

**TUGAS AKHIR**

**RANCANG BANGUN *SEMI-AUTOMATIC NAVIGATION ROBOT*  
MENGUNAKAN *REMOTE CONTROL* DAN SENSOR ULTRASONIK  
BERBASIS *FUZZY LOGIC CONTROL***

Disusun Sebagai Salah Satu Persyaratan dalam Memperoleh Gelar Sarjana (S-1)  
pada Jurusan Teknik Elektro



Oleh :  
**FARHAN WILDANI**  
**1177070027**

**JURUSAN TEKNIK ELEKTRO  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SUNAN GUNUNG DJATI  
BANDUNG  
2022**

**LEMBAR PENGESAHAN TUGAS AKHIR**

**RANCANG BANGUN *SEMI-AUTOMATIC NAVIGATION ROBOT*  
MENGUNAKAN *REMOTE CONTROL* DAN SENSOR ULTRASONIK  
BERBASIS *FUZZY LOGIC CONTROL***

Oleh:

**FARHAN WILDANI**

**1177070027**

Telah disetujui dan disahkan sebagai Tugas Akhir Jurusan Teknik Elektro  
di Bandung, pada tanggal 24 Februari 2022

Menyetujui,

Pembimbing I

Pembimbing II

**Dr. Rina Mardiaty, M.T.**  
NIP. 198409042009122002

**Edi Mulyana, S.T., M.T.**  
NIP. 197001062008011025

Mengesahkan,

Dekan Fakultas Sains dan Teknologi  
UIN SGD Bandung

Ketua Jurusan Teknik Elektro

**Dr. Hasniah Aliah, M.Si.**  
NIP. 197806132005012014


**Nanang Ismail, M.T.**  
NIP. 197505262011011002



***Karya ini dipersembahkan untuk :***

*Kedua orang tua, baik ibu ataupun ayah dan keluarga tercinta.  
Terlebih kepada diri sendiri yang telah berjuang dan bertahan hingga saat ini.  
Serta, kepada rekan-rekan yang senantiasa memberikan dukungan selama ini.*

uin  
UNIVERSITAS ISLAM NEGERI  
SUNAN GUNUNG DJATI  
BANDUNG

 <b>KEMENTERIAN AGAMA</b> <b>UNIVERSITAS ISLAM NEGERI</b> <b>SUNAN GUNUNG DJATI</b> <b>FAKULTAS SAINS DAN TEKNOLOGI</b> <small>Jl. AH Nasution No. 105 Bandung</small>	<b>FORM (FR)</b>	No. Dok : FST-TU-AKM-FR-F.29
		Tgl. Terbit : 1 September 2014
		No. Revisi: : 00
		Hal : ii/75
<b>SURAT PERNYATAAN KARYA SENDIRI</b>		

*Bismillahirraahmanirrahim,*

Saya yang bertanda tangan di bawah ini,

Nama : Farhan Wildani  
NIM : 1177070027  
Tempat, Tanggal Lahir : Bekasi, 18 Februari 1999  
Jurusan : Teknik Elektro  
Fakultas : Sains dan Teknologi  
Judul Tugas Akhir : Rancang Bangun *Semi-Automatic Navigation Robot* Menggunakan *Remote Control* dan Sensor Ultrasnoik Berbasis *Fuzzy Logic Control*

Dengan ini menyatakan bahwa:

1. Tugas Akhir ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Universitas Islam Negeri (UIN) Sunan Gunung Djati Bandung maupun di Perguruan Tinggi lain.
2. Karya tulis ini adalah murni gagasan, rumusan dan penelitian saya, tanpa bantuan pihak lain, kecuali arahan Tim Pembimbing dan masukan Tim Penelaah.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali tertulis dengan jelas dicantumkan dalam daftar pustaka sebagai acuan dalam naskah dengan menyebutkan nama pengarangnya.
4. Pernyataan ini saya buat dengan sesungguhnya, dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya ini, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi ini.

Bandung, 24 Februari 2022

Farhan Wildani  
NIM. 1177070027

## ABSTRAK

Seiring dengan berkembangnya zaman, teknologi pada bidang *mobile robot* terus mengalami kemajuan yang sangat pesat. Dalam perkembangan robot otonom, navigasi menjadi salah satu bagian yang memiliki peran penting. Oleh karena itu, *mobile robot* harus mampu beradaptasi dengan lingkungannya. Sehingga diperlukan suatu metode pengendalian yang dapat membantu robot dalam proses penyesuaian dinamika lingkungan sekitarnya. Pada penelitian ini robot *semi-automatic navigation* mengadopsi kecerdasan buatan logika *Fuzzy* sebagai pemroses keluaran yang akan dihasilkan oleh robot. Logika *Fuzzy* pada robot ini digunakan untuk mengendalikan kecepatan motor berdasarkan jarak *obstacle* yang terbaca oleh sensor dan masukan yang diberikan oleh *remote control*. Pada penelitian ini digunakan satu buah sensor ultrasonik HC-SR04 yang dipasang pada bagian depan robot dan *remote control* untuk memberikan perintah kepada robot serta Arduino MEGA 2560 sebagai mikrokontrolernya. Untuk membuat pergerakan robot lebih stabil saat menghindari rintangan, diterapkan algoritma logika *Fuzzy* untuk mengontrol PWM motor kanan dan kiri. Pengujian sistem *Fuzzy* robot dilakukan dengan skenario robot mendeteksi halangan/*obstacle* pada jarak 4cm dan remote memberikan input nilai 1870Hz. Didapatkan hasil yang ditampilkan pada serial monitor aplikasi Arduino IDE sebesar 62,5 PWM untuk motor kanan dan 103,53 PWM untuk motor kiri sedangkan hasil simulasi pada aplikasi Matlab didapatkan PWM motor kanan sebesar 62,5 PWM dan motor kiri sebesar 104 PWM. Dengan membandingkan *output* dari *semi-automatic navigation robot* berbasis *Fuzzy* dengan keluaran hasil simulasi, didapatkan bahwa logika *Fuzzy* telah berhasil diimplementasikan pada robot dengan tingkat keberhasilan 100% untuk motor kanan dan 99,995% untuk motor kiri. Berikutnya pada pengujian perbandingan manuver robot menggunakan metode *Fuzzy* dengan robot tanpa menggunakan *Fuzzy* didapatkan hasil bahwa robot yang menggunakan metode *Fuzzy logic control* lebih stabil dalam bermanuver menghindari rintangan karena robot yang menggunakan *Fuzzy* dapat menyesuaikan terhadap dinamika lingkungan yang dihadapinya.

Kata kunci: Robot navigasi, logika *Fuzzy*, *remote control*, sensor ultrasonik.

## **ABSTRACT**

*With the development of the times, technology in the field of mobile robots continues to experience very fast progress. In the development of autonomous robots, navigation goes wrong one part that has an important role. Therefore, mobile robots must be able to adapt to their environment. So we need a control method that can help the robot in the process of adjusting the dynamics of the surrounding environment. This semi-automatic navigation robot adopts fuzzy logic artificial intelligence as a tool processing output to be generated by the robot. Fuzzy logic in this robot is used to control the motor speed based on the obstacle distance read by the sensor and the input given by the remote control. In this study, one fruit was used ultrasonic sensor HC-SR04 mounted on the front of the robot and remote control to give commands to the robot and Arduino MEGA 2560 as the microcontroller. In order to make the robot's movement more stable when avoiding obstacles, implemented fuzzy logic algorithm to control right and left motor PWM. Test fuzzy robot system is carried out with the scenario of a robot detecting obstacles in the the distance is 4cm and the remote gives an input value of 1870Hz. Displayed results obtained on the Arduino IDE application serial monitor of 62.5 PWM for the right motor and 103.53 PWM for the left motor while the simulation results in the Matlab application get PWM the right motor is 62.5 PWM and the left motor is 104 PWM. By comparing the output of a fuzzy-based semi-automatic navigation robot with the output of the simulation results, It was found that fuzzy logic has been successfully implemented on a robot with a level of 100% success for the right motor and 99.995% for the left motor. Next on comparative testing of robot maneuvers using the fuzzy method with robots without using fuzzy, it was found that the robot using the fuzzy logic method control is more stable in maneuvering to avoid obstacles because the robot that uses fuzzy can adapt to the dynamics of the environment it faces.*

*Keywords:* Navigation robot, fuzzy logic, remote control, ultrasonic sensor.

## KATA PENGANTAR

*Assalamualaikum Warohmatullahi Wabarokatuh,*

Alhamdulillah, puji dan syukur penulis panjatkan ke hadirat Allah SWT yang telah memberikan rahmat, nikmat sehat dan segala kemudahan-Nya sehingga penulis dapat menyelesaikan tugas akhir ini yang berjudul Rancang Bangun *Semi-Automatic Navigation Robot* Menggunakan *Remote Control* dan Sensor Ultrasnoik Berbasis *Fuzzy Logic Control*

Pada kesempatan ini juga penulis ingin mengucapkan rasa terima kasih dan apresiasi atau penghargaan yang sebesar-besarnya kepada pihak-pihak yang sudah membantu, membimbing, serta memberi dorongan kepada penulis sehingga tugas akhir ini dapat terselesaikan, yakni kepada:

1. Ibu Dr. Hasniah Aliah, M.Si., selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Gunung Djati Bandung.
2. Bapak Nanang Ismail, M.T., selaku ketua Jurusan Teknik Elektro Fakultas Sains dan Teknologi UIN Sunan Gunung Djati Bandung.
3. Ibu Dr. Rina Mardiaty, M.T., selaku Dosen Pembimbing Pertama yang selalu sabar membimbing dan memberikan waktu, ilmu, tenaga dan pikiran selama melakukan penelitian.
4. Bapak Edi Mulyana, S.T., M.T., selaku Dosen Pembimbing Kedua yang selalu sabar membimbing dan memberikan waktu, tenaga dan pikiran dalam penyelesaian tugas akhir.
5. Kedua Orang Tua yang selalu memberikan semangat dan dukungan baik berupa moral maupun materil serta tidak henti-hentinya untuk berdoa, demi keberhasilan penulis.
6. Aan Eko Setiawan, S.T., selaku Dosen praktikum yang senantiasa memberi arahan dalam proses pengerjaan Skripsi ini.
7. Rekan-rekan Teknik Elektro 2017 yang senantiasa memberikan doa serta dukungan sehingga tugas akhir ini dapat terselesaikan.
8. Serta kepada pihak-pihak yang tidak bisa disebutkan satu per satu namanya yang ikut andil membantu dalam menyelesaikan tugas akhir ini, penulis mengucapkan banyak terima kasih.

Akhir kata, kami berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan kepada semua pihak yang telah membantu. Saya menyadari bahwa tugas akhir ini masih banyak kekurangan yang perlu diperbaiki. Maka dari itu, kritik dan saran dari berbagai pihak sangat diharapkan.

*Wassalamualaikum Warohmatullahi Wabarokatuh.*

Bandung, 24 Februari 2022

Penulis,



**Farhan Wildani**  
**NIM. 1177070027**



## DAFTAR ISI

HALAMAN PENGESAHAN.....	i
HALAMAN PERUNTUKAN .....	ii
LEMBAR PERNYATAAN KARYA SENDIRI .....	iii
ABSTRAK .....	iv
<i>ABSTRACT</i> .....	v
KATA PENGANTAR .....	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR .....	xi
DAFTAR TABEL.....	xiii
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang Masalah.....	1
1.2 <i>State of The Art</i> .....	2
1.3 Rumusan Masalah .....	6
1.4 Tujuan.....	6
1.5 Manfaat Penelitian.....	6
1.5.1 Manfaat Akademis .....	6
1.5.2 Manfaat Praktis .....	6
1.6 Batasan Masalah.....	6
1.7 Kerangka Berpikir .....	7
1.8 Sistematika Penulisan.....	8
<b>BAB II TEORI DASAR .....</b>	<b>9</b>
2.1 Sistem Kendali .....	9
2.1.1 Sistem Kontrol Loop Terbuka.....	9
2.1.2 Sistem Kontrol Loop Tertutup .....	9
2.2 Respon Sistem .....	10
2.3 <i>Fuzzy Logic Control</i> .....	11
2.4 <i>Fuzzy Logic Control</i> Metode Mamdani .....	12
2.4.1 Pembentukan Himpunan <i>Fuzzy</i> .....	12
2.4.2 Aplikasi Fungsi Implikasi .....	16
2.4.3 Komposisi Aturan .....	18
2.4.4 Defuzzisikasi .....	18

2.5	Mikrokontroler .....	19
2.6	Arduino Mega 2560.....	19
2.7	Sensor Ultrasonik HC-SR04 .....	21
2.8	Bahasa C.....	22
2.9	Motor DC ( <i>Direct Current</i> ).....	23
2.10	Motor <i>Driver Shield</i> L293D.....	24
2.11	<i>Remote Control</i> 2.4GHz.....	24
BAB III METODE PENELITIAN .....		26
3.1	Metode Penelitian.....	26
3.2	Studi Literatur .....	26
3.3	Identifikasi Masalah .....	27
3.4	Analisis Kebutuhan .....	27
3.5	Perancangan .....	27
3.6	Implementasi Sistem .....	28
3.7	Integrasi Sistem .....	28
3.8	Pengujian Sistem.....	28
3.9	Analisis Hasil .....	28
BAB IV PERANCANGAN DAN IMPLEMENTASI.....		29
4.1	Analisis Kebutuhan .....	29
4.2	Perancangan .....	29
4.3	Perancangan Sistem Kendali <i>Fuzzy Logic</i> .....	31
4.3.1	Pemodelan <i>Fuzzy Logic</i> .....	31
4.3.2	<i>Membership Function Input</i> .....	32
4.3.3	<i>Membership Function Output</i> .....	35
4.3.4	<i>Fuzzy Rule Base</i> .....	38
4.4	Perancangan <i>Hardware</i> .....	39
4.4.1	Konfigurasi <i>Motor Driver Shield</i> L293D dengan Arduino MEGA 2560.....	39
4.4.2	Instalasi Sensor Ultrasonik HC-SR04 .....	40
4.4.3	Instalasi <i>Receiver</i> 2.4GHz 3 Kanal .....	41
4.4.4	Instalasi Motor DC.....	42
4.4.5	Instalasi Baterai Lipo 11V .....	43
4.4.6	Skema Rangkaian Robot .....	44
4.5	Implementasi .....	44

4.6	Implementasi Rancangan Program.....	44
4.6.1	<i>Source Code</i> Variabel <i>Input</i> dan <i>Output</i> .....	45
4.6.2	Pemrograman Model <i>Fuzzy</i> Pada Arduino IDE.....	46
4.7	Implementasi Rancangan <i>Hardware</i> .....	47
BAB V	PENGUJIAN DAN ANALISIS .....	48
5.1	Pengujian .....	48
5.2	Pengujian <i>Hardware</i> .....	48
5.2.1	Pengujian Sensor Ultrasonik .....	48
5.2.2	Pengujian <i>Remote Control</i> .....	49
5.2.3	Pengujian <i>Motor Driver Shield L293D</i> .....	50
5.3	Pengujian Model <i>Fuzzy Logic Control</i> .....	51
5.3.1	Pengujian <i>Fuzzy</i> Pada Sistem Robot .....	52
5.3.2	Perhitungan <i>Fuzzy</i> Secara Manual .....	53
5.3.3	Pengujian <i>Fuzzy</i> Menggunakan Aplikasi Simulasi .....	57
5.3.4	Perbandingan Hasil Pengujian .....	57
5.4	Pengujian Respon Sistem .....	58
5.5	Pengujian Manuver dan Perbandingan Robot .....	63
5.6	Analisis Pengujian.....	66
BAB VI	KESIMPULAN DAN SARAN.....	69
6.1	Kesimpulan.....	69
6.2	Saran.....	70
DAFTAR PUSTAKA	.....	71
LAMPIRAN	.....	74

## DAFTAR GAMBAR

Gambar 1.1	Hubungan Penelitian.....	5
Gambar 1.2	Kerangka Berpikir.....	7
Gambar 2.1	Sistem Kontrol .....	9
Gambar 2.2	Blok Diagram Sistem Kontrol Loop Terbuka .....	9
Gambar 2.3	Blok Diagram Sistem Kontrol Loop Tertutup.....	10
Gambar 2.4	Tanggapan transien dan <i>steady state</i> .....	10
Gambar 2.5	Struktur Dasar <i>Fuzzy Logic Control</i> .....	12
Gambar 2.6	Representasi Linear Naik.....	13
Gambar 2.7	Representasi Linear Turun.....	13
Gambar 2.8	Representasi Kurva Segitiga .....	14
Gambar 2.9	Representasi Kurva Trapesium .....	15
Gambar 2.10	Representasi Kurva Trapesium .....	16
Gambar 2.11	Fungsi Implikasi MIN.....	17
Gambar 2.12	Fungsi Implikasi DOT .....	18
Gambar 2.13	Konfigurasi pin Arduino Mega 2560 .....	21
Gambar 2.14	Prinsip kerja sensor ultrasonik .....	21
Gambar 2.15	Motor DC.....	23
Gambar 2.16	Motor Driver Shield L293D.....	24
Gambar 2.17	<i>Remote Control</i> 2.4 GHZ.....	25
Gambar 3.1	Blok Diagram Proses Penelitian .....	26
Gambar 4.1	Blok rancangan robot.....	30
Gambar 4.2	Blok diagram sistem kendali robot .....	31
Gambar 4.3	<i>Flowchart</i> program robot.....	31
Gambar 4.4	Pemodelan <i>Fuzzy Logic</i> .....	32
Gambar 4.5	Grafik <i>Membership Function</i> Sensor Ultrasonik .....	33
Gambar 4.6	Grafik <i>Membership Function Remote Control</i> .....	34
Gambar 4.7	Grafik <i>Membership Function</i> Motor DC Kanan.....	36
Gambar 4.8	Grafik <i>Membership Function</i> Motor DC Kiri.....	37
Gambar 4.9	Konfigurasi <i>Motor Driver Shield</i> L293D Dengan Arduino MEGA.....	39
Gambar 4.10	Instalasi Sensor Ultrasonik HC-SR04.....	40

Gambar 4.11	Blok Diagram <i>Transmitter Remote Control</i> .....	41
Gambar 4.12	Blok Diagram <i>Receiver</i> .....	41
Gambar 4.13	Instalasi <i>Receiver 2.4GHz 3 Kanal</i> .....	42
Gambar 4.14	Instalasi Motor DC.....	42
Gambar 4.15	Instalasi Baterai Lipo 11V .....	43
Gambar 4.16	Skematik Rangkaian Robot .....	44
Gambar 4.17	(a) Robot tampak depan, (b) Robot tampak atas.....	47
Gambar 5.1	Grafik kecepatan PWM motor terhadap Waktu.....	51
Gambar 5.2	Pengujian <i>Fuzzy</i> pada sistem robot .....	52
Gambar 5.3	Hasil Perhitungan <i>Fuzzy</i> pada Aplikasi Arduino IDE .....	53
Gambar 5.4	Hasil Perhitungan <i>Fuzzy</i> pada Aplikasi Simulasi .....	57
Gambar 5.5	Grafik respon sistem sensor kecepatan motor Lambat .....	60
Gambar 5.6	Grafik respon sistem sensor kecepatan motor Normal .....	61
Gambar 5.7	Grafik respon sistem sensor kecepatan motor Cepat .....	63
Gambar 5.8	Desain lingkungan pengujian manuver robot .....	64
Gambar 5.9	Kondisi lingkungan pengujian manuver robot.....	65



## DAFTAR TABEL

Tabel 1.1	Referensi Jurnal.....	3
Tabel 2.1	Spesifikasi Arduino MEGA 2560 .....	20
Tabel 4.1	Alat dan Bahan .....	29
Tabel 4.2	Linguistik <i>Input</i> Sensor Ultrasonik .....	32
Tabel 4.3	Linguistik <i>Input Remote Control</i> .....	33
Tabel 4.4	Linguistik <i>Output</i> Motor DC.....	35
Tabel 4.5	<i>Fuzzy Rule Base</i> .....	38
Tabel 4.6	Komponen Robot .....	39
Tabel 4.7	Instalasi Pin <i>Motor Driver Shield</i> L293D .....	40
Tabel 4.8	Instalasi Sensor Ultrasonik HC-SR04.....	41
Tabel 4.9	Instalasi <i>Receiver</i> 2.4GHz 3 Kanal.....	42
Tabel 4.10	Instalasi Motor Kiri .....	43
Tabel 4.11	Instalasi Motor Kanan .....	43
Tabel 4.12	<i>Library</i> program pada Arduino IDE.....	45
Tabel 5.1	Pengujian sensitifitas jarak pantul sensor ultrasonik .....	49
Tabel 5.2	Sensitivitas material pantul .....	49
Tabel 5.3	Pengujian konektivitas robot dengan <i>remote control</i> .....	50
Tabel 5.4	Pengujian Kecepatan PWM Motor.....	51
Tabel 5.5	Hasil Perhitungan <i>Fuzzy</i> .....	57
Tabel 5.6	Hasil respon sistem sensor kecepatan motor Lambat.....	59
Tabel 5.7	Hasil respon sistem sensor kecepatan motor Normal.....	60
Tabel 5.8	Hasil respon sistem sensor kecepatan motor Cepat.....	62
Tabel 5.9	<i>Rules</i> robot tanpa <i>Fuzzy</i> .....	63
Tabel 5.10	Pengujian Pergerakan Robot Dengan <i>Fuzzy</i> .....	65
Tabel 5.11	Pengujian Pergerakan Robot Tanpa <i>Fuzzy</i> .....	66

## BAB I PENDAHULUAN

### 1.1 Latar Belakang Masalah

Perkembangan teknologi pada bidang robotika terus mengalami kemajuan yang semakin pesat sejak ditemukannya piranti semikonduktor. Piranti semikonduktor memungkinkan penciptaan robot yang memiliki ukuran yang lebih kecil, kinerja robot yang lebih baik serta tingkat akurasi robot yang lebih tinggi. *Obstacle avoidance* adalah salah satu syarat utama untuk *mobile robot* termasuk robot navigasi otomatis. Robot penghindar rintangan dirancang untuk memungkinkan robot melakukan navigasi di lingkungan yang tidak diketahui dengan menghindari terjadinya tabrakan [1]. Robot penghindar rintangan akan mendeteksi rintangan yang berada di jalur lalu menghindarinya. Ada beberapa jenis robot yang menggunakan sistem navigasi seperti *wall-trace*, pendeteksi tepi dan *line-following* [1].

Sensor memegang peranan penting dalam sistem pengukur jarak yang dirancang. Sensor yang digunakan untuk pengukuran jarak di antaranya adalah sensor ultrasonik, sensor inframerah dan sinar laser [2]. Sensor inframerah dan sinar laser menggunakan cahaya sebagai media penjarannya, sedangkan sensor ultrasonik menggunakan gelombang akustik sebagai media perantara. Sensor ultrasonik banyak digunakan dalam perancangan sistem pengukur jarak karena sensor ultrasonik tidak terpengaruh terhadap warna objek pemantul, kelunakan objek pemantul, dan aman dari gangguan atau noise gelombang-gelombang lain [2].

Sensor ultrasonik merupakan jenis sensor non-kontak yang dapat digunakan untuk mengukur jarak serta kecepatan suatu benda. Sensor Ultrasonik bekerja berdasarkan sifat-sifat gelombang suara dengan frekuensi lebih besar daripada rentang suara manusia [3]. Dengan menggunakan sensor ultrasonik, robot dapat mendeteksi rintangan yang berada disekitar jalurnya lalu memberikan input untuk segera menghindari objek tersebut. Namun, sistem kendali digital hanya bekerja pada logika 0 dan 1, jalan atau berhenti. Hal itu juga yang menjadi masalah pada mobilitas robot yang akan berhenti secara mendadak sedangkan rintangan yang dihadapi masih berada diluar jalur robot.

Logika *Fuzzy* adalah salah satu pengendali modern yang mampu bekerja baik pada sistem-sistem non-linier dengan menawarkan kemudahan dalam perancangan program karena tidak memerlukan model matematis dari proses [4]. Pengendali

berbasis *Fuzzy* merupakan metode pengendali yang termasuk golongan *Artificial intelligence*. Berbeda dengan kendali PID (*Proportional Integral Derivative*) yang hanya menggunakan perhitungan matematis, pengendali *Fuzzy* menggabungkan perhitungan matematis dan juga algoritma untuk mengendalikan perilaku robot terhadap dinamika lingkungan [5].

Meskipun *automatic navigation robot* ini bekerja secara otomatis tapi pada kasus ini *user* dan *remote control* diasumsikan sebagai sebuah sistem yang memberikan perintah terhadap robot tersebut. Pada beberapa kasus, permasalahan yang muncul adalah *input* yang diberikan untuk robot berlebih sehingga terjadinya salah deteksi dari sistem yang menyebabkan manuver robot yang tidak sempurna bahkan dalam beberapa kasus robot menabrak benda yang berada dijaluannya. Dari permasalahan tersebut maka diperlukan adanya sebuah *warning system* agar robot dapat bertindak ketika ada kesalahan *input* dari sistem yang mengendalikan robot tersebut.

Berdasarkan pada latar belakang tersebut penelitian ini dilakukan perancangan *semi-automatic navigation robot* dengan *remote control* dan sensor ultrasonik berbasis *Fuzzy logic control* sebagai . Pada penelitian ini digunakan sensor ultrasonik sebagai pendeteksi objek dan *Fuzzy logic control* sebagai metode untuk mengontrol kecepatan motor pada robot.

Pada penelitian ini akan diimplementasikan sebuah sistem metode *Fuzzy Logic Control* untuk mengendalikan kecepatan putaran Motor DC yang diakibatkan oleh kelalaian dalam memberikan perintah kepada robot. Pada dasarnya penelitian ini akan diterapkan pada pengendalian menggunakan *computer vision*. Namun peran dari *computer vision* digantikan oleh *user* yang memberikan perintah menggunakan *remote control*.

## **1.2 State of The Art**

Dalam tahap ini akan diuraikan secara singkat penelitian sebagai bentuk memperkuat alasan mengapa penelitian ini akan dilakukan. Berikut referensi jurnal penelitian yang dapat dilihat pada Tabel 1.1.



Tabel 1.1 Referensi Jurnal

Nama Peneliti	Tahun	Judul Penelitian
Kirti Bhagat, Sayalee Deshmukh, Dhonde Shraddha, Sneha Ghag	2016	<i>Obstacle Avoidance Robot.</i>
Andri Dwi Prasetyo, Edi Mulyana, Ir. Neni Utami Adiningsih	2016	Rancang Bangun Sistem Deteksi Halang Rintang Robot Quadcopter Menggunakan Arduino Uno.
Raden Hilman Suryanegara, Eka Purwa Laksana	2019	Rancang Bangun Robot Pembersih Lantai Otomatis Berbasis Mikrokontroler ATmega 16A.
Deni Kurnia	2019	Rancang Bangun Robot Pemadam Api Menggunakan Kontrol <i>Bluetooth</i> dan <i>Virtual Reality</i> .
Zamroni Adeyanto, Ahmad Izzuddin, Nuzul Hikmah	2020	Rancang Bangun Robot <i>Vacuum Cleaner</i> Dengan Menerapkan <i>Propositional Logic</i> Untuk Pengaturan Navigasi.
Alwan Abdul Zaki, Rina Mardiaty, Edi Mulyana, Ulfiah	2020	<i>Modelling Wall Tracer Robot Motion Based On Fuzzy Logic Control.</i>

Penelitian mengenai robot *obstacle avoidance* dan *automatic navigation robot* baik dikendalikan secara otomatis maupun manual dengan pengendali nirkabel telah dilakukan oleh berbagai lembaga, baik universitas maupun lembaga riset. Pada Tabel 1.1 diperlihatkan masing-masing penelitian yang berkaitan dengan navigasi otomatis dan penghindaran halang rintang.

Penelitian yang dilakukan oleh Kirti Bhagat di India pada tahun 2016 membahas tentang rancang bangun robot yang menghindari rintangan menggunakan sensor ultrasonik untuk mengatur pergerakannya. Mikrokontroler mampu mendeteksi lingkungan sekitar melalui masukan dari sensor. Keluaran dari komparator diberikan ke mikrokontroler yang kemudian menggerakkan aktuator ke arah kiri dan kanan dengan mengalirkan daya melalui motor DC [1].

Penelitian kedua dilakukan pada tahun 2016 oleh Andri Dwi Prasetyo dkk. dengan meneliti deteksi halang rintang pada robot quadcopter. Penelitian ini bertujuan untuk merancang bangun alat dengan cara membuat perangkat keras

dan perangkat lunak serta pengolahan data tersebut. Sistem deteksi halang rintang robot quadcopter menggunakan Arduino uno Atmega 328P-PU, Sensor Ultrasonik, *Receiver Remote Control*, dan *Flight Controller*. Dapat mendeteksi halang rintang dan bermanuver sebaliknya/menghindar. Sensor Ultrasonik sebagai pendeteksi halang rintang, *Receiver Remote Control* sebagai penerima sinyal yang diberikan oleh *remote control* yang menghasilkan sinyal berupa pulsa pwm (*pulse width modulation*). Arduino Uno berperan sebagai pengolah data yang diberikan oleh input dari Sensor Ultrasonik dan *Receiver Remote Control*. Hasil data yang diolah akan dikirimkan pada *Flight Controller* yang berperan sebagai penerima sinyal berupa pulsa pwm [6].

Raden Hilman S dan Eka Purwa Laksanan melakukan penelitian robot pembersih lantai otomatis dengan menggunakan ATmega 16A sebagai mikrokontroler dan sensor HC-SR04 untuk mendeteksi halangan didepan robot. Robot akan membersihkan lantai mulai dari pojok ruangan dengan penggunaan yang menentukan posisi awal robot. Sistem akan berjalan dengan proses penyapuan, penyedotan, pengepelan dan pengeringan. Setelah robot melakukan pembersihan dengan mengelilingi ruangan secara otomatis, maka pengguna akan menonaktifkan robot [7].

Pada penelitian keempat Zamroni Adeyanto dkk merancang dan membangun sebuah robot *vacuum cleaner* yang dapat bernavigasi secara otomatis dan menerapkan logika *propositional logic* pada robot sehingga robot dapat kembali ke jalur setelah melewati rintangan agar tidak terdapat daerah yang tidak dilewati oleh robot *vacuum cleaner* tersebut [8].

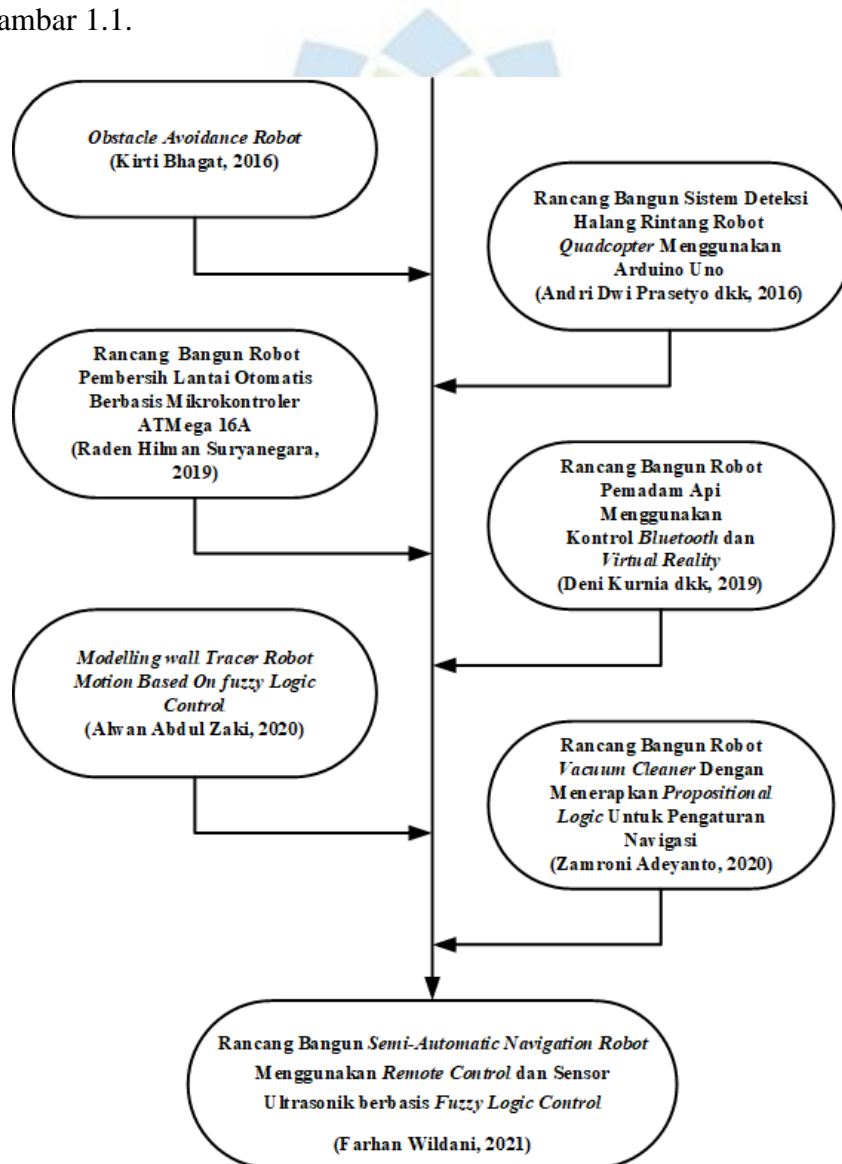
Pada penelitian yang dilakukan oleh Deni Kurnia dkk., dibuat rancang bangun robot pemadam api menggunakan kontrol bluetooth dan Virtual Reality dimana Rancang bangun robot ini menggunakan mikrokontroler Arduino Uno, pada sistem kontrol dari robot ini menggunakan bluetooth dan kamera Virtual Reality, sensor ultrasonik disini sebagai sistem proteksi robot agar tidak menabrak halangan dan Virtual Reality Box sebagai piranti penglihatan kamera robot untuk pengguna robot dalam menelusuri ruangan dan mencari sumber api, setelah mendeteksi api robot secara otomatis akan memadamkan api tersebut. [9].

Penelitian terakhir yang dilakukan oleh Alwan Abdul Zaki pada tahun 2020 menerapkan metode *Fuzzy* dengan tipe Mamdani sebagai sistem kendali dan

Mikrokontroler Arduino Uno sebagai controller lalu sensor Ultrasonik sebagai nilai input jarak dapat menghindari halangan yang dipasang pada masing- masing lintasan dengan baik [10].

Berdasarkan penelitian yang telah dilakukan sebelumnya, pada penelitian ini akan dilakukan penelitian untuk meningkatkan efisiensi kinerja motor menggunakan metode *Fuzzy logic* yang mampu meningkatkan respon sistem dengan baik serta memasang sensor pada robot.

*State of The Arts* penelitian ini menggunakan satu rujukan jurnal internasional dan empat jurnal nasional yang berhubungan dengan penelitian ini diperlihatkan pada Gambar 1.1.



Gambar 1.1 Hubungan Penelitian

### 1.3 Rumusan Masalah

Berdasarkan latar belakang permasalahan, ada beberapa masalah yang perlu dirumuskan:

1. Bagaimana rancang bangun *semi-automatic navigation robot* menggunakan *remote control* dan sensor ultrasonik berbasis *Fuzzy logic control*?
2. Bagaimana kinerja dari *semi-automatic navigation robot* yang menggunakan metode *Fuzzy logic control*?

### 1.4 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

1. Implementasi *semi-automatic navigation robot* menggunakan *remote control* dan sensor ultrasonik dengan berbasis *Fuzzy logic control*.
2. Melakukan analisis hasil kinerja dari *semi-automatic navigation robot* yang menggunakan metode *Fuzzy logic control*.

### 1.5 Manfaat Penelitian

Dari penelitian ini diharapkan ada manfaat yang didapatkan oleh semua kalangan. Berikut manfaat penelitian yang diharapkan :

#### 1.5.1 Manfaat Akademis

Adapun manfaat akademis yang didapatkan dari penelitian ini dapat berkontribusi dalam memperkaya khasanah keilmuan pada bidang kontrol khususnya metode *Fuzzy logic*.

#### 1.5.2 Manfaat Praktis

Adapun manfaat praktis yang didapatkan dari penelitian ini adalah dapat memberikan solusi kepada pekerjaan manusia dalam melakukan proses navigasi robot.

### 1.6 Batasan Masalah

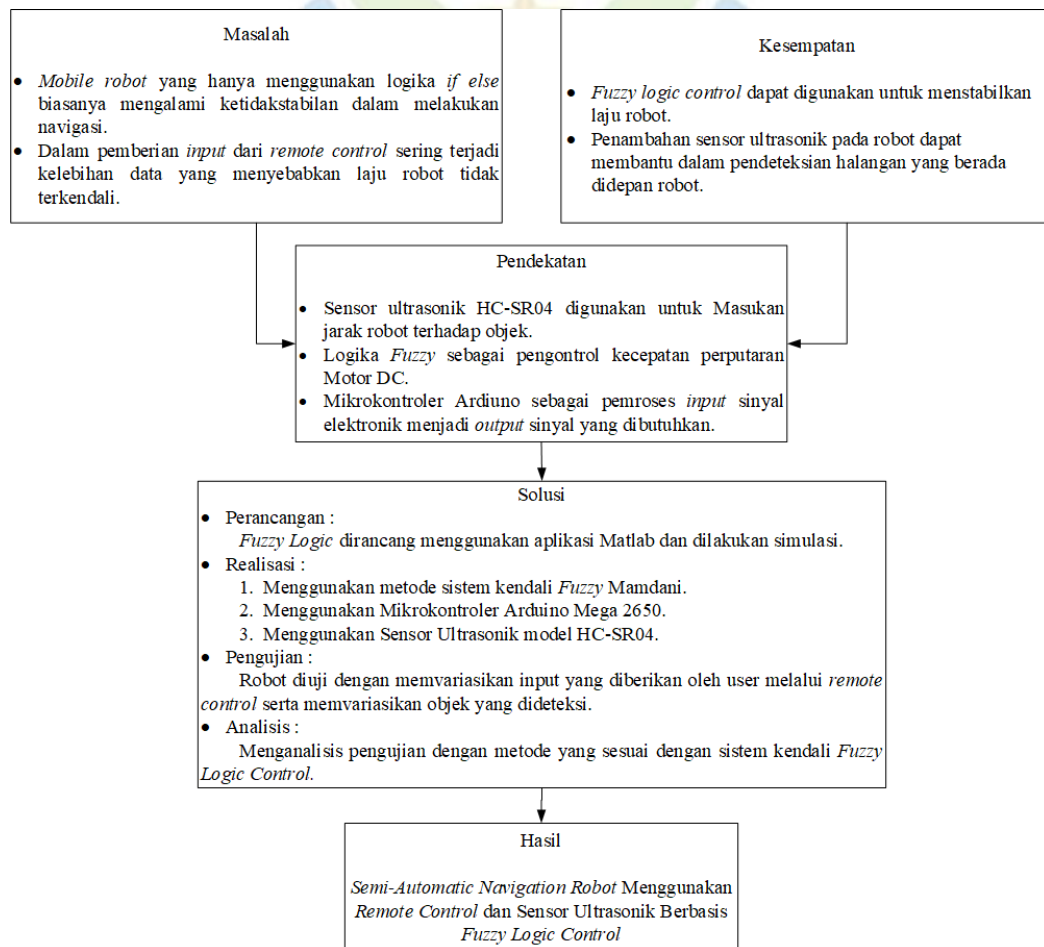
Penulisan akan dibatasi pada bagian berikut :

1. Sensor ultrasonik yang digunakan adalah tipe HC-SR04.
2. Menggunakan *Remote Control* tiga kanal dengan frekuensi 2,4 GHz.
3. *Driver motor* dengan arus 2A dan tegangan 5V.
4. Motor DC dengan tegangan 12V.

5. Menggunakan Arduino Mega dengan mikrokontroler ATmega 2560.
6. Bahasa C sebagai bahasa pemrograman yang digunakan.
7. Arduino IDE untuk meng-*compile* program.
8. Robot ini dikendalikan dengan *remote control* oleh *user* namun tetap mendapatkan *input* dari sensor maka dari itu dinamakan robot *semi-automatic*.
9. Robot hanya akan berbelok ke arah kanan ketika bermanuver untuk menghindari *obstacle*
10. *User* dan *Remote control* diasumsikan sebagai sistem yang mengendalikan robot.

## 1.7 Kerangka Berpikir

Kerangka berpikir dijelaskan pada Gambar 1.2.



Gambar 1.2 Kerangka Berpikir

## **1.8 Sistematika Penulisan**

Sistematika penulisan tugas akhir ini terdiri dari tiga bab yang menguraikan permasalahan yang dibahas. Berikut sistematika penulisan proposal tugas akhir ini:

### **BAB I PENDAHULUAN**

Pada bab ini meliputi latar belakang, *State of The Art*, rumusan masalah, tujuan, manfaat, batasan masalah, kerangka berpikir dan sistematika penulisan.

### **BAB II TEORI DASAR**

Pada bab ini menjelaskan tentang hal-hal pokok sebelum melakukan penelitian. Karena menyangkut dengan penelitian perlu adanya penguasaan teori yang berhubungan dengan penelitian yang dilakukan mengenai penelitian ini.

### **BAB III METODOLOGI PENELITIAN**

Pada bab ini menguraikan tentang metode penelitian yang digunakan pada penelitian ini dan jadwal penelitian untuk proposal penelitian Rancang Bangun *Semi-Automatic Navigation Robot* menggunakan *Remote Control* dan Sensor Ultrasonik berbasis *Fuzzy Logic Control*.

### **BAB IV PERANCANGAN DAN IMPLEMENTASI ALAT**

Pada bab ini menjelaskan alur tahap-tahap perancangan, mulai dari analisis kebutuhan hingga implementasi baik dalam segi *software* maupun *hardware* untuk rancang bangun *semi-automatic navigation robot* menggunakan *remote control* dan sensor ultrasonik berbasis *Fuzzy logic control*.

### **BAB V PENGUJIAN DAN ANALISIS**

Pada bab ini memaparkan hasil pengujian-pengujian yang telah dilakukan serta menganalisis data yang diperoleh pada saat pengujian *semi-automatic navigation robot* menggunakan *remote control* dan sensor ultrasonik berbasis *Fuzzy logic control*.

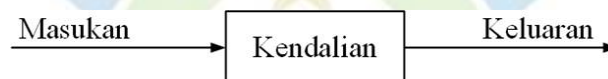
### **BAB VI PENUTUP**

Pada bab ini menjelaskan tentang bagian penutup dari penelitian. Pada bagian ini terdapat kesimpulan, serta saran untuk penelitian-penelitian selanjutnya.

## BAB II TEORI DASAR

### 2.1 Sistem Kendali

Sistem kendali atau sistem kontrol (*control system*) merupakan suatu alat untuk mengendalikan, memerintah dan mengatur keadaan dari suatu sistem. Sistem kendali dapat dikatakan sebagai hubungan antara komponen yang membentuk sebuah konfigurasi sistem, yang akan menghasilkan tanggapan sistem yang diharapkan. Masukan dan keluaran merupakan variabel atau besaran fisis. Keluaran adalah suatu hal yang dapat dihasilkan oleh kendalian, artinya yang dikendalikan, sedangkan masukan adalah suatu hal yang mempengaruhi kendalian, yang mengatur keluaran [11]. Sistem kontrol dibedakan atas 2 bagian, yaitu : sistem kontrol loop terbuka (*Open loop control*) dan sistem kontrol loop tertutup (*Close loop control*). Berikut blok diagram suatu sistem kontrol dapat dilihat pada Gambar 2.1[12].



Gambar 2.1 Sistem Kontrol

#### 2.1.1 Sistem Kontrol Loop Terbuka

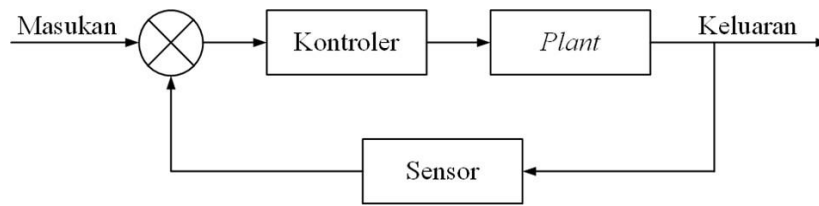
Sistem kontrol loop terbuka adalah suatu sistem kontrol yang keluarannya tidak mempengaruhi aksi dari pengontrolnya. Keluaran pada sistem kontrol loop terbuka tidak diukur dibandingkan dengan masukannya. Berikut blok diagram sistem kontrol loop terbuka dapat dilihat pada Gambar 2.2[12].



Gambar 2.2 Blok Diagram Sistem Kontrol Loop Terbuka

#### 2.1.2 Sistem Kontrol Loop Tertutup

Sistem kontrol loop tertutup merupakan suatu sistem yang sinyal keluarannya mempunyai pengaruh langsung terhadap aksi pengontrolan. Dengan kata lain istilah loop tertutup berarti menggunakan istilah umpan balik untuk memperkecil kesalahan pada sistem. Berikut blok diagram sistem kontrol loop tertutup dapat dilihat pada Gambar 2.3[12].



Gambar 2.3 Blok Diagram Sistem Kontrol Loop Tertutup

## 2.2 Respon Sistem

Respon sistem merupakan perubahan kondisi *output* terhadap perubahan sinyal *input*. Respon sistem pada sistem kendali dapat dibagi menjadi dua bagian dilihat dari segi waktunya, yaitu tanggapan transien (*transient response*) dan tanggapan keadaan tunak (*steady state response*). Tanggapan transien terjadi pada saat bermula hingga tanggapan sistem berada pada nilai yang diinginkan. Tanggapan keadaan tunak/*steady state response* dimulai ketika tanggapan mendekati nilai yang diinginkan sampai waktu yang tak terhingga. Jadi respon sistem  $c(t)$  dapat ditulis :

$$c(t) = c_{tr} + c_{ss} \quad (2.1)$$

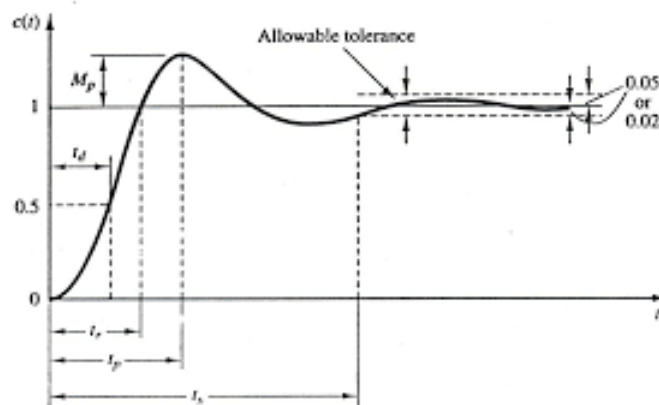
Keterangan :

$c(t)$  = Respon Sistem

$c_{ss}$  = Respon Transien Sistem

$c_{ss}$  = Respon *Steady State* Sistem

Gambar 2.4 memperlihatkan tanggapan transien dan tanggapan *steady state* [12].



Gambar 2.4 Tanggapan transien dan *steady state*



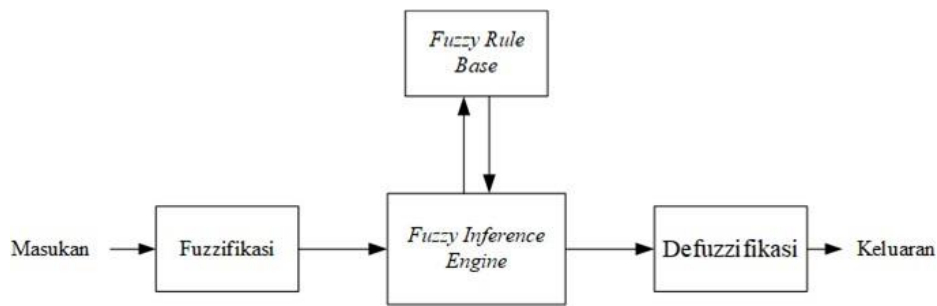
### 2.3 *Fuzzy Logic Control*

*Fuzzy* secara bahasa diartikan sebagai kabur atau samar-samar. Suatu nilai dapat bernilai besar atau salah secara bersamaan. Dalam *Fuzzy* dikenal derajat keanggotaan (*membership degree*) yang memiliki rentang nilai 0 (nol) hingga 1 (satu). Berbeda dengan himpunan tegas yang hanya memiliki nilai 1 atau 0 (ya atau tidak). Logika *Fuzzy* merupakan sesuatu logika yang memiliki nilai kekaburan atau kesamaran (*Fuzzyness*) antara benar atau salah. Dalam teori logika *Fuzzy* suatu nilai bias bernilai benar atau salah secara bersama. Namun berapa besar keberadaan dan kesalahan suatu tergantung pada bobot keanggotaan yang dimilikinya.

Logika *Fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang input kedalam suatu ruang output, mempunyai nilai kontinyu. *Fuzzy* dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang sama [13].

Logika *Fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang input kedalam suatu ruang output, mempunyai nilai kontinyu. *Fuzzy* dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang sama [14]. Logika *Fuzzy* memungkinkan nilai keanggotaan antara 0 dan 1, tingkat keabuan dan juga hitam dan putih, dan dalam bentuk linguistik, konsep tidak pasti seperti "sedikit", "lumayan" dan "sangat" [13]. Pada operasi himpunan *Fuzzy* diperlukan proses inferensi atau penalaran. Berikut adalah beberapa hal yang perlu dipahami dalam sistem kendali *Fuzzy* [13]:

1. Variabel *Fuzzy* merupakan variabel yang akan dibahas dalam suatu sistem *Fuzzy*, seperti : kecepatan, suhu, kelembaban dan sebagainya.
2. Himpunan *Fuzzy* merupakan suatu grup yang mewakili suatu kondisi tertentu dalam suatu variabel *Fuzzy*.
3. Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *Fuzzy*.
4. Domain himpunan *Fuzzy* adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan, domain merupakan himpunan bilangan riil yang senantiasa naik secara monoton dari kiri ke kanan dan dapat berupa bilangan negatif maupun positif.



Gambar 2.5 Struktur Dasar *Fuzzy Logic Control*

Pada Gambar 2.5 ditunjukkan struktur dasar dari sistem kendali *Fuzzy*. Berikut penjelasan dari struktur atau elemen dasar dari logika *Fuzzy* [13]:

1. *Knowledge base* atau *rule base*, berisi aturan-aturan secara linguistik yang bersumber dari para pakar.
2. Mekanisme pengambil keputusan (*inference engine*), merupakan bagaimana para pakar mengambil suatu keputusan dengan menerapkan pengetahuan (*knowledge*).
3. Proses fuzzifikasi (*fuzzifier*), yaitu mengubah nilai dari himpunan tegas ke nilai *Fuzzy*.
4. Proses defuzzifikasi (*defuzzifier*), yaitu mengubah nilai *Fuzzy* hasil inferensi menjadi nilai tegas.

## 2.4 *Fuzzy Logic Control* Metode Mamdani

Metode Mamdani sering juga dikenal dengan nama metode MIN - MAX. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Terdapat 4 tahapan untuk mendapatkan *output* pada metode *Fuzzy*, yaitu [15]:

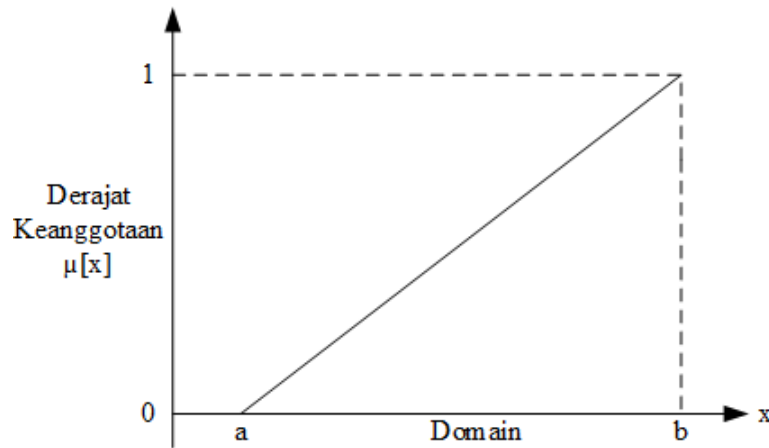
### 2.4.1 Pembentukan Himpunan *Fuzzy*

Berdasarkan metode Mamdani, variabel *input* dan *output* dibagi menjadi satu atau lebih himpunan *Fuzzy*, dan setiap variabel *input* dan *output* memiliki variabel linguistik. Fungsi keanggotaan adalah kurva yang menunjukkan penetapan titik masukan data dalam nilai keanggotaannya dengan interval antara 0 dan 1 [13]. Ada beberapa fungsi yang bisa digunakan, yaitu :

1. Representasi Linier

Pada representasi linier, pemetaan masukan ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Ada dua keadaan himpunan *Fuzzy*

linier, yaitu linier naik dan linier turun. Gambar 2.6 merepresentasikan himpunan *Fuzzy* linier naik [13].



Gambar 2.6 Representasi Linear Naik

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0 & ; x \geq a \\ \frac{x-a}{b-a} & ; a \geq x \geq b \\ 1 & ; x \leq b \end{cases} \quad (2.2)$$

Keterangan :

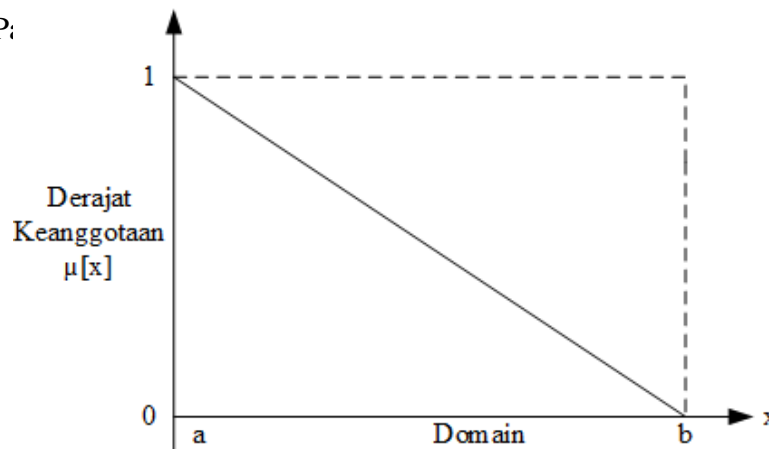
a = nilai *domain* yang mempunyai derajat keanggotaan nol

b = nilai *domain* yang mempunyai derajat keanggotaan satu

x = nilai *input* yang akan diubah ke dalam bilangan *Fuzzy*

P:

un [13].



Gambar 2.7 Representasi Linear Turun

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} \frac{b-x}{b-a} & ; a \geq x \geq b \\ 0 & ; x \leq b \end{cases} \quad (2.3)$$

Keterangan :

a = nilai *domain* yang mempunyai derajat keanggotaan satu

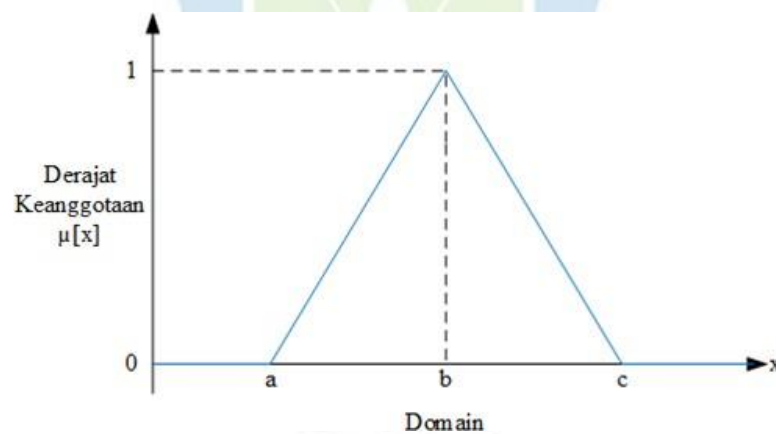
b = nilai *domain* yang mempunyai derajat keanggotaan nol

x = nilai *input* yang akan diubah ke dalam bilangan *Fuzzy*

## 2. Representasi Kurva Segitiga

Pada dasarnya kurva segitiga merupakan gabungan antara 2 garis (linier).

Untuk representasinya dapat dilihat pada Gambar 2.8 [13].



Gambar 2.8 Representasi Kurva Segitiga

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0 & ; x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a} & ; a \leq x \leq b \\ \frac{c-x}{c-b} & ; b \leq x \leq c \end{cases} \quad (2.4)$$

Keterangan :

a = nilai *domain* terkecil yang mempunyai derajat keanggotaan nol

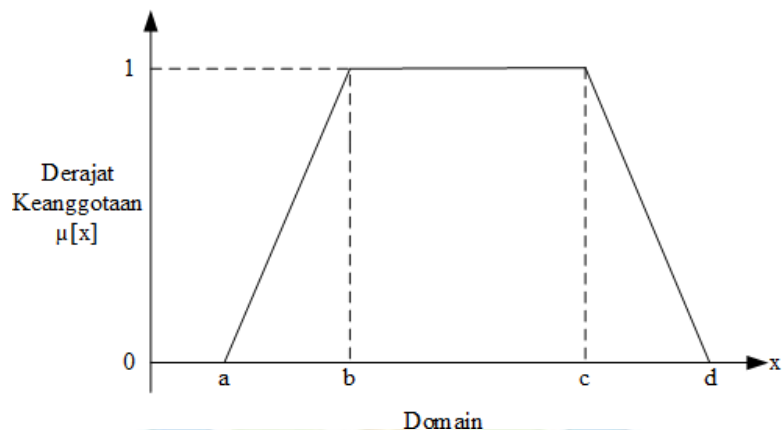
b = nilai *domain* yang mempunyai derajat keanggotaan satu

c = nilai *domain* terbesar yang mempunyai derajat keanggotaan nol

x = nilai *input* yang akan diubah ke dalam bilangan *Fuzzy*

### 3. Representasi Kurva Trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga karena merupakan gabungan dari dua garis (linier), hanya terdapat beberapa titik dengan nilai keanggotaan 1. Representasi kurva trapesium dapat dilihat pada Gambar 2.9 [13].



Gambar 2.9 Representasi Kurva Trapesium

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0 & ; x \leq a \text{ atau } x \geq d \\ \frac{x-a}{b-a} & ; a < x < b \\ 1 & ; b \leq x \leq c \\ \frac{d-x}{d-c} & ; x \geq c \end{cases} \quad (2.5)$$

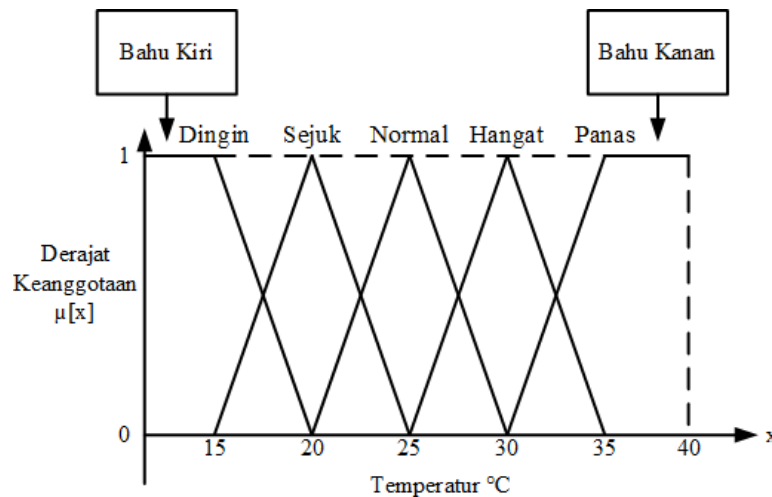
Keterangan :

- a = nilai *domain* terkecil yang mempunyai derajat keanggotaan nol
- b = nilai *domain* terkecil yang mempunyai derajat keanggotaan satu
- c = nilai *domain* terbesar yang mempunyai derajat keanggotaan satu
- d = nilai *domain* terbesar yang mempunyai derajat keanggotaan nol
- x = nilai *input* yang akan diubah ke dalam bilangan *Fuzzy*

### 4. Representasi Kurva Bahu

Bentuk kurva bahu berbeda dari kurva segitiga di mana satu sisi variabel berubah naik atau turun dan sisi lainnya tidak. Himpunan *Fuzzy* bahu juga digunakan untuk mengakhiri variabel suatu daerah *Fuzzy*. Bahu kiri bergerak dari arah yang benar ke arah yang salah, sedangkan bahu kanan

bergerak dari arah yang salah ke arah yang benar. Gambar 2.10 menunjukkan variabel temperatur dengan daerah bahunya [13].



Gambar 2.10 Representasi Kurva Trapesium

#### 2.4.2 Aplikasi Fungsi Implikasi

Setelah variabel *input* dan *output* didapatkan, langkah selanjutnya pada metode Mamdani adalah menentukan aplikasi fungsi implikasi. Fungsi implikasi adalah struktur logis yang terdiri dari seperangkat premis dan kesimpulan. Fungsi implikasi berguna untuk mengetahui hubungan antara premis dan kesimpulan. Bentuk dari fungsi implikasi ini adalah dengan pernyataan *IF is A THEN is B*, dengan  $x$  dan  $y$  merupakan skalar, serta  $A$  dan  $B$  merupakan himpunan *Fuzzy* [16]. Dalam istilah logika *Fuzzy*, proposisi yang mengikuti *IF* disebut dengan antisenden, sedangkan proposisi yang mengikuti *THEN* disebut dengan konsekuen. Proposisi atau aturan *Fuzzy* ini dapat diperluas dengan menggunakan penghubung *Fuzzy AND* (interseksi).

Pada *Fuzzy Mamdani*, ada dua fungsi implikasi yang dapat digunakan, yaitu *MIN* dan *DOT*. Pada Gambar 2.11 dan Gambar 2.12 studi kasus yang diambil adalah *IF* permintaan *TINGGI AND* biaya produksi *SEDANG THEN* produksi barang *NORMAL*. Berikut adalah contoh penggunaan fungsi implikasi *MIN* dan *DOT* :

1. *MIN* (Minimum)

Pengambilan keputusan dengan fungsi min, yaitu dengan cara mencari nilai minimum berdasarkan aturan ke- $i$  dan dapat dinyatakan dengan :

$$\alpha_i \cap \mu_{C_i}(Z) \tag{2.6}$$

dimana :

$$\alpha_i = \mu_{A_i}(x) \cap \mu_{B_i}(x) = \min \{ \mu_{A_i}(x), \mu_{B_i}(x) \} \quad (2.7)$$

Keterangan:

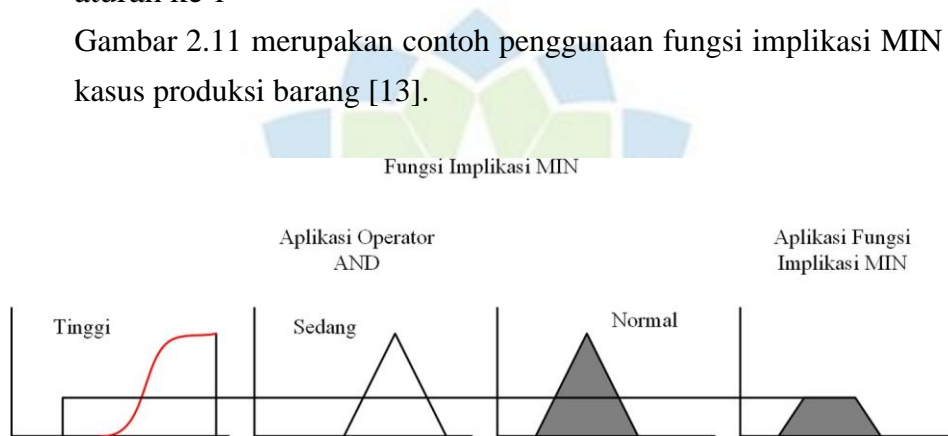
$\alpha_i$  = nilai minimum dari himpunan Fuzzy A dan B pada aturan ke-i

$\mu_{A_i}(x)$  = derajat keanggotaan x dari himpunan Fuzzy A pada aturan ke-i

$\mu_{B_i}(x)$  = derajat keanggotaan x dari himpunan Fuzzy B pada aturan ke-i

$\mu_{C_i}(x)$  = derajat keanggotaan konsekuen pada himpunan Fuzzy C pada aturan ke-i

Gambar 2.11 merupakan contoh penggunaan fungsi implikasi MIN pada kasus produksi barang [13].



Gambar 2.11 Fungsi Implikasi MIN

## 2. DOT (*Product*)

Pengambilan keputusan dengan fungsi DOT yang didasarkan pada aturan ke-i dinyatakan dengan :

$$\alpha_i \cdot \mu_{C_i}(Z) \quad (2.8)$$

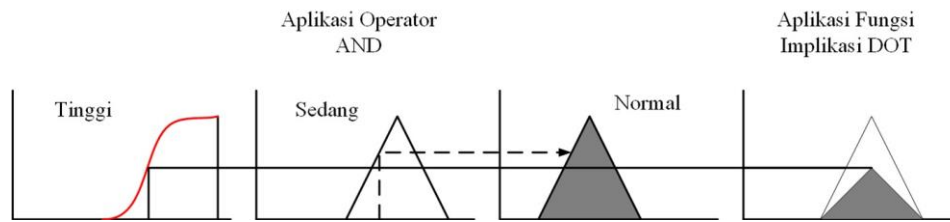
Keterangan :

$\alpha_i$  = nilai minimum dari himpunan Fuzzy A dan B pada aturan ke-i

$\mu_{C_i}(x)$  = derajat keanggotaan konsekuen pada himpunan Fuzzy C pada aturan ke-i.

Penggunaan fungsi DOT dapat dilihat pada Gambar 2.12 dengan studi kasus yang sama dengan fungsi MIN [13].

### Fungsi Implikasi DOT



Gambar 2.12 Fungsi Implikasi DOT

### 2.4.3 Komposisi Aturan

Setelah didapatkan hasil dari fungsi implikasi, langkah selanjutnya adalah menentukan komposisi dari setiap aturan dan metode yang digunakan untuk melakukan inferensi sistem *Fuzzy* yaitu metode MAX.

Solusi himpunan *Fuzzy* dalam metode ini diperoleh dengan mengambil nilai maksimum dari aturan, kemudian menggunakannya untuk memodifikasi daerah *Fuzzy* dan menerapkannya pada *output* dengan menggunakan operator OR (*union*). Ketika semua proposisi telah ditinjau, *output* akan berisi himpunan *Fuzzy* yang mewakili kontribusi setiap proposisi. Secara umum dapat dirumuskan sebagai berikut :

$$U_{s_f}[x_i] = \max(U_{s_f}[x_i], U_{k_f}[x_i]) \quad (2.9)$$

Keterangan :

$U_{s_f}[x_i]$  = nilai keanggotaan solusi *Fuzzy* sampai aturan ke-i

$U_{k_f}[x_i]$  = nilai keanggotaan konsekuen *Fuzzy* aturan ke-i

### 2.4.4 Defuzzifikasi

Proses defuzzifikasi dipergunakan untuk menafsirkan nilai keanggotaan *Fuzzy* menjadi keputusan tertentu atau bilangan real [13]. *Input* dari proses defuzzifikasi adalah himpunan *Fuzzy*, sedangkan *output* yang dihasilkan berupa bilangan pada domain himpunan *Fuzzy*. Metode yang digunakan dalam proses defuzzifikasi ini adalah metode defuzzifikasi *Centroid* (titik pusat). Metode ini memperhatikan kondisi setiap area yang buram untuk menghasilkan hasil yang lebih akurat [16]. Metode *centroid* dapat disebut sebagai *center of area* (pusat gravitasi) dan merupakan metode yang paling umum dan banyak diusulkan digunakan oleh banyak peneliti. Metode *centroid* yaitu suatu metode dimana semua daerah *Fuzzy* dari hasil komposisi aturan digabungkan dengan tujuan untuk membentuk hasil yang optimal dan mengambil titik pusat daerah *Fuzzy*. Solusi dalam metode ini



diperoleh dengan mengambil pusat dari area *Fuzzy*. Metode ini dapat dirumuskan sebagai berikut :

$$Z = \frac{\int_a^b z \cdot \mu_{(z)} dz}{\int_a^b \mu_{(z)} dz} \text{ untuk domain kontinyu} \quad (2.10)$$

Keterangan :

$Z$  = nilai domain ke- $i$

$\mu_{(z)}$  = derajat keanggotaan titik tersebut

$Z_0$  = nilai hasil defuzzifikasi

## 2.5 Mikrokontroler

Mikrokontroler adalah sebuah komputer kecil yang dikemas dalam bentuk chip IC (Integrated Circuit) dan dirancang untuk melakukan tugas atau operasi tertentu. Pada dasarnya, sebuah IC Mikrokontroler terdiri dari satu atau lebih Inti Prosesor (CPU), Memori (RAM dan ROM) serta perangkat *INPUT* dan *OUTPUT* yang dapat diprogram [17]. Dalam pengaplikasiannya, Pengendali Mikro yang dalam bahasa Inggris disebut dengan *Microcontroller* ini digunakan dalam produk ataupun perangkat yang dikendalikan secara otomatis seperti sistem kontrol mesin mobil, perangkat medis, pengendali jarak jauh, mesin, peralatan listrik, mainan dan perangkat-perangkat yang menggunakan sistem tertanam lainnya.

## 2.6 Arduino Mega 2560

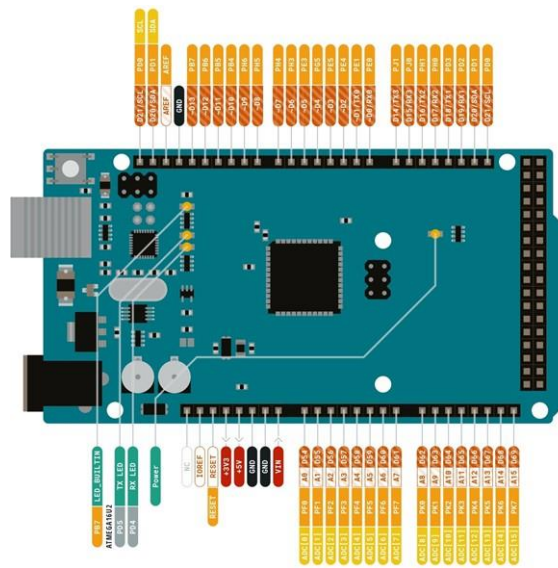
Arduino adalah pengendali mikro *singleboard* yang bersifat *open-source* yang dirancang untuk memudahkan penggunaan elektronik berbagai bidang. *Controller* arduino menggunakan keluarga mikrokontroler Atmega yang dirilis oleh Atmel. Penggunaan *controller* arduino menggunakan bahasa pemrograman tersendiri, namun masih memiliki kemiripan *syntax* dengan bahasa pemrograman C. Arduino mega 2560 menggunakan Atmega 2560 sebagai prosesornya. Arduino mega 2560 memiliki pin *input* dan *output* sebanyak 54 pin I/O, yang terdiri dari 16 pin analog *input/output* berlabel A0 sampai A15 sebagai ADC dengan resolusi 10 bit [10]. Pada Tabel 2.1 dan Gambar 2.13 menunjukkan spesifikasi dan konfigurasi pin Arduino MEGA 2560.

Tabel 2.1 Spesifikasi Arduino MEGA 2560

Mikrokontroler	ATMega 2560.
Tegangan Pengoperasian	5 V.
Tegangan <i>Input</i> yang disarankan	7-12 V.
Batas Tegangan <i>Input</i>	6-20 V.
Jumlah pin I/O digital	54 pin digital (15 diantaranya menyediakan keluaran PWM)
Jumlah pin input Analog	16 pin.
Arus DC tiap pin I/O	20 mA.
Arus DC untuk pin 3,3 V	50 mA.
Memori <i>Flash</i>	256 KB (ATMega 2560) sekitar 8 KB digunakan oleh <i>bootloader</i> .
SRAM	8 KB (ATMega 2560).
EPROM	4 KB (ATMega 2560).
<i>Clock Speed</i>	16 Mhz.

Pada arduino mega 2560 pin-pin yang konfigurasi yang berada pada board *interface* arduino tersebut memiliki fungsi masing-masing, antara lain adalah sebagai berikut [10] :

1. Serial : 0(RX) dan 1(TX), 19(RX) dan 18(TX), 17(RX) dan 16(TX), dan 15(RX) dan 14(TX), Digunakan untuk menerima (RX) dan mengirim (TX) TTL Data Serial.
2. *Interupt Eksternal Interupts* : pin 2, pin 3, pin 18, pin 19, pin 20, dan 21. Pin dapat dikonfigurasi untuk men-*trigger* sebuah *interupt* pada *lowvalue*, *rising* atau *falling-edge*.
3. PWM: 3,5,6,9,10,11,12,13 dan 44,45,46. Mendukung 8- *bit output* PWM dengan fungsi *analog write*.
4. SPI : 53 (SS), 51 (MOSI), 50 (MISO), 52 (SCK).pin ini men-*textsupport* komunikasi SPI, yan mana masih mendukung *hardware*, yang tidak termasuk pada bahasa Arduino.
5. LED : 13 adalah dibuat untuk koneksi LED ke digital pin 13. Ketika pin bernilai *HIGH LED* hidup, dan ketika pin *LOW LED* mati.



3. Tegangan kerja 5V DC.
4. Resolusi 1 cm.
5. Frekuensi Ultrasonik 40 kHz.
6. Dapat dihubungkan langsung ke kaki mikrokontroler.

## 2.8 Bahasa C

Bahasa pemrograman C merupakan salah satu bahasa pemrograman komputer. Dibuat pada tahun 1972 oleh Dennis Ritchie untuk Sistem Operasi Unix di Bell Telephone Laboratories. Meskipun Bahasa C dibuat untuk memprogram sistem dan jaringan komputer namun bahasa ini juga sering digunakan dalam mengembangkan software aplikasi. Bahasa C juga banyak dipakai oleh berbagai jenis platform sistem operasi dan arsitektur komputer, bahkan terdapat beberapa compiler yang sangat populer telah tersedia. Bahasa C secara luar biasa memengaruhi bahasa populer lainnya, terutama C++ yang merupakan ekstensi dari C. Adapun struktur dasar bahasa C yaitu [19] :

1. *Header*, berguna untuk mengisi dengan *library* sesuai yang akan digunakan dan harus sesuai dengan program yang sedang digunakan untuk koding saat itu.
2. Tipe data, merupakan program yang paling penting, karena dapat mempengaruhi setiap intruksi yang akan dieksekusi oleh komputer.
3. Konstanta, merupakan nilai yang tidak dapat diubah selama proses program berlangsung. Konstanta harus didefinisikan terlebih dahulu diawal program.
4. Fungsi *main* (), merupakann fungsi yang wajib ada, karena titik awal dan akhir eksekusi program.
5. Praprosesor include, merupakan pembaca suatu file yang diantaranya berisi deklarasasi fungsi dan definisi konstanta.
6. Komentar, digunakan untuk keperluan dokumentasi agar program mudah dipahami.

Bahasa C memiliki kelebihan yang bersifat portabel dan fleksibel untuk semua jenis komputer. Bahasa C hanya menyediakan sedikit kata-kata kunci. Proses *executable* program bahasa C lebih cepat dan juga dukungan pustaka yang banyak.

## 2.9 Motor DC (*Direct Current*)

Motor DC adalah peralatan elektronika dasar yang memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Dalam motor DC terdapat dua kumparan yaitu kumparan medan yang berfungsi untuk menghasilkan medan magnet dan kumparan jangkar yang berfungsi sebagai tempat terbentuknya gaya gerak listrik (GGL E). Jika arus dalam kumparan jangkar berinteraksi dengan medan magnet, akan timbul torsi yang akan memutar motor berfungsi untuk mengubah tenaga listrik menjadi tenaga mekanik yang desain awalnya diperkenalkan oleh Michel Faraday lebih dari seabad yang lalu [3]. Gambar 2.15 menunjukkan Motor DC.



Sumber : [www.hsiangneng.com](http://www.hsiangneng.com)

Gambar 2.15 Motor DC

Pada prinsipnya motor DC memiliki dua bagian dasar yaitu :

1. Bagian yang tetap atau stasioner disebut stator, stator ini menggunakan medan magnet, baik yang dibangkitkan dari sebuah koil (elektromagnetik) atau magnet permanen.
2. Bagian yang berputar disebut rotor atau armature. Rotor ini berupa sebuah koil dimana arus listrik mengalir. Jenis motor dibedakan berdasarkan pengaturan listrik dan konstruksi.

Motor DC bekerja bila pada kedua kaki stator diberikan tegangan atau beda potensial sehingga pada rotor terjadi perubahan energi listrik menjadi energi mekanik yang mengakibatkan terjadinya tolak menolak antara rotor dan stator, karena kedua bagian ini dipasang secara berdampingan dengan kutub yang berbeda. Dari tolak menolak yang terjadi pada kedua magnet yang ditimbulkan oleh stator dan rotor maka terjadi suatu pergerakan yang mengakibatkan rotor akan berputar

sesuai dengan pemberian tegangan pada kaki stator [3].

### 2.10 Motor Driver Shield L293D

IC L293D yang terdapat pada motor *driver shield* adalah IC yang dirancang secara khusus sebagai pengendali motor DC dan dapat dikendalikan dengan rangkaian mikrokontroler. Motor DC yang dikendalikan dengan *driver* IC L293D dapat dihubungkan ke *ground* maupun ke sumber tegangan positif karena didalam sistem *driver* L293D yang digunakan adalah *totem pool*. Dalam satu buah IC L293D terdiri dari 4 buah *driver* motor DC yang berdiri sendiri-sendiri. Sehingga dapat digunakan untuk membuat *driver H-bridge* untuk 2 buah motor DC. Modul arduino motor *driver shield* L293D ini menggunakan dua buah IC L293D sehingga terdapat empat buah *H-bridge* yang dapat digunakan mengendalikan empat buah motor DC secara dua arah dengan arus untuk setiap *textH-bridge* ialah 0,6 A. Secara *default* tegangan kerja motor yang dapat dikendalikan ialah berkisar antara 4,5 Volt hingga 16 Volt. Selain motor DC, modul *driver* motor ini juga dapat digunakan untuk mengendalikan hingga dua buah *motor stepper* dan juga dua buah motor servo [3]. Gambar 2.16 menunjukan *Motor Driver Shield L293D*.



Sumber : [www.adafruit.com](http://www.adafruit.com)

Gambar 2.16 Motor Driver Shield L293D

### 2.11 Remote Control 2.4GHz

*Remote Control* merupakan bagian yang berinteraksi langsung dengan pengguna untuk memberikan sinyal perintah-perintah untuk menggerakkan robot dalam arah gerakan arah naik, turun, maju, mundur, kiri dan kanan. Frekuensi yang digunakan sebagai media transmisi adalah gelombang radio pada frekuensi 2,4

Ghz. Pada kondisi diluar ruangan penggunaan frekuensi jika dibandingkan dengan menggunakan sinyal infra merah, sinar matahari sering memberikan gangguan terhadap sinyal infra merah yang sangat mempengaruhi proses kendali, maka gelombang radio merupakan pilihan tepat. Selain itu penggunaan gelombang radio mempunyai keunggulan dimana data yang dipancarkan dapat dikirim pada jarak yang cukup jauh [6]. Gambar 2.17 menunjukkan *remote control* 2.4 GHz.



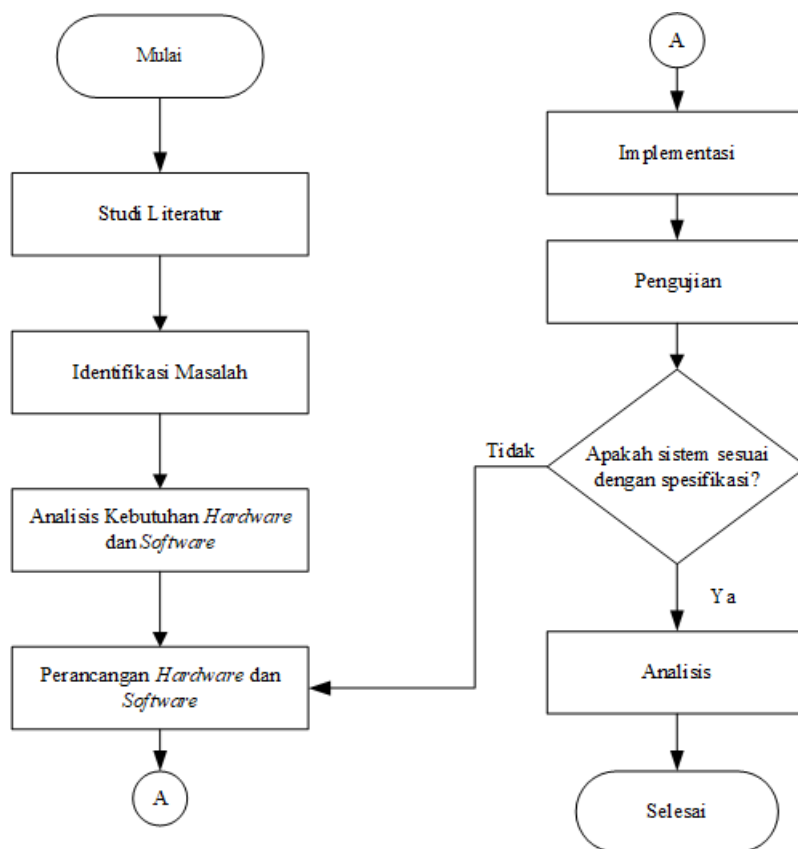
Sumber : [www.flysky-cn.com](http://www.flysky-cn.com)

Gambar 2.17 *Remote Control* 2.4 GHz

## BAB III METODE PENELITIAN

### 3.1 Metode Penelitian

Metode yang digunakan pada penelitian ini yaitu melakukan studi literatur dari berbagai penelitian yang sebelumnya telah dilakukan, kemudian tahap perumusan masalah serta menganalisis kebutuhan hingga perancangan alat untuk memperoleh hasil dan kesimpulan. Berikut tahapan proses pada penelitian ini dijabarkan pada Gambar 3.1.



Gambar 3.1 Blok Diagram Proses Penelitian

### 3.2 Studi Literatur

Studi literatur merupakan suatu proses pengumpulan data yang berkaitan dengan penelitian yang akan dilakukan. Dimana alat yang akan dibuat mengacu pada referensi yang sudah ada sebelumnya untuk dilakukan pengembangan lebih lanjut. Sumber data yang diambil berasal dari referensi jurnal ilmiah, mengkaji buku-buku yang berkaitan dengan sistem yang akan dirancang, dan juga dari berbagai sumber di internet.



### 3.3 Identifikasi Masalah

Identifikasi masalah merupakan tahapan yang berkaitan dengan permasalahan yang timbul serta mencari cara untuk menyelesaikan permasalahan yang timbul tersebut. Identifikasi masalah pada penelitian Tugas Akhir ini telah dijabarkan secara lengkap pada bagian rumusan masalah. Untuk hipotesis penelitian Tugas Akhir ini dapat dilihat pada bagian manfaat penelitian. Adapun dari hasil identifikasi permasalahan yang timbul yaitu merancang dan menganalisis Sistem Rancang Bangun *Automatic Navigation Robot* Menggunakan *Remote Control* dan Sensor Ultrasonik berbasis *Fuzzy Logic Control* agar bekerja sesuai yang ditentukan.

### 3.4 Analisis Kebutuhan

Penelitian yang akan dilakukan membutuhkan beberapa komponen dan perangkat yang menunjang penelitian. Komponen dan perangkat yang dibutuhkan untuk melakukan penelitian ini diantaranya sebagai berikut:

1. Perangkat Keras (*Hardware*)
  - (a) Arduino MEGA 2560
  - (b) Sensor Ultrasonik HC-SR04
  - (c) Motor DC
  - (d) *Motor Driver*
  - (e) *Remote Control* dengan frekuensi 2.4 GHz
  - (f) *Receiver* 2.4 GHz 3 kanal
  - (g) *Chasis Aluminium Alloy*
2. Perangkat Lunak (*Software*)
  - (a) IDE Arduino yang digunakan untuk membuat program pada *Software* Arduino Uno dan mengirimkan hasil kompilasi ke Arduino Uno.
  - (b) Aplikasi Matlab untuk membuat *membership function* dan *if then rule*.

### 3.5 Perancangan

Perancangan *Sketch Program* dilakukan di Arduino IDE untuk memprogram mikro kontroler sistem pada Rancang Bangun *Automatic Navigation Robot* Menggunakan *Remote Control* dan Sensor Ultrasonik berbasis *Fuzzy Logic Control*. Perancangan *membership function* dan *if then rule* menggunakan Aplikasi Matlab.

Pada penelitian ini, metode *Fuzzy Logic* diterapkan pada pergerakan motor setelah mendapatkan masukan dari sensor ultrasonik yang membaca jarak *obstacle*.

Perancangan hardware dilakukan berdasarkan prosedur untuk setiap instalasi blok perangkatnya. Pada perancangan hardware ini terdapat beberapa komponen yang akan digunakan yaitu satu buah sensor ultrasonik HC-SR04 dan *receiver* 2.4 GHz sebagai *input* yang terhubung dengan mikrokontroler Arduino Mega 2560. Kemudian terdapat satu buah *driver shield* L293D yang akan dihubungkan dengan dua buah motor DC sebagai *output*.

### **3.6 Implementasi Sistem**

Setelah semua proses dilakukan, hasil dari perencanaan dan analisis kebutuhan sudah sesuai akan dilakukan pembuatan alat berdasarkan rancangan pada perancangan sistem, perangkat keras juga perangkat lunak. Tahap ini dilakukan untuk mencapai hasil dari perancangan sebelumnya sehingga alat tersebut dapat dipergunakan.

### **3.7 Integrasi Sistem**

Kombinasi perangkat lunak dan perangkat keras adalah proses integrasi sistem. Proses ini dijalankan setelah sistem berfungsi normal, dan meminimalkan kerusakan dan ketidaksesuaian dalam proses desain. Proses ini memastikan bahwa desain perangkat lunak dan perangkat keras berfungsi dengan baik.

### **3.8 Pengujian Sistem**

Pada tahap ini dilakukan pengujian sistem *hardware* dan *software*. Pada pengambilan data ini dilakukan pengujian dari robot yang telah dirancang. Pengujian robot terdiri dari pengujian respon dua sensor robot dan *remote control* serta manuver pergerakan robot menghindari *obstacle* saat sebelum dan sesudah diimplementasikan metode *Fuzzy logic*.

### **3.9 Analisis Hasil**

Pada bagian analisis data dilakukan perbandingan dari hasil pengujian dengan data referensi. Analisis data meliputi bagian sensor sebagai pengindra dan perbandingan hasil saat diimplementasikan *Fuzzy logic control* dari *tools* yang mumpuni dengan implementasi manuver pada robot.

## BAB IV PERANCANGAN DAN IMPLEMENTASI

### 4.1 Analisis Kebutuhan

Sebelum melakukan perancangan dan implementasi maka dilakukan terlebih dahulu analisis kebutuhan untuk menunjang kelancaran dan keberhasilan penelitian ini. Analisis kebutuhan terdiri dari alat dan bahan yang terbagi menjadi dua yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*) seperti pada Tabel 4.1.

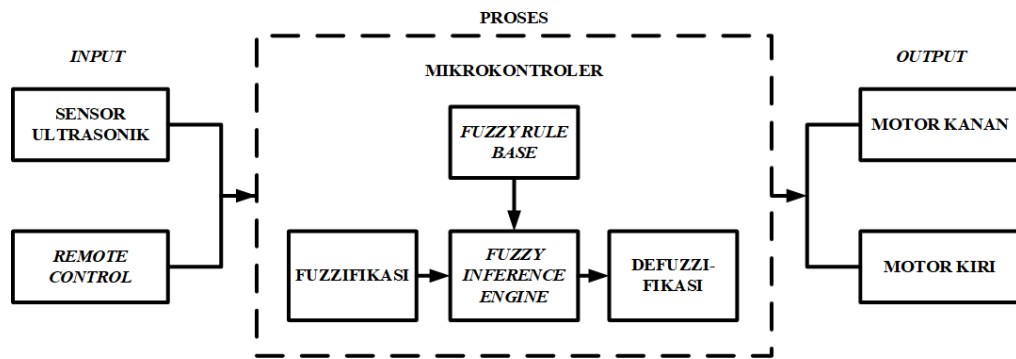
Tabel 4.1 Alat dan Bahan

No.	Alat	Bahan
1	Laptop	Arduino MEGA 2560
2	<i>Software</i> Matlab	<i>Driver Motor Shield</i> L293D
3	<i>Software</i> Arduino IDE	Sensor Ultrasonik HC-SR04
4	Obeng	Baterai Lipo 11V
5	Tang Potong	Motor DC
6	<i>Remote Control</i>	<i>Receiver</i> 2.4 GHz 3 kanal
7	Kabel USB to Arduino	<i>Chassis Aluminium Alloy</i>
8	Solder	Kabel <i>Jumper</i>

### 4.2 Perancangan

Perancangan dilakukan untuk menentukan proses pengimplementasian alat yang telah dirancang. Tahap perancangan sistem ini meliputi perancangan perangkat keras dan perangkat lunak, dimana bagian perancangan perangkat keras menjelaskan rancangan skema komponen elektronik robot dan konfigurasi perangkat Arduino MEGA 2560. Sedangkan pada perancangan perangkat lunak akan dibahas mengenai perancangan program pada Arduino IDE dan proses simulasi *Fuzzy Logic* pada aplikasi Matlab.

Pada perancangan robot ini terdapat 3 tahapan sistem yaitu *input*, proses dan *output*. Pada proses *input* digunakan sensor ultrasonik HC-SR04 yang dipasang pada bagian depan robot serta satu buah *receiver* 2.4 GHz yang mendapatkan masukan dari *remote control*. Hasil pembacaan dari remote control dan sensor ultrasonik akan diproses oleh Arduino MEGA 2560 yang telah berbasis sistem kendali *Fuzzy Logic* dan nantinya akan mempengaruhi hasil output pada dua buah motor DC yang dipasang pada robot. Pada Gambar 4.1 dapat dilihat blok diagram perancangan perangkat keras dari robot ini.



Gambar 4.1 Blok rancangan robot

Cara kerja robot ini meliputi beberapa bagian yaitu *input*, proses dan *output*. Secara keseluruhan setiap bagian ini saling berhubungan satu sama lain sehingga robot yang digunakan dapat tercipta.

### 1. Bagian *Input*

Bagian *input* pada robot ini adalah *remote control* dan sensor ultrasonik HC-SR04. *Remote control* berfungsi sebagai pemberi sinyal masukan kepada motor DC. *Remote control* juga mempengaruhi kecepatan motor DC karena semakin dalam *toggle*/tuas *remote control* ditekan maka semakin cepat perputaran motor DC yang dihasilkan. Sedangkan sensor ultrasonik berfungsi untuk pemberi informasi jarak antara robot dengan *obstacle* yang ada dihadapannya.

### 2. Bagian Proses

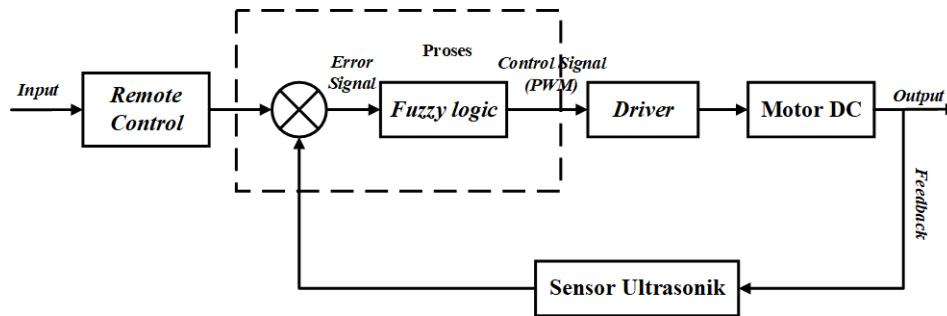
Bagian proses merupakan bagian utama pada robot ini, karena robot ini bekerja berdasarkan pengukuran jarak oleh sensor terhadap *obstacle*. Bagian ini menggunakan mikrokontroler Arduino MEGA 2560 untuk mengontrol logika *Fuzzy* untuk memproses hasil pengukuran jarak pada sensor. Data yang diterima akan diproses yang nantinya akan membuat keputusan tentang output sesuai dengan *input* program. Masukan dari remote control yang berupa sinyal analog juga diproses menggunakan logika *Fuzzy* untuk mengatur perputaran motor DC.

### 3. Bagian *Output*

Bagian ini merupakan hasil akhir dari penerimaan data masukan oleh remote control dan sensor. Pada output digunakan dua buah motor DC yang dibagi menjadi motor kanan dan motor kiri. Output yang dihasilkan adalah PWM atau perputaran masing-masing motor DC. Motor

DC digunakan untuk menghindari rintangan dengan cara bermanuver yang sesuai dengan hasil dari bagian proses.

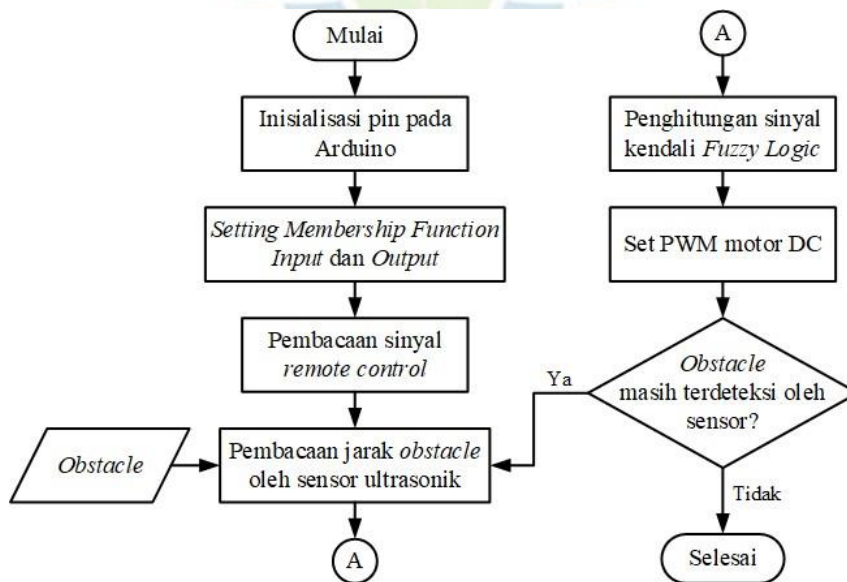
Pada Gambar 4.2 dilihat diagram sistem kendali pada robot ini.



Gambar 4.2 Blok diagram sistem kendali robot

### 4.3 Perancangan Sistem Kendali *Fuzzy Logic*

Pada penelitian ini *software* Arduino IDE digunakan untuk memrogram Arduino MEGA 2560 yang berbasis logika *Fuzzy*. Pada Gambar 4.3 dijelaskan alur dari program yang dirancang pada robot ini.

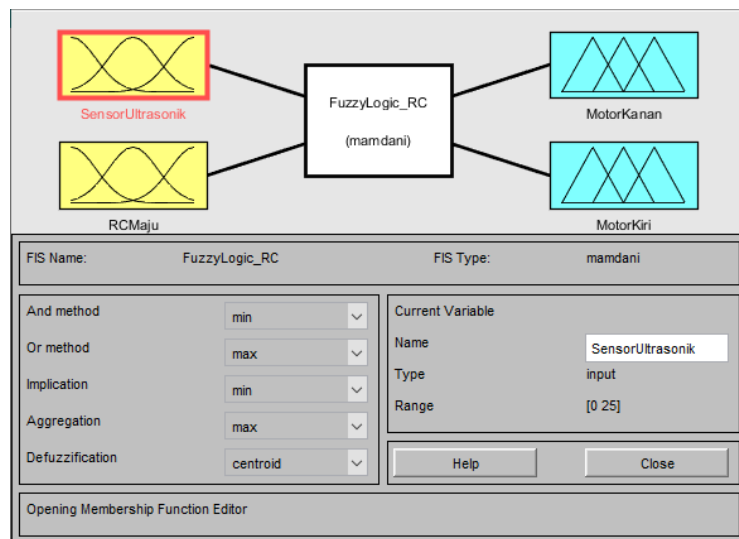


Gambar 4.3 Flowchart program robot

#### 4.3.1 Pemodelan *Fuzzy Logic*

Pemodelan sistem kendali *Fuzzy Logic* pada penelitian ini digunakan metode Mamdani yang dimana terbentuknya sebuah himpunan fungsi keanggotaan

(*membership function*) adalah dari tiap-tiap variabel yang digunakan, baik *input* maupun *output*. Pada bagian *input* digunakan satu buah sensor ultrasonik dan *remote control* yang memiliki fungsi keanggotaan masing-masing. Pada bagian output sendiri digunakan dua buah motor DC yang juga memiliki fungsi keanggotaan ditiap motornya. Pada Gambar 4.4 dapat dilihat hasil pemodelan *Fuzzy Mamdani*.



Gambar 4.4 Pemodelan *Fuzzy Logic*

### 4.3.2 *Membership Function Input*

Pada robot ini *input* yang digunakan ada dua buah, yaitu sensor ultrasonik HC-SR04 dan *remote control*. Kedua input tersebut memiliki variabel yang berbeda maka dari itu dibutuhkannya istilah linguistik untuk memudahkan perancangan sistem kendali pada penelitian ini. Istilah linguistik pada fungsi keanggotaan sensor ultrasonik dapat dilihat pada Tabel 4.2.

Tabel 4.2 Linguistik *Input* Sensor Ultrasonik

Jarak (cm)	Istilah Linguistik
[0 0 25 30]	Dekat (De)
[25 30 35 40]	Tengah (T)
[35 40 45 45]	Jauh (J)

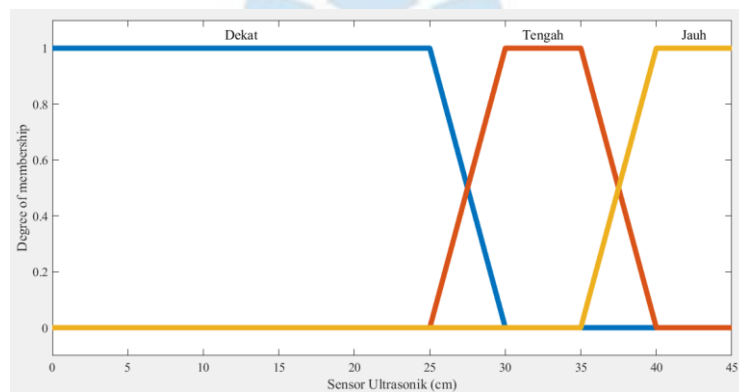
Pada Tabel 4.3 dijelaskan istilah linguistik untuk *membership function* dari *remote control*.

Tabel 4.3 Linguistik *Input Remote Control*

<i>Toggle Level</i>	<b>Istilah Linguistik</b>
[1500 1500 1650 1700]	Dangkal (Dkl)
[1650 1700 1850 1900]	Sedang (Sdg)
[1850 1900 2050 2050]	Dalam (Dlm)

### 1. *Membership Function Sensor Ultrasonik*

Pada proses *Fuzzy logic* ditentukan sebuah himpunan *Fuzzy* yang diatur pada aplikasi simulasi. Gambar 4.5 menunjukkan grafik *membership function* dari sensor ultrasonik.



Gambar 4.5 Grafik *Membership Function* Sensor Ultrasonik

Berdasarkan pada Gambar 4.5 didapat persamaan sebagai berikut :

$$\mu_{Dekat} = \begin{cases} 1 & ; x \leq 10 \\ \frac{15-x}{5} & ; 10 \leq x \leq 15 \\ 0 & ; x \geq 15 \end{cases} \quad (4.1)$$

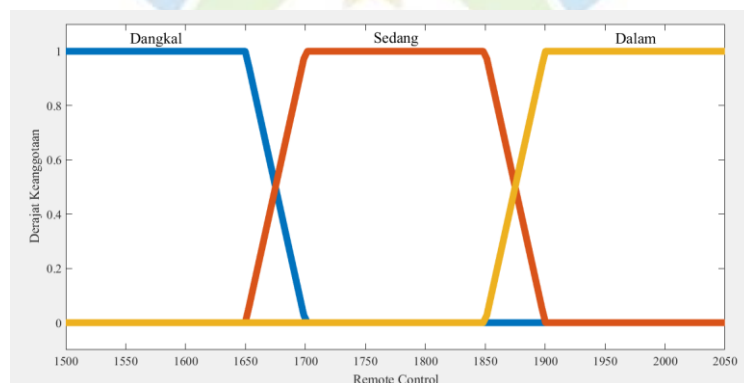
$$\mu_{Tengah} = \begin{cases} \frac{x-10}{5} & ; 10 \leq x \leq 15 \\ 1 & ; 15 \leq x \leq 20 \\ \frac{25-x}{5} & ; 20 \leq x \leq 25 \\ 0 & ; x \geq 25 \end{cases} \quad (4.2)$$

$$\mu_{Jauh} = \begin{cases} \frac{x-20}{5} & ; 20 \leq x \leq 25 \\ 1 & ; x \geq 30 \end{cases} \quad (4.3)$$

Fungsi keanggotaan untuk variabel sensor ultrasonik diatur dengan *range* [0 30]. Fungsi keanggotaan diubah namanya menjadi Dekat (De), menggunakan tipe trapmf (fungsi keanggotaan trapesium), dan kemudian memasukkan nilai [0 0 10 15] untuk nilai parameter, himpunan *Fuzzy* Tengah (T) memiliki nilai parameter [10 15 20 25] dan untuk himpunan *Fuzzy* Jauh (J) memiliki nilai parameter [20 25 30 30]. Persamaan fungsi keanggotaan sensor ultrasonik dapat dilihat pada Persamaan (4.1), Persamaan (4.2) dan Persamaan (4.3).

## 2. *Membership Function Remote Control*

Variabel pada *remote control* meliputi tiga himpunan yaitu Dangkal, Sedang dan Dalam. Pengambilan nama tersebut berdasarkan aktivitas penekanan *toggle* pada *remote control* seperti halnya yang dilakukan terhadap pedal gas kendaraan bermotor. Pada Gambar 4.6 diperlihatkan grafik fungsi keanggotaan *input* dari *remote control*.



Gambar 4.6 Grafik *Membership Function Remote Control*

Berdasarkan pada Gambar 4.6 didapat persamaan sebagai berikut :

$$\mu_{Dangkal} = \begin{cases} 1 & ; x \leq 1650 \\ \frac{1700-x}{50} & ; 1650 \leq x \leq 1700 \\ 0 & ; x \geq 1700 \end{cases} \quad (4.4)$$



$$\mu_{Sedang} = \begin{cases} \frac{x-1650}{50} & ; 1650 \leq x \leq 1700 \\ 1 & ; 1700 \leq x \leq 1850 \\ \frac{1900-x}{50} & ; 1850 \leq x \leq 1900 \end{cases} \quad (4.5)$$

$$\mu_{Dalam} = \begin{cases} \frac{x-1850}{50} & ; 1850 \leq x \leq 1900 \\ 1 & ; x \geq 2050 \end{cases} \quad (4.6)$$

Fungsi keanggotaan untuk variabel *remote control* diatur dengan range [1500 2050]. Hal ini dikarenakan nilai *toggle level remote control* untuk perintah maju berada pada rentang nilai 1500 sampai 2050. Fungsi keanggotaan diubah namanya menjadi Dangkal (Dkl), menggunakan tipe trapmf (fungsi keanggotaan trapesium), dan kemudian dimasukkan nilai [1500 1500 1650 1700] sebagai nilai parameter, himpunan *Fuzzy Sedang* (Sdg) memiliki nilai parameter [1650 1700 1850 1900] dan untuk himpunan *Fuzzy Dalam* (Dlm) memiliki nilai parameter [1850 1900 2050 2050]. Persamaan fungsi keanggotaan sensor ultrasonik dapat dilihat pada Persamaan (4.4), Persamaan (4.5) dan Persamaan (4.6).

### 4.3.3 Membership Function Output

Pada robot ini dibuat membership function pada dua buah motor DC sebagai penggerak robot. Karena banyak variabel yang digunakan, istilah linguistik harus digunakan untuk memudahkan sistem kendali yang dirancang dalam penelitian ini. Istilah linguistik pada fungsi keanggotaan motor DC dapat dilihat pada Tabel 4.4.

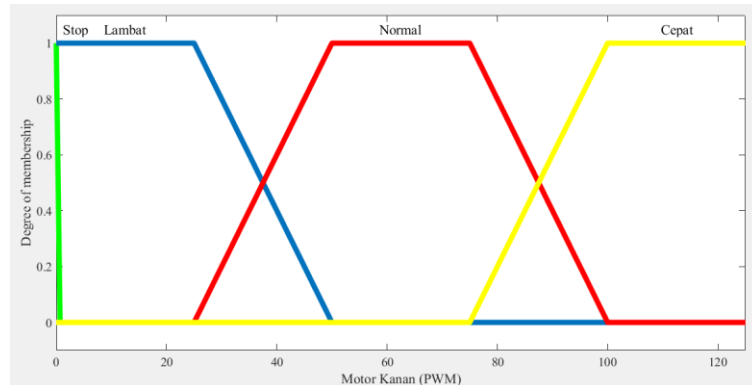
Tabel 4.4 Linguistik *Output* Motor DC

Kecepatan Motor (PWM)	Istilah Linguistik	
	Motor Kiri	Motor kanan
[0 0 25 50]	Lambat (Lki)	Lambat (Lka)
[25 50 75 100]	Normal (Nki)	Normal (Nka)
[75 100 125 125]	Cepat (Cki)	Cepat (Cka)

#### 1. Membership Funtion Fuzzy Motor Kanan

Pada proses *Fuzzy logic* ditentukan sebuah himpunan *Fuzzy* yang diatur

pada aplikasi simulasi. Gambar 4.7 menunjukkan grafik *membership function* dari motor DC kanan.



Gambar 4.7 Grafik *Membership Function* Motor DC Kanan

Berdasarkan pada Gambar 4.7 didapat persamaan sebagai berikut :

$$\mu_{Stop} = \begin{cases} 1 & ; x = 0 \\ 0 & ; x > 25 \end{cases} \quad (4.7)$$

$$\mu_{Lambat} = \begin{cases} 1 & ; x \leq 25 \\ \frac{50-x}{25} & ; 25 \leq x \leq 50 \\ 0 & ; x \geq 50 \end{cases} \quad (4.8)$$

$$\mu_{Normal} = \begin{cases} 0 & ; x < 25 \\ \frac{x-25}{25} & ; 25 \leq x \leq 50 \\ 1 & ; 50 \leq x \leq 75 \\ \frac{75-x}{25} & ; 75 \leq x \leq 100 \\ 0 & ; x \geq 100 \end{cases} \quad (4.9)$$

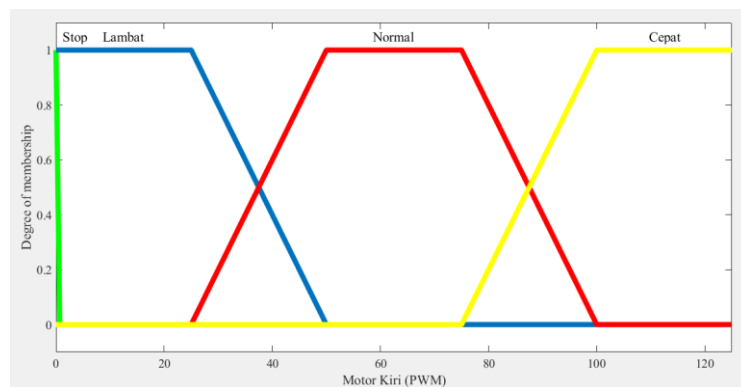
$$\mu_{Cepat} = \begin{cases} 0 & ; x < 75 \\ \frac{x-75}{25} & ; 75 \leq x \leq 100 \\ 1 & ; x \geq 100 \end{cases} \quad (4.10)$$

Fungsi keanggotaan untuk variabel motor DC kanan diatur dengan *range* [0 125]. Fungsi keanggotaan diubah namanya menjadi Lambat (Lka), menggunakan tipe trapmf (fungsi keanggotaan trapesium), dan kemudian memasukkan nilai [0 0 25 50] untuk nilai parameter, himpunan *Fuzzy* Normal (Nka) memiliki nilai parameter [25 50 75 100] dan untuk

himpunan *Fuzzy* Cepat (Cka) memiliki nilai parameter [75 100 125 125]. Persamaan fungsi keanggotaan motor DC kanan dapat dilihat pada Persamaan (4.8), Persamaan (4.9) dan Persamaan (4.10).

## 2. Membership Funtion Fuzzy Motor Kiri

Pada proses *Fuzzy logic* ditentukan sebuah himpunan *Fuzzy* yang diatur pada aplikasi simulasi. Gambar 4.8 menunjukkan grafik *membership function* dari motor DC kiri.



Gambar 4.8 Grafik *Membership Function* Motor DC Kiri

Berdasarkan pada Gambar 4.8 didapat persamaan sebagai berikut :

$$\mu_{Stop} = \begin{cases} 1 & ; x = 0 \\ 0 & ; x > 0 \end{cases} \quad (4.11)$$

$$\mu_{Lambat} = \begin{cases} 1 & ; x \leq 25 \\ \frac{50-x}{25} & ; 25 \leq x \leq 50 \\ 0 & ; x \geq 50 \end{cases} \quad (4.12)$$

$$\mu_{Normal} = \begin{cases} \frac{x-25}{25} & ; 25 \leq x \leq 50 \\ 1 & ; 50 \leq x \leq 75 \\ \frac{75-x}{150-x} & ; 75 \leq x \leq 100 \\ 0 & ; x \geq 100 \end{cases} \quad (4.13)$$

$$\mu_{Cepat} = \begin{cases} \frac{x-75}{25} & ; 75 \leq x \leq 100 \\ 1 & ; x \geq 125 \end{cases} \quad (4.14)$$

Fungsi keanggotaan untuk variabel motor DC kanan diatur dengan *range*

[0 125]. Fungsi keanggotaan diubah namanya menjadi Lambat (Lki), menggunakan tipe trapmf (fungsi keanggotaan trapesium), dan kemudian memasukkan nilai [0 0 25 50] untuk nilai parameter, himpunan *Fuzzy* Normal (Nki) memiliki nilai parameter [25 50 75 100] dan untuk himpunan *Fuzzy* Cepat (Cki) memiliki nilai parameter [75 100 125 125]. Persamaan fungsi keanggotaan motor DC kanan dapat dilihat pada Persamaan (4.12), Persamaan (4.13) dan Persamaan (4.14).

#### 4.3.4 *Fuzzy Rule Base*

Setelah pendefinisian membership function pada sensor ultrasonik serta remote control sebagai input dan output dari kedua motor DC tersebut, maka aturan *Fuzzy* didefinisikan sebagai probabilitas sensor dan motor yang menjadi parameter untuk proses penentuan output. Pada Tabel 4.5 terdapat sembilan kemungkinan yang digunakan pada penelitian ini.

Tabel 4.5 *Fuzzy Rule Base*

<i>Rule</i>	<i>Input</i>		<i>Output</i>	
	Sensor	Remote	Motor Kanan	Motor Kiri
R1	Dekat	Dangkal	Normal	Cepat
R2	Dekat	Sedang	Normal	Cepat
R3	Dekat	Dalam	Normal	Cepat
R4	Tengah	Dangkal	Lambat	Lambat
R5	Tengah	Sedang	Normal	Normal
R6	Tengah	Dalam	Normal	Normal
R7	Jauh	Dangkal	Normal	Normal
R8	Jauh	Sedang	Normal	Normal
R9	Jauh	Dalam	Cepat	Cepat

Pada Tabel 4.5 dapat dilihat ketika motor kiri lebih cepat dari motor kanan maka menandakan bahwa kondisi robot dalam keadaan sedang berbelok ke kiri dan ketika kecepatan motor sama menandakan bahwa robot berjalan dalam keadaan lurus.

#### 4.4 Perancangan *Hardware*

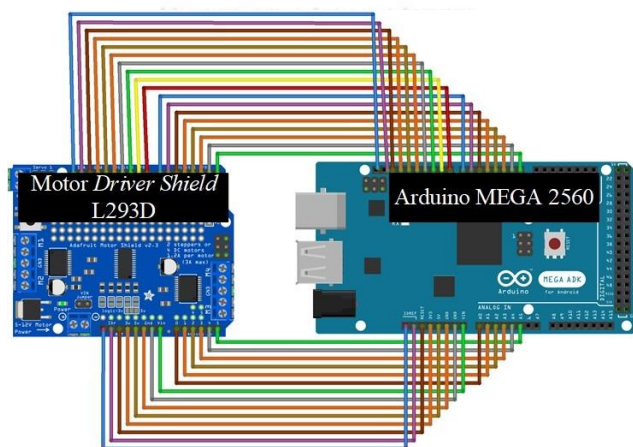
Pada perancangan robot ini digunakan beberapa komponen elektronik maupun non-elektronik pendukung. Dapat dilihat pada Tabel 4.6 nama dan jumlah komponen dan perangkat yang digunakan pada penelitian ini.

Tabel 4.6 Komponen Robot

No.	Nama	Jumlah
1	<i>Chassis aluminium alloy</i>	1 buah
2	Arduino MEGA 2560	1 buah
3	<i>Driver Motor Shield L293D</i>	1 buah
4	Motor DC	2 buah
5	Batere Lipo 1 Volt	1 buah
6	<i>Receiver 2.4 GHz 3 kanal</i>	1 buah
7	Sensor Ultrasonik HC-SR04	1 buah
8	<i>Remote Control 2.4 GHz</i>	1 buah
9	Kabel <i>Jumper</i>	20 buah

##### 4.4.1 Konfigurasi *Motor Driver Shield L293D* dengan Arduino MEGA 2560

Pada penelitian ini, motor driver shield L293D digunakan sebagai pengendali motor DC. Motor driver ini dikonfigurasi dengan Arduino MEGA 2560 seperti yang dapat dilihat pada Gambar 4.9.



Gambar 4.9 Konfigurasi *Motor Driver Shield L293D* Dengan Arduino MEGA

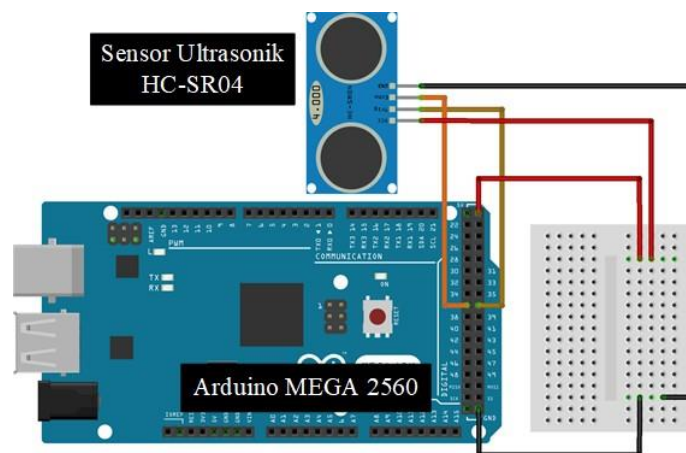
Instalasi pemasangan motor driver shield L293D pada Arduino MEGA 2560 dapat dilihat pada Tabel 4.7.

Tabel 4.7 Instalasi Pin *Motor Driver Shield L293D*

No.	Pin Arduino MEGA 2560	Motor Driver Shield L293D	No.	Pin Arduino MEGA 2560	Motor Driver Shield L293D
1	Analog In A5	Analog 5	14	1	1
2	Analog In A4	Analog 4	15	2	2
3	Analog In A3	Analog 3	16	3	3
4	Analog In A2	Analog 2	17	4	4
5	Analog In A1	Analog 1	18	5	5
6	Analog In A0	Analog 0	19	6	6
7	VIN	Vin	20	7	7
8	GND	Gnd	21	8	8
9	5 V	5v	22	9	9
10	3 V 3	3v	23	10	10
11	RESET	RST	24	11	11
12	IOREF	Ior	25	12	12
13	0	0	26	13	13

#### 4.4.2 Instalasi Sensor Ultrasonik HC-SR04

Pada penelitian ini digunakan satu buah sensor ultrasonik HC-SR04 sebagai pendeteksi jarak robot ke *obstacle*. Sensor tersebut dipasang pada bagian depan robot. Pada Gambar 4.10 dapat dilihat konfigurasi sensor ultrasonik HC-SR04.



Gambar 4.10 Instalasi Sensor Ultrasonik HC-SR04

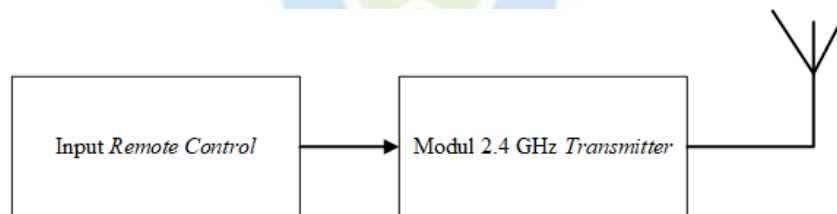
Pemasangan sensor ultrasonik HC-SR04 dibagian depan robot yang berposisi ditengah. Pada Tabel 4.8 dapat dilihat instalasi sensor ultrasonik HC-SR04 dengan Arduino MEGA 2560.

Tabel 4.8 Instalasi Sensor Ultrasonik HC-SR04

No.	Pin Arduino Mega 2560	Sensor Ultrasonik HC-SR04
1	5V	VCC
2	37	Trigger
3	36	Echo
4	GND	GND

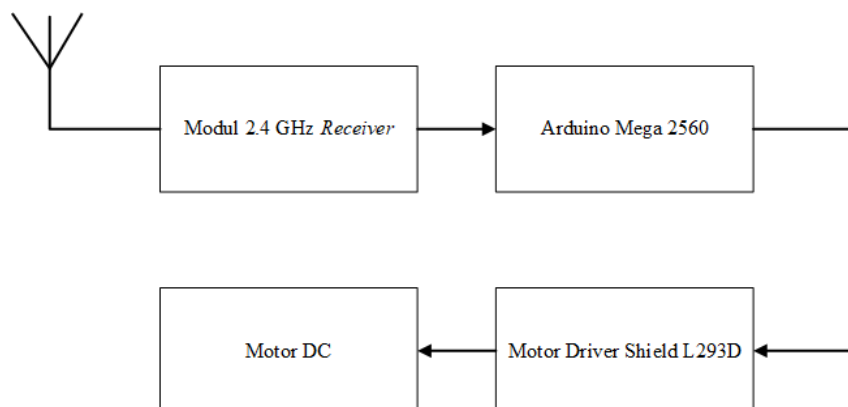
#### 4.4.3 Instalasi Receiver 2.4GHz 3 Kanal

Pada penelitian ini digunakan receiver 2.4GHz 3 kanal sebagai penerima sinyal masukan dari remote control untuk memberikan perintah kepada robot. Receiver tersebut dipasang pada bagian tengah robot. Berikut Gambar 4.11 merupakan blok diagram transmitter remote control pada sistem robot.



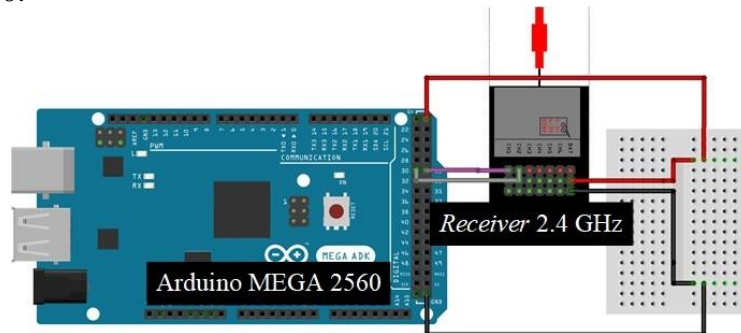
Gambar 4.11 Blok Diagram Transmitter Remote Control

Gambar 4.12 dibawah ini adalah blok diagram receiver pada sistem robot.



Gambar 4.12 Blok Diagram Receiver

Pada Gambar 4.13 ditunjukkan penginstalasian *receiver* dengan Arduino MEGA 2560.



Gambar 4.13 Instalasi *Receiver* 2.4GHz 3 Kanal

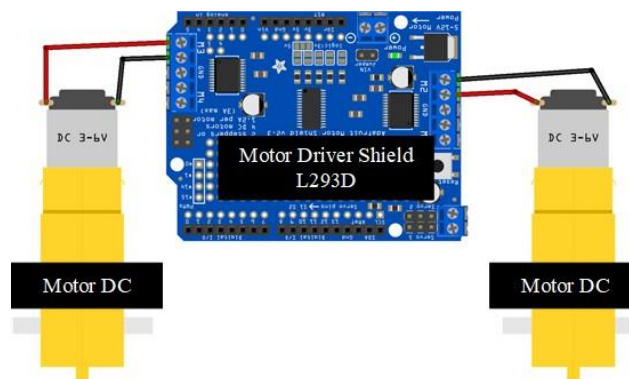
Pemasangan *receiver* ini diletakkan dibagian tengah robot atau lebih tepatnya dibelakang sensor ultrasonik. Pada Tabel 4.9 dapat dilihat instalasi *receiver* 2,4GHz 3 kanal dengan Arduino MEGA 2560.

Tabel 4.9 Instalasi *Receiver* 2.4GHz 3 Kanal

No.	Pin Arduino Mega 2560	<i>Receiver</i> 2.4 GHz
1	5V	VCC
2	30	<i>Channel</i> 1
3	31	<i>Channel</i> 2
4	GND	GND

#### 4.4.4 Instalasi Motor DC

Pada penelitian ini digunakan Motor DC sebagai penggerak robot yang dapat ditunjukkan pada Gambar 4.14.



Gambar 4.14 Instalasi Motor DC



Pemasangan Motor DC ke Pin motor driver shield dapat dilihat pada Tabel 4.10 untuk motor kiri dan Tabel 4.11 untuk motor kanan.

Tabel 4.10 Instalasi Motor Kiri

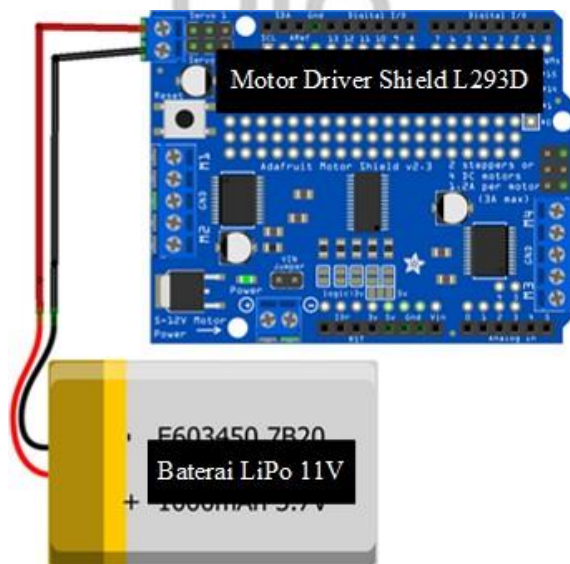
No.	Pin Motor Driver Shield L293D	Pin Motor DC
1	M2+	+
2	M2-	-

Tabel 4.11 Instalasi Motor Kanan

No.	Pin Motor Driver Shield L293D	Pin Motor DC
1	M3+	+
2	M3-	-

#### 4.4.5 Instalasi Baterai Lipo 11V

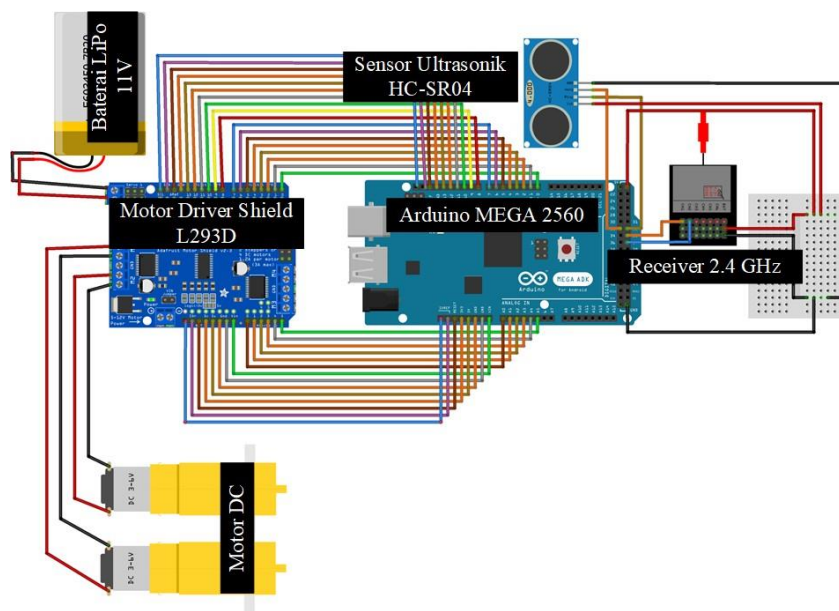
Pada penelitian ini digunakan Baterai Lipo 11V sebagai sumber tegangan robot untuk menggerakkan motor DC. Pada Gambar 4.15 ditunjukkan skema rangkaian instalasi baterai.



Gambar 4.15 Instalasi Baterai Lipo 11V

#### 4.4.6 Skema Rangkaian Robot

Pada penelitian ini digunakan komponen yang terdiri dari mikrokontroler Arduino MEGA 2560, *motor driver shield* L293D, sensor ultrasonik, motor DC, baterai Lipo 11V serta receiver 2.4GHz sebagai penerima sinyal masukan dari remote control. Skematik rangkaian robot untuk penelitian ini dapat dilihat pada Gambar 4.16.



Gambar 4.16 Skematik Rangkaian Robot

#### 4.5 Implementasi

Implementasi merupakan penerapan yang mengarah pada tindakan, aktivitas atau mekanisme yang dirancang secara terperinci dan telah disusun secara matang. Implementasi pada penelitian ini merupakan realisasi dari hasil yang telah dibahas sebelumnya. Hasil implementasi perangkat lunak tersebut kemudian diterapkan pada perangkat keras yang saling terhubung menjadi sebuah sistem.

#### 4.6 Implementasi Rancangan Program

Implementasi perangkat lunak pada penelitian ini berupa program mikrokontroler dengan aplikasi Arduino IDE. Implementasi perangkat lunak pada penelitian ini merupakan implementasi model *Fuzzy logic* yang telah dirancang sebelumnya pada bab perancangan. Pada aplikasi ini dirancang sebuah program untuk mengatur proses pada input sehingga menghasilkan output yang diinginkan oleh pengguna.

Pengimplementasian perangkat lunak pada sistem yang dirancang ini terdiri dari pembuatan *source code* pada Arduino IDE dan pemodelan *Fuzzy*.

Dibutuhkan *library* untuk memudahkan pemrograman yang akan dijalankan oleh pengguna. *Library* secara istilah merupakan sekumpulan kode yang mempunyai fungsi tertentu, sehingga fungsi yang kompleks terasa lebih mudah dengan bantuan fungsi *library* ini. Pada penelitian digunakan dua *library* pada pemrograman robon ini. *Library* yang digunakan pada penelitian ini ditunjukkan Tabel 4.12

Tabel 4.12 *Library* program pada Arduino IDE

No.	<i>Library</i>	Keterangan
1	<AFMotor.h>	<i>Library</i> ini memungkinkan untuk mengendalikan motor DC dengan menggunakan <i>Motor Driver Shield</i>
2	<Fuzzy.h>	<i>Library</i> ini memungkinkan untuk menyatukan <i>input</i> , <i>ouput</i> , dan <i>rule Fuzzy logic</i> .

#### 4.6.1 *Source Code Variabel Input dan Output*

Pada bagian *input* digunakan satu buah sensor ultrasonik yang berfungsi sebagai penentu jarak antara robot dengan *obstacle* dan *remote control* sebagai pemberi perintah kepada robot untuk bergerak maju. Penggunaan *library* AFMotor.h dalam penelitian ini sebagai penguasai *motor driver shield* L293D. Kemudian variable motor 2 dan motor 3 diinisialisasi sebagai variabel motor yang akan digunakan. Berikut pengimplementasian *source code* yang digunakan untuk mengatur *input* dan *output* pada penelitian ini.

---

#### **Algorithm 1** Variabel Linguistik

---

1: **procedure**

2: Fuzzy \*fuzzy = new Fuzzy();

3:

4: // Fuzzy Input Sensor Ultrasonik

5: FuzzySet \*Dekat = new FuzzySet(0, 0, 5, 10);

6: FuzzySet \*Tengah = new FuzzySet(5, 10, 15, 20);

7: FuzzySet \*Jauh = new FuzzySet(15, 20, 25, 25);

---

---

**Algorithm 2** Variabel Linguistik

---

```
1: procedure
2:   // Fuzzy Input Remote Control
3:   FuzzySet *Dangkal = new FuzzySet(1500, 1500, 1650, 1700);
4:   FuzzySet *Sedang = new FuzzySet(1650, 1700, 1850, 1900);
5:
6:   FuzzySet *Dalam = new FuzzySet(1850, 1900, 2050, 2050);
7:   // Fuzzy Output Motor Kanan
8:   FuzzySet *Lambat = new FuzzySet(0, 0, 25, 50);
9:   FuzzySet *Normal = new FuzzySet(25, 50, 75, 100);
10:  FuzzySet *Cepat = new FuzzySet(75, 100, 125, 125);
11:
12:  // Fuzzy Output Motor Kiri
13:  FuzzySet *Lambat = new FuzzySet(0, 0, 25, 50);
14:  FuzzySet *Normal = new FuzzySet(25, 50, 75, 100);
15:  FuzzySet *Cepat = new FuzzySet(75, 100, 125, 125);
16:  void setup()
```

---

#### 4.6.2 Pemrograman Model *Fuzzy* Pada Arduino IDE

Pada penelitian ini digunakan sembilan *rule base* sesuai himpunan *Fuzzy* yang telah dirancang. Penerapannya pada aplikasi arduino IDE dapat dilihat pada Algorithm 3.

---

**Algorithm 3** Implementasi *Fuzzy Rules*

---

```
1: procedure
2:   // Building fuzzy rule
3:   FuzzyRuleAntecedent *pelanAnddekat = new FuzzyRuleAntecedent();
4:   pelanAnddekat->joinWithAND(ma_pelan, de_dekat);
5:
6:   FuzzyRuleConsequent *O1_kanan = new FuzzyRuleConsequent();
7:   O1_kanan->addOutput(vr_lambat);
8:   O1_kanan->addOutput(vl_cepat);
```

---

---

**Algorithm 4** Implementasi *Fuzzy Rules*

---

1: **procedure**

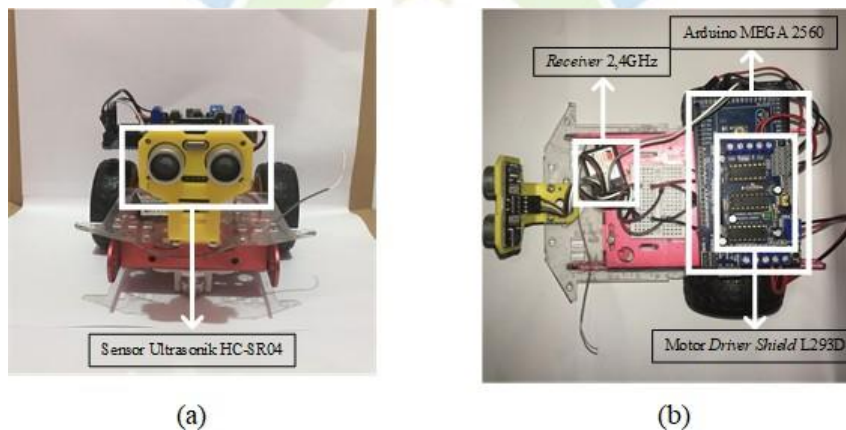
2: FuzzyRule \*fuzzyRule1 = new FuzzyRule(1, pelanAnddekat, O1\_kanan);

3: fuzzy->addFuzzyRule(fuzzyRule1);

---

#### 4.7 Implementasi Rancangan *Hardware*

Pada implementasi *hardware* ini dilakukan pemasangan perangkat yang sebelumnya telah disebutkan pada Tabel 4.6 sesuai dengan skema yang telah digambarkan pada Gambar 4.16. Setelah pemasangan seluruh komponen robot, dapat dilihat pada Gambar 4.17 (a) tampak depan dari robot dengan satu buah sensor ultrasonik yang terpasang pada bagian depan robot. Untuk tampak atas robot dapat dilihat pada Gambar 4.17 (b). Pada gambar tersebut terlihat *receiver* 2.4 GHz dan motor *driver shield* L293D yang terpasang pada Arduino MEGA 2560 serta dihubungkan dengan semua komponen yang terpasang pada robot.



Gambar 4.17 (a) Robot tampak depan, (b) Robot tampak atas

## BAB V PENGUJIAN DAN ANALISIS

### 5.1 Pengujian

Proses perancangan dan implementasi perangkat lunak dan perangkat keras pada sistem robot telah selesai dilakukan. Setelah semua bagian telah dirancang dan dirakit, langkah selanjutnya adalah tahap pengujian. Secara umum tujuan dari pengujian ini adalah untuk mengetahui apakah alat yang dibuat dapat bekerja dengan baik. Ada beberapa tahap pengujian pada alat sesuai fungsi yang telah ditentukan. Berikut hal yang akan diuji pada penelitian ini :

1. Pengujian *Hardware*

Dalam pengujian *hardware*, beberapa pengujian akan dilakukan pada beberapa komponen robot seperti sensor ultrasonik, *remote control* dan motor DC.

2. Pengujian Sistem Kendali *Fuzzy Logic*

Pada pengujian perangkat lunak akan dilakukan tiga pengujian berdasarkan hasil penerapan metode Fuzzy dalam *software* simulasi, perhitungan secara manual dan pengujian sistem yang dibangun.

3. Pengujian Respon Sistem

Pada pengujian respon sistem akan dilakukan sebuah skenario pengujian untuk mengetahui nilai *overshoot*, *settling time*, dan *error steady state*.

4. Pengujian Manuver Robot

Pada pengujian manuver robot akan dilakukan pengujian perbandingan PWM hasil pergerakan robot ketika menggunakan metode *Fuzzy logic control* dan robot tanpa metode *Fuzzy logic control*.

### 5.2 Pengujian *Hardware*

Pengujian *hardware* dilakukan pada komponen-komponen yang akan diterapkan pada robot ini. Pada bagian ini komponen yang akan diuji adalah sensor ultrasonik, *remote control* dan pengujian motor.

#### 5.2.1 Pengujian Sensor Ultrasonik

Pada penelitian ini dilakukan pengujian sensor ultrasonik HC-SR04 yang bertujuan untuk mengetahui apakah sensor bekerja dengan baik atau tidak. Pengujian integrasi sensor ultrasonik ini meliputi :

### 1. Sensitivitas Jarak Pantul

Pengujian ini dilakukan dengan menempatkan sebuah rintangan di depan sensor robot dengan jarak variabel dari 0 sampai 30 cm. Pada Tabel 5.1 menunjukkan pengujian sensitivitas jarak pantul dari sensor.

Tabel 5.1 Pengujian sensitivitas jarak pantul sensor ultrasonik

Jarak Sensor ke Rintangan (cm)		Selisih Jarak (cm)
Meteran	Ultrasonik	
5	5.00	0.00
10	10.00	0.00
15	14.95	0.05
20	20.00	0.00
25	25.00	0.00
30	30.08	0.08
<b>Rata-rata selisih jarak</b>		<b>0.02</b>

### 2. Sensitivitas Material Pantul

Setelah dilakukan pengujian sensitivitas jarak pantul, maka akan dilakukan pengujian sensitivitas bahan pantul. Tes ini melibatkan beberapa jenis bahan/material. Tujuan dari pengujian ini adalah untuk mengidentifikasi objek yang dapat dideteksi oleh sensor tergantung dari jenis materialnya. Tabel 5.2 menampilkan beberapa bahan dan hasil dari pengujian material pantul ini.

Tabel 5.2 Sensitivitas material pantul

No.	Jenis Material	Status
1	Kardus	Terdeteksi
2	Plastik	Terdeteksi
3	Kaca	Terdeteksi
4	Tembok	Terdeteksi
5	Kertas	Terdeteksi
6	Kayu	Terdeteksi

#### 5.2.2 Pengujian Remote Control

Setelah dilakukan pengujian terhadap sensor, tahap selanjutnya adalah pengujian *remote control*. Pengujian ini dilakukan dengan mengukur jarak

komunikasi antara robot dengan *remote control*. Pengujian jarak ini dilakukan dengan menambah jarak 10 meter hingga robot tidak lagi terhubung/dikendalikan oleh *remote control*. Berikut Tabel 5.3 merupakan hasil dari pengujian konektivitas robot.

Tabel 5.3 Pengujian konektivitas robot dengan *remote control*

Percobaan	Jarak (M)	Konektivitas	Keterangan
1	10	Terhubung	Robot dapat dikendalikan
2	20	Terhubung	Robot dapat dikendalikan
3	30	Terhubung	Robot dapat dikendalikan
4	40	Terhubung	Robot dapat dikendalikan
5	50	Terhubung	Robot dapat dikendalikan
6	60	Terhubung	Robot dapat dikendalikan
7	70	Terhubung	Robot dapat dikendalikan
8	80	Terhubung	Robot dapat dikendalikan
9	90	Terhubung	Robot dapat dikendalikan
10	100	Terhubung	Robot dapat dikendalikan
11	110	Terhubung	Robot dapat dikendalikan
12	120	Terhubung	Robot dapat dikendalikan
13	130	Terhubung namun kurang merespon	Robot sedikit dapat dikendalikan dengan adanya hilang kendali
14	140	Terhubung namun kurang merespon	Robot sedikit dapat dikendalikan dengan adanya hilang kendali
15	≥ 150	Terputus	Robot tidak dapat dikendalikan

Dari hasil pengujian pada Tabel 5.3 dapat diketahui bahwa jangkauan maksimal komunikasi antara robot dan *remote control* kurang dari 150 meter. Pada saat jarak antara robot dengan *remote control* mulai memasuki 130 meter, robot mulai mengalami kurang responsifnya terhadap perintah yang diberikan oleh *remote control*. Memasuki jarak 150 meter robot mengalami putusnya koneksi dengan *remote control*.

### 5.2.3 Pengujian Motor Driver Shield L293D

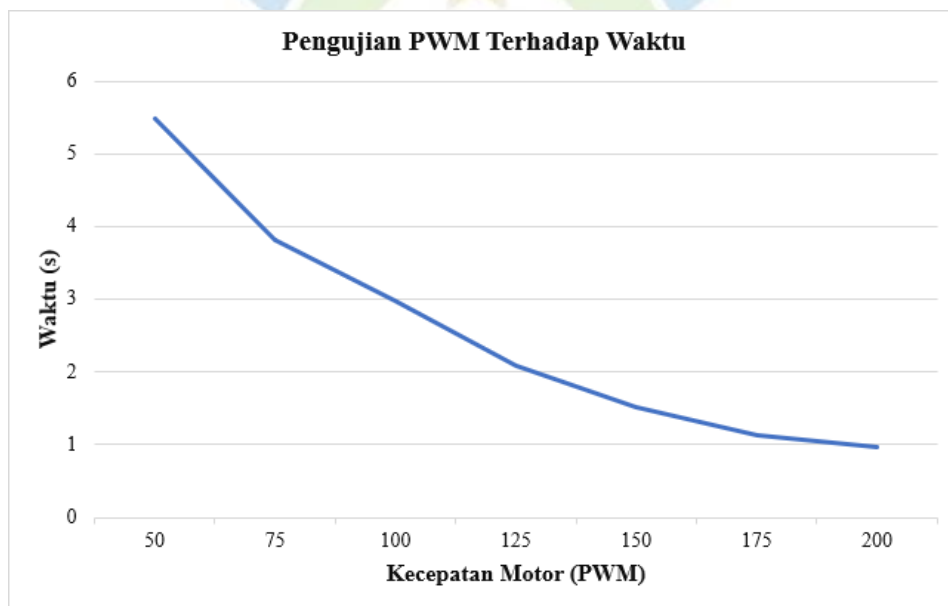
Pengujian motor *driver shield* L293D bertujuan untuk mengetahui hasil kecepatan PWM (*Pulse Width Modulation*) yang dihasilkan oleh motor DC 6-



12 volt. Pengujian dilakukan dengan memvariasikan kecepatan PWM motor dan mengukur waktu tempuh robot dari jarak 100cm. Tabel 5.4 menunjukkan hasil pengujian waktu tempuh motor driver shield L293D untuk jarak 100cm.

Tabel 5.4 Pengujian Kecepatan PWM Motor

No.	Kecepatan (PWM)	Waktu (s)
1	50	5,49
2	75	3,81
3	100	2,98
4	125	2,09
5	150	1,51
6	175	1,13
7	200	0,97



Gambar 5.1 Grafik kecepatan PWM motor terhadap Waktu

Berdasarkan pada Gambar 5.1 didapatkan hasil pengujian PWM motor bahwa semakin tinggi nilai PWM, semakin cepat robot dapat mencapai jarak 100cm.

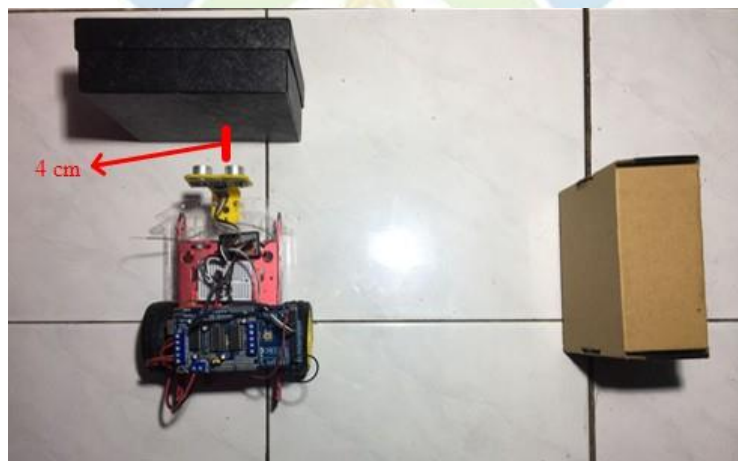
### 5.3 Pengujian Model *Fuzzy Logic Control*

Pada pengujian model *Fuzzy logic* ini bertujuan untuk memverifikasi kesesuaian sistem kendali yang telah dirancang pada robot ini dan dibandingkan

dengan hasil perhitungan manual dan hasil perhitungan menggunakan aplikasi simulasi. Pada pengujian ini diambil sebuah kasus dengan kondisi sensor ultrasonik mendapatkan *input* jarak sebesar 4cm dari *obstacle* dan *remote control* memberikan input sebesar 1870. Dengan menggunakan sistem kendali *Fuzzy*, akan dicari nilai keluaran motor kiri (PWM) dan motor kanan (PWM) untuk robot. Pengujian model *Fuzzy* yang dibangun dilakukan dengan melakukan 3 perbandingan hasil perhitungan, yaitu : Hasil perhitungan *Fuzzy* manual, hasil simulasi, hasil perhitungan sistem pada robot telah dibangun.

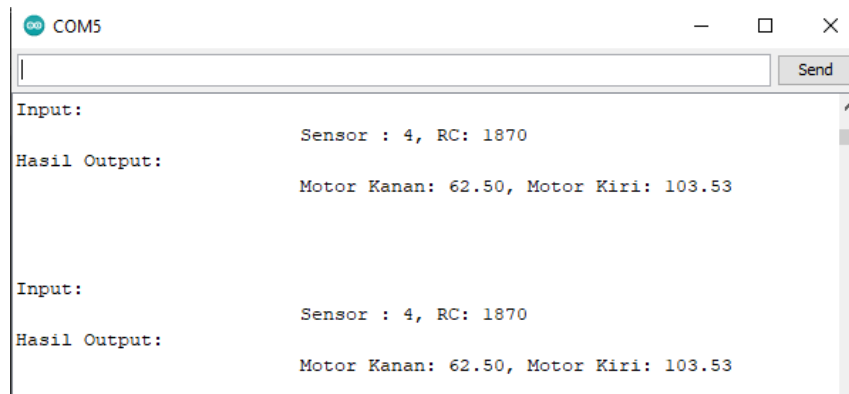
### 5.3.1 Pengujian *Fuzzy* Pada Sistem Robot

Pada pengujian model *Fuzzy* pada sistem robot semi-automatic navigation ini didapatkan hasil yang ditampilkan dari serial monitor aplikasi Arduino IDE dengan studi kasus pada sensor ultrasonik sebesar 4cm dan 1870 *input* dari *remote control* seperti pada Gambar 5.2.



Gambar 5.2 Pengujian *Fuzzy* pada sistem robot

Berdasarkan pada pengujian *Fuzzy* pada sistem robot ini didapatkan hasil output pada PWM motor kanan sebesar 62.5 dan pada motor kiri PWM yang dihasilkan sebesar 103.53. Gambar 5.3 menunjukkan hasil keluaran *Fuzzy* pada serial monitor aplikasi Arduino IDE.



Gambar 5.3 Hasil Perhitungan *Fuzzy* pada Aplikasi Arduino IDE

### 5.3.2 Perhitungan *Fuzzy* Secara Manual

Dalam perhitungan manual sistem kendali *Fuzzy* dengan contoh kasus di atas, perlu dilakukan beberapa langkah dan akan dibahas lebih detail.

#### 1. Menentukan Himpunan *Fuzzy*

##### Derajat Kebebasan Sensor Ultrasonik

Pada contoh kasus telah ditentukan jarak sensor ultrasonik terhadap *obstacle*, yaitu 4 cm. Dilihat dari Gambar 4.5, 4 cm termasuk kedalam himpunan *Fuzzy* Dekat. Berdasarkan pada Persamaan 4.1 yang telah dijabarkan sebelumnya, maka diperoleh :

$$\mu_{Dekat}(4) = 1 \quad (5.1)$$

##### Derajat Kebebasan *Remote Control*

Pada contoh kasus telah ditentukan juga nilai yang diberikan oleh *remote control* sebesar 1870 yang jika dilihat pada Gambar 4.6 maka 1870 termasuk kedalam himpunan *Fuzzy* Sedang dan Dalam. Berdasarkan pada Persamaan 4.5 dan Persamaan 4.6 yang telah dijabarkan sebelumnya, maka diperoleh :

$$\mu_{Sedang}(1870) = \frac{1900 - 1870}{50} = 0.6 \quad (5.2)$$

$$\mu_{Dalam}(1870) = \frac{1870 - 1850}{50} = 0.4 \quad (5.3)$$

#### 2. Aplikasi Fungsi Implikasi

Fungsi implikasi yang digunakan dalam proses ini adalah fungsi *MIN*

karena *rule base Fuzzy* yang dibangun adalah fungsi *AND*. Dengan fungsi *MIN* maka di ambil derajat keanggotaan minimum dari variabel *input* sebagai *outputnya*. Berdasarkan *rule base* yang ditunjukkan pada Tabel 4.5 kondisi pada kasus ini menunjukkan ada 2 aturan yang memberikan nilai (memenuhi) yaitu [R2] dan [R3].

[R2] = Jika sensor ultrasonik Dekat dan *remote control* Sedang maka motor kanan Normal dan motor kiri Cepat.

$$\begin{aligned}\lambda_{[R2]} &= \mu_{Dekat} \cap \mu_{Sedang} \\ \lambda_{[R2]} &= \min \{1, 0.6\} \\ \lambda_{[R2]} &= 0.6\end{aligned}\tag{5.4}$$

[R3] = Jika sensor ultrasonik Dekat dan *remote control* Dalam maka motor kanan Normal dan motor kiri Cepat.

$$\begin{aligned}\lambda_{[R3]} &= \mu_{Dekat} \cap \mu_{Dalam} \\ \lambda_{[R3]} &= \min \{1, 0.4\} \\ \lambda_{[R3]} &= 0.4\end{aligned}\tag{5.5}$$

### 3. Komposisi Aturan

Komposisi aturan adalah kesimpulan keseluruhan dengan mengambil derajat keanggotaan maksimum dari setiap konsekuensi aplikasi fungsi implikasi dengan menggabungkan semua kesimpulan dari setiap aturan, sehingga diperoleh daerah solusi *Fuzzy* sebagai berikut :

[R2] = Jika sensor ultrasonik Dekat dan *remote control* Sedang maka motor kanan Normal dan motor kiri Cepat.

Motor Kanan :

$$\begin{aligned}\frac{\alpha_{21} - 25}{25} &= 0.6 \\ \alpha_{21} &= 35\end{aligned}\tag{5.6}$$

Motor Kiri :

$$\begin{aligned}\alpha_{22} &= (0.6 \times 25)100 \\ \alpha_{22} &= 85\end{aligned}\tag{5.7}$$

[R3] = Jika sensor ultrasonik Dekat dan *remote control* Dalam maka motor kanan Normal dan motor kiri Cepat.

Motor Kanan :

$$\begin{aligned}\alpha_{31} &= (0.4 \times 25) + 25 \\ \alpha_{31} &= 35\end{aligned}\tag{5.8}$$

Motor Kiri :

$$\begin{aligned}\alpha_{32} &= (0.4 \times 25) - 100 \\ \alpha_{32} &= 65\end{aligned}\tag{5.9}$$

#### 4. Defuzzifikasi Motor Kanan

Langkah selanjutnya adalah mencari luas daerah dan momen keluaran PWM kecepatan motor yang sesuai, dalam hal ini terdapat dua luas daerah pada setiap keluaran dan momen pada keluaran PWM kecepatan motor. Luas daerah output PWM kecepatan motor kanan :

$$\begin{aligned}L_{A1} &= \frac{(40 - 25) \times 0.6}{2} = 4.5 \\ L_{A2} &= (85 - 40) \times 0.6 = 27 \\ L_{A3} &= \frac{(100 - 85) \times 0.6}{2} \\ L_{Total} &= 4.5 + 27 + 4.5 \\ &= 36\end{aligned}\tag{5.10}$$

Momen *output* PWM Motor Kanan :

$$\begin{aligned}M_1 &= \int_{25}^{40} \left(\frac{1}{25}z - 1\right)z dz = 157.5 \\ M_2 &= \int_{40}^{85} 0.6z dz = 1687.5 \\ M_3 &= \int_{85}^{100} \left(-\frac{1}{25}z + 4\right)z dz = 405 \\ M_{Total} &= 2250\end{aligned}\tag{5.11}$$

Untuk mengetahui nilai *output* PWM kecepatan motor kanan yaitu menggunakan Persamaan 5.12 seperti dibawah ini :

$$Z^* = \frac{2250}{36} = 62.5 \text{ PWM} \quad (5.12)$$

### 5. Defuzzifikasi Motor Kiri

Langkah selanjutnya adalah mencari luas daerah dan momen keluaran PWM kecepatan motor yang sesuai, dalam hal ini terdapat dua luas daerah pada setiap keluaran dan momen pada keluaran PWM kecepatan motor. Luas daerah output PWM kecepatan motor kiri :

$$\begin{aligned} L_{A1} &= \frac{(90 - 75) \times 0.6}{2} = 4.5 \\ L_{A2} &= (125 - 90) \times 0.6 = 21 \\ L_{Total} &= 4.5 + 21 \\ &= 25.5 \end{aligned} \quad (5.13)$$

Momen *output* PWM Motor kiri :

$$\begin{aligned} M_1 &= \int_{75}^{90} \frac{1}{(25z - 3)z} dz = 382.5 \\ M_2 &= \int_{90}^{125} 0.6z dz = 2257.5 \\ M_{Total} &= 2640 \end{aligned} \quad (5.14)$$

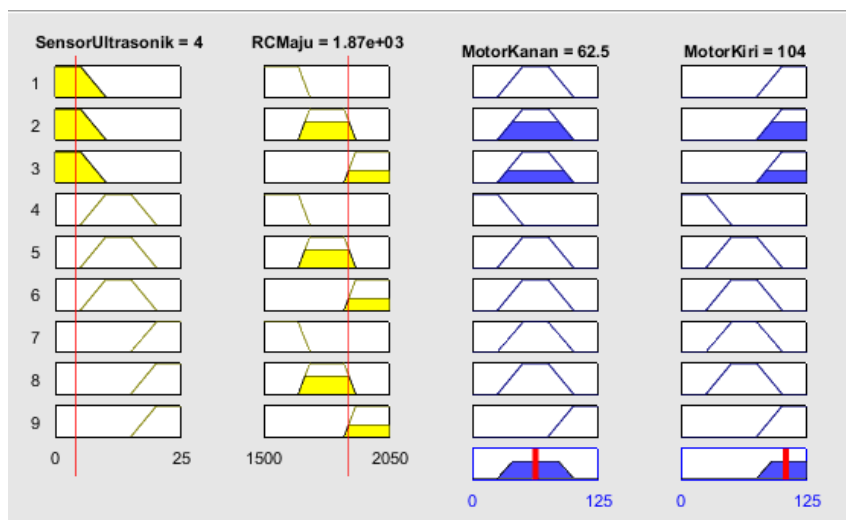
Untuk mengetahui nilai *output* PWM kecepatan motor kiri yaitu menggunakan Persamaan 5.15 seperti dibawah ini :

$$Z^* = \frac{2640}{25.5} = 103.53 \text{ PWM} \quad (5.15)$$

Setelah dilakukan perhitungan secara manual nilai PWM yg dihasilkan dengan studi kasus sensor ultrasonik berjarak 4 cm dan remote control memberikan perintah senilai 1870 didapatkan hasil PWM kecepatan motor kanan sebesar 62.5 PWM dan motor kiri sebesar 103.53 PWM

### 5.3.3 Pengujian *Fuzzy* Menggunakan Aplikasi Simulasi

Pada pengujian *Fuzzy* menggunakan aplikasi simulasi ini dilakukan dengan studi kasus yang sama dimana sensor ultrasonik mendeteksi *obstacle* yang berjarak 4 cm dan remote control memberikan perintah senilai 1870. Dari pengujian tersebut didapatkan hasil output PWM motor kanan sebesar 62.5 dan PWM motor kiri sebesar 104 seperti yang dilihat pada Gambar 5.4.



Gambar 5.4 Hasil Perhitungan *Fuzzy* pada Aplikasi Simulasi

### 5.3.4 Perbandingan Hasil Pengujian

Terjadi selisih perbedaan nilai *output* PWM antara simulasi *software* dan sistem yang dibangun. Perbedaan yang di tunjukan pada Tabel 5.5 tidak terlalu signifikan, dan masih sesuai dengan aturan *Fuzzy* yang telah ditentukan. Perbedaan hasil output tersebut dapat disebabkan beberapa faktor yang salah satunya adalah kesalahan dalam pembulatan bilangan. Pada Tabel 5.5 ditunjukkan hasil perhitungan pada pengujian secara manual, melalui aplikasi simulasi dan pada sistem yang dibangun.

Tabel 5.5 Hasil Perhitungan *Fuzzy*

<i>Output</i>	Perhitungan Manual	Hasil Simulasi	Sistem yang Dibangun
<b>PWM Kanan</b>	62.5	62.5	62.5
<b>PWM Kiri</b>	103.53	104	103.53

Terdapat perbedaan keluaran PWM motor kanan dan motor kiri pada sistem robot penghindar halangan dengan hasil pada software simulasi. Maka dari itu, perlu dicari nilai error pada sistem yang dibangun dengan persamaan berikut :

$$\%error = \left| \frac{x - x}{x} \right| \times 100\% \quad (5.16)$$

Keterangan :

$x$  = Hasil pegujian pada sistem

$x$  = Hasil pengujian pada aplikasi simulasi

1. Persentase nilai *error* motor kanan :

$$\%error = \left| \frac{62.5 - 62.5}{62.5} \right| \times 100\% = 0\% \quad (5.17)$$

2. Persentase nilai *error* motor kiri :

$$\%error = \left| \frac{103.53 - 104}{104} \right| \times 100\% = 0.005\% \quad (5.18)$$

Berdasarkan hasil pengujian perbandingan sistem robot dengan hasil pemodelan *Fuzzy* pada aplikasi simulasi didapatkan *error* sebesar 0.005% pada motor kiri dan 0% pada motor kanan. Dengan hasil nilai *error* tersebut maka metode *Fuzzy logic control* berhasil diimplementasikan pada robot ini dengan persentase keberhasilan 100% untuk motor kanan dan 99.995% pada motor kiri.

#### 5.4 Pengujian Respon Sistem

Pengujian respon sistem menggunakan waktu sebagai variabel bebasnya dalam sistem kendali, evaluasi respon keadaan dan keluaran terhadap waktu untuk menentukan karakteristik sinyal dari hasil data yang didapat. Pengujian respon sistem pada robot *semi-automatic navigation* dilakukan dengan cara menentukan titik awal robot terhadap *obstacle*. Dapat diketahui bahwa robot *semi-automatic navigation* pada penelitian ini memiliki tiga level kecepatan motor yang diperintahkan oleh *remote control* yaitu Cepat, Normal dan Lambat. Maka dari itu pengujian respon sistem robot *semi-automatic navigation* ini akan diuji untuk masing-masing kecepatan motor. Melalui hasil pembacaan sensor ultrasonik yaitu jarak titik awal robot yang telah ditentukan adalah 60cm sedangkan *setpoint* 35cm. Sehingga dilakukan pendekatan untuk mendapatkan karakteristik respon



sistem berdasarkan teori respon sistem yaitu dengan cara menentukan nilai balik berdasarkan perhitungan pembacaan sensor seperti pada Persamaan 5.19.

$$\text{Nilai Balik Sensor} = \text{Titik Awal Robot} - \text{Nilai Pembacaan Sensor} \quad (5.19)$$

Dari Persamaan 5.19 didapatkan hasil pada masing-masing kecepatan motor sebagai berikut :

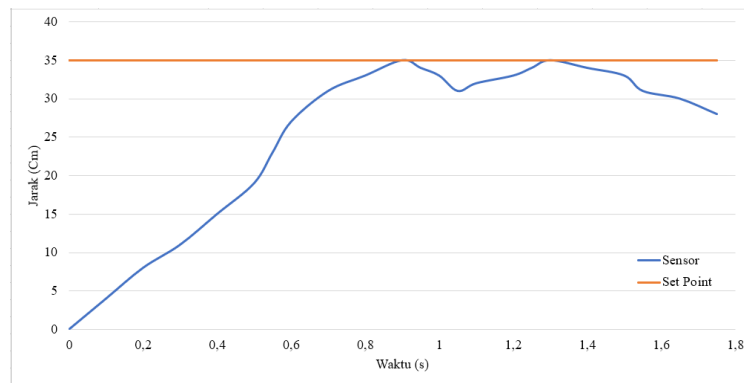
**1. Pengujian Respon Sistem Sensor Dengan Kecepatan Motor Lambat**

Dari Persamaan 5.19 didapatkan hasil pengujian respon sistem untuk kecepatan motor Lambat seperti pada Tabel 5.6.

Tabel 5.6 Hasil respon sistem sensor kecepatan motor Lambat

Waktu (s)	Sensor Ultrasonik (Cm)	Nilai Balik Sensor Ultrasonik (Cm)	Set Point (Cm)
0	60	0	35
0.1	56	4	35
0.2	52	8	35
0.3	49	11	35
0.4	45	15	35
0.5	41	19	35
0.55	37	23	35
0.6	33	27	35
0.7	29	31	35
0.8	27	33	35
0.9	25	35	35
0.95	26	34	35
1	27	34	35
1.05	29	31	35
1.1	28	32	35
1.2	27	33	35
1.25	26	34	35
1.3	25	35	35
1.4	26	34	35
1.5	27	33	35
1.55	29	31	35

Waktu (s)	Sensor Ultrasonik (Cm)	Nilai Balik Sensor Ultrasonik (Cm)	Set Point (Cm)
1.65	30	30	35
1.75	32	28	35



Gambar 5.5 Grafik respon sistem sensor kecepatan motor Lambat

Dari Tabel 5.6 dan Gambar 5.5 dapat diperoleh nilai *over shoot*, *error steady state* dan *settling time* sebagai berikut :

$$\begin{aligned} \text{Overshoot} &= \frac{35 - 35}{35} \times 100\% = 0\% \\ \text{ErrorSteadyState} &= \left| \frac{34 - 35}{35} \right| \times 100\% = 2.8\% \\ \text{SettlingTime} &= 1.3\text{s} \end{aligned} \quad (5.20)$$

Respon robot pada pengujian respon sistem sensor untuk kecepatan motor Lambat mencapai *steady state* respon dengan memakan waktu 0.8s selain itu didapatkan nilai *error steady state* sebesar 2.8% dan memiliki hasil nilai *overshoot* sebesar 0%. Hal tersebut dapat dilihat pada Gambar 5.5.

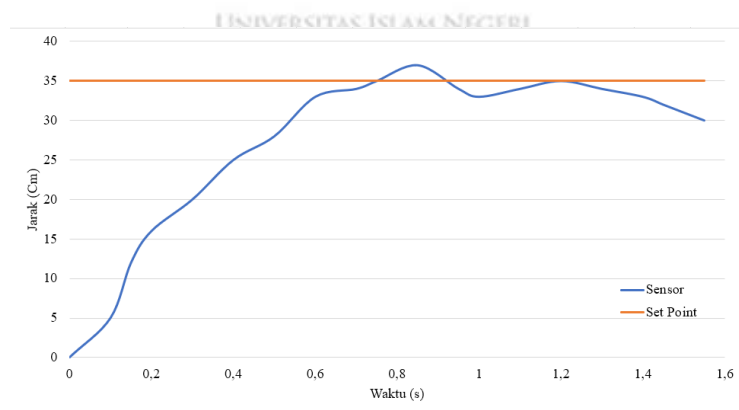
## 2. Pengujian Respon Sistem Sensor Dengan Kecepatan Motor Normal

Dari Persamaan 5.19 didapatkan hasil pengujian respon sistem untuk kecepatan motor Lambat seperti pada Tabel 5.7.

Tabel 5.7 Hasil respon sistem sensor kecepatan motor Normal

Waktu (s)	Sensor Ultrasonik (Cm)	Nilai Balik Sensor Ultrasonik (Cm)	Set Point (Cm)
0	60	0	35
0.1	55	5	35

Waktu (s)	Sensor Ultrasonik (Cm)	Nilai Balik Sensor Ultrasonik (Cm)	Set Point (Cm)
0.15	48	12	35
0.2	44	16	35
0.3	40	20	35
0.4	35	25	35
0.5	32	28	35
0.6	27	33	35
0.7	26	34	35
0.75	25	35	35
0.85	23	37	35
0.95	26	34	35
1	27	33	35
1.1	26	34	35
1.2	25	35	35
1.3	26	34	35
1.4	27	33	35
1.45	28	32	35
1.5	29	31	35
1.55	30	30	35



Gambar 5.6 Grafik respon sistem sensor kecepatan motor Normal

Dari Tabel 5.6 dan Gambar 5.5 dapat diperoleh nilai *over shoot*, *error steady state* dan *settling time* sebagai berikut :

$$Overshoot = \frac{37 - 35}{35} \times 100\% = 5.7\%$$

$$ErrorSteadyState = \left| \frac{34 - 35}{35} \right| \times 100\% = 2.9\% \quad (5.21)$$

$$SettlingTime = 1.2s$$

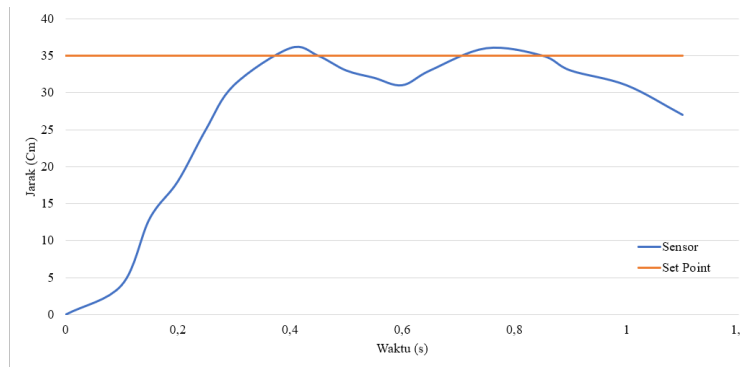
Respon robot pada pengujian respon sistem sensor untuk kecepatan motor Normal mencapai *steady state* respon dengan memakan waktu 0.75s selain itu didapatkan nilai *error steady state* sebesar 2.9% dan memiliki hasil nilai *overshoot* sebesar 5.7%. Hal tersebut dapat dilihat pada Gambar 5.6.

### 3. Pengujian Respon Sistem Sensor Dengan Kecepatan Motor Cepat

Dari Persamaan 5.19 didapatkan hasil pengujian respon sistem untuk kecepatan motor Lambat seperti pada Tabel 5.8.

Tabel 5.8 Hasil respon sistem sensor kecepatan motor Cepat

Waktu (s)	Sensor Ultrasonik (Cm)	Nilai Balik Sensor Ultrasonik (Cm)	Set Point (Cm)
0	60	0	35
0.1	56	4	35
0.15	47	13	35
0.2	42	18	35
0.25	35	25	35
0.3	29	31	35
0.4	24	36	35
0.45	25	35	35
0.5	27	33	35
0.55	28	32	35
0.6	29	31	35
0.65	27	33	35
0.75	24	36	35
0.85	25	35	35
0.9	27	33	35
1	29	31	35
1.1	33	27	35



Gambar 5.7 Grafik respon sistem sensor kecepatan motor Cepat

Dari Tabel 5.8 dan Gambar 5.7 dapat diperoleh nilai *over shoot*, *error steady state* dan *settling time* sebagai berikut :

$$\begin{aligned}
 \text{Overshoot} &= \frac{36 - 35}{35} \times 100\% = 2.9\% \\
 \text{ErrorSteadyState} &= \left| \frac{33 - 35}{35} \right| \times 100\% = 5.7\% \\
 \text{SettlingTime} &= 0.85s
 \end{aligned}
 \tag{5.22}$$

Respon robot pada pengujian respon sistem sensor untuk kecepatan motor Cepat mencapai *steady state* respon dengan memakan waktu 0.45s selain itu didapatkan nilai *error steady state* sebesar 5.7% dan memiliki hasil nilai *overshoot* sebesar 2.9%. Hal tersebut dapat dilihat pada Gambar 5.7.

## 5.5 Pengujian Manuver dan Perbandingan Robot

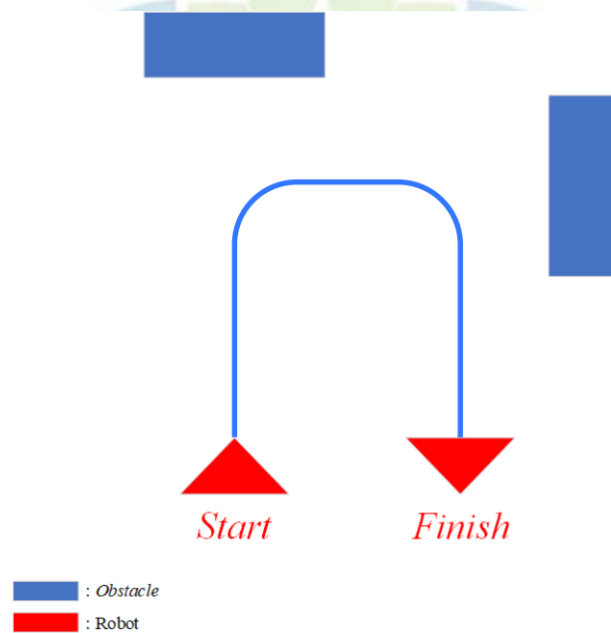
Pada pengujian manuver robot akan dilakukan terhadap robot serupa yang tidak menggunakan metode *Fuzzy* pada sitem kendalinya. Pada Tabel 5.9 dillihatkan *rule* untuk robot tanpa metode *Fuzzy*.

Tabel 5.9 *Rules* robot tanpa *Fuzzy*

No.	Sensor Ultrasonik (cm)	Remote Control	Motor Kanan (PWM)	Motor Kiri (PWM)
1	>35	>1750	100	100
2	>35	<1750	75	75
3	<35	>1750	50	100
4	<35	<1750	50	75

Setelah dilakukan pengujian *hardware* dan *software*, dilakukan pengujian perbandingan robot antara robot yang menggunakan logika Fuzzy dan robot yang tidak menggunakan logika Fuzzy, pengujian ini dapat mengetahui apakah program yang dihasilkan bekerja dengan baik atau tidak. Pengujian ini dilakukan dengan melihat respon robot saat menghindari sebuah objek kardus berbentuk kotak, kemudian menghitung dengan stopwatch berapa lama waktu yang dibutuhkan robot untuk bekerja dari titik awal hingga titik akhir ketika robot bermanuver menghindari objek.

Pengujian ini dilakukan dengan media *box* berbentuk kubus dengan tiap sisinya berukuran 15cm sebagai media pendeteksi robot. Pada pengujian ini kondisi lingkungan telah didesain sedemikian rupa seperti pada Gambar 5.8.



Gambar 5.8 Desain lingkungan pengujian manuver robot

Pada Gambar 5.8 akan diuji robot yang menggunakan metode *Fuzzy* dan robot tanpa menggunakan metode *Fuzzy*. Robot akan mulai bergerak dari titik *start* dan akan berakhir dititik *finish*. Gambar 5.9 menunjukkan kondisi lingkungan pengujian manuver robot.



Gambar 5.9 Kondisi lingkungan pengujian manuver robot

Pada Tabel 5.10 ditunjukkan hasil pengujian robot yang menggunakan metode *Fuzzy*.

Tabel 5.10 Pengujian Pergerakan Robot Dengan *Fuzzy*

Jarak Sensor (cm)	Remote Control	Motor Kanan (PWM)	Motor Kiri (PWM)	Keterangan
15	1910	62.5	62.5	Lurus Kecepatan Normal
18	1590	62.5	62.5	Lurus Kecepatan Normal
7	1830	62.5	82.6	Belok Kanan
9	1620	32.4	38.7	Belok Kanan Kecepatan Lambat
21	1765	62.5	62.5	Lurus Kecepatan Normal
4	1870	62.5	104	Belok Kanan Kecepatan Cepat
10	1740	62.5	62.5	Lurus Kecepatan Normal

Pada pengujian manuver robot dengan menggunakan metode *Fuzzy* didapatkan hasil seperti pada Tabel 5.10. Robot yang dibangun telah berhasil menghindari rintangan yang berada didepan dengan berbelok kearah kanan sesuai

dengan sistem yang telah dibangun. Tabel 5.10 menunjukkan bahwa hasil PWM sangat bervariasi, yang menunjukkan bahwa robot yang menggunakan metode *Fuzzy logic control* lebih stabil karena robot yang menggunakan metode *Fuzzy* dapat beradaptasi dengan lingkungan sekitar. Selanjutnya dilakukan pengujian pergerakan robot tanpa metode *Fuzzy* dan didapat hasil seperti pada Tabel 5.11.

Tabel 5.11 Pengujian Pergerakan Robot Tanpa *Fuzzy*

Jarak Sensor (cm)	Remote Control	Motor Kanan (PWM)	Motor Kiri (PWM)	Keterangan
15	1910	100	100	Lurus Kecepatan Cepat
18	1590	75	75	Lurus Kecepatan Normal
7	1830	50	100	Menabrak
9	1620	50	75	Belok Kanan Kecepatan Normal
21	1765	100	100	Lurus Kecepatan Cepat
4	1870	50	100	Menabrak
10	1740	75	75	Lurus Kecepatan Normal

Dari hasil pengujian pada Tabel 5.11, robot tanpa *Fuzzy* beberapa kali menabrak *obstacle* yang menghalanginya dan didapatkan hasil PWM yang tidak bervariasi karena pada robot tanpa *Fuzzy* nilai PWM ditentukan sama besar pada semua kondisi pengujian. Oleh karena itu, robot yang tidak menggunakan *Fuzzy logic control* dapat dikatakan tidak stabil karena tidak dapat beradaptasi dengan dinamika lingkungan.

## 5.6 Analisis Pengujian

Pada penelitian ini telah dilakukan beberapa pengujian, yaitu pengujian sensor ultrasonik HC-SR04 yang dilakukan dengan membandingkan jarak yang ditentukan menggunakan meteran dengan hasil pembacaan oleh sensor ultrasonik yang memiliki rata-rata selisih jarak sebesar 0,02cm dan pengujian material yang dapat dideteksi oleh sensor. Pengujian *remote control* dilakukan dengan mengukur



jarak komunikasi antara robot dan *remote control* dengan menambah jarak 10 meter hingga robot tidak lagi terhubung oleh *remote control*. Pengujian motor *diver shield* L293D dilakukan dengan dua cara yaitu mengukur waktu tempuh robot dengan PWM yang telah ditentukan. Selanjutnya pengujian *software* dilakukan dengan membandingkan hasil perhitungan manual secara matematis, dengan hasil simulasi pada aplikasi Matlab dan perhitungan *Fuzzy* dengan sistem yang telah dibangun. Pengujian yang terakhir, yaitu perbandingan manuver robot yang telah diimplementasikan *Fuzzy logic* sebagai sistem kendalinya dengan robot yang tidak menggunakan *Fuzzy logic control*.

Pada pengujian model *Fuzzy* robot ini telah dibandingkan dengan tiga hasil pengujian dengan studi kasus yang telah ditentukan sebelumnya, yaitu sensor ultrasonik yang mendeteksi *obstacle* pada jarak 4cm dan *remote control* yang memberikan perintah sebesar 1870. Pada bagian ini dilakukan tiga pengujian, yaitu perhitungan secara manual yang didapatkan hasil nilai keluaran berupa PWM motor kanan sebesar 62,5 PWM dan motor kiri sebesar 103,53 PWM. Dari pengujian pada aplikasi simulasi didapatkan nilai *output* PWM pada motor kanan sebesar 62,5 PWM dan motor kiri sebesar 104 PWM. Pada pengujian dari sistem yang telah dibangun didapatkan hasil keluaran sebesar 62,5 PWM untuk motor kanan dan 103,53 PWM untuk motor kiri. Pengujian metode *Fuzzy logic control* ini dapat dinyatakan berhasil berdasarkan pengujian yang dilakukan berjalan dengan baik, karena didapatkan nilai *error* sebesar 0,005% untuk motor kiri dan 0% untuk nilai *error* pada motor kanan.

Pada pengujian respon sistem robot dilakukan pengujian pada sensor ultrasonik dengan jarak awal robot ke rintangan sejauh 60cm dan dilakukan dengan tiga kecepatan motor yang dipengaruhi oleh masukan dari *remote control* yaitu Lambat, Normal dan Cepat. Pengujian respon sistem pertama dilakukan pada kecepatan robot Lambat didapatkan hasil nilai *overshoot* sebesar 0% dan nilai *error steady state* sebesar 2.8% selain itu robot pada kecepatan Lambat diperlukan waktu 0.8s untuk mencapai *steady state*. Pengujian respon sistem kedua dilakukan pada kecepatan robot Normal didapatkan hasil nilai *overshoot* sebesar 5.7% dan nilai *error steady state* sebesar 2.9% selain itu robot pada kecepatan Lambat diperlukan waktu 0.75s untuk mencapai *steady state*. Pengujian respon sistem ketiga dilakukan pada kecepatan robot Cepat didapatkan hasil nilai *overshoot* sebesar 2.9% dan nilai *error steady state* sebesar 5.7% selain itu robot pada kecepatan Cepat diperlukan

waktu 0.45s untuk mencapai *steady state*.

Pengujian perbandingan robot yang menggunakan metode *Fuzzy* dengan robot yang tidak menggunakan metode *Fuzzy* telah dilakukan dengan hasil bahwa robot yang menggunakan metode *Fuzzy* dapat menghindari rintangan yang ditentukan dengan hasil PWM yang bervariasi seperti yang ditunjukkan pada Tabel 5.10. Hal tersebut menunjukkan bahwa metode *Fuzzy* yang diterapkan pada robot dapat membuat robot berjalan lebih stabil karena robot dapat beradaptasi dengan lingkungan sekitar. Penyesuaian ini karena proses pada metode *Fuzzy* mempertimbangkan tingkat keabuan, yang berarti mempertimbangkan semua nilai (dari 0 hingga 1) yang diterima oleh sensor. Namun hasilnya berbeda saat pengujian robot tanpa menggunakan metode *Fuzzy* pada Tabel 5.11, beberapa hasil menunjukkan bahwa robot tidak menghindari rintangan. Hal ini dikarenakan proses penentuan nilai PWM yang tidak bervariasi dalam menghadapi segala kondisi yang membuat robot tidak stabil karena tidak dapat beradaptasi dengan lingkungan yang dihadapinya.



## BAB VI KESIMPULAN DAN SARAN

### 6.1 Kesimpulan

Berdasarkan penjelasan di atas dan hasil pembuatan rancang bangun *semi-automatic navigation robot* menggunakan *remote control* dan sensor ultrasonik berbasis *fuzzy logic control*, kesimpulan yang didapat adalah:

1. Penelitian ini telah berhasil merancang sebuah *semi-automatic navigation robot* berbasis *fuzzy logic control* sesuai spesifikasi yang diinginkan.
2. Pada penelitian ini telah dilakukan pengujian sistem *semi-automatic navigation robot* berbasis *Fuzzy Logic Control* sehingga didapatkan hasil analisis sebagai berikut :
  - (a) Berdasarkan pengujian perangkat keras didapatkan bahwa akurasi sensor hanya memiliki rata-rata *error* sebesar 0,02cm yang mempengaruhi pengujian gerak robot sehingga membuat robot bekerja dengan baik.
  - (b) Dalam proses penerapan algoritma *fuzzy logic control* untuk *semi-automatic navigation robot*, dilakukan perbandingan hasil perhitungan fuzzy pada sistem yang dibangun dengan hasil perhitungan pada aplikasi simulasi dan perhitungan manual. Berdasarkan perbandingan tersebut, penerapan metode *fuzzy* pada *semi-automatic navigation robot* ini dapat dikatakan berhasil, karena nilai *error* pada motor kanan adalah 0% dan nilai *error* pada motor kiri adalah 0,005%.
  - (c) Pada pengujian respon sistem robot dapat memberikan respon yang sesuai meskipun masih memiliki nilai *overshoot*. Didapatkan nilai *overshoot* 0%, *error steady state* 2.8% dan *settling time* selama 1,3 detik untuk kecepatan Lambat Pada kecepatan normal didapatkan nilai *overshoot* 5.7%, *error steady state* 2.8% dan *settling time* selama 1,2 detik. Untuk kecepatan Cepat didapatkan nilai *overshoot* 2.9%, *error steady state* 5.7% dan *settling time* selama 0.85 detik.
  - (d) Pada pengujian perbandingan robot didapatkan bahwa robot yang menggunakan metode *fuzzy logic control* bekerja lebih stabil karena dapat beradaptasi dengan lingkungan yang menghalanginya dan dapat menghindari *obstacle*.

## 6.2 Saran

Berdasarkan hasil penelitian ditemukan beberapa kekurangan yang terdapat pada penelitian ini, sehingga peneliti mengajukan beberapa saran demi pengembangan penelitian ini. Saran tersebut antara lain sebagai berikut :

1. Sistem robot masih dapat dikembangkan dengan menambah satu atau dua buah sensor ultrasonik HC-SR04 atau dapat menggantikannya dengan sensor lain seperti LIDAR.
2. Pengimplementasian sistem atau metode dapat dikembangkan dengan menggabungkan metode *fuzzy* dan metode lainnya seperti PID agar dapat menerapkan konsep "*Hybrid control system*".



## DAFTAR PUSTAKA

- [1] K. Bhagat, S. Deshmukh, S. Dhonde, and S. Ghag, "Obstacle avoidance robot," *International Journal of Science, Engineering and Technology Research (IJSETR)*, vol. 5, no. 2, pp. 439–442, 2016.
- [2] W. S. Pambudi, "Rancang bangun 3 wheels omni-directional mobile robot menggunakan sensor position sensitive device (psd) serta sensor vision dengan metode kendali fuzzy logic controller (flc) untuk menghindari halangan," *Semantik*, vol. 1, no. 1, 2011.
- [3] M. Jimmy and Y. Y. Christianto, "Pembuatan pengendali manual nirkabel untuk automated guided vehicle (agv) menggunakan modul radio zig-bee di pt. astra otoparts divisi winteq," *Technologic*, vol. 2, no. 1, 2011.
- [4] A. I. Amilah *et al.*, "Simulasi sistem kendali kestabilan motor dc menggunakan kendali pid dan fuzzy logic cocontrol (flc)," Ph.D. dissertation, Universitas Negeri Semarang, 2016.
- [5] A. Najmurrokhman, K. Kusnandar, B. H. Wibowo, and M. H. R. Amarullah, "Desain pengendali logika fuzzy tipe takagi-sugeno-kang untuk mengatur kecepatan gerak mobile robot," *Prosiding Semnastek*, 2018.
- [6] A. D. Prasetyo, "Rancang bangun sistem deteksi halang rintang robot quadcopter menggunakan arduino uno," Ph.D. dissertation, UIN Sunan Gunung Djati Bandung, 2016.
- [7] R. H. Suryanegara and E. P. Laksana, "Rancang bangun robot pembersih lantai otomatis berbasis mikrokontroler atmega-16a," *Arsitron*, vol. 9, no. 01, pp. 27–35, 2019.
- [8] Z. Adeyanto, A. Izzuddin, and N. Hikmah, "Rancang bangun robot vacuum cleaner dengan menerapkan proportional logic untuk pengaturan navigasi," *Mnemonic: Jurnal Teknik Informatika*, vol. 3, no. 2, pp. 15–20, 2020.

- [9] D. Kurnia, R. Mardiaty, M. R. Efendi, and A. E. Setiawan, "Rancang bangun robot pemadam api menggunakan kontrol bluetooth dan virtual reality," *Jurnal Telekomunikasi, Elektronika, Komputasi dan Kontrol Vol. 5 No. 2*, 2019.
- [10] A. A. Zaki, E. Mulyana, R. Mardiaty *et al.*, "Modeling wall tracer robot motion based on fuzzy logic control," in *2020 6th International Conference on Wireless and Telematics (ICWT)*. IEEE, 2020, pp. 1–6.
- [11] A. Triwiyatno, "Konsep umum sistem kontrol," *Universitas Diponegoro*, 2011.
- [12] K. Ogata, *Modern control engineering*. Prentice hall India, 2010, vol. 5.
- [13] S. Kusumadewi, S. Hartati, A. Harjoko, and R. Wardoyo, "Fuzzy multi-attribute decision making," *Yogyakarta: Graha Ilmu*, pp. 78–79, 2006.
- [14] N. Ismail, I. Nursalim, H. M. Saputra, and T. S. Gunawan, "Implementation of fuzzy logic control system on rotary car parking system prototype," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 12, no. 2, pp. 706–715, 2018.
- [15] I. Iancu, "A mamdani type fuzzy logic controller," *Fuzzy logic-controls, concepts, theories and applications*, pp. 325–350, 2012.
- [16] A. Lahsasna and W. C. Seng, "An improved genetic-fuzzy system for classification and data analysis," *Expert Systems with Applications*, vol. 83, pp. 49–62, 2017.
- [17] R. P. Indahningrum, "Rancang bangun wall tracer robot menggunakan sensor ultrasonic hc-sr04 berbasis fuzzy logic control," Ph.D. dissertation, UIN Sunan Gunung Djati Bandung, 2020.
- [18] E. S. Purnomo, A. Rochim, and E. D. Widiyanto, "Handsight: Hand-mounted device untuk membantu tunanetra berbasis ultrasonic dan arduino," *Jurnal Teknologi dan Sistem Komputer*, vol. 3, pp. 51–57, 01 2016.

- [19] M. Firdaus, M. Syaryadhi, and A. Rahman, “Pengendalian robot mobil otonom pemotong rumput menggunakan metode logika fuzzy,” *Jurnal Karya Ilmiah Teknik Elektro*, vol. 2, no. 2, 2017.



## LAMPIRAN

### Lampiran A : Kode Program untuk Semi-Automatic Navigation Robot

Pada bagian ini dijabarkan *source code* yang digunakan untuk memrogram Semi-Automatic Navigation Robot agar berjalan sesuai yang telah dirancang.

```
1 #include <Fuzzy.h>
2 #include <AFMotor.h>
3 // For scope, instantiate all objects you will need to access in loop()
4 // It may be just one Fuzzy, but for demonstration, this sample will print
5 // all FuzzySet pertinence
6 #define echoPin1 24 //Echo Pin depan
7 #define trigPin1 25 //Trigger Pin
8
9 #define ch1 30 //ch1 gas toogle
10 #define ch2 31 //ch2 spinner
11
12 int vch1;
13 int vch2;
14
15 int maximumRange = 200; //kebutuhan akan maksimal range
16 int minimumRange = 00; //kebutuhan akan minimal range
17 long duration1, distancel; //waktu untuk kalkulasi jarak
18
19 AF_DCMotor roda_kanan(1);
20 AF_DCMotor roda_kiri(4);
21 // Fuzzy
22 Fuzzy *fuzzy = new Fuzzy();
23
24 // FuzzyInput 1 - CH1 Maju
25 FuzzySet *ma_pelan = new FuzzySet(1500, 1500, 1650, 1700);
26 FuzzySet *ma_sedang = new FuzzySet(1650, 1700, 1850, 1900);
27 FuzzySet *ma_dalam = new FuzzySet(1850, 1900, 2050, 2050);
28
29 // FuzzyInput 2 - Sensor depan
```

UNIVERSITAS ISLAM NEGERI  
SUNAN GUNUNG DJATI  
BANDUNG



```

30 FuzzySet *de_dekat = new FuzzySet(0, 0, 25, 30);
31 FuzzySet *de_tengah = new FuzzySet(25, 30, 35, 40);
32 FuzzySet *de_jauh = new FuzzySet(35, 40, 45, 45);
33
34 // FuzzyOutput 1 - Roda Kiri
35 FuzzySet *vl_stop = new FuzzySet(0, 0, 0, 0);
36 FuzzySet *vl_lambat = new FuzzySet(0, 0, 45, 70);
37 FuzzySet *vl_normal = new FuzzySet(45, 70, 95, 120);
38 FuzzySet *vl_cepat = new FuzzySet(95, 120, 145, 145);
39
40 // FuzzyOutput 2 - Roda Kanan
41 FuzzySet *vr_stop = new FuzzySet(0, 0, 0, 0);
42 FuzzySet *vr_lambat = new FuzzySet(0, 0, 25, 50);
43 FuzzySet *vr_normal = new FuzzySet(25, 50, 75, 100);
44 FuzzySet *vr_cepat = new FuzzySet(75, 100, 125, 125);
45
46 void setup()
47 {
48     // Set the Serial output
49     Serial.begin(9600);
50     // Set a random seed
51     randomSeed(analogRead(0));
52
53     // Every setup must occur in the function setup()
54
55     // FuzzyInput
56     FuzzyInput *sensorchmaju = new FuzzyInput(1);
57
58     sensorchmaju->addFuzzySet(ma_pelan);
59     sensorchmaju->addFuzzySet(ma_sedang);
60     sensorchmaju->addFuzzySet(ma_dalam);
61     fuzzy->addFuzzyInput(sensorchmaju);
62
63     // FuzzyInput
64     FuzzyInput *sensordepan = new FuzzyInput(2);
65
66     sensordepan->addFuzzySet(de_dekat);
67     sensordepan->addFuzzySet(de_tengah);
68     sensordepan->addFuzzySet(de_jauh);
69     fuzzy->addFuzzyInput(sensordepan);
70
71
72     // FuzzyOutput
73     FuzzyOutput *rodakanan = new FuzzyOutput(1);
74
75     rodakanan->addFuzzySet(vr_stop);
76     rodakanan->addFuzzySet(vr_lambat);
77     rodakanan->addFuzzySet(vr_normal);
78     rodakanan->addFuzzySet(vr_cepat);
79     fuzzy->addFuzzyOutput(rodakanan);
80
81     // FuzzyOutput
82     FuzzyOutput *rodakiri = new FuzzyOutput(2);
83

```



```

84   rodakiri->addFuzzySet(vl_stop);
85   rodakiri->addFuzzySet(vl_lambat);
86   rodakiri->addFuzzySet(vl_normal);
87   rodakiri->addFuzzySet(vl_cepat);
88   fuzzy->addFuzzyOutput(rodakiri);
89
90   //Building FuzzyRule
91   FuzzyRuleAntecedent *pelanAnddekat = new FuzzyRuleAntecedent();
92   pelanAnddekat->joinWithAND(ma_pelan, de_dekat);
93
94   FuzzyRuleConsequent *O1_kanan = new FuzzyRuleConsequent();
95   O1_kanan->addOutput(vr_stop);
96   O1_kanan->addOutput(vl_cepat);
97
98   FuzzyRule *fuzzyRule1 = new FuzzyRule(1, pelanAnddekat, O1_kanan);
99   fuzzy->addFuzzyRule(fuzzyRule1);
100
101  //detebede
102  FuzzyRuleAntecedent *sedangAnddekat = new FuzzyRuleAntecedent();
103  sedangAnddekat->joinWithAND(ma_sedang, de_dekat);
104
105  FuzzyRuleConsequent *O2_kanan = new FuzzyRuleConsequent();
106  O2_kanan->addOutput(vr_stop);
107  O2_kanan->addOutput(vl_cepat);
108
109  FuzzyRule *fuzzyRule2 = new FuzzyRule(2, sedangAnddekat, O2_kanan);
110  fuzzy->addFuzzyRule(fuzzyRule2);
111
112  //dejabede
113  FuzzyRuleAntecedent *dalamAnddekat = new FuzzyRuleAntecedent();
114  dalamAnddekat->joinWithAND(ma_dalam, de_dekat);
115
116  FuzzyRuleConsequent *O3_kanan = new FuzzyRuleConsequent();
117  O3_kanan->addOutput(vr_stop);
118  O3_kanan->addOutput(vl_cepat);
119
120  FuzzyRule *fuzzyRule3 = new FuzzyRule(3, dalamAnddekat, O3_kanan);
121  fuzzy->addFuzzyRule(fuzzyRule3);
122
123  //dedebete
124  FuzzyRuleAntecedent *pelanAndtengah = new FuzzyRuleAntecedent();
125  pelanAndtengah->joinWithAND(ma_pelan, de_tengah);
126
127  FuzzyRuleConsequent *O4_lurus = new FuzzyRuleConsequent();
128  O4_lurus->addOutput(vr_lambat);
129  O4_lurus->addOutput(vl_lambat);
130
131  FuzzyRule *fuzzyRule4 = new FuzzyRule(4, pelanAndtengah, O4_lurus);
132  fuzzy->addFuzzyRule(fuzzyRule4);
133
134  //betedete
135  FuzzyRuleAntecedent *sedangAndtengah = new FuzzyRuleAntecedent();
136  sedangAndtengah->joinWithAND(ma_sedang, de_tengah);
137

```

```

138 FuzzyRuleConsequent *O5_lurus = new FuzzyRuleConsequent();
139 O5_lurus->addOutput(vr_lambat);
140 O5_lurus->addOutput(vl_normal);
141
142 FuzzyRule *fuzzyRule5 = new FuzzyRule(5, sedangAndtengah, O5_lurus);
143 fuzzy->addFuzzyRule(fuzzyRule5);
144
145 //dejabete
146 FuzzyRuleAntecedent *dalamAndtengah = new FuzzyRuleAntecedent();
147 dalamAndtengah->joinWithAND(ma_dalam, de_tengah);
148
149 FuzzyRuleConsequent *O6_lurus = new FuzzyRuleConsequent();
150 O6_lurus->addOutput(vr_stop);
151 O6_lurus->addOutput(vl_lambat);
152
153 FuzzyRule *fuzzyRule6 = new FuzzyRule(6, dalamAndtengah, O6_lurus);
154 fuzzy->addFuzzyRule(fuzzyRule6);
155
156 //dedebeja
157 FuzzyRuleAntecedent *pelanAndjauh = new FuzzyRuleAntecedent();
158 pelanAndjauh->joinWithAND(ma_pelan, de_jauh);
159
160 FuzzyRuleConsequent *O7_lurus = new FuzzyRuleConsequent();
161 O7_lurus->addOutput(vr_lambat);
162 O7_lurus->addOutput(vl_lambat);
163
164 FuzzyRule *fuzzyRule7 = new FuzzyRule(7, pelanAndjauh, O7_lurus);
165 fuzzy->addFuzzyRule(fuzzyRule7);
166
167 //detebeja
168 FuzzyRuleAntecedent *sedangAndjauh = new FuzzyRuleAntecedent();
169 sedangAndjauh->joinWithAND(ma_sedang, de_jauh);
170
171 FuzzyRuleConsequent *O8_lurus = new FuzzyRuleConsequent();
172 O8_lurus->addOutput(vr_normal);
173 O8_lurus->addOutput(vl_normal);
174
175 FuzzyRule *fuzzyRule8 = new FuzzyRule(8, sedangAndjauh, O8_lurus);
176 fuzzy->addFuzzyRule(fuzzyRule8);
177
178 //dejabeja
179 FuzzyRuleAntecedent *dalamAndjauh = new FuzzyRuleAntecedent();
180 dalamAndjauh->joinWithAND(ma_dalam, de_jauh);
181
182 FuzzyRuleConsequent *O9_lurus = new FuzzyRuleConsequent();
183 O9_lurus->addOutput(vr_cepat);
184 O9_lurus->addOutput(vl_cepat);
185
186 FuzzyRule *fuzzyRule9 = new FuzzyRule(9, dalamAndjauh, O9_lurus);
187 fuzzy->addFuzzyRule(fuzzyRule9);
188
189 pinMode(trigPin1, OUTPUT);
190 pinMode(echoPin1, INPUT);
191 pinMode(ch1, INPUT);

```

```

190 pinMode(echoPin1, INPUT);
191 pinMode(ch1, INPUT);
192 pinMode(ch2, INPUT);
193 roda_kanan.run(FORWARD);
194 roda_kiri.run(FORWARD);
195 }
196
197 void loop()
198 {
199
200 digitalWrite(trigPin1, LOW);delayMicroseconds(2);
201 digitalWrite(trigPin1, HIGH);delayMicroseconds(10);
202 digitalWrite(trigPin1, LOW);
203 duration1 = pulseIn(echoPin1, HIGH);
204
205 //perhitungan untuk dijadikan jarak
206 distancel = duration1/58.2;
207
208 vch1=pulseIn(ch1,HIGH);
209 //=====Pembacaan Nilai Gelombang Ch2
210 vch2=pulseIn(ch2,HIGH);
211 //=====Konversi nilai gelombang
212
213 Serial.print("RC1 : ");
214 Serial.println(vch1);
215 Serial.print("distancel : ");
216 Serial.println(distancel);
217
218
219 fuzzy->setInput(1, vch1);
220 fuzzy->setInput(2, distancel);
221
222 fuzzy->fuzzify();
223
224 float output_vr = fuzzy->defuzzify(1);
225 float output_vl = fuzzy->defuzzify(2);
226
227 roda_kanan.setSpeed(output_vr);
228 roda_kiri.setSpeed(output_vl);
229
230 Serial.println("Result: ");
231 Serial.print("\t\t\t Vr: ");
232 Serial.print(output_vr);
233 Serial.print(", and Vl: ");
234 Serial.println(output_vl);
235
236 delay(50);
237
238 }

```

