol pergerakan robot.



Gambar 3. 7 Desain Penempatan *Odometry*

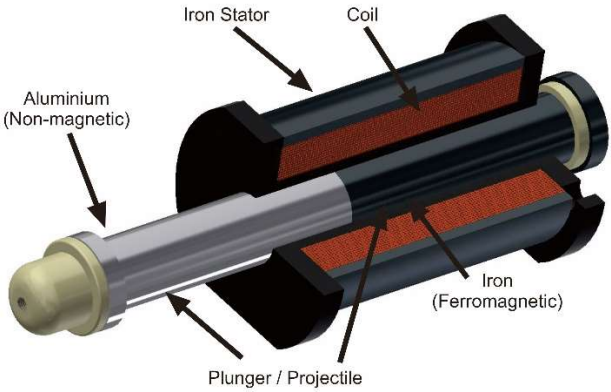
Pada penelitian ini, sistem *odometry* inilah yang akan digunakan untuk menentukan nilai jarak robot ke target.

3.2.3 Desain Mekanisme Penendang

Mekanisme penendang merupakan mekanisme yang memiliki peranan sangat penting dalam robot sepak bola beroda. Hal ini dikarenakan tim dapat dikatakan menang ketika tim tersebut mencetak gol lebih banyak daripada tim lawan. Gol atau bola masuk ke dalam gawang terjadi hanya diperbolehkan dari hasil tendangan yang dilakukan oleh robot, bola tidak boleh digiring sampai masuk ke dalam gawang. Maka dari itu mekanisme penendang berperan sangat penting dalam menentukan kemenangan suatu tim.

Mekanisme penendang yang digunakan dalam penelitian ini adalah solenoid. Solenoid bekerja sesuai dengan prinsip kerja medan magnet sehingga ketika dialiri arus dapat menghasilkan medan magnet yang cukup besar untuk menggerakkan proyektil yang berada di dalam solenoid tersebut. Proyektil digunakan berbentuk silinder pejal dengan diameter 30 mm dengan panjang proyektil adalah 248 mm. Proyektil terbagi menjadi dua bagian, yaitu

sepanjang 124 mm bagian depan merupakan bahan bersifat feromagnetik, pada penelitian ini menggunakan besi, dan sepanjang 124 mm sisanya merupakan bahan bersifat nonmagnetik, pada penelitian ini menggunakan aluminium. Lihat **Gambar 3.8**. Tujuan dari pembuatan proyektil yang terdiri dari jenis bahan adalah agar gaya yang dihasilkan oleh medan magnet lebih optimal.

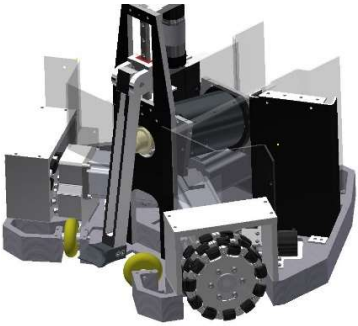


Gambar 3. 8 Ilustrasi Solenoid dan Proyektil

Dalam pengimplementasian solenoid sebagai mekanisme penendang, solenoid diletakkan pada bagian depan-tengah robot, pada bagian yang paling mudah untuk menjangkau bola dan melakukan tendangan. Eksekusi tendangan yang dilakukan tidak hanya dengan mengandalkan dari pergerakan proyektil saja, tetapi ujung dari proyektil dihubungkan pada sebuah lempengan aluminium yang cukup tebal dan dapat bergerak mengayun seperti 'kaki', sehingga gaya yang diberikan oleh proyektil dapat tersalurkan pada bola dengan lebih optimal. Lihat **Gambar 3.9**. Ilustrasi gerakan menendang pada robot dapat dilihat pada **Gambar 3.10**. Kemudian 'kaki penendang' tersebut dihubungkan dengan motor DC menggunakan *gear* ber-*ratio* 1:1. Pergerakan motor DC tersebut dikontrol melalui *microcontroller* STM32F407 *Discovery*, sehingga 'kaki penendang' dapat digerakkan naik-turun dengan bebas. Lihat **Gambar 3.11**. Tujuan dari dibuatnya mekanisme yang dapat bergerak naik turun ini adalah agar robot dapat mengatur

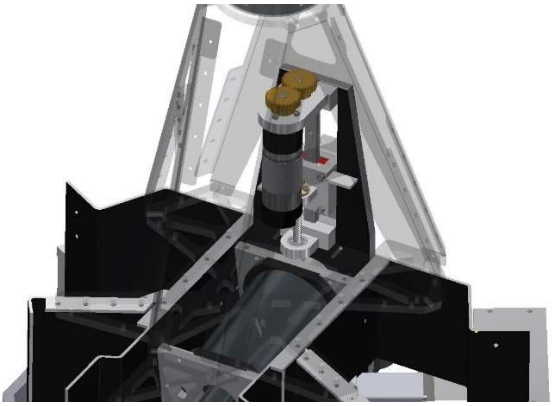
ketinggian tendangan sesuai dengan ketinggian target yang diinginkan. Semakin rendah posisi penendang, maka semakin besar sudut tendangan yang dihasilkan dan semakin tinggi bola akan melambung. Begitu juga sebaliknya, semakin tinggi posisi penendang, maka semakin kecil sudut tendangan yang dihasilkan dan rendah pula tinggi lambungan bola (semakin datar). Hal ini sesuai dengan hukum fisika pada teori gerak parabola.

Selanjutnya mekanisme naik-turun inilah yang akan dikontrol oleh sistem logika *fuzzy* dalam penelitian ini.



Gambar 3. 9 Mekanisme Penendang

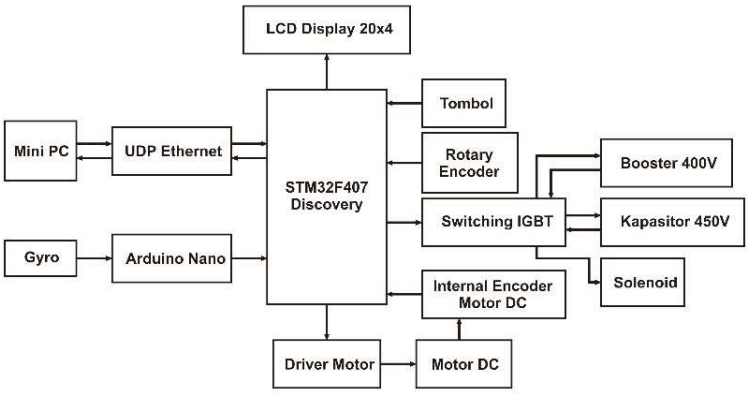
Gambar 3. 10 Ilustrasi Robot Menendang



Gambar 3. 11 Mekanisme Naik-Turun Penendang

3.3 Perancangan Elektronik Robot

Pada subbab ini akan dijelaskan tentang komponen elektronik yang digunakan pada robot, khususnya yang digunakan dalam penelitian ini, beserta dengan rangkaiannya. Berikut merupakan gambar diagram blok dari keseluruhan sistem elektronik.

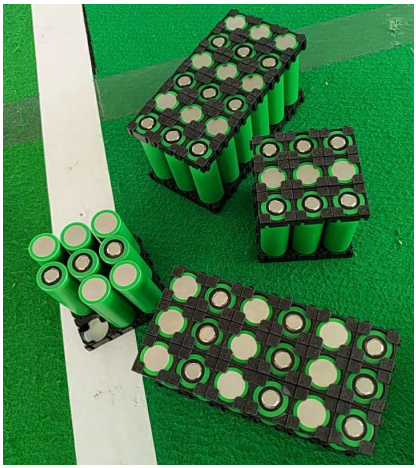


Gambar 3. 12 Diagram Blok Sistem Elektronik

Berikut merupakan penjelasan untuk beberapa komponen elektronik yang berperan penting dalam pelaksanaan penelitian ini.

3.3.1 Power Suplai

Setiap komponen yang digunakan pada robot memiliki spesifikasi khusus untuk nilai suplai yang digunakan agar komponen dapat digunakan secara efektif. Suplai yang digunakan robot adalah 2 baterai 5 sel berjenis LiFePo yang diparalel menjadi satu sehingga menghasilkan tegangan keluarannya bernilai 34V. Penggunaan jenis, jumlah dan kapasitas baterai dipilih berdasarkan pertimbangan kapasitas ruang yang tersedia pada robot, efektivitas kerja baterai, durasi penggunaan baterai dan nilai tegangan yang dibutuhkan oleh setiap komponen. Salah satu penyebab utamanya adalah karena motor penggerak utama dan *driver motor*-nya membutuhkan suplai 36V. Tetapi karena dalam penelitian ini tidak menggunakan motor penggerak utama, maka suplai baterai dapat digantikan dengan baterai 6 sel jenis LiIon dengan tegangan keluarannya 24V. Penyusunan modul baterai LiIon dapat dilihat pada **Gambar 3.13**.



Gambar 3. 13 Baterai LiIon

Kemudian, dari baterai LiIon 24V tersebut diparalel dan dihubungkan pada tiga buah *buck converter*, yang bernilai 19V untuk suplai PC, 12V untuk suplai *switch* IGBT dan *booster*, dan

terakhir 5V sebagai suplai *microcontroller*. Untuk motor DC yang menggerakkan penendang naik-turun beserta *driver motor*-nya mendapatkan suplai tegangan langsung dari baterai.

3.3.2 Sistem Penendang

Sistem elektronik pada mekanisme penendang melibatkan sejumlah komponen. Komponen dasar yang digunakan adalah *booster*, IGBT sebagai komponen *switching*-nya dan solenoid.

a. Rangkaian Penaik Tegangan (DC to DC *Booster*)

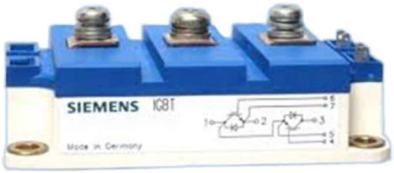
Rangkaian penaik tegangan DC atau biasa disebut dengan *DC to DC booster* merupakan rangkaian elektronik yang digunakan untuk menghasilkan tegangan keluaran yang bernilai lebih besar daripada nilai tegangan masukannya [14]. Pada penelitian ini tegangan masukan *booster* berasal dari tegangan yang keluar dari *buck converter* senilai 12V, sementara nilai keluarannya diatur sebesar 400V. Nilai tegangan 400V inilah yang nantinya digunakan untuk mengisi kapasitor.

Pada penelitian ini *booster* yang digunakan adalah modul *booster* yang sudah jadi, dengan tegangan masukannya bernilai 12V dan keluaran tegangannya dapat diatur hingga mencapai 450V. Namun pada penelitian ini tegangan keluaran *booster* hanya dibatasi sampai 400V dengan alasan keamanan. Gambar modul *booster* dapat dilihat pada **Gambar 3.14**. Kemudian, keluaran dari *booster* dihubungkan dengan rangkaian *switching* IGBT yang juga dihubungkan secara paralel dengan kapasitor berkapasitas 450V. Hal ini bertujuan agar tegangan dari *booster* disimpan terlebih dahulu dalam sebuah kapasitor yang nantinya akan digunakan untuk mengaliri solenoid sehingga terjadi tendangan.

Gambar 3. 14 Modul *Booster* 450V

b. Rangkaian *switching* IGBT

Rangkaian *switching* IGBT merupakan raingkaian yang berperan sebagai perantara yang menghubungkan antara rangkaian penaik tegangan dan kapasitor. Selain itu rangkaian ini juga berperan sebagai saklar yang menghubungkan antara kapasitor dan solenoid. Pada penelitian ini, komponen yang digunakan untuk melakukan *switching* adalah MOSFET *Gate Drive Circuit* dari SIEMENS. Cara kerja MOSFET ini ialah sama dengan cara kerja MOSFET pada umumnya. MOSFET ini berfungsi sebagai saklar untuk mengalirkan tegangan yang disimpan oleh kapasitor ke solenoid, sehingga terjadi sebuah tendangan. MOSFET ini akan aktif ketika diberi sinyal PWM (*Pulse Width Modulation*) dari *microcontroller* STM32F407, sehingga tendangan hanya terjadi ketika sinyal dari STM32F407 aktif. Rangkaian *switching* IGBT dapat dilihat pada **Gambar 3.15**.



Gambar 3. 15 Rangkaian *Switching* IGBT

c. **Solenoid**

Pada bagian ini akan dijelaskan perancangan dan detail menyeluruh tentang solenoid yang digunakan dalam penelitian ini. Di sini penulis mengutip dari jurnal yang ditulis oleh B.P.T. van Goch dari University Technology of Eindhoven, dengan judul "*Optimizing a solenoid for a Robocup kicker*" [15]. Pada jurnal tersebut dijelaskan bahwa energi yang dibutuhkan bola untuk mencapai suatu nilai kecepatan tertentu dapat dihitung dari persamaan berikut.

$$E_{ball} = \frac{1}{2} \cdot m_{ball} \cdot {v_{ball}}^2 + \frac{1}{2} \cdot J_{ball} \cdot {\omega_{ball}}^2 \qquad (3.2)$$

di mana,

m_{ball} = massa bola (kg)

v_{ball} = kecepatan bola (m/s)
J_{ball} $= \frac{2}{3} \cdot m_{ball} \cdot {r_{ball}}^2$
ω_{ball} $= \frac{v_{ball}}{r_{ball}}$
r_{ball} = jari-jari bola

Nilai massa dan jari-jari bola dapat diketahui dari peraturan resmi Kontes Robot Sepak Bola Beroda Indonesia dan peraturan internasional RoboCup *Middle Size League* (MSL) yaitu.

m_{ball} = 0,45 kg
r_{ball} = 0,11 m

Maka, dari persamaan tersebut dapat diketahui nilai energi yang dibutuhkan atau dihasilkan oleh solenoid.

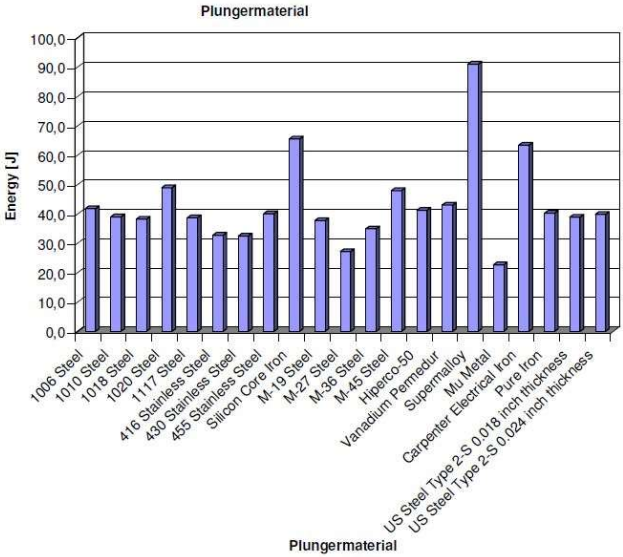
$$E_{solenoid} = E_{ball} \tag{3.3}$$

Setelah dilakukan sejumlah perhitungan, maka tahap selanjutnya adalah tahap perancangan solenoid. Solenoid terbagi menjadi tiga bagian penting, yaitu *plunger* (proyektil), *shielding* dan *coil* (kumparan). Berikut merupakan proses perancangan dari ketiga bagian penting dari solenoid.

- ***Plunger***

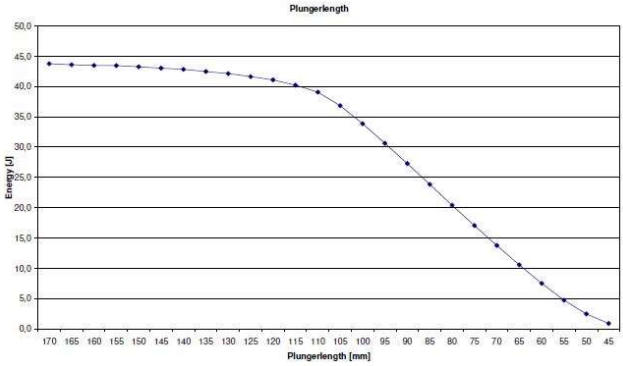
  Hal pertama yang perlu dilakukan adalah menentukan jenis bahan yang digunakan sebagai proyektil. Sebagaimana yang telah diketahui bahwa bahan yang digunakan sebagai proyektil haruslah material dengan sifat feromagnetik. Berdasarkan kutipan jurnal didapatkan grafik antara jenis material dari proyektil terhadap energi yang dihasilkan sebagai berikut.

  Dari data yang disajikan pada **Gambar 3.16** di bawah, dapat diketahui bahwa tiga jenis material yang menghasilkan energi terbesar secara berurutan adalah *Supermalloy*, *Silicon Core Iron* dan *Carpenter Electrical Iron*. Dikarenakan ketiga jenis material tersebut sangat susah untuk didapatkan, maka material terbaik yang memungkinkan untuk dipilih adalah material pada nomor urut ke-4 setelah *Carpenter Electrical Iron*, yaitu jenis *1020 Steel*. Material jenis *1020 Steel* ini lah yang akan digunakan dalam penelitian ini.



Gambar 3. 16 *Plunger Material* [15, p21]

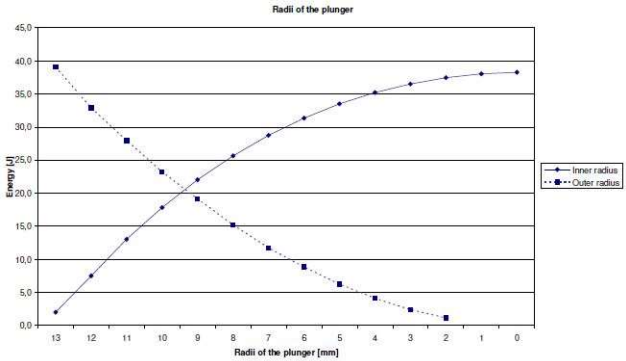
Selanjutnya adalah menentukan ukuran panjang dan diameter dari proyektil yang akan digunakan. Berdasarkan kutipan jurnal maka didapatkan grafik antara panjang proyektil terhadap jumlah energi yang dihasilkan sebagai berikut.



Gambar 3. 17 Grafik *Plunger Length* [15, p19]

Dari data di atas dapat disimpulkan bahwa semakin panjang proyektil maka akan menyerap energi lebih banyak daripada proyektil yang berukuran pendek. Dari data tersebut juga menunjukkan bahwa grafik menjadi *asymptotic* saat proyektil memiliki panjang lebih dari 110 mm. Hal ini dikarenakan di atas nilai tertentu, sisa panjang proyektil bergerak pada daerah yang cukup jauh dari kumparan, di mana medan magnet di area tersebut hampir bernilai 0 dan membutuhkan sedikit ekstra energi untuk dapat menggerakkan proyektil di area tersebut. Maka dari itu, dalam penelitian ini menggunakan proyektil dengan panjang 124 mm. Tapi, dikarenakan 124 mm terlalu pendek untuk mencapai 'kaki penendang', maka proyektil diperpanjang dengan menambahkan 124 mm aluminium (non-magnetik) pada salah satu ujungnya. Sehingga total panjang proyektil menjadi 248 mm. Gambar proyektil dapat dilihat pada **Gambar 3.8**.

Kemudian berikutnya adalah menganalisis ukuran diameter dari proyektil. Berdasarkan kutipan jurnal didapatkan grafik perbandingan antara radius dalam dan luar proyektil terhadap jumlah energi yang dihasilkan sebagai berikut.



Gambar 3. 18 Grafik Diameter Proyektil [15, p20]

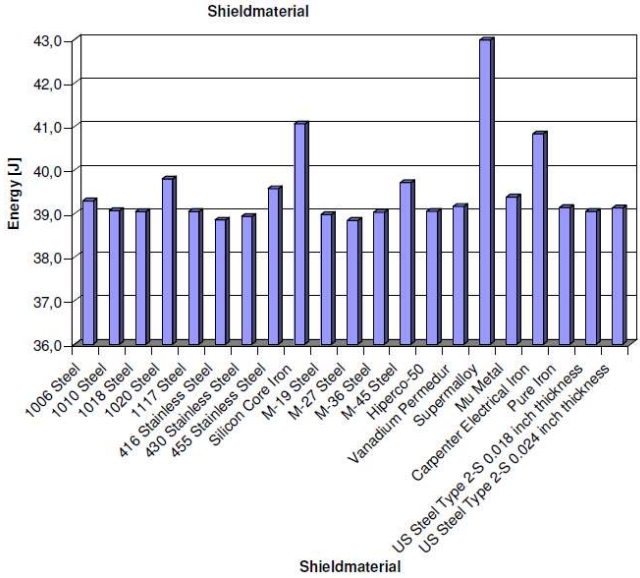
Dari data di atas dapat ditarik kesimpulan bahwa energi yang dihasilkan oleh besarnya radius luar akan berbanding terbalik dengan besarnya radius dalam. Hal ini berarti bahwa, radius dalam haruslah sekecil mungkin dan radius luar dapat berukuran besar. Pada penelitian ini ukuran radius yang

digunakan adalah 15 mm (30 mm diameter) untuk radius luar dan 1 mm (2 mm diameter) untuk radius dalam.

- ***Shielding***

Hal berikutnya yang perlu dipertimbangkan adalah masalah *shielding*. Hal ini menjadi cukup penting dikarenakan akan membahayakan sistem apabila terjadi kebocoran medan magnet dikarenakan dapat memengaruhi kinerja dan pembacaan dari komponen elektronik lainnya yang terpasang pada robot. Pada bagian ini hanya ada 2 hal yang perlu dipertimbangkan, yaitu bahan material dan ketebalannya.

Berdasarkan kutipan jurnal didapatkan bahwa pengujian jenis material *shielding* mirip dengan pengujian pada jenis material proyektil. Hal ini dikarenakan keduanya memiliki tujuan yang sama untuk menyerap energi sebanyak mungkin. Grafik antara jenis material *shielding* terhadap jumlah energi yang diserap adalah sebagai berikut.



Gambar 3. 19 Grafik *Shielding Material* [15, p23]

Dari data di atas dapat disimpulkan jenis material yang efektif untuk digunakan sebagai *shielding* adalah jenis material

yang sama seperti yang digunakan pada proyektil, yaitu 1020 Steel.

Selanjutnya adalah menentukan ketebalan dari *shield*. Untuk menganalisis membutuhkan teori reluktansi, yang mana ini mirip dengan teori pada kelistrikan. Dengan mengekuivalensikan resistansi dengan reluktansi ($\Re$), arus dengan flux (F), dan tegangan dengan *magneto motive force* (mmf), M. Maka didapatkan persamaan sebagai berikut.

$$I = \frac{V}{R} \cong F = \frac{M}{\Re} \tag{3.4}$$

Persamaan reluktansi didefinisikan sebagai berikut.

$$\Re = \frac{l}{A_c} . \frac{1}{\mu} \tag{3.5}$$

dengan l adalah panjang jalur medan magnet, A_c adalah area di mana medan magnet berada dan μ merupakan material parameter yang disebut dengan *permeability*. Dikarenakan *coil* tidak berubah saat ditambahkan *shield*, total reluktansi dalam sistem menjadi :

$$\Re = \Re_1 + \Re_2 + \Re_3 + \Re_4 \tag{3.6}$$

Berdasarkan hasil perhitungan dan pertimbangan dari jenis dan ukuran *coil* yang digunakan, maka ketebalan *shield* yang digunakan dalam penelitian ini adalah 5,2 mm mengelilingi *coil*.

- ***Coil***

Coil atau kumparan memiliki sejumlah parameter yang perlu diperhatikan, seperti jumlah lilitan, jumlah layer, panjang kawat dan ketebalan kawat. Berdasarkan parameter-parameter tersebut didapatkan sebuah persamaan untuk menghitung intensitas medan magnet di dalam kumparan (H) sebagai berikut.

$$H = \frac{NI}{4b} . \left[\frac{x+b}{\sqrt{(x+b)^2 + a^2}} - \frac{x-b}{\sqrt{(x-b)^2 + a^2}} \right] \tag{3.7}$$

Dengan N adalah jumlah lilitan, I adalah arus, b adalah setengah panjang dari kumparan, a adalah jari-jari dan x

adalah posisi dalam kumparan (pada hal ini $x = 0$ karena berada di tengah kumparan).

Berdasarkan persamaan tersebut didapatkan bahwa kumparan yang digunakan pada penelitian ini memiliki spesifikasi sebagai berikut.

Panjang kumparan	= 120 mm
Diameter dalam	= 40 mm
Diameter luar	= 56,8 mm
Tipe tembaga	= AWG 16 (1,29 mm)
Lilitan	= 1400
Layer	= 14

3.3.3 Human Machine Interface (HMI) pada Robot

Pada robot terdapat sebuah *LCD Display* berukuran 20x4 yang berfungsi untuk menampilkan nilai-nilai hasil perhitungan algoritma robot. Selain itu, *LCD* juga menampilkan sejumlah status dari algoritma yang sedang berjalan. Hal ini bertujuan untuk memastikan algoritma pada robot telah berfungsi dengan baik. Pada robot juga dilengkapi dengan lima buah tombol (*push button*) dan tiga buah *on/off button* yang digunakan untuk memberi perintah/kondisi tertentu secara manual pada robot atau juga dapat digunakan untuk mengatur parameter kontrol. Tampilan HMI pada robot dapat dilihat pada **Gambar 3.20**.

Gambar 3. 20 Tampilan HMI Pada Robot

3.4 Algoritma dan Sistem Perangkat Lunak

Pada subbab ini akan dijelaskan tentang pembuatan algoritma pada robot secara menyeluruh mulai dari algoritma pembacaan sistem *odometry*, algoritma sistem tendangan dan algoritma logika *fuzzy*.

3.4.1 Sistem Pembacaan *Odometry*

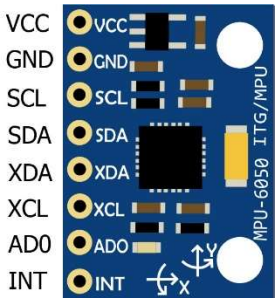
Sistem *odometry* yang digunakan adalah dua buah *rotary encoder* berjenis *incremental*. *Encoder* jenis ini bekerja dengan cara menghasilkan sinyal keluaran setiap kali porosnya berputar. Ketelitian dari sinyal keluaran bergantung pada resolusi sensor. *Rotary encoder* yang digunakan memiliki resolusi 360 PPR (*Pulse per Revolution*). Hal ini berarti bahwa *encoder* tersebut dapat menghasilkan 360 pulsa dalam 1 putaran penuh. Pada penelitian ini *encoder* yang dihubungkan pada *microcontroller* STM32F407, mengaktifkan kedua *timer* (T1 dan T2) yang terdapat pada *pin timer* STM32F407, dengan begitu kedua *timer* tersebut akan melakukan perhitungan pulsa secara bersamaan. Konfigurasi pembacaan sinyal pada *microcontroller* adalah menggunakan pembacaan perubahan sinyal *HIGH to LOW* dan *LOW to HIGH* (*CHANGE*), sehingga dalam satu kali keluaran sinyal pulsa dalam satu *timer* akan menghasilkan 2 sinyal/nilai. Dengan kata lain, ketelitian pada pembacaan sinyal *encoder* telah ditingkatkan menjadi 4 kali nilai semula, yaitu menjadi 1.440 sinyal pulsa dalam satu kali putaran.

Konfigurasi pemasangan kedua buah *rotary encoder* dipasang secara terpisah dengan berjarak 90 deajat antar *encoder*-nya. Sementara jarak roda *omniwheel* pada *encoder* terhadap titik pusat *odometry* adalah 164,4366 mm. Penggunaan kedua buah *rotary encoder* secara terpisah bertujuan untuk mendapatkan data posisi X dan Y seakurat mungkin. Sistem *odometry* ini hanya digunakan untuk mendapatkan data koordinat posisi X dan Y saja. Untuk data sudut/arah robot didapatkan dari sensor IMU yang telah terdapat giroskop dan akselerometer di dalamnya. Pada penelitian ini menggunakan MPU 6050, lihat **Gambar 3.21**. Sensor IMU tersebut berfungsi untuk menunjukkan kecepatan sudut robot terhadap porosnya dan menunjukkan akselerasi robot pada sumbu tertentu, lihat **Gambar 3.22**. Kemudian sensor tersebut dihubungkan dengan *microcontroller* Arduino Nano yang digunakan untuk menerima sekaligus mengolah data dari sensor. Kemudian data yang sudah jadi dalam berupa nilai sudut radian dan

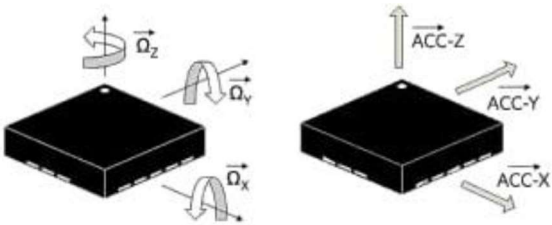
derajat dikirimkan melalui serial ke *microcontroller* STM32F407 untuk diolah lebih lanjut. Nilai data perubahan sudut tersebut digunakan sebagai orientasi sistem *odometry* dalam melakukan perhitungan saat robot melakukan pergerakan translasi sekaligus rotasi.

Selanjutnya dengan konfigurasi pemasangan *rotary encoder* yang telah disebutkan di atas, maka nilai posisi X dan Y dapat ditentukan melalui persamaan berikut.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos(\theta + 45°) & \cos(\theta + 135°) \\ \sin(\theta + 45°) & \sin(\theta + 135°) \end{bmatrix} \tag{3.8}$$



Gambar 3. 21 Sensor IMU MP-6050



Gambar 3. 22 Giroskop 3 Sumbu dan Akselerometer 3 Sumbu

3.4.2 Sistem Kontrol Penendang

Sistem penendang dikontrol melalui *microcontroller* STM32F407 berupa sinyal pulsa. Sinyal pulsa ini memicu untuk mengalirkan tegangan yang disimpan dalam kapasitor menuju

solenoid sehingga terjadi tendangan. Proses tendangan ini dieksekusi melalui perintah dari tombol pada HMI robot.

Saat tombol ditekan dan *microcontroller* memberikan instruksi 'tendang', sinyal pulsa pemicu tendangan tidak akan langsung dikeluarkan begitu saja. Tetapi *microcontroller* akan melakukan kontrol posisi penendang terlebih dahulu. Kontrol posisi penendang ini dilakukan dengan memberikan PWM pada *driver motor* BTN 7970 beserta direksi putaran motornya untuk menggerakkan motor yang terpasang pada sistem naik-turun penendang. Perputaran motor dibaca melalui *internal encoder* yang terdapat pada motor dan datanya dikirim ke *microcontroller* STM32F407 untuk mendapatkan data posisi penendang. Target nilai posisi penendang didapatkan dari hasil perhitungan logika *fuzzy* yang akan dijelaskan pada subbab berikutnya. Ketika posisi penendang telah sesuai, maka selanjutnya miktokontroler mengirimkan sinyal pulsa pemicu untuk melakukan eksekusi tendangan.

3.4.3 Algoritma Logika *Fuzzy*

Pada penelitian ini, sistem logika *fuzzy* berperan sebagai sistem yang melakukan kontrol terhadap jarak tendang bola. algoritma logika *fuzzy* ditempatkan pada microcontroller STMF407, hal ini bertujuan untuk memudahkan komputasi dan pengiriman data antara *microcontroller* dengan PC. *Software* yang digunakan adalah STM32 Cube MX dan Atollic TrueSTUDIO yang dirancang khusus sebagai *software* pemrograman *microcontroller* STM32.

Logika *fuzzy* ini menggunakan dua parameter sebagai masukannya, yaitu berupa data jarak robot terhadap target dan ketinggian target. Keluaran dari sistem ini berupa jarak tendang bola yang diharapkan sesuai degan target yang di-*input*-kan ke dalam sistem. Parameter jarak tendang bola yang diamati dalam penelitian ini adalah titik jatuh bola.

Sistem kontrol ini merupakan sistem yang bersifat *open-loop*, tidak ada *feedback* yang digunakan untuk mengoreksi nilai *error* yang terjadi. Sehingga, diharapkan sistem dapat menghasilkan nilai seakurat mungkin dengan melakukan *tuning* pada nilai-nilai *membership function* beserta *rule*-nya.

Nilai yang digunakan sebagai masukan sistem didapatkan dari pembacaan sistem *odometry* sehingga didapatkan koordinat

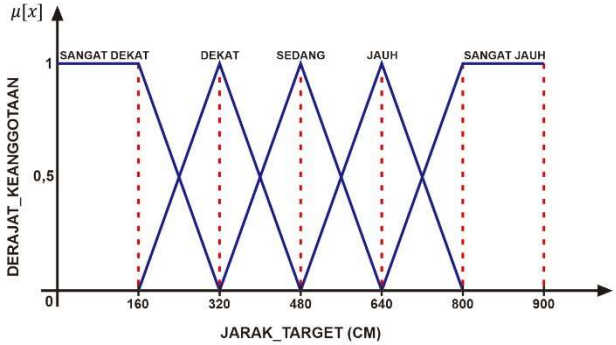
posisi robot saat ini. Kemudian dilakukan perhitungan jarak antara koordinat robot terhadap target. Dari persamaan tersebut didapatkan nilai jarak target yang akan digunakan sebagai masukan sistem. Sementara untuk nilai ketinggian target dimasukkan secara manual oleh *user*, sebagaimana yang telah dirumuskan pada batasan masalah penelitian ini.

Selanjutnya adalah memproses nilai masukan tersebut ke dalam sistem logika *fuzzy*. Proses yang dilakukan sistem dapat dilihat pada **Gambar 3.23**. Metode yang digunakan pada penelitian ini adalah metode Tsukamoto.

Gambar 3. 23 Diagram Blok Sistem Logika *Fuzzy*

a. Fuzzifikasi

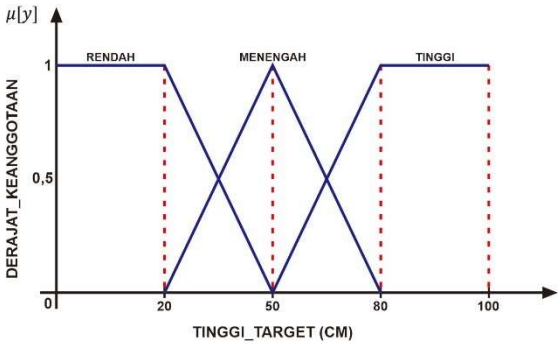
Proses fuzzifikasi merupakan tahapan untuk menentukan nilai fungsi keanggotaan dari setiap masukannya menjadi sebuah nilai 0 sampai dengan 1. Hal pertama yang perlu dilakukan ialah menentukan variabel dan jumlah himpunan *fuzzy* dari setiap variabel. Karena pada penelitian ini menggunakan dua jenis masukan, maka ada dua variabel dalam proses fuzzifikasi, yaitu variabel jarak dan tinggi.



Gambar 3. 24 Fungsi Keanggotaan Variabel Jarak

Selanjutnya adalah membagi jumlah kelas/himpunan *fuzzy* dari setiap variabel. Untuk variabel jarak, di sini peneliti mengacu pada ukuran lapangan yang tertera pada aturan Kontes Robot Sepak Bola Beroda Indonesia, yaitu berukuran 8 x 12 meter. Nilai ukuran lapangan tersebut digunakan sebagai batas jarak maksimal yang akan dimasukkan ke dalam sistem. Kemudian, untuk pembagian jumlah himpunan *fuzzy*, peneliti membagi variabel jarak menjadi lima kelas seperti yang terlihat pada **Gambar 3.24**. Namun, pada akhirnya nilai-nilai batas yang digunakan di setiap *membership function* didapatkan dari hasil konfigurasi sistem yang akan dibahas lebih lanjut pada Bab IV.

Sementara untuk variabel tinggi, peneliti mengacu pada ketinggian gawang yang tertera pada aturan KRSBI Beroda, yaitu 100 cm (1 meter). Nilai ketinggian gawang tersebut selanjutnya digunakan sebagai batas nilai maksimum pada masukan nilai variabel tinggi. Kemudian, untuk pembagian jumlah himpunan *fuzzy*-nya, peneliti membagi menjadi tiga kelas seperti yang terlihat pada **Gambar 3.25**. Sementara untuk nilai batas yang digunakan pada masing-masing himpunan *fuzzy* ditentukan berdasarkan toleransi ketinggian tendangan, yaitu sesuai dengan diameter bola $\pm$20 cm.



Gambar 3. 25 Fungsi Keanggotaan Variabel Tinggi

Setelah nilai dari variabel jarak dan tinggi di-*input*-kan ke dalam sistem, selanjutnya sistem akan melakukan komputasi dari nilai-nilai tersebut untuk didapatkan nilai derajat keanggotaannya pada setiap *membership function*. Nilai derajat keanggotaan atau

disebut juga α-predikat dari setiap *membership function* tersebut nantinya akan diteruskan ke proses selanjutnya, yaitu aturan *fuzzy*.

b. *Rule Fuzzy*

Setelah melalui proses fuzzifikasi, maka akan didapatkan nilai derajat keanggotaan atau α-predikat pada setiap *membership function*. Kemudian setiap nilai α-predikat dilakukan pengecekan terhadap aturan *fuzzy* yang telah dibuat. Kemudian, dua nilai α-predikat dari variabel jarak dan tinggi akan diambil nilai terkecilnya dan disimpan. Nilai α-predikat terkecil dari setiap kondisi ini yang nantinya akan digunakan pada proses defuzzifikasi untuk mendapatkan nilai keluaran sistem.

Aturan *fuzzy* ini memiliki tujuan untuk memutuskan jenis eksekusi yang akan dilakukan selanjutnya. Jenis eksekusi yang dilakukan sistem berupa nilai posisi penendang yang harus dicapai agar hasil tendangan dapat sesuai dengan target yang diinginkan. Nilai posisi penendang tersebut berupa nilai dalam satuan *encoder*. Tabel aturan *fuzzy* dapat dilihat pada **Tabel 3.1**.

Tabel 3. 1 Aturan Logika *Fuzzy*

Jarak / Tinggi	Sangat Dekat	Dekat	Sedang	Jauh	Sangat Jauh
Rendah	Sangat Pelan	Pelan	Sedang-1	Sedang-2	Kencang Sekali
Menengah	Sedang-1	Sedang-1	Sedang-2	Cukup Kencang	Kencang Sekali
Tinggi	Kencang	Sedang-2	Sedang-2	Sangat Kencang	Kencang Sekali

c. Defuzzifikasi

Proses selanjutnya adalah defuzzifikasi. Setelah didapatkan jenis eksekusi tendangan beserta nilai α-predikat minimal dari setiap kondisi, maka selanjutnya adalah mengubah nilai α-predikat menjadi nilai yang dapat dieksekusi oleh robot, yang mana adalah nilai keluaran *fuzzy*. Penentuan nilai keluaran ditentukan dari masing-masing nilai α-predikat dikalikan dengan nilai himpunan *fuzzy* keluarannya. Grafik himpunan *fuzzy* keluaran dapat dilihat pada **Gambar 3.26**.

Gambar 3. 26 Fungsi Keanggotaan Keluaran *Fuzzy*

Tahap selanjutnya adalah penentuan nilai keluaran *fuzzy*. Dikarenakan penelitian ini menggunakan metode Tsukamoto, maka nilai keluaran *fuzzy* ditentukan berdasarkan rata-rata dari nilai α-predikat. Berikut merupakan persamaan untuk menentukan nilai keluaran *fuzzy*.

$$z_{total} = \frac{\sum(\alpha_{prdikat_i} \cdot z_i)}{\sum \alpha_{prdikat_i}} \tag{3.8}$$

BAB IV
HASIL DAN ANALISA DATA

Pada bab ini akan dibahas tentang pengujian sistem, hasil pengujian beserta analisa hasil terhadap keseluruhan sistem yang telah dibuat. Pengujian sistem dilakukan secara bertahap, yaitu dengan menguji masing-masing sistem hingga sampai pengujian keseluruhan sistem. Pengujian ini dilakukan di ruang *workshop*, Gedung Pusat Robotika ITS.

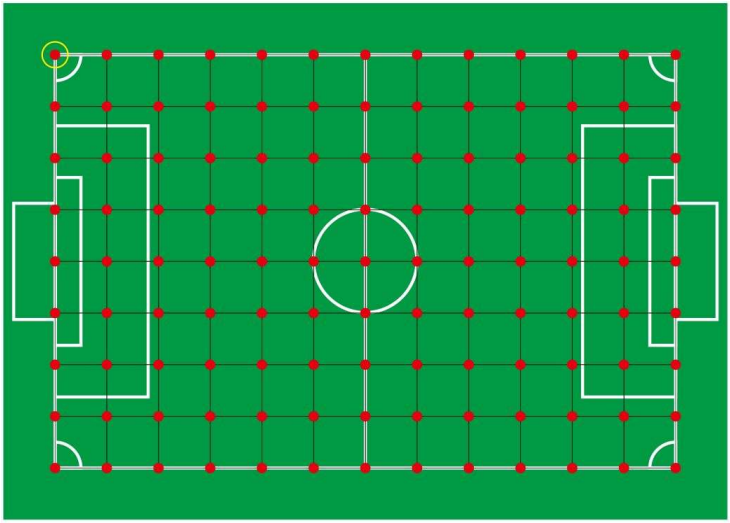
Gambar 4. 1 Robot dan Lokasi Pengujian

Pada **Gambar 4.1** menunjukkan bentuk robot (keseluruhan sistem) yang digunakan dalam penelitian ini beserta lapangan yang digunakan sebagai lokasi pengujian.

4.1 Pengujian Sistem *Odometry*

Pengujian sistem *odometry* dilakukan dengan tujuan untuk mengetahui tingkat akurasi dan nilai eror yang dihasilkan dari pembacaan *rotary encoder* terhadap pergerakan robot. Pertama, pengujian dilakukan dengan melakukan pengecekan terhadap perubahan nilai X, Y dan *tetha*

robot. Jika nilai dan arah perubahan telah sesuai maka dilakukan pengujian lebih lanjut. Pengujian lebih lanjut ini dilakukan dengan menggerakkan robot ke sejumlah titik yang telah dipetakan pada lapangan. Kemudian, membandingkan nilai pembacaan sistem pada titik-titik tersebut terhadap nilai posisi yang sesungguhnya. Pengujian ini dilakukan pada lapangan yang berukuran 8 x 12 meter, kemudian lapangan tersebut dibagi ke dalam bentuk plot berukuran 1 x 1 meter, sehingga terdapat 117 titik pengujian. Persebaran titik pengujian dapat dilihat pada **Gambar 4.2**.



Gambar 4. 2 Persebaran Titik Pengujian *Odometry*

Pengujian dilakukan sebanyak satu kali pada 117 titik tersebut. Sebelum dilakukan pengujian telah ditentukan sebuah titik yang digunakan sebagai *checkpoint* atau kalibrasi posisi robot. Titik kalibrasi tersebut berada pada koordinat (0,0), yaitu pada sisi ujung kiri belakang lapangan. Titik kalibrasi posisi dapat dilihat pada **Gambar 4.2**, titik tersebut diberi tanda lingkaran kuning. Proses pengujian dimulai dari titik koordinat (0,0) dan berjalan ke arah sumbu X sampai ke koordinat (800,0). Kemudian, setelah sumbu X selesai barulah robot bergerak ke arah sumbu Y mulai dari (0,100) sampai dengan koordinat (800,100). Setiap perpindahan posisi Y dilakukan kalibrasi lagi terhadap titik (0,0),

sehingga sistem dikalibrasi setiap pengambilan 9 data atau sebelum melakukan perpindahan posisi Y. Proses pengujian ini dilakukan hingga robot mencapai koordinat (800,1200), yaitu telah melakukan seluruh pengujian terhadap 117 titik.

Hasil dari pengujian ini dapat dilihat pada **Tabel 4.1** berikut.

Tabel 4. 1 Data RMSE Sistem *Odometry*

RMSE Sumbu-X	RMSE Sumbu-Y	RMSE Nilai Jarak
15,556	5,627	9,207
18,333	11,215	15,320
17,694	20,634	24,44
25,623	30,673	35,528
11,518	10,878	6,778
21,102	7,172	13,674
19,717	1,527	9,013
33,913	14,783	26,151
26,865	2,624	15,564
31,531	13,828	27,204
32,530	11,165	28,738
25,249	4,558	21,805
26,818	16,289	26,704
Rata-Rata RMSE		
23,5735	11,6137	20,1007

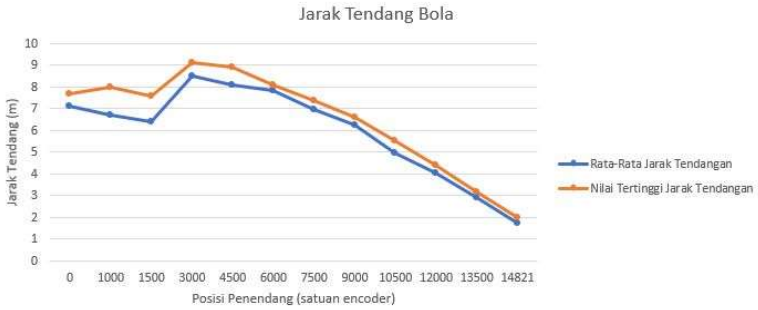
Pada data di atas menunjukkan hasil perhitungan RMSE (*Root Mean Square Error*) dari setiap data yang diambil dari keseluruhan 117 titik. Satu baris data RMSE di atas mewakili 9 data yang diambil dalam satu koordinat Y (satu kali kalibrasi), sehingga menghasilkan 13 data RMSE sebagai perwakilan dari 13 koordinat Y yang diuji (13 kali kalibrasi). Untuk data yang lebih lengkap terkait seluruh nilai pembacaan koordinat X, Y dan jarak hasil pembacaan sistem terhadap nilai asli yang terukur dapat dilihat pada **Lampiran 1**.

4.2 Pengujian Sistem Penendang

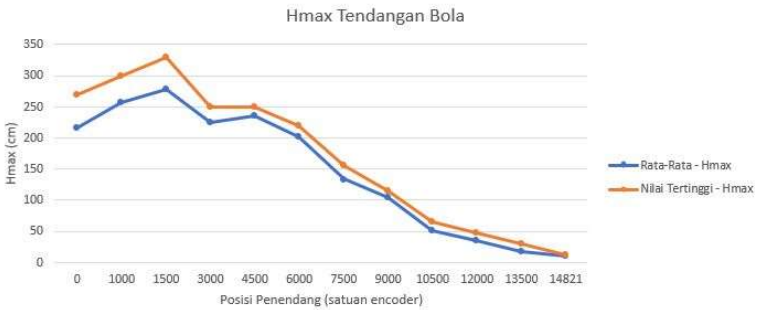
Pengujian sistem penendang dilakukan dengan tujuan untuk mengetahui batas kemampuan tendangan robot. Selain itu, pengujian ini digunakan untuk menentukan batas-batas nilai pada fungsi keanggotaan logika *fuzzy* serta untuk menentukan aturan *fuzzy* dan jumlah keluarannya.

Ada beberapa parameter yang akan diamati dalam pengujian ini, yaitu batas maksimum jarak tendang bola yang dapat dicapai, ketinggian maksimum bola, lebar pulsa yang digunakan dan posisi penendang.

Setelah dilakukan pengujian sebanyak 120 kali, yaitu 10 jenis pulsa yang digunakan antara 0 – 50 ms divariasikan dengan 12 macam posisi penendang, dapat disimpulkan bahwa lebar pulsa yang digunakan hampir tidak memengaruhi jarak tendang bola. Hal ini dikarenakan dari 10 jenis pulsa yang digunakan menghasilkan bola jatuh pada jarak yang relatif sama di setiap percobaannya (tidak ada variasi). Sementara perubahan jarak tendang bola terjadi karena dipengaruhi oleh posisi ketinggian penendang. Selain itu posisi ketinggian penendang juga memengaruhi ketinggian (Hmax) tendangan yang dihasilkan. Sehingga, posisi ketinggian penendang digunakan sebagai keluaran logika *fuzzy*, berperan melakukan kontrol terhadap jarak tendang bola.



Gambar 4. 3 Grafik Jarak Tendang Bola



Gambar 4. 4 Grafik Ketinggian Maksimum Tendangan

Hasil dari pengujian ini dapat dilihat pada **Gambar 4.3** dan **Gambar 4.4** yang menunjukkan hubungan antara nilai posisi penendang terhadap jarak tendang bola dan ketinggian tendangan (Hmax). Pada kedua grafik tersebut diperlihatkan, untuk kurva berwarna biru merupakan rata-rata dari 10 data yang diambil menggunakan pulsa antara 0 - 50 ms. Sementara kurva berwarna oranye merupakan nilai tertinggi yang diperoleh dari 10 data tersebut.

Dari kedua macam grafik di atas, grafik antara nilai tertinggi dengan nilai rata-rata memiliki bentuk yang cenderung stabil (sama), hanya saja nilai rata-rata lebih rendah daripada nilai tertinggi. Sehingga data rata-rata dapat digunakan sebagai acuan untuk proses dan pengujian selanjutnya. Hal ini dikarenakan nilai rata-rata cenderung lebih representatif daripada nilai tertinggi.

4.3 Pendefinisian dan Pengujian Sistem *Fuzzy Logic*

Setelah dilakukan pengujian terhadap sistem penendang dan telah diketahui sejauh apa kemampuan tendangan yang dapat dihasilkan, tahap selanjutnya adalah mendefinisikan nilai-nilai yang nantinya akan menjadi parameter dalam sistem logika *fuzzy*.

4.3.1 Pendefinisian Parameter Sistem

Tahapan awal adalah pendefinisian parameter sistem, yaitu proses penentuan nilai batas yang akan digunakan pada setiap himpunan *fuzzy*. Dari data yang disajikan pada **Gambar 4.3**, dapat diketahui bahwa jarak maksimum yang dapat dicapai bola hanya 9 meter. Sehingga nilai batas himpunan *fuzzy* dari variabel jarak ditentukan bernilai setiap kelipatan 160 cm. Hal ini dikarenakan terdapat 5 kelas dalam variabel jarak dan peneliti mengambil jarak maksimumnya, yaitu bernilai 8 meter sebagai nilai aman. Sementara untuk variabel tinggi, peneliti hanya membagi menjadi 3 kelas dengan masing-masing bernilai 20, 50 dan 80 cm. Nilai terendah 20 cm dan tertinggi 80 cm didapatkan dari diameter bola yang bernilai $\pm$20 cm dan ketinggian maksimal robot yang diizinkan bermain di lapangan adalah 80 cm.

Setelah nilai batas ditentukan, selanjutnya adalah menentukan aturan *fuzzy* dan jumlah kelas pada keluarannya beserta masing-masing nilainya. Penentuan ini dilakukan dengan cara melakukan pengujian tendangan pada sistem dan mencatat nilai posisi penendang pada setiap kondisi nilai jarak dan tinggi tertentu.

Nilai kondisi jarak dan tinggi didapatkan dari setiap *membership function* yang memiliki α-predikat bernilai 1. Hal ini dikarenakan nilai kondisi jarak dan tinggi pada setiap *membership function* yang memiliki α-predikat bernilai 1, dianggap telah merepresentasikan setiap kondisi jarak maupun tinggi secara keseluruhan.

Pengujian ini dilakukan dengan melakukan sebanyak 83 kali tendangan dengan menggunakan objek yang memiliki 3 macam ketinggian yang berbeda. Dari 83 kali percobaan didapatkan data sebagai berikut.

Tabel 4. 2 Pengujian Nilai Batas Himpunan *Fuzzy*

Ketinggian (cm)	Jarak (cm)	Posisi Penendang (satuan *encoder*)
20	160	13000
	320	12200
	480	10200
	640	8900
	800	5500
	900	4500
50	160	10200
	320	10200
	480	8800
	640	7750
	800	4500
	900	4500
80	160	8250
	320	9000
	480	8900
	640	6825
	800	4500
	900	4500

Dari data di atas terdapat 3 nilai yang berwarna merah, hal ini dikarenakan pada ketiga nilai tersebut, tendangan yang dihasilkan oleh sistem tidak mampu mencapai kondisi jarak maupun tinggi yang ditentukan. Maka dari itu untuk ketiga kondisi tersebut langsung diberi nilai tertinggi yaitu '4500'.

Selanjutnya nilai-nilai pada kolom '*Encoder*' yang ditunjukkan pada **Tabel 4.2** digunakan untuk menyusun aturan *fuzzy*. Hasil penyusunan aturan *fuzzy* dapat dilihat pada **Tabel 4.3**.

Tabel 4. 3 Penyusunan Aturan *Fuzzy*

	Sangat Dekat	Dekat	Sedang	Jauh	Sangat Jauh
Rendah	13000	12200	10200	8900	5500
Menengah	10200	10200	8800	7750	4500
Tinggi	8250	9000	8900	6825	4500

Dari data yang disajikan pada **Tabel 4.3** di atas, dapat dilihat bahwa aturan *fuzzy* yang dihasilkan memiliki terlalu banyak kelas/nilai. Maka dari itu dilakukan pemilahan dan pemangkasan pada nilai-nilai yang dianggap serupa, sehingga didapatkan aturan dengan 8 macam keluaran yang setiap kelasnya ditandai dengan warna yang berbeda-beda. Hasil dari pemilahan jumlah kelas beserta masing-masing nilainya akan digunakan untuk menentukan *membership function* dari keluaran sistem sebagaimana yang telah diperlihatkan pada Bab sebelumnya, yaitu pada **Tabel 3.1** dan **Gambar 3.26**.

4.3.2 Pengujian Sistem Logika *Fuzzy*

Tahap selanjutnya adalah pengujian sistem logika *fuzzy*. Sebelumnya telah dilakukan sejumlah konfigurasi mulai dari perancangan sistem *fuzzy*, pengambilan data tendangan hingga penyesuaian sistem *fuzzy* dengan data tendangan yang ada. Pada tahap kali ini akan dilakukan pengujian sistem *fuzzy* yang telah dibuat dengan menyesuaikan data tendangan yang dihasilkan. Hasil dari pengujian ini akan menentukan apakah sistem sudah berjalan sesuai dengan yang diinginkan atau perlu dilakukan *tuning* ulang terhadap sistem.

Pengujian ini dilakukan dengan membawa robot ke beberapa titik sampel dengan nilai jarak tertentu. Kemudian, nilai data tinggi dimasukkan secara manual melalui tombol yang ada pada HMI robot. Nilai keluaran *fuzzy* ditampilkan melalui *LCD Display* yang terpasang pada HMI robot. Kemudian nilai yang ditampilkan dari setiap pengujian yang dilakukan dicatat dan dibandingkan dengan nilai hasil konfigurasi sistem. Jika nilai telah dianggap sesuai, maka pengujian sistem dapat dilanjutkan ke tahap berikutnya, yaitu pengujian keseluruhan sistem.

Tabel 4. 4 Pengujian Sistem Logika *Fuzzy*

Jarak (cm)	Tinggi (cm)	Keluaran (satuan *encoder*)
400	0	11200
	35	10466
	65	9188
	80	8900
500	0	10037
	35	9409
	65	8577
	80	8640
600	0	9225
	35	8810
	65	7763
	80	7343
700	0	7625
	35	7180
	65	6468
	80	6328
800	0	5500
	35	5500
	65	5500
	80	5500

Dari hasil pengujian yang ditampilkan pada **Tabel 4.4** dapat dilihat bahwa sistem logika *fuzzy* dapat berfungsi dengan baik. Nilai keluaran yang ditampilkan juga sesuai dengan hasil konfigurasi.

4.4 Pengujian Keseluruhan Sistem

Setelah melakukan sejumlah pengujian terhadap masing-masing sistem, tahap selanjutnya adalah melakukan pengujian keseluruhan sistem. Pada pengujian ini akan melibatkan seluruh sistem mulai dari sistem *odometry*, sistem penendang dan sistem logika *fuzzy*. Pengujian keseluruhan sistem dilakukan dalam 2 tahapan, yaitu pengujian awal dan akhir.

4.4.1 Pengujian Awal

Sebelum dilakukannya pengambilan data untuk pengujian keseluruhan sistem, dilakukan sedikit percobaan untuk menguji apakah sistem telah layak untuk dilakukan uji coba secara menyeluruh atau masih perlu dilakukan konfigurasi ulang. Uji coba

ini dilakukan sebanyak 12 kali tendangan dengan jarak dan ketinggian target yang berbeda-beda. Hasil dari uji coba ini dapat dilihat pada tabel berikut.

Tabel 4. 5 Uji Coba Awal Sistem

Koordinat X	Koordinat Y	Tinggi (cm)	Keberhasilan
400	500	0	
		35	
		65	
		80	
	600	0	
		35	
		65	
		80	
	700	0	
		35	
		65	
		80	

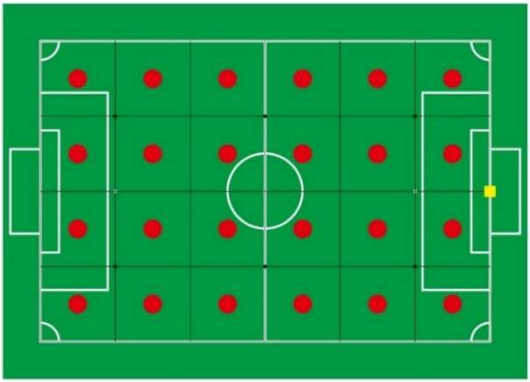
Dari hasil uji coba awal, di dapatkan bahwa hanya ada 1 yang gagal dan 11 lainnya berhasil. Kegagalan ini dikarenakan bola yang ditendang oleh robot mengenai mistar gawang sehingga tidak terjadi gol. Maka dari itu, dapat disimpulkan bahwa sistem telah berjalan cukup baik dan pengujian dapat dilanjutkan.

4.4.2 Pengujian Akhir

Pengujian akhir keseluruhan sistem dilakukan dengan membagi lapangan berukuran 8 x 12 meter menjadi sebuah plot dengan ukuran 2 x 2 meter, sehingga terdapat sejumlah 24 plot. Lihat **Gambar 4.5**. Kemudian dalam setiap plot akan dilakukan tendangan ke arah target, sehingga akan menghasilkan 24 tendangan dengan nilai jarak yang berbeda-beda. Selanjutnya, 24 macam nilai jarak tersebut divariasikan lagi dengan 4 macam nilai tinggi, sehingga menghasilkan 96 kali percobaan tendangan. Jumlah 24 macam nilai jarak dan 4 macam nilai tinggi digunakan sebagai perwakilan data dari setiap himpunan *fuzzy* variabel jarak dan tinggi.

Pengujian dilakukan dengan membawa robot pada titik dan kondisi yang ditentukan, sebagaimana yang telah dijelaskan sebelumnya, untuk melakukan 96 kali tendangan ke arah target.

Target yang digunakan adalah benda-benda yang memiliki tinggi sesuai dengan kriteria yang telah disebutkan sebelumnya. Kemudian, benda-benda tersebut diletakkan secara bergantian di titik tengah gawang lawan seolah-olah menjadi robot penjaga gawang lawan. Pada **Gambar 4.5** posisi benda ditempatkan pada kotak yang berwarna kuning.



Gambar 4. 5 Peta Persebaran Titik Pengujian

Tabel 4. 6 Hasil Pengujian Keseluruhan Sistem-1

Tinggi Target = 0 cm	
Persentase Keberhasilan	92%
Persentase Keakuratan	92%
Persentase Kegagalan	8%
Tinggi Target = 35 cm	
Persentase Keberhasilan	96%
Persentase Keakuratan	67%
Persentase Kegagalan	4%
Tinggi Target = 65 cm	
Persentase Keberhasilan	54%
Persentase Keakuratan	33%
Persentase Kegagalan	46%
Tinggi Target = 80 cm	
Persentase Keberhasilan	58%
Persentase Keakuratan	25%
Persentase Kegagalan	42%

Setelah dilakukan pengujian sebanyak 96 kali tendangan, didapatkan hasil yang kurang memuaskan. Hal ini dikarenakan masih banyak terjadinya kegagalan dan kecilnya tingkat keberhasilan dan akurasi dari sistem. Hasil dari pengujian ini dapat dilihat pada **Tabel 4.6**.

Dari hasil pengujian yang disajikan dalam **Tabel 4.6** dapat disimpulkan bahwa tingkat keberhasilan dan akurasi tendangan menurun cukup signifikan dengan seiring bertambahnya ketinggian target. Hal ini dibuktikan dengan kecilnya persentase keberhasilan pada tinggi target 65 dan 80 cm dan besarnya persentase kegagalannya. Tingginya nilai persentase kegagalan disebabkan karena banyak tendangan yang terlalu tinggi dan menyebabkan bola mengenai bagian atas mistar gawang atau bahkan melampaui ketinggian gawang sehingga tidak terjadi gol. Selain itu, untuk tendangan dengan jarak di atas 600 cm menghasilkan tendangan yang sudutnya terlalu besar, sehingga bola melambung terlalu tinggi dan saat bola menyentuh tanah gaya dorong terhadap sumbu X telah habis. Hal ini menyebabkan bola berhenti dan diam di tengah lapangan atau bola menggelinding ke arah yang tak menentu.

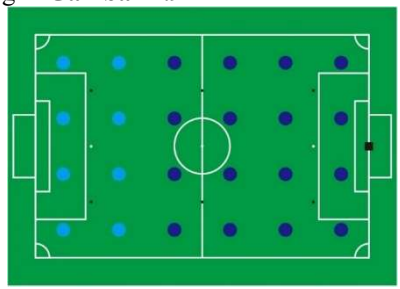
Penyebab dari hasil yang kurang memuaskan ini adalah pendefinisian nilai logika *fuzzy* yang digunakan hanya mengandalkan pendekatan regresi pada kurva hasil tendangan yang disajikan pada **Gambar 4.3** dan **Gambar 4.4**, sehingga hasilnya kurang merepresentasikan setiap kondisi jarak dan tinggi yang ada. Solusi dari permasalahan ini adalah melakukan konfigurasi ulang pada aturan *fuzzy* dan nilai-nilai yang digunakan sebagai parameter sistem logika *fuzzy*, sebagaimana yang telah dijelaskan pada Subbab 4.3 sebelumnya. Tujuan dari dilakukannya konfigurasi ulang adalah untuk mendapatkan nilai keluaran yang merepresentasikan dari setiap kondisi jarak maupun tinggi yang ada, sehingga dapat diperoleh hasil tendangan yang optimal.

Setelah dilakukan sejumlah konfigurasi pada aturan *fuzzy* dan nilai parameter sistem, pengujian kembali dilakukan dengan metode yang sama seperti sebelumnya. Penggunaan metode yang sama bertujuan untuk membandingkan hasil dari sebelum dan sesudah dilakukannya konfigurasi sistem. Hasil dari pengujian kedua dapat dilihat pada **Tabel 4.7**.

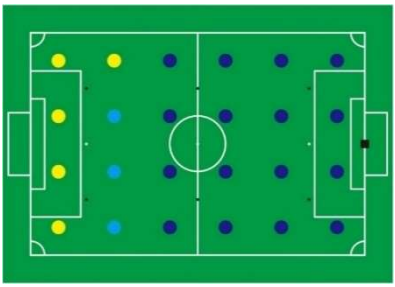
Tabel 4. 7 Hasil Pengujian Keseluruhan Sistem-2

Tinggi Target = 0 cm	
Persentase Keberhasilan	100%
Persentase Keakuratan	100%
Persentase Kegagalan	0%
Tinggi Target = 35 cm	
Persentase Keberhasilan	100%
Persentase Keakuratan	79%
Persentase Kegagalan	0%
Tinggi Target = 65 cm	
Persentase Keberhasilan	96%
Persentase Keakuratan	71%
Persentase Kegagalan	4%
Tinggi Target = 80 cm	
Persentase Keberhasilan	83%
Persentase Keakuratan	50%
Persentase Kegagalan	17%

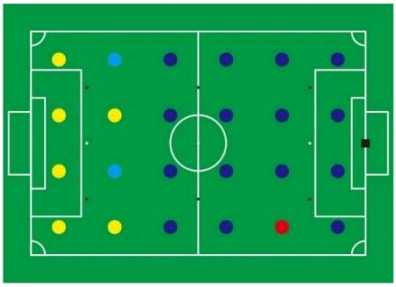
Dari hasil pengujian yang disajikan pada **Tabel 4.7**, menunjukkan terjadinya peningkatan yang cukup signifikan pada tingkat keberhasilan dan akurasi tendangan, khususnya pada ketinggian 65 dan 80 cm. Namun, tetap saja masih ada kegagalan yang terjadi. Kegagalan tersebut disebabkan tendangan yang dihasilkan terlalu tinggi sehingga menyebabkan bola mengenai bagian atas mistar gawang. Total kegagalan yang terjadi ada 5, yaitu 4 kegagalan terjadi pada ketinggian 80 cm dan 1 kegagalan pada ketinggian 65 cm. Detail hasil pengujian dapat dilihat pada **Gambar 4.6** sampai dengan **Gambar 4.9**.



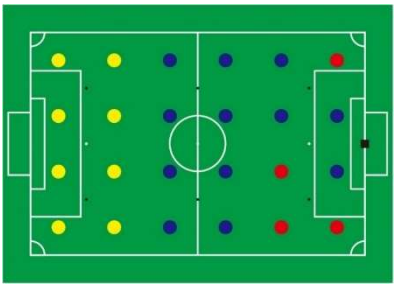
Gambar 4. 6 Hasil Pengujian Tinggi 0



Gambar 4. 7 Hasil Pengujian Tinggi 35



Gambar 4. 8 Hasil Pengujian Tinggi 65



Gambar 4. 9 Hasil Pengujian Tinggi 80

Dari keempat gambar yang telah disajikan di atas merepresentasikan hasil pengujian keseluruhan sistem. Titik-titik berjumlah 24 buah yang tersebar di seluruh lapangan merupakan titik koordinat pengujian tendangan dan warna pada titik tersebut merupakan status keberhasilan dari tendangan. Definisi dari warna yang digunakan pada titik-titik tersebut dapat dilihat pada **Tabel 4.8**.

Tabel 4. 8 Definisi Warna Titik

	= Bola jatuh sesuai dengan titik dan ketinggian yang telah ditentukan
	= Bola jatuh sesuai dengan titik dan ketinggian yang telah ditentukan, tetapi melalui pantulan kedua/ketiga/dst.
	= Bola jatuh tidak sesuai dengan titik dan ketinggian yang ditentukan, tetapi berhasil masuk ke gawang dengan cara menggelinding
	= Bola tidak masuk ke gawang

Tujuan dari penelitian ini adalah melakukan kontrol tendangan agar bola dapat masuk ke gawang. Maka, tingkat keberhasilan tendangan diperoleh dari jumlah warna 'Biru Tua', 'Biru Muda' dan 'Kuning'. Sementara untuk tingkat akurasi dari tendangan diperoleh dari jumlah warna 'Biru Tua' dan 'Biru Muda' saja. Hasil nilai persentase tingkat keberhasilan dan akurasi tendangan telah disajikan pada **Tabel 4.7**.

Berikutnya adalah melakukan pengujian kembali terhadap titik-titik yang masih mengalami kegagalan, khususnya pada ketinggian 65 dan 80 cm. Hal ini bertujuan untuk menekan angka persentase kegagalan menjadi sekecil mungkin. Setelah dilakukan pengujian kembali didapatkan hasil sebagai berikut.

Tabel 4. 9 Hasil Pengujian Akhir

Tinggi Target = 65 cm	
Persentase Keberhasilan	100%
Persentase Keakuratan	75%
Persentase Kegagalan	0%
Tinggi Target = 80 cm	
Persentase Keberhasilan	100%
Persentase Keakuratan	67%
Persentase Kegagalan	0%

Dari **Tabel 4.8** di atas, menunjukkan bahwa sistem berhasil melakukan tendangan ke gawang dalam seluruh pengujian.

BAB V
PENUTUP

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari melakukan sejumlah perancangan dan pengujian pada penelitian ini adalah sebagai berikut.

1. Sistem logika *fuzzy* mampu mengikuti karakteristik dari hubungan antara posisi penendang terhadap hasil tendangan. Hal ini ditunjukkan dengan diperolehnya akurasi yang mencapai 100% dan kegagalan 0% pada pengujian akhir. Keberhasilan dari sistem ditentukan dari proses konfigurasi sistem itu sendiri. Hal ini disebabkan sistem pada penelitian ini yang bersifat *open-loop*, sehingga tidak ada *feedback* yang mengoreksi nilai *error* yang terjadi.
2. Besar energi yang dihasilkan oleh solenoid untuk melakukan tendangan bergantung pada karakteristik solenoid itu sendiri. Pada penelitian ini kemampuan tendangan yang dihasilkan oleh solenoid hanya mampu mencapai jarak 9 meter.
3. Lebar pulsa yang digunakan untuk memicu terjadinya tendangan tidak mempengaruhi jarak tendang bola dan ketinggian bola, sehingga perlu parameter lain yang digunakan untuk mengontrol tendangan, yaitu menggunakan mekanisme naik-turun pada penendang.
4. Karakteristik hubungan antara posisi penendang terhadap hasil tendangan menunjukkan kurva logaritmik. Hal ini ditunjukkan dengan pergerakan kurva yang terus naik pada nilai 0 – 3000, kemudian kembali turun hingga mencapai 0.
5. Penggunaan sistem *odometry* cukup efektif untuk memetakan lapangan dan membaca pergerakan robot. Namun, sistem ini tidak cocok digunakan dalam jangka waktu yang lama karena robot yang aktif bergerak di dalam lapangan menyebabkan semakin bertambahnya nilai eror dari waktu ke waktu. Semakin banyak pergerakan robot, maka semakin besar nilai erornya. Hal ini terbukti dengan nilai rata-rata RMSE yang didapatkan adalah sebesar 23,573 untuk koordinat X dan 11,613 untuk koordinat Y.

5.2 Saran

Saran yang dapat diberikan untuk melakukan pengembangan terhadap penelitian ini adalah sebagai berikut.

1. Perlu dikembangkannya sistem yang dapat membaca posisi robot terhadap lapangan dengan lebih presisi agar tidak perlu melakukan kalibrasi secara rutin.
2. Perlu dirancang mekanisme atau rangkaian yang mampu membatasi tegangan yang digunakan solenoid. Hal ini bertujuan agar tidak hanya ketinggian dan jarak jatuh bola saja yang dapat dikontrol, tetapi kecepatan bola juga dapat dikontrol.
3. Penggunaan modul *booster* kurang efektif karena dalam pengisian kapasitor mulai dari nol hingga penuh membutuhkan waktu yang cukup lama. Sehingga, perlu dirancang rangkaian *booster* yang memiliki durasi *charging*-nya cukup cepat.

DAFTAR PUSTAKA

[1] A. K. Mulya, F. Ardilla, dan D. Pramadihanto, “Ball tracking and goal detection for middle size soccer robot using omnidirectional camera,” dalam *2016 International Electronics Symposium (IES)*, Denpasar, Indonesia, Sep 2016, hlm. 432–437, doi: 10.1109/ELECSYM.2016.7861045.

[2] R. Dias, J. Silva, J. L. Azevedo, B. Cunha, A. J. R. Neves, dan N. Lau, “Improving the kicking accuracy in a soccer robot,” dalam *Proceedings of the 30th Annual ACM Symposium on Applied Computing - SAC ’15*, Salamanca, Spain, 2015, hlm. 295–300, doi: 10.1145/2695664.2695862.

[3] R. Saputra, *Buku Ajar Fisika*. 2016.

[4] J. Walker, R. Resnick, dan D. Halliday, *Fundamentals of Physics, Halliday & Resnick 10th Edition*, 10th edition. Hoboken, NJ: Wiley, 2014.

[5] S. Ho, S. Foong, C. Lim, C. Lim, K. Lin, dan L. Kuppan, “Projectile motion on an inclined misty surface: I. Capturing and analysing the trajectory,” *Phys. Educ.*, vol. 44, hlm. 253, Apr 2009, doi: 10.1088/0031-9120/44/3/004.

[6] National Research Council (U.S.), Ed., *Opportunities in High Magnetic Field Science*. Washington, D.C: National Academies Press, 2005.

[7] S. H. M. Kasaei, S. M. M. Kasaei, dan S. A. M. Kasaei, “Design and Implementation Solenoid Based Kicking Mechanism for Soccer Robot Applied in Robocup-MSL,” *Int. J. Adv. Robot. Syst.*, vol. 7, no. 4, hlm. 30, Des 2010, doi: 10.5772/10490.

[8] E. Trillas dan L. Eciolaza, “An Introduction to Fuzzy Control,” dalam *Fuzzy Logic*, vol. 320, Cham: Springer International Publishing, 2015, hlm. 175–202.

[9] E. Trillas dan L. Eciolaza, *Fuzzy Logic*, vol. 320. Cham: Springer International Publishing, 2015.

[10] J. M. Mendel, “Fuzzy Logic Systems For Engineering : A Tutorial,” *Proc. IEEE*, vol. 83, no. 3, hlm. 345–377, Mar 1995, doi: 10.1109/5.364485.

[11] F. Nita, H. Thee, dan Saufiah, “Bab VII : Logika Samar (Fuzzy Logic),” 2017, hlm. 1.

[12] E. E. Perepelkin, A. N. Petersky, I. P. Yudin, R. V. Polyakova, dan V. A. Panacik, “The solenoid type magnetic field detector modeling,” dalam *2014 20th International Workshop on Beam Dynamics and Optimization (BDO)*, Saint-Petersburg, Russia, Jun 2014, hlm. 1–2, doi: 10.1109/BDO.2014.6890065.

[13] K. J. Meessen, J. J. H. Paulides, dan E. A. Lomonova, “A football kicking high speed actuator for a mobile robotic application,” dalam *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, Glendale, AZ, USA, Nov 2010, hlm. 1659–1664, doi: 10.1109/IECON.2010.5675433.

[14] H. D. Yohanes, D. S. Fransiscus, D. Gunawan, “Rancang Bangun Mekanisme Penendang Bola Berbasis Elektromagnetik untuk Robot Sepakbola Beroda R2C-*Warrior*,” dalam *Teche Jurnal Ilmiah Elektronika*, April 2018, vol. 17, no. 1, hlm. 49-57.

[15] B. P. T. van Goch, “Optimizing a solenoid for a Robocup kicker,” May 2006, DCT - No. 2006.051

LAMPIRAN

LAMPIRAN 1

Data pembacaan sistem *odometry*

Y_*Odometry*	X_*Odometry*	X_Terukur	Y_Terukur	Jarak_*Odometry*	Jarak_Terukur
0	0	0	0	1264	1264,911064
0	100	98	3	1240	1236,931688
0	200	194	3	1221	1216,552506
0	300	291	3	1208	1204,159458
0	400	386	3	1203	1200
0	500	484	0	1202	1204,159458
0	600	580	4	1208	1216,552506
0	700	677	8	1223	1236,931688
0	800	774	13	1244	1264,911064
100	0	1	102	1169	1170,469991
100	100	95	104	1137	1140,175425
100	200	191	105	1114	1118,033989
100	300	287	107	1098	1104,536102
100	400	384	108	1091	1100
100	500	480	110	1092	1104,536102
100	600	576	112	1101	1118,033989
100	700	674	117	1116	1140,175425
100	800	771	121	1140	1170,469991
200	0	0	200	1076	1077,032961
200	100	93	201	1044	1044,030651
200	200	190	204	1017	1019,803903
200	300	287	208	997	1004,987562
200	400	384	213	986	1000
200	500	481	219	983	1004,987562
200	600	577	226	990	1019,803903
200	700	675	232	1005	1044,030651
200	800	773	239	1030	1077,032961
300	0	-9	302	986	984,8857802
300	100	87	308	944	948,6832981
300	200	183	312	913	921,9544457
300	300	279	317	890	905,5385138
300	400	376	322	877	900
300	500	472	327	875	905,5385138
300	600	570	336	879	921,9544457
300	700	665	347	893	948,6832981
300	800	762	357	917	984,8857802
400	0	8	400	890	894,427191
400	100	103	400	852	854,4003745
400	200	200	396	827	824,6211251
400	300	297	395	810	806,2257748
400	400	393	390	809	800
400	500	489	387	817	806,2257748
400	600	587	387	833	824,6211251
400	700	683	385	862	854,4003745

400	800	778	381	901	894,427191
500	0	-4	500	808	806,2257748
500	100	90	500	764	761,5773106
500	200	188	501	729	728,0109889
500	300	284	504	705	707,1067812
500	400	381	506	693	700
500	500	477	508	696	707,1067812
500	600	573	509	711	728,0109889
500	700	672	511	740	761,5773106
500	800	767	512	779	806,2257748
600	0	-2	600	721	721,1102551
600	100	94	600	673	670,8203932
600	200	190	600	635	632,455532
600	300	286	602	608	608,276253
600	400	383	600	599	600
600	500	479	600	605	608,276253
600	600	575	600	624	632,455532
600	700	672	601	657	670,8203932
600	800	768	604	700	721,1102551
700	0	-17	697	653	640,3124237
700	100	78	700	593	583,0951895
700	200	174	704	544	538,5164807
700	300	270	708	508	509,9019514
700	400	368	713	487	500
700	500	465	715	488	509,9019514
700	600	560	718	507	538,5164807
700	700	657	722	542	583,0951895
700	800	752	726	590	640,3124237
800	0	-10	797	574	565,6854249
800	100	84	798	510	500
800	200	181	798	457	447,2135955
800	300	278	799	418	412,3105626
800	400	375	800	400	400
800	500	471	801	404	412,3105626
800	600	567	803	430	447,2135955
800	700	664	803	476	500
800	800	762	805	535	565,6854249
900	0	-16	895	515	500
900	100	81	900	437	424,2640687
900	200	177	903	370	360,5551275
900	300	273	906	319	316,227766
900	400	368	910	290	300
900	500	466	914	293	316,227766
900	600	562	917	325	360,5551275
900	700	660	921	381	424,2640687
900	800	757	925	451	500
1000	0	-15	994	463	447,2135955
1000	100	79	996	379	360,5551275
1000	200	176	999	300	282,8427125
1000	300	271	1001	235	223,6067977
1000	400	369	1005	196	200
1000	500	466	1009	201	223,6067977

1000	600	562	1012	247	282,8427125
1000	700	658	1017	316	360,5551275
1000	800	754	1023	396	447,2135955
1100	0	-12	1092	426	412,3105626
1100	100	85	1093	331	316,227766
1100	200	183	1094	241	223,6067977
1100	300	279	1096	158	141,4213562
1100	400	375	1097	105	100
1100	500	473	1098	125	141,4213562
1100	600	570	1100	197	223,6067977
1100	700	667	1100	285	316,227766
1100	800	764	1103	376	412,3105626
1200	0	-14	1192	414	400
1200	100	83	1189	316	300
1200	200	181	1187	219	200
1200	300	277	1184	123	100
1200	400	374	1183	30	0
1200	500	472	1182	74	100
1200	600	569	1182	170	200
1200	700	666	1180	266	300
1200	800	761	1179	362	400

LAMPIRAN 2

Data pengujian tendangan

Posisi Penendang = 0			Posisi Penendang = 1000		
Pulsa Tendangan (ms)	Jarak (m)	Hmax (cm)	Pulsa Tendangan (ms)	Jarak (m)	Hmax (cm)
5	7,1	260	5	6,8	280
10	7,1	245	10	6	260
15	7,7	220	15	7,4	230
20	7,7	190	20	7,1	230
25	7,5	170	25	7	250
30	7	180	30	8	225
35	6	200	35	6,6	250
40	7	170	40	5,8	280
45	7,3	250	45	6,8	265
50	6,9	270	50	5,45	300
Rata-Rata	7,13	215,5	Rata-Rata	6,695	257
Heighest	7,7	270	Heighest	8	300
Posisi Penendang = 1500			Posisi Penendang = 3000		
Pulsa Tendangan (ms)	Jarak (m)	Hmax (cm)	Pulsa Tendangan (ms)	Jarak (m)	Hmax (cm)
5	5,4	300	5	8,6	250
10	4,1	330	10	8,4	230
15	7,2	240	15	9,1	225
20	7,3	260	20	8,5	230
25	6	280	25	9	210
30	7,35	245	30	7,9	200
35	5,8	310	35	8,9	220
40	6,2	300	40	8,2	230
45	7,6	240	45	8,8	225
50	6,9	280	50	7,8	230
Rata-Rata	6,385	278,5	Rata-Rata	8,52	225
Heighest	7,6	330	Heighest	9,1	250
Posisi Penendang = 4500			Posisi Penendang = 6000		
Pulsa Tendangan (ms)	Jarak (m)	Hmax (cm)	Pulsa Tendangan (ms)	Jarak (m)	Hmax (cm)
5	7,5	250	5	7,9	210
10	7,9	220	10	8,1	200
15	7,9	240	15	7,8	210
20	8,2	235	20	7,9	200
25	8,1	230	25	7,5	210
30	8,9	235	30	7,6	195
35	8,6	240	35	7,8	220
40	8,9	210	40	7,9	200
45	7,9	240	45	7,8	180

50	6,9	250	50	8,1	200
Rata-Rata	8,08	235	Rata-Rata	7,84	202,5
Heighest	8,9	250	Heighest	8,1	220
Posisi Penendang = 7500			Posisi Penendang = 9000		
Pulsa Tendangan (ms)	Jarak (m)	Hmax (cm)	Pulsa Tendangan (ms)	Jarak (m)	Hmax (cm)
5	7	135	5	6,6	105
10	6,7	140	10	6,6	105
15	6,4	110	15	6,4	110
20	7,1	135	20	5,9	95
25	6,8	130	25	5,8	100
30	7,2	140	30	6,3	110
35	7,2	155	35	6,2	115
40	7,4	145	40	6,5	115
45	6,9	110	45	6	90
50	7,2	135	50	6,2	100
Rata-Rata	6,99	133,5	Rata-Rata	6,25	104,5
Heighest	7,4	155	Heighest	6,6	115
Posisi Penendang = 10.500			Posisi Penendang = 12000 (76 mm)		
Pulsa Tendangan (ms)	Jarak (m)	Hmax (cm)	Pulsa Tendangan (ms)	Jarak (m)	Hmax (cm)
5	5,55	65	5	4,05	35
10	4,9	50	10	4,25	34
15	5,2	55	15	4,3	42
20	5,15	53	20	4,35	40
25	5,25	50	25	4,1	38
30	5,05	52	30	4,4	47
35	4,85	48	35	3,85	26
40	4,65	42	40	3,35	24
45	4,35	40	45	4,05	34
50	4,8	49	50	3,7	23
Rata-Rata	4,975	50,4	Rata-Rata	4,04	34,3
Heighest	5,55	65	Heighest	4,4	47
Posisi Penendang = 13.500			Posisi Penendang = 14.821 (MAX = 95mm)		
Pulsa Tendangan (ms)	Jarak (m)	Hmax (cm)	Pulsa Tendangan (ms)	Jarak (m)	Hmax (cm)
5	2,9	17	5	2	12
10	2,6	16	10	2	4
15	2,65	15	15	1,9	5
20	2,85	17	20	1,6	12
25	3,15	17	25	1,8	10
30	2,6	30	30	1,6	10
35	3,2	16	35	1,85	9
40	3,2	20	40	1,7	11
45	2,8	16	45	1,45	12

50	3,1	14	50	1,7	9
Rata-Rata	2,905	17,8	Rata-Rata	1,76	9,4
Heighest	3,2	30	Heighest	2	12

LAMPIRAN 3

Data konfigurasi parameter *fuzzy*

Encoder	Tinggi In	Tinggi Out	Jarak In	Jarak Out
14750	20	3	0	100
14000	20	10	0	100
14000	20	10	160	160
13500	20	14	160	160
13000	20	20	160	160
12500	20	0	320	310
12000	20	35	320	320
12250	20	16	320	320
12150	20	28	320	320
12200	20	22	320	320
12225	20	0	320	315
10500	20	0	480	450
10000	20	30	480	480
10250	20	18	480	480
10150	20	35	480	480
10200	20	0	480	460
10175	20	0	480	470
10200	20	25	480	480
8250	20	50	640	640
9000	20	16	640	640
8800	20	16	640	640
8750	20	28	640	640
8950	20	16	640	640
8850	20	25	640	640
8900	20	22	640	640
5500	20	20	800	800
3000	20	0	900	0
4000	20	0	900	0
4500	20	0	900	0
5000	20	0	900	0
12750	50	20	160	160
10000	50	50	160	160
10200	50	50	160	160
10200	50	50	320	320
8250	50	110	480	480
9000	50	25	480	480
8800	50	95	480	480
8900	50	30	480	480
8850	50	30	480	480
8800	50	50	480	480
5500	50	999	640	640
7250	50	60	640	640

7500	50	40	640	640
7400	50	80	640	640
7450	50	70	640	640
7500	50	70	640	640
7750	50	50	640	640
5500	50	0	800	780
4750	50	0	800	750
6000	50	20	800	800
6250	50	30	800	800
6500	50	0	800	780
4500	50	50	800	800
4500	50	0	900	840
3750	50	0	900	0
8900	80	70	160	160
8750	80	70	160	160
8250	80	80	160	160
8250	80	110	320	320
8800	80	110	320	320
9000	80	80	320	320
8250	80	120	480	480
8500	80	100	480	480
8800	80	100	480	480
9000	80	75	480	480
8900	80	100	480	480
8950	80	70	480	480
8900	80	60	480	480
8900	80	80	480	480
8250	80	60	640	640
7750	80	60	640	640
7000	80	50	640	640
5500	80	160	640	640
8900	80	30	640	640
6500	80	160	640	640
6800	80	120	640	640
7000	80	65	640	640
6900	80	75	640	640
6850	80	75	640	640
6825	80	80	640	640
5500	80	65	800	800
4500	80	65	800	800
6000	80	40	800	800

LAMPIRAN 4

Program sistem logika *fuzzy*

```
// BismillahTA.cpp : This file contains the 'main' function.
Program execution begins and ends there.
//

#include <iostream>
#include <stdio.h>
#include <math.h>
using namespace std;

#define PI 3.1415926536
#define x_gawang 400
#define y_gawang 1200

float jarak, tinggi, posisi_x, posisi_y,
sudut_radian, sudut_derajat, sudut_heading,
uMember, uNilaiZ,
uSangatDekat, uDekat, uSedang, uJauh, uSangatJauh,
uRendah, uMenengah, uTinggi,
Output, Output_Invers;

float alphapred[3][5], Z[3][5];
string Keluaran, kondisiJarak, kondisiTinggi;
//float jarakuji[5] = { 400, 500, 600, 700 };
//float tinggiuji[4] = { 0, 35, 65, 80 };
//float OUTPUT[5][4];
//string Keterangan[5][4];

// Menghitung Jarak dan Sudut
void perhitungan()
{
        jarak = sqrt(powf((x_gawang - posisi_x), 2) +
powf((y_gawang - posisi_y), 2));
        sudut_radian = atan2f((y_gawang - posisi_y), (x_gawang
- posisi_x));
        sudut_derajat = sudut_radian * 180 / PI;
        sudut_heading = sudut_derajat;
}

// Rumus Mencari Nilai Alpha dengan Inputan Nilai X
void kurva_member(int pilih, float nilai, float A, float B,
float C)
{
        switch (pilih)
        {
```

```
        case 1:
                if ((nilai >= A) && (nilai <= B)) uMember = 1;
                if ((nilai > B) && (nilai < C)) uMember = (C -
nilai) / (C - B);
                if (nilai >= C) uMember = 0;
                break;
        case 2:
                if ((nilai <= A) || (nilai >= C)) uMember = 0;
                if ((nilai > A) && (nilai < B)) uMember =
(nilai - A) / (B - A);
                if ((nilai > B) && (nilai < C)) uMember = (C -
nilai) / (C - B);
                if (nilai == B) uMember = 1;
                break;
        case 3:
                if (nilai <= A) uMember = 0;
                if ((nilai > A) && (nilai < B)) uMember =
(nilai - A) / (B - A);
                if (nilai >= B) uMember = 1;
                break;
        }
}

void fuzzifikasi()
{
        // Tinggi
        uMember = 0;
        kurva_member(1, tinggi, 0, 25, 50);
        uRendah = uMember;
        kurva_member(2, tinggi, 25, 50, 75);
        uMenengah = uMember;
        kurva_member(3, tinggi, 50, 75, 100);
        uTinggi = uMember;

        // Jarak
        uMember = 0;
        kurva_member(1, jarak, 0, 160, 320);
        uSangatDekat = uMember;
        kurva_member(2, jarak, 160, 320, 480);
        uDekat = uMember;
        kurva_member(2, jarak, 320, 480, 640);
        uSedang = uMember;
        kurva_member(2, jarak, 480, 640, 800);
        uJauh = uMember;
        kurva_member(3, jarak, 640, 800, 900);
        uSangatJauh = uMember;
}
```

```
void rulebase_fuzzy()
{
        if (kondisiJarak == "SangatDekat" && kondisiTinggi ==
"Rendah") Keluaran = "Sangat Pelan";
        else if (kondisiJarak == "SangatDekat" &&
kondisiTinggi == "Menengah") Keluaran = "Sedang-1";
        else if (kondisiJarak == "SangatDekat" &&
kondisiTinggi == "Tinggi") Keluaran = "Kencang";

        else if (kondisiJarak == "Dekat" && kondisiTinggi ==
"Rendah") Keluaran = "Pelan";
        else if (kondisiJarak == "Dekat" && kondisiTinggi ==
"Menengah") Keluaran = "Sedang-1";
        else if (kondisiJarak == "Dekat" && kondisiTinggi ==
"Tinggi") Keluaran = "Sedang-2";

        else if (kondisiJarak == "Sedang" && kondisiTinggi ==
"Rendah") Keluaran = "Sedang-1";
        else if (kondisiJarak == "Sedang" && kondisiTinggi ==
"Menengah") Keluaran = "Sedang-2";
        else if (kondisiJarak == "Sedang" && kondisiTinggi ==
"Tinggi") Keluaran = "Sedang-2";

        else if (kondisiJarak == "Jauh" && kondisiTinggi ==
"Rendah") Keluaran = "Sedang-2";
        else if (kondisiJarak == "Jauh" && kondisiTinggi ==
"Menengah") Keluaran = "Cukup Kencang1";
        else if (kondisiJarak == "Jauh" && kondisiTinggi ==
"Tinggi") Keluaran = "Sangat Kencang";

        else if (kondisiJarak == "SangatJauh" && kondisiTinggi
== "Rendah") Keluaran = "Kencang Sekali";
        else if (kondisiJarak == "SangatJauh" && kondisiTinggi
== "Menengah") Keluaran = "Kencang Sekali";
        else if (kondisiJarak == "SangatJauh" && kondisiTinggi
== "Tinggi") Keluaran = "Kencang Sekali";
}

//===== Untuk Output Berbentuk Diagram Batang =====
int Fuzzy_Set[3][5] =
{
        // vDekat | Dekat | Sedang | Jauh | vJauh
        {13000, 12200, 10200, 8900, 5500}, // | Rendah
        {10200, 10200, 8900, 7750, 5500}, // | Menengah
        {8250, 8900, 8900, 6825, 5500}  // | Tinggi
```

```
// Nilai Kurva 8 Output (After Tuned)
        // Sangat Pelan = 13000
        // Pelan = 12200
        // Sedang-1 = 10200
        // Sedang-2 = 8900
        // Kencang 1 = 8250
        // Cukup Kencang = 7750
        // Sangat Kencang = 6825
        // Kencang Sekali = 5500

};

int Invers_Fuzzyset[3][5] =
{
        // Menggunakan Nilai Invers dari Kurva Output (14500 -
nilai output)
        // vDekat | Dekat | Sedang | Jauh | vJauh
        {0, 0, 2000, 4250, 6500}, // | Rendah
        {0, 2000, 4250, 6500, 11000}, // | Menengah
        {2000, 4250, 6500, 11000, 11000}  // | Tinggi

        // Nilai Kurva 7 Output
        // Sangat Pelan = 14750
        // Pelan = 12750
        // Sedang = 10500
        // Kencang = 8250
        // Sangat Kencang 1 = 5500
        // Sangat Kencang 2 = 4750
        // Sangat Kencang 3 = 3750
};


//===== Untuk Output Berbentuk Kurva =====
// Rumus Invers Mencari Nilai X dengan Inputan Alpha
void nilai_z(int pilih, float Y, float A, float B, float C)
{
        switch (pilih)
        {
        case 1:
                if (Y == 1) uNilaiZ = B;
                else if (Y == 0) uNilaiZ = C;
                else uNilaiZ = C + (B * Y) - (C * Y);
                break;
        case 2:
                if (Y == 1) uNilaiZ = B;
```

```
                if (Y <= A) uNilaiZ = A;
                if (Y >= C) uNilaiZ = C;
                if ((Y > A) && (Y < B)) uNilaiZ = (B * Y) - (A
* Y) + A;
                if (Y == B) uNilaiZ = B;
                if ((Y > B) && (Y < C)) uNilaiZ = C + (B * Y)
- (C * Y);
                break;
        case 3:
                if (Y == 1) uNilaiZ = B;
                else if (Y == 0) uNilaiZ = A;
                else uNilaiZ = (B * Y) - (A * Y) + A;
                break;
        }
}

void defuzzy()
{
        float pembilang = 0, penyebut = 0, Ztotal = 0;
        float data_uJarak[5] = { uSangatDekat, uDekat,
uSedang, uJauh, uSangatJauh };
        float data_uTinggi[3] = { uRendah, uMenengah, uTinggi
};
        string DataJarak[5] = { "SangatDekat", "Dekat",
"Sedang", "Jauh", "SangatJauh" };
        string DataTinggi[3] = { "Rendah", "Menengah",
"Tinggi" };

        // Menampilkan Nilai Setiap Variabel, for Troubleshoot
Only
        // Nilai Fuzzifikasi
        cout << "Nilai uSangatDekat = " << uSangatDekat <<
endl;
        cout << "Nilai uDekat = " << uDekat << endl;
        cout << "Nilai uSedang = " << uSedang << endl;
        cout << "Nilai uJauh = " << uJauh << endl;
        cout << "Nilai uSangatJauh = " << uSangatJauh << endl;

        cout << endl << "Nilai uRendah = " << uRendah << endl;
        cout << "Nilai uMenengah = " << uMenengah << endl;
        cout << "Nilai uTinggi = " << uTinggi << endl << endl;
        for (int i = 0; i < 3; i++)
        {
                for (int j = 0; j < 5; j++)
                {
```

```
                        kondisiJarak = DataJarak[j];
                        kondisiTinggi = DataTinggi[i];
                        // Menampilkan Semua Nilai Alpha Jarak
dan Tinggi di Setiap Kondisi
                        cout << "-----------------------------
--" << endl;
                        cout << "Alpha Jarak [" << i + 1 << j
+ 1 << "] : " << data_uJarak[j] << endl;
                        cout << "Alpha Tinggi [" << i + 1 << j
+ 1 << "] : " << data_uTinggi[i] << endl;
                        rulebase_fuzzy();
                        alphapred[i][j] =
std::fminf(data_uJarak[j], data_uTinggi[i]);
                        // Menampilkan Setiap Nilai Alpha
Minimal di Setiap Kondisi
                        cout << "Alpha Predikat (Alpha
Minimal) [" << i + 1 << j + 1 << "] : " << alphapred[i][j] <<
endl;

                        //===== Program Untuk Output Berbentuk
Kurva =====
                        //if (Keluaran == "SangatPelan")
nilai_z(1, alphapred[i][j], 0, 10, 20);
                        //else if (Keluaran == "Pelan")
nilai_z(2, alphapred[i][j], 10, 20, 30);
                        //else if (Keluaran == "Sedang")
nilai_z(2, alphapred[i][j], 20, 30, 40);
                        //else if (Keluaran == "Kencang")
nilai_z(2, alphapred[i][j], 30, 40, 50);
                        //else if (Keluaran ==
"SangatKencang") nilai_z(3, alphapred[i][j], 40, 50, 60);
                        //Z[i][j] = uNilaiZ;
                        //// Menampilkan Setiap Nilai Z dan
Output Rulebase dari Setiap Kondisi
                        //cout << "Z [" << i + 1 << j + 1 <<
"] : " << Z[i][j] << endl;
                        //cout << "Ouput Fuzzy [" << i + 1 <<
j + 1 << "] : " << Keluaran << endl;
                        //pembilang += alphapred[i][j] *
Z[i][j];
                        //penyebut += alphapred[i][j];
                                                //=====
Program Untuk Output Berbentuk Diagram Batang =====
                        Z[i][j] = alphapred[i][j] *
Fuzzy_Set[i][j];
```

```
pembilang += Z[i][j];
                        penyebut += alphapred[i][j];
                        cout << "Nilai Z [" << i + 1 << j + 1
<< "] : " << Z[i][j] << endl;
                        cout << "Ouput Fuzzy [" << i + 1 << j
+ 1 << "] : " << Keluaran << endl;
                }
        }
        // Proses Menghitung Ztotal Metode Tsukamoto
        Ztotal = pembilang / penyebut;
        Output = Ztotal;
        cout << "-------------------------------" << endl;
}

int main()
{
    // Masukkan nilai awal untuk jarak dan ketinggian target
        cout << "Program Fuzzy Kontrol Tendangan" << endl <<
endl;
        cout << "Pilih Jenis Inputan yang Digunakan : " <<
endl;
        cout << "1. Inputan Berupa Nilai Jarak" << endl;
        cout << "2. Inputan Berupa Nilai Posisi XY Robot" <<
endl;
        cout << "(Jawab) = ";
        int mode;
        cin >> mode;
        switch (mode)
        {
        case 1 :
                cout << endl << "Rentang Nilai Jarak : 0 -
1200 cm" << endl << "Rentang Nilai Tinggi : 0 - 100 cm" <<
endl << endl;
                cout << "Input Nilai Jarak = ";
                cin >> jarak;
                break;
        case 2 :
                cout << endl << "Rentang Nilai Posisi X : 0 -
800 cm" << endl  << "Rentang Nilai Posisi Y : 0 - 1200 cm" <<
endl << "Rentang Nilai Tinggi : 0 - 100 cm" << endl << endl;
                cout << "Input Nilai Posisi X = ";
                cin >> posisi_x;
                cout << "Input Nilai Posisi Y = ";
                cin >> posisi_y;
                //Menghitung Nilai Jarak dan Sudut
                perhitungan();
```

```
                break;
        }

        cout << "Input Nilai Ketinggian = ";
        cin >> tinggi;

        // Menampilkan Nilai Jarak, Tinggi dan Arah Heading
Robot
        cout << endl << "Nilai Jarak = " << jarak << endl;
        cout << "Nilai Tinggi = " << tinggi << endl;
        cout << "Arah Heading Robot = " << sudut_heading <<
endl << endl;

        if (jarak > 850)
        {
                Output = 3000;
        }
        else
        {
                //Proses Sistem Fuzzy
                fuzzifikasi();
                defuzzy();
        }

        //Menampilkan Hasil Output dari Fuzzy
        cout << endl << "Keluaran = " << Keluaran << endl;
        cout << endl << "Nilai Output = " << Output << endl;
}
```

BIODATA PENULIS

Muhammad Revo Khairullah, dilahirkan ibunya di Kota Surabaya, Provinsi Jawa Timur. Ia lahir pada hari Sabtu Legi, tanggal 16 Mei 1998 pada saat krisis moneter dan masa Reformasi. Semasa sekolah, penulis mengenyam pendidikannya di Kabupaten Jember, Provinsi Jawa Timur. Penulis merupakan alumni dari TK Al Amien Jember, SD Al – Baitul Amien Jember, SMPN 3 Jember dan SMAN 1 Jember. Setelah lulus SMA, penulis merantau ke Surabaya untuk melanjutkan jenjang pendidikannya di Institut Teknologi Sepuluh Nopember Surabaya, di jurusan Teknik Elektro. Selama masa perkuliahannya, penulis telah menghabiskan waktu 3 tahun dalam menekuni bidang robotika dan tergabung dalam salah satu tim robotika ITS. Selama tergabung dalam tim robot, penulis telah berhasil mengikuti dan menjuarai berbagai kompetisi regional, nasional dan internasional pada tahun 2017 sampai dengan 2019. Pada tahun 2020 penulis vakum dari dunia robotika dikarenakan adanya pandemi dan larangan untuk berkegiatan. Penulis selalu bercita-cita untuk menjadi pribadi yang selalu bisa memberikan manfaat kepada orang di sekitarnya.

Email : revo.khairullah@gmail.com