



TESIS KI - 142502

***REMOTE CONTROL KOMUNIKASI ROBOT
BERBASIS PERGERAKAN TANGAN PADA
SMARTPHONE DENGAN METODE LOGIKA FUZZY***

**TENGGU MUSRI
NRP. 5114201064**

**DOSEN PEMBIMBING
Dr. Darlis Heru Murti, S.Kom, M.Kom.
Abdul Munif, S. Kom, M. Sc**

**PROGRAM MAGISTER
BIDANG KEAHLIAN KOMPUTASI CERDAS DAN VISUALISASI
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016**



TESIS KI 142502

**REMOTE CONTROL OF ROBOT COMMUNICATION
BASED ON HAND MOTION IN SMARTPHONE
USING FUZZY LOGIC METHOD**

TENGGU MUSRI
NRP. 5114201064

SUPERVISOR
Dr. Darlis Heru Murti, S.Kom, M.Kom.
Abdul Munif, S. Kom, M. Sc

**MAGISTER PROGRAM
INTELLIGEN COMPUTATION AND VISUALIZATION
DEPARTMENT OF INFORMATIC ENGINEERING
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016**

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)
di
Institut Teknologi Sepuluh Nopember Surabaya

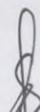
oleh:
TENGGU MUSRI
Nrp. 5114201064

Dengan judul :
Remote Control Komunikasi Robot Berbasis Pergerakan Tangan Pada Smartphone Dengan
Metode Logika Fuzzy

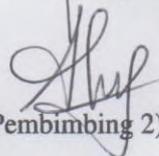
Tanggal Ujian : 1 Juli 2016
Periode Wisuda : September 2016

Disetujui oleh:

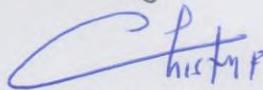
Dr. Darlis Heru Murti, S.Kom, M.Kom
NIP. 197712172003121001


(Pembimbing 1)

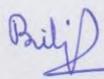
Abdul Munif, S.Kom, M.Sc
NIP. 198608232015041004


(Pembimbing 2)

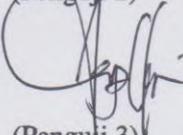
Dr. Eng. Chastine Fatichah, S.Kom, M.Kom
NIP. 197512202001122002


(Penguji 1)

Bilqis Amaliah, S.Kom, M.Kom
NIP. 197509172001122002

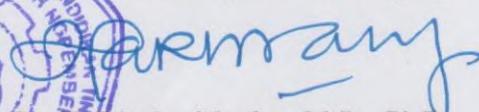

(Penguji 2)

Arya Yudhi Wijaya, S.Kom, M.Kom
NIP. 198409042010121002


(Penguji 3)



Direktur Program Pasca Sarjana,


Prof. Dr. Djauhar Manfaat, M.Sc., Ph.D.
NIP. 196012021987011001

Remote Control Komunikasi Robot Berbasis Pergerakan Tangan Pada Smartphone Dengan Metode Logika Fuzzy

Nama mahasiswa : Tengku Musri
NRP : 5114201064
Pembimbing I : Dr. Darlis Heru Murti, S.Kom., M.Kom
Pembimbing II : Abdul Munif, S.Kom., M.Sc

ABSTRAK

Robot mobil (*mobile robot*) adalah sebuah mesin yang mampu bergerak pada suatu kondisi tertentu. Robot mobil dapat diklasifikasi berdasarkan dua hal, yaitu menurut lingkungan tempat robot tersebut bekerja dan alat kendali yang digunakan untuk menggerakkan robot. Terdapat berbagai macam media yang digunakan untuk mengendalikan robot, yaitu *remote control*, *computer*, *joystick*, atau *smartphone* yang memiliki sensor *accelerometer*. Sensor *accelerometer* merupakan sebuah sensor yang digunakan untuk merubah tampilan pada *smartphone* menjadi *portrait* dan *landscape*, selain itu biasa juga digunakan untuk permainan berjenis simulasi. Sensor ini dapat mengukur kemiringan suatu benda karena memiliki 3 sumbu axis X, Y, dan Z. Ketiga sumbu tersebut dapat digunakan untuk mengendalikan sebuah robot dengan koneksi antara dua *platform* yang berbeda menggunakan media komunikasi *bluetooth*. Pada robot beroda, pengendalian perintah robot untuk menggerakkan robot maju, mundur, belok kiri, dan belok kanan dapat dilakukan berdasarkan kemiringan yang terdapat pada *accelerometer* yang tertanam pada *smartphone*. Masih terdapat kelemahan pada metode pengendalian ini, yaitu kecepatan robot yang konstan ketika digerakkan sehingga kemungkinan robot menabrak halangan yang ada di depan sangatlah besar.

Penelitian ini mengusulkan sebuah pengontrolan *mobile robot* melalui gerakan tangan memanfaatkan sensor *accelerometer* yang terdapat pada *smartphone* menggunakan media komunikasi *bluetooth*. Kecepatan motor DC dikontrol dengan menggunakan logika *fuzzy* untuk menentukan kecepatan yang sesuai dengan keadaan sehingga robot mobil tidak akan menabrak halangan. Kecepatan motor DC digunakan sebagai masukan pada metode PWM (*Pulse Width Modulation*) sebagai penggerak motor. Masukan jarak pada mobil robot didapatkan dari sensor ultrasonik akan diproses oleh mikrokontroler. Hasil keluaran berupa kecepatan putaran motor DC yang berubah secara adaptif berdasarkan informasi jarak yang di kirim oleh sensor *ultrasonik*. Penggunaan metode tersebut membuat pergerakan robot lebih halus dan lebih persisi pada saat berhenti ketika menemukan halangan.

Dalam skenario ujicoba, digunakan beberapa kecepatan dan jarak pada sebuah ruangan dengan tujuan untuk mengetahui performa gerakan robot ketika di kontrol. Kecepatan yang digunakan adalah 160, 200 dan 240 pwm. Jarak yang digunakan adalah 20, 30, 40, 50, 60 dan 70 cm. Berdasarkan pengujian pada metode usulan, diperoleh hasil nilai *error* terkecil yaitu pada jarak 50 cm dengan kecepatan 200 pwm. Hasil yang didapatkan dari sepuluh kali percobaan, robot berhenti secara otomatis dengan nilai *error* melewati jarak yang ditentukan rata-rata 1.45 cm dari

threshold jarak 50 cm. Hasil pengujian menunjukkan bahwa tingkat akurasi kontrol menggunakan logika *fuzzy* memberikan hasil yang lebih baik dari segi akurasi kontrol.

Kata Kunci: *Mobile robot, fuzzy logic, Android, accelerometer, bluetooth, pwm*

Remote Control of Robot Communication Based On Hand Motion in Smartphone Using Fuzzy Logic Method

Nama mahasiswa : Tengku Musri
NRP : 5114201064
Pembimbing I : Dr. Darlis Heru Murti, S.Kom., M.Kom
Pembimbing II : Abdul Munif, S.Kom., M.Sc

ABSTRACT

Robot car is a machine that is able to move in a certain condition. Robot cars can be classified based on two things, depending on the environment the robot works and control device that used to move the robot. There are various kinds of media are used to control the robot, i.e. remote control, computer, joystick, or a smartphone which has an accelerometer sensor. The accelerometer sensor is a sensor that is used to change the screen orientation on a smartphone, but it is also commonly used for game with simulation genre. This sensor can measure the slope of an object because it has a 3-axis axis X, Y, and Z. All three of these axes can be used to control a robot with the connection between the two different platforms using bluetooth. On wheeled robots, control command the robot to move the robot forward, backward, turn left, and turn right to do based on the slope contained in accelerometer embedded on the smartphone. There is a weakness in this control method, the robot speed constant when it is moved so there is high possibility of robots to bumping into the obstacles.

This study proposes a mobile robot control through hand movements utilize the accelerometer sensor which is embedded on the smartphone using Bluetooth. DC motor speed is controlled by using fuzzy logic to determine the speed appropriate to the circumstances so that the robot car will not bump the obstacle. DC motor speed are used as input to the method of PWM (Pulse Width Modulation) as a motor drive. The distance on the car robot as an input obtained from the ultrasonic sensors processed by the microcontroller. The result of the output of the DC motor rotation speed is changed adaptively based on the distance information that is sent by the ultrasonic sensor. The use of this method makes the movement of the robot is smoother and more precise to stop when it finds an obstacle.

In the test scenario, used some of the speed and distance of a room in order to determine the performance of the robot's movement when in control. The speed used was 160, 200 and 240 pwm. The distances used are 20, 30, 40, 50, 60 and 70 cm. Based on testing on the proposed method, the smallest error value is at a distance of 50 cm with a speed of 200 pwm. Results that obtained from ten experiments, the robot stopped automatically with the error value passes a specified distance average of 1.45 cm from a distance of 50 cm threshold. The test results show that the level of accuracy control using fuzzy logic gives better results in terms of control accuracy.

Keywords : Mobile robot, fuzzy logic, Android, accelerometer, bluetooth, pwm.

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

ABSTRAK	iii
ABSTRACT	v
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
BAB 1 PENDAHULUAN	1
1.1 Pendahuluan	1
1.2 Perumusan Masalah	3
1.3 Tujuan Penelitian	4
1.4 Manfaat Penelitian	4
1.5 Kontribusi	4
1.6 Batasan Penelitian	4
1.7 Sistematika Penulisan	5
BAB 2 KAJIAN PUSTAKA	7
2.1 <i>Mobile Robot</i>	7
2.2 Arduino	7
2.2.1 Daya (USB / <i>Barrel jack</i>)	8
2.2.2 Pin (5v, 3.3v, GND, Analog, Digital, PWM, <i>Aref</i>)	9
2.3 Sensor <i>Accelerometer</i>	11
2.4 <i>Bluetooth</i>	11
2.4.1 Cara Kerja <i>Bluetooth</i>	12
2.4.2 Kekurangan dan Kelebihan <i>Bluetooth</i>	13
2.5 Logika <i>Fuzzy</i>	13
2.5.1 Kendali logika <i>Fuzzy</i>	14
2.6 Android	16
2.7 Sensor Ultrasonik	17
BAB 3 METODOLOGI PENELITIAN	21
3.1 Tahapan Penelitian	21
3.2 Studi Literatur	21
3.3 Rancangan Penelitian	22

3.3.1	Perancangan Sistem.....	23
3.3.2	Inisialisasi Penggunaan Pin pada Motor DC.....	24
3.3.3	Inisialisasi Penggunaan Pin pada Sensor Ultrasonik.....	25
3.3.4	Inisialisasi Penggunaan Pin untuk <i>Bluetooth</i>	25
3.4	Desain Komunikasi dan Navigasi Robot.....	26
3.5	Desain Metode <i>Fuzzy</i>	29
3.6	Implementasi Penelitian.....	32
3.7	Uji Coba dan Analisis.....	33
3.7.1	Skenario dan Uji Coba.....	33
3.7.2	Ruang Uji Coba.....	34
BAB 4 HASIL DAN PEMBAHASAN.....		37
4.1	Lingkungan Uji Coba.....	37
4.2	Pengujian Robot Otomatis Pada Ruangan Bebas.....	38
4.2.1	Halangan Berada di Depan dan Samping Kanan Robot.....	38
4.2.2	Halangan Berada di Depan dan Samping Kiri Robot.....	40
4.2.3	Halangan Berada di Depan, Kiri dan Kanan Robot.....	41
4.2.4	Halangan Berada di Depan Robot, Melewati Persimpangan.....	42
4.2.5	Halangan Berada di Depan Robot Berbentuk Segi Tiga.....	44
4.3	Pengujian Kontrol Manual Menggunakan Logika <i>Fuzzy</i>	45
4.3.1	<i>Membership function</i> jarak yang digunakan.....	46
4.3.2	<i>Membership Function</i> Untuk Kecepatan.....	49
4.3.3	Pengujian Dengan Jarak 20 Cm Menggunakan Logika <i>Fuzzy</i>	50
4.3.4	Pengujian Dengan Jarak 30 Cm Menggunakan Logika <i>Fuzzy</i>	52
4.3.5	Pengujian dengan jarak 40 cm menggunakan logika <i>fuzzy</i>	52
4.3.6	Pengujian Dengan Jarak 50 Cm Menggunakan Logika <i>Fuzzy</i>	53
4.3.7	Pengujian Dengan Jarak 60 Cm Menggunakan Logika <i>Fuzzy</i>	54
4.3.8	Pengujian Dengan Jarak 70 Cm Menggunakan Logika <i>Fuzzy</i>	55
4.4	Pengujian Kontrol Manual Tanpa <i>Fuzzy</i>	56
4.4.1	Pengujian Dengan Kecepatan 160 Pwm.....	56
4.4.2	Pengujian Dengan Kecepatan 200 pwm.....	58
4.4.3	Pengujian Dengan Kecepatan 240 pwm.....	59
4.5	Analisis Hasil dan Pembahasan.....	60
BAB 5 KESIMPULAN DAN SARAN.....		63

5.1	Kesimpulan.....	63
5.2	Saran.....	63
DAFTAR PUSTAKA		65

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Arduino Mega	8
Gambar 2.2 Sensor <i>accelerometer</i>	11
Gambar 2.3 <i>Bluetooth</i> (Sumber: www.intorobotics.com)	12
Gambar 2.4 Mekanisme kendali logika <i>fuzzy</i>	14
Gambar 2.5 <i>Fuzzy set</i> untuk <i>output</i> dengan variabel <i>membership</i>	15
Gambar 2.6 Grafik perubahan <i>du</i> terhadap <i>error</i>	15
Gambar 2.7 Sensor jarak ultrasonik	17
Gambar 2.8 Instalasi sensor ultrasonik	17
Gambar 2.9 Diagram waktu sensor ultrasonik.....	18
Gambar 2.10 Jarak Ukur Sensor Ultrasonik	18
Gambar 3.1 Tahapan Metodologi Penelitian	21
Gambar 3.2 Diagram alir sistem komunikasi android dan mikrokontroler	22
Gambar 3.3 Rancangan Arsitektur robot	23
Gambar 3.4 Rangkain skematik robot.....	24
Gambar 3.5 Diagram alir data axis Y dan Z pada <i>smartphone</i>	26
Gambar 3.6 Proses penerimaan data mikrokontroler	27
Gambar 3.7 Pengolahan data yang diterima	28
Gambar 3.8 <i>Scanning</i> jarak robot.....	28
Gambar 3.9 Desain antarmuka pada <i>smartphone</i>	29
Gambar 3.10 <i>Membership function</i> untuk input jarak dari sensor ultrasonik	30
Gambar 3.11 <i>Membership function</i> untuk kecepatan.....	31
Gambar 3.12 Gerakan tangan sumbu x dan y <i>accelerometer</i>	33
Gambar 3.13 Ruang uji coba.....	34
Gambar 4.1 Robot ketika menemukan halangan disisi kanan dan di depan.....	38
Gambar 4.2 Robot bermanuver kekiri.....	39
Gambar 4.3 Robot ketika menemukan halangan di sisi kiri dan di depan.....	40
Gambar 4.4 Robot bermanuver kekanan.....	40
Gambar 4.5 Robot ketika diberikan halangan di sisi kiri, kanan dan depan.....	41
Gambar 4.6 Robot bermanuver ke belakang.....	42

Gambar 4.7 Robot melewati persimpangan	43
Gambar 4.8 Robot ketika halangan dari semua sisi.....	43
Gambar 4.9 Robot bermanuver ke kiri	44
Gambar 4.10 Robot melewati halangan di sisi depan berbentuk kerucut	45
Gambar 4.11 <i>Membership function</i> untuk jarak 20 cm	46
Gambar 4.12 <i>Membership function</i> untuk jarak 30 cm	47
Gambar 4.13 <i>Membership function</i> pada jarak 30 cm.....	47
Gambar 4.14 <i>Membership function</i> untuk jarak 50 cm	48
Gambar 4.15 <i>Membership function</i> untuk jarak 60 cm	48
Gambar 4.16 <i>Membership function</i> untuk jarak 70 cm	49
Gambar 4.17 <i>Membership function</i> pada kecepatan 160 pwm	49
Gambar 4.18 <i>Membership function</i> pada kecepatan 200 pwm	50
Gambar 4.19 <i>Membership function</i> pada kecepatan 240 pwm	50
Gambar 4.20 Grafik hasil pengujian pada jarak 20 cm	51
Gambar 4.21 Grafik hasil pengujian pada jarak 30 cm	52
Gambar 4.22 Grafik hasil pengujian pada jarak 40 cm	53
Gambar 4.23 Grafik hasil pengujian pada jarak 50 cm	54
Gambar 4.24 Grafik hasil pengujian pada jarak 60 cm	55
Gambar 4.25 Grafik hasil pengujian pada jarak 70 cm	56
Gambar 4.26 Grafik hasil pengujian pada kecepatan 160 pwm	57
Gambar 4.27 Grafik hasil pengujian dengan kecepatan 200 pwm	58
Gambar 4.28 Grafik hasil pengujian dengan kecepatan 240 pwm	56
Gambar 4.29 Hasil keseluruhan pengujian menggunakan logika <i>fuzzy</i>	60
Gambar 4.30 Hasil keseluruhan pengujian tanpa logika <i>fuzzy</i>	61

DAFTAR TABEL

Tabel 3.1 Membership function untuk input kecepatan dan jarak.....	30
Tabel 3.2 Rule evaluation untuk sistem <i>fuzzy</i> (Martin 2009).....	31
Tabel 4.1 Spesifikasi perangkat keras yang digunakan	37
Tabel 4.2 Spesifikasi perangkat lunak yang digunakan	38
Tabel 4.3 Hasil Pembacaan Sensor untuk Halangan di Depan dan Kanan	39
Tabel 4.4 Hasil Pembacaan Sensor untuk Halangan di Depan dan Kiri.....	41
Tabel 4.5 Tabel hasil percobaan robot bermanuver kebelakang.....	42
Tabel 4.6 Percobaan Robot bermanuver ke kiri dan kanan	44
Tabel 4.7 Hasil Pengujian pada jarak 20 cm.....	51
Tabel 4.8 Hasil Pengujian pada jarak 30 cm.....	52
Tabel 4.9 Hasil pengujian menggunakan <i>fuzzy</i> pada jarak 40 cm.....	53
Tabel 4.10 Hasil pengujian menggunakan <i>fuzzy</i> pada jarak 50 cm.....	53
Tabel 4.11 Hasil pengujian menggunakan <i>fuzzy</i> pada jarak 60 cm.....	55
Tabel 4.12 Hasil pengujian menggunakan <i>fuzzy</i> pada jarak 70 cm.....	55
Tabel 4.13 Hasil Pengujian tanpa <i>fuzzy</i> dengan kecepatan 160 pwm.....	57
Tabel 4.14 Hasil Pengujian dengan Kecepatan 200 pwm.....	58
Tabel 4.15 Hasil Pengujian Dengan Kecepatan 240 pwm.....	59

[Halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

1.1 Pendahuluan

Dewasa ini perkembangan teknologi *wireless* seperti teknologi *Bluetooth*, pada ponsel sudah berkembang sangat pesat dan telah banyak di implementasikan di berbagai bidang seperti menggunakan *bluetooth* sebagai penghubung atau perantara antara dua perangkat agar dapat saling berkomunikasi atau melakukan pertukaran data. Dalam pengembangannya komunikasi antara perangkat *Bluetooth* dikembangkan pada dunia robotika (Indar Sugiarto dkk, 2008), pada sebuah *mobile robot* biasanya dikendalikan dengan menggunakan sebuah *remote control* yang terhubung dengan sebuah robot melalui media kabel atau melalui media tanpa kabel, namun *remote control* tersebut masih berupa *remote control* konvensional atau bahkan *remote control* khusus yang dibuat untuk robot itu sendiri.

Banyak penelitian yang telah dilakukan salah satunya dilakukan ada tiga item hasil yang diperoleh dari pengontrolan robot menggunakan gerakan tangan dibandingkan pengontrolan menggunakan PC. Dimana peneliti menyajikan desain sistem pengendalian penggerak robot rumah (Wu dkk, 2013). Mengusulkan kontrol robot berbasis gerakan tangan dengan memanfaatkan sensor *accelerometer* yang terdapat pada *smartphone* dimana *bluetooth* sebagai media penghubung antara robot dan *smartphone* bagi pengguna untuk mengontrol robot dengan umpan balik enam gerakan tangan didefinisikan untuk navigasi robot. Sumbu 3-axis *accelerometer* digunakan untuk mendeteksi gerakan tangan pengguna. Hasil percobaan navigasi menunjukkan bahwa tingkat keberhasilan mengontrol melalui gerakan tangan 13,33% lebih efektif dibandingkan mengontrol menggunakan PC. Ketepatan kontrol tangan 15,4% dibandingkan menggunakan PC dan waktu pelaksanaan 24.7% lebih cepat. Ini berarti bahwa kontrol menggunakan gerakan tangan lebih efisien dan fleksibel dibandingkan kontrol robot lewat PC.

Namun pada penelitian ini masih terdapat kelemahan karena kecepatan gerak robot masih bersifat konstan ketika dikontrol. Apabila robot berjalan dengan kecepatan penuh maka robot akan berhenti secara mendadak ketika menemukan

halangan sehingga terjadinya dorongan yang kuat dari motor jadi kemungkinan terjadinya pergeseran robot berhenti dari tempat yang ditentukan akan sangatlah besar. Oleh karena itu perlunya sebuah sistem pengontrol yang bisa mengatur kecepatan gerak robot secara adaptif agar robot bisa mengambil keputusan secara otomatis mengurangi kecepatan berdasarkan informasi jarak yang yang diterima dari sensor ultrasonik. Sehingga kemungkinan menabrak halangan yang ada di depan sangat kecil.

Penelitian berikutnya menjelaskan penerapan logika *fuzzy* untuk memecahkan masalah dari *mobile* robot otonom melakukan pelacakan jalan dalam sebuah labirin (Indar Sugiarto dkk. 2008). Robot dilengkapi dengan *transceiver* ultrasonik untuk mengukur jarak antara robot dan dinding. Informasi dari jarak akan diproses menggunakan logika *fuzzy* yang ditanam dalam dua *chip* FPGA (*Field-Programmable Gate Array*). FPGA memiliki tanggung jawab untuk memformulasikan aturan *fuzzy* dan menghasilkan sinyal PWM untuk motor robot sebagai hasil dari inferensi *fuzzy*. Melaksanakan semua algoritma logika *fuzzy* yang diperlukan akan membutuhkan banyak sumber daya FPGA. Oleh karena itu, menggunakan dua FPGA: XC4010E dan XC4005 di dalamnya. Yang pertama untuk *fuzzification* dan evaluasi aturan dasar, dan mengkonsumsi 98% dari sumber daya (CLBs). Yang kedua untuk defuzzifikasi dan PWM generasi *output*, yang memanfaatkan 78% dari semua CLBs. Dengan menerapkan logika *fuzzy* menggunakan FPGA, robot mencapai pelacakan relatif aman jalan secara *real-time*. Namun pada penelitian ini masih terdapat kelemahan bilamana robot melewati persimpangan atau jalan buntu. Respon robot tidak begitu baik saat bertemu persimpangan dengan belokan tajam model T, dimana robot harus segera memutuskan apakah belok ke kanan atau ke kiri, kelemahan kedua yang ditemukan adalah apabila robot bertemu dengan jalan buntu dalam hal ini belum diberikan alternatif solusi yang memuaskan atas permasalahan kedua.

Penelitian berikutnya mengusulkan sebuah robot *mobile* yang dapat digunakan untuk mendeteksi bom yang ada di bawah mobil (Abdulsada, 2015) dengan menggunakan algoritma logika *fuzzy* sebagai pengontrol gerak robot untuk maju, mundur, belok kiri dan belok kanan menggunakan algoritma ini adalah pilihan yang lebih baik untuk beberapa alasan seperti kelancaran transisi dalam

gerakan robot selama navigasi. Menggunakan algoritma *subsumption* membantu untuk menyeimbangkan gerakan empat roda kendaraan sebagai penggerak robot, membuat robot dapat menyelesaikan dan mencari ke arah horizontal setelah menyelesaikan vertikal arah atau sebaliknya (ketika robot mulai *scan* dari lebar kendaraan daripada di arah vertikal). Semua fitur digabungkan dengan OpenCV agar akurasi dalam pengontrolan lebih baik.

Dari permasalahan yang telah dipaparkan maka penelitian ini berfokus pada membangun sebuah robot yang bisa dikontrol secara manual menggunakan gerakan tangan dengan memanfaatkan sensor *accelerometer* yang terdapat pada *smartphone* dan bisa berjalan secara otomatis dengan menggunakan logika *fuzzy* sebagai kelancaran transisi gerakan robot berdasarkan data jarak yang dikirim oleh tiga buah sensor ultrasonik yang dipasang pada sisi kiri, sisi kanan dan sisi depan pada robot supaya robot bisa mengambil keputusan secara cepat dan tepat ketika berhadapan persimpangan dan jalan yang buntu apakah harus bergerak cepat, pelan, sedang atau belok kiri atau kekanan tergantung medan yang dilewati oleh robot. Dari metode yang diusulkan diharapkan dapat memperhalus gerakan robot ketika berjalan sehingga dapat meningkatkan akurasi dalam mengontrol robot menggunakan gerakan tangan.

Pada beberapa penelitian terkait dapat memberikan referensi untuk merancang sistem navigasi robot pada penelitian ini. Untuk lebih lanjut mengenai penelitian ini akan dibahas pada paparan metodologi di bawah ini.

1.2 Perumusan Masalah

Berdasarkan uraian yang telah dijelaskan pada latar belakang, maka permasalahan yang akan diselesaikan dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang dan mengimplementasikan mekanisme kontrol robot menggunakan sensor *accelerometer* pada *smartphone* dengan menerapkan logika *fuzzy*.
2. Bagaimana melakukan integrasi antara perangkat robot dan *smartphone* dari segi ketepatan pergerakan tangan.
3. Bagaimana akurasi robot berhenti ketika menemukan halangan.

4. Bagaimana mengatur kecepatan motor berdasarkan informasi jarak yang di terima dari sensor ultrasonik.
5. Bagaimana merancang sistem kendali otomatis dan manual pada robot.

1.3 Tujuan Penelitian

Berdasarkan perumusan masalah yang telah diuraikan, maka muncul beberapa tujuan yang ingin dicapai. Adapun tujuan dalam penelitian ini antara lain:

1. Membangun perangkat robot yang dapat dikontrol melalui *smartphone* dengan menggunakan logika *fuzzy*.
2. Menerapkan algoritma *fuzzy* untuk ketepatan kontrol dari pergerakan tangan.
3. Perangkat *smartphone* dan perangkat robot yang dihubungkan menggunakan media komunikasi *bluetooth*.
4. Menerapkan sistem kontrol otomatis dan manual pada robot.

1.4 Manfaat Penelitian

Penelitian ini memberikan gambaran manfaat dalam melakukan kontrol antara *smartphone* dan perangkat robot dengan menggunakan logika *fuzzy* jika dibandingkan tanpa menggunakan metode logika *fuzzy*.

1.5 Kontribusi

Pada penelitian ini mengusulkan sebuah pengontrolan *mobile* robot melalui gerakan tangan memanfaatkan sensor *accelerometer* yang terdapat pada *smartphone* dengan media komunikasi *bluetooth* menggunakan pengatur kecepatan motor DC yang dikontrol dengan menggunakan logika *fuzzy*. Penggerak motor menggunakan metode PWM (*Pulse Width Modulation*). *Input* jarak yang diterima dari sensor ultrasonik akan diproses oleh mikrokontroler. Hasil keluaran berupa kecepatan putaran motor DC yang berubah secara adaptif berdasarkan informasi jarak yang dikirim oleh sensor ultrasonik sehingga pergerakan robot lebih halus dan lebih persisi pada saat berhenti ketika menemukan halangan.

1.6 Batasan Penelitian

Batasan dalam penelitian ini adalah sebagai berikut:

1. Penelitian dilakukan pada perangkat *smartphone* dengan memanfaatkan sensor *accelerometer* sebagai pengontrol robot.
2. Implementasi menggunakan *bluetooth* sebagai media komunikasi antara *smartphone* dengan robot.
3. Implementasi dilakukan pada perangkat Arduino mega sebagai otak dari robot.
4. Implementasi dilakukan pada ruangan *indoor*.

1.7 Sistematika Penulisan

Sistematika penulisan proposal tesis ini adalah:

1. **BAB I PENDAHULUAN.** Bab ini berisi pendahuluan yang menjelaskan latar belakang, perumusan masalah, batasan masalah, tujuan dan manfaat, kontribusi penelitian dan sistematika penulisan terkait dengan penelitian.
2. **BAB II KAJIAN PUSTAKA.** Bab ini berisi dasar-dasar teori dan kajian pustaka terkait penelitian, di antaranya, *mobile* robot, arduino, sensor *accelerometer*, logika *fuzzy*, Android, sensor ultrasonik.
3. **BAB III METODOLOGI PENELITIAN.** Bab ini berisi tahapan penelitian meliputi studi literatur, rancangan penelitian dan skenario pengujian.
4. **BAB IV UJI COBA DAN HASIL ANALISIS.** Bab ini berisi tahapan skenario uji coba metode usulan dan analisa hasil.
5. **BAB V KESIMPULAN DAN SARAN.** Bab ini berisi tahapan menjelaskan tentang kesimpulan dan saran dari hasil penelitian.

[Halaman ini sengaja dikosongkan]

BAB 2

KAJIAN PUSTAKA

Pada bagian ini akan dijelaskan konsep dasar tentang teori yang digunakan sebagai dasar dalam melakukan penelitian.

2.1 Mobile Robot

Robot mobil (*mobile robot*) adalah sebuah mesin otomatis yang mampu bergerak pada suatu kondisi tertentu. Robot *mobile* di klasifikasikan menjadi dua, yaitu menurut lingkungan tempat robot tersebut bekerja dan alat navigasi yang digunakan untuk bergerak. Berdasarkan lingkungan tempat robot tersebut bekerja, robot *mobile* terbagi menjadi empat macam: robot yang bekerja di atas permukaan tanah (*land robot*), robot udara yang biasa disebut *unmanned aerial vehicle* (UAV). (Krofitsch et al. 2013) *autonomous underwater vehicles* (AUVs), dan robot yang bekerja pada lingkungan kutub robot yang bekerja pada kondisi permukaan tanah yang dilapisi es (*polar robots*). Sedangkan berdasarkan alat yang digunakan untuk bergerak, robot mobil terbagi menjadi robot berlengan atau berkaki atau kaki menyerupai manusia ataupun hewan dan robot beroda *wheeled mobile robot* (WMR) (Wikipedia, Mobile Robot, 2016).

Robot *mobile* tersebut merupakan Wahana Nir Awak (WaNA) yang telah menjadi sarana yang sering digunakan oleh pihak militer maupun pihak sipil untuk melakukan pengintaian, penjelajahan, dan pengawasan ke tempat-tempat yang berbahaya bagi manusia. Kemampuannya yang dapat dikontrol dari jauh atau bahkan dirancang agar dapat bergerak sendiri sesuai dengan lintasan tertentu akan sangat menguntungkan bagi manusia. Beberapa keuntungan yang dapat diperoleh antara lain meminimalisasi resiko, meminimalisasi objektif, dan lain sebagainya.

2.2 Arduino

Arduino adalah pengendali mikro *single-board* yang bersifat *open source* dirancang untuk memudahkan pengembang dalam membuat proyek-proyek elektronika diberbagai bidang. Arduino terdiri dari dua bagian utama yaitu sebuah papan sirkuit fisik (sering disebut juga dengan mikrokontroler) dan sebuah perangkat

lunak atau IDE (*Integrated Development Environment*) yang berjalan pada komputer. Mikrokontroler adalah komputer kecil pada sebuah chip atau IC (*integrated circuit*) yang berisi prosesor inti, memori, dan *input* atau *output* yang dapat diprogram. Program yang ditanamkan pada mikrokontroler inilah yang akan memerintahkan *hardware* untuk melakukan proses tertentu seperti membaca *input*, memproses data *input* dan menampilkan *output* sesuai kebutuhan. Arduino IDE adalah *software* yang digunakan untuk menulis dan meng-*upload* kode dari komputer ke papan fisik (*hardware*) Arduino (Chauhan et al. 2013).

Platform Arduino sekarang ini menjadi sangat populer dengan pertambahan jumlah pengguna baru yang terus meningkat. Hal ini karena kemudahannya dalam penggunaan dan penulisan kode. Tidak seperti kebanyakan papan sirkuit pemrograman sebelumnya, Arduino tidak lagi membutuhkan perangkat keras terpisah (disebut *programmer* atau *downloader*) untuk memuat atau meng-*upload* kode baru ke dalam mikrokontroler. Cukup dengan menggunakan kabel USB untuk mulai menggunakan Arduino.

Ada banyak jenis papan Arduino yang dapat digunakan untuk tujuan yang berbeda. Beberapa papan memiliki ukuran, jumlah pin dan mikrokontroler yang berbeda seperti pada Gambar 2.1. Namun sebagian besar Arduino memiliki komponen utama salah satu Arduino yang umum digunakan yaitu Arduino MEGA.



Gambar 2.1 Arduino Mega

2.2.1 Daya (USB / Barrel jack)

Setiap papan Arduino membutuhkan jalur untuk terhubung ke sumber listrik. Arduino Mega dapat diaktifkan melalui kabel USB yang berasal dari komputer atau

power supply terpisah yang dihubungkan ke *Barrel Jack*. Pada Gambar 2.1 koneksi USB diberi label (1) dan *Barrel Jack* diberi label (2).

2.2.2 Pin (5v, 3.3v, GND, Analog, Digital, PWM, Aref)

Pin pada Arduino adalah tempat menghubungkan kabel untuk membuat suatu rangkaian (menghubungkan satu titik dengan titik lainnya pada *breadboard* dengan sejumlah kabel penghubung) (Gibb, 2010). Arduino memiliki beberapa jenis pin yang masing-masing diberi label pada *board* Arduino dan digunakan untuk fungsi yang berbeda-beda.

GND (3): GND merupakan singkatan dari GROUND. Ada beberapa pin GND pada Arduino, dan semuanya dapat digunakan untuk hubungan ke *ground*. 5V (4) dan 3.3V (5): Pin 5V digunakan untuk menyediakan tegangan sebesar 5 Volt. Sedangkan pin 3.3V digunakan untuk menyediakan tegangan 3,3 Volt. Sebagian besar komponen sederhana yang digunakan bersama dengan Arduino berjalan pada kisaran tegangan 3,3 Volt hingga 5 Volt.

1. Analog (6): Pin yang berada pada area berlabel “ANALOG IN” (A0 sampai A5 pada Arduino Uno) digunakan sebagai pin analog. Yaitu pin yang digunakan untuk membaca sinyal-sinyal analog dari sensor-sensor analog (misal; sensor suhu) dan mengubahnya menjadi nilai digital yang dapat kita baca.
2. Digital (7): Pin digital berada di seberang pin analog (0 sampai 13 pada Arduino). Pin ini dapat digunakan dalam 2 arah digital yaitu *input* digital (misal untuk melihat kondisi bahwa tombol sedang ditekan) dan *output* digital (memberikan tegangan sebuah LED).
3. PWM (8): Perhatikanlah simbol *tilde* (~) terdapat beberapa pin digital (yaitu pada pin 3, 5, 6, 9, 10 dan 11 pada Arduino Uno). Pin ini dapat digunakan sebagai pin digital biasa, tetapi juga dapat digunakan sebagai pin PWM (*Pulse Width Modulation*). PWM biasanya digunakan sebagai pin yang mampu mensimulasikan *output* analog (seperti mengatur pemudaran cahaya dan warna LED saat datang dan pergi).
4. AREF (9): Kepanjangan dari *Analog Reference* atau Referensi Analog. Pin ini terkadang digunakan sebagai referensi dalam mengatur tegangan eksternal

(antara 0 sampai 5 Volt) untuk memberikan limit (batasan akhir) pada *input* pin analog.

5. Tombol Reset

Menekan tombol Reset (10) beberapa saat pada papan Arduino akan menghubungkan pin reset ke *Ground*. Tombol ini digunakan untuk me-*restart* kode yang telah dimuat pada Arduino. Tombol ini sangat bermanfaat untuk menguji ulang kode, jika kode tersebut ditulis tanpa pengulangan (*repeat*) atau sekali jalan.

LED Indikator daya tepat di bawah pada sebelah kanan tulisan “UNO” pada papan Arduino Uno, terdapat sebuah LED kecil berlabel “ON” (11). LED ini harus menyala setiap kali menghubungkan papan Arduino pada sumber listrik. Jika lampu LED ini tidak menyala, kemungkinan ada rangkaian yang mengalami kerusakan atau terjadi kesalahan.

TX-RX LEDTX adalah singkatan dari ‘*Transmit*’ (kirim), dan RX adalah singkatan dari ‘*Receive*’ (terima). LED indikator TX dan RX (12) ini akan berkedip redup atau terang dengan jeda tidak tentu untuk memberitahukan bahwa telah terjadi komunikasi pada serial. Kedipan LED ini sebagai indikasi visual yang merupakan pertanda bahwa telah terjadi pengiriman dan penerimaan data pada papan Arduino (misal ketika kita meng-*upload* kode baru ke dalam papan Arduino).

6. IC Utama

Komponen yang bentuknya persegi panjang dan berwarna hitam dengan semua kakinya terbuat dari logam, inilah yang disebut IC atau *Integrated Circuit* (13). IC adalah otak dari papan Arduino. IC utama Arduino berbeda-beda sesuai dengan jenis papan, tetapi biasanya memiliki ciri khas bertuliskan ATmega yang merupakan IC buatan perusahaan ATMEL. Hal ini merupakan sangat penting untuk memahami jenis IC (sesuai juga dengan jenis papan Arduino) sebelum memuat atau meng-*upload* sekumpulan kode program yang telah dibuat dari perangkat lunak Arduino IDE kedalam papan Arduino. Informasi tentang tipe IC dapat ditemukan pada permukaan IC tersebut.

7. Regulator Tegangan

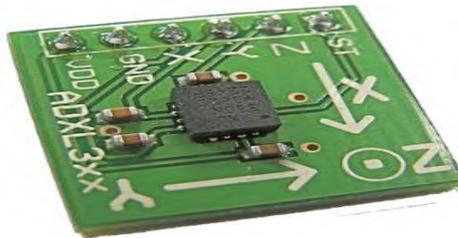
Voltage Regulator (14) bukanlah komponen yang mampu berinteraksi dengan papan Arduino. *Voltage Regulator* atau regulator tegangan akan mengalirkan sejumlah tegangan teregulasi ke dalam papan Arduino. *Voltage Regulator* berperan

sebagai penjaga pintu gerbang dan akan membalikkan atau membuang tegangan berlebihan yang bisa membahayakan rangkaian. Batas tegangan yang masuk ke dalam papan tidak lebih besar dari 20 volt.

2.3 Sensor Accelerometer

Accelerometer modern tidak lain adalah MEMS (*micro electro mechanical system*) berskala kecil. *Accelerometer* adalah suatu alat untuk mengukur percepatan, mendeteksi dan mengukur getaran, atau untuk mengukur percepatan gravitasi (inklinasi). Pengukurannya bisa secara analog maupun digital. *Accelerometer* dapat digunakan untuk mengukur percepatan baik statis maupun dinamis. *Accelerometer* akan mengalami percepatan dalam kisaran dari -1g sampai +1g (9.8m/s), dan hingga kemiringan 180°.

Accelerometer merupakan komponen elektronik yang mengukur kemiringan dan gerak (Krofitsch dkk, 2013). *Accelerometer* juga mampu untuk membaca gerakan rotasi dan gerakan seperti berayun atau bergetar. Pada *smartphone accelerometer* merupakan sensor yang bisa membaca pergerakan sehingga dapat mengubah tampilan layar dari posisi *landscape* ke *portrait* atau sebaliknya dengan cukup memiringkan badan ponsel.



Gambar 2.2 Sensor *accelerometer*

Sumber: <http://www.electronics-lab.com/blog/?p=24>

2.4 Bluetooth

Bluetooth adalah suatu peralatan media komunikasi yang dapat digunakan untuk menghubungkan sebuah perangkat komunikasi dengan perangkat komunikasi lainnya, *bluetooth* umumnya digunakan di *handphone*, komputer atau pc, tablet, dan lain-lain. Fungsi *bluetooth* yaitu untuk mempermudah berbagi atau *sharing file*,

audio, menggantikan penggunaan kabel dan lain-lain. Saat ini sudah banyak sekali perangkat yang menggunakan *bluetooth*.

Definisi *bluetooth* (Lee et al. 2014) yang lainnya adalah sebuah teknologi komunikasi *wireless* atau tanpa kabel yang beroperasi dalam pita frekuensi 2,4 GHz (antara 2.402 GHz s/d 2.480 GHz) dengan menggunakan sebuah *frequency hop ultrasonik tranceiver* yang mampu menyediakan layanan komunikasi data dan juga suara secara *real-time* antara *host-host bluetooth* dengan jarak jangkauan layanan yang terbatas.

Pada dasarnya teknologi *bluetooth* ini diciptakan bukan hanya untuk menggantikan atau menghilangkan penggunaan media kabel dalam melakukan pertukaran data atau informasi, tetapi juga mampu menawarkan fitur yang bagus atau baik untuk teknologi *mobile wireless* atau tanpa kabel, dengan biaya yang relatif rendah, konsumsi daya rendah, *interoperability* yang sangat menjanjikan, mudah dalam pengoperasiannya dan juga mampu menyediakan berbagai macam layanan.

2.4.1 Cara Kerja *Bluetooth*

Sistem *bluetooth* terdiri atas sebuah radio *transceiver*, *baseband link Management dan Control*, *Baseband (processor core, SRAM, UART, PCM USB Interface)*, *flash* dan *voice codec*.

1. *Baseband link controller* menghubungkan *hardware* atau perangkat keras radio ke *baseband processing* dan juga layer protokol fisik.
2. *Link manager* melakukan aktivitas protokol tingkat tinggi, yaitu seperti melakukan *link setup*, autentikasi dan juga konfigurasi.



Gambar 2.3 *Bluetooth* (Sumber: www.intorobotics.com)

2.4.2 Kekurangan dan Kelebihan *Bluetooth*

Kelebihan:

1. Bisa menembus rintangan, misalnya seperti dinding, kotak, dan sebagainya walaupun jarak transmisinya hanya 10 M.
2. Tidak memerlukan media kabel ataupun kawat.
3. Dapat mensinkronisasi data dari *handphone* ke komputer atau laptop.
4. Dapat dipakai sebagai perantara modem.
5. Praktis dan tidak ribet dalam penggunaannya.

Kekurangan:

1. Memakai frekuensi yang sama dengan gelombang WiFi.
2. Apabila terlalu banyak koneksi *bluetooth* di dalam satu ruangan, akan sulit untuk menemukan penerima yang dituju.
3. Sering beredar virus-virus yang disebarkan melalui *bluetooth*, khususnya dari *handphone*.
4. Cukup banyak mekanisme keamanan yang harus diperhatikan untuk mencegah kegagalan pengiriman data atau penerimaan data maupun informasi.
5. Kecepatan dalam transfer data tidak tetap, tergantung dari perangkat yang dipakai untuk mengirim dan yang menerima data maupun informasi.

2.5 Logika *Fuzzy*

Himpunan *fuzzy* merupakan salah satu cara matematis yang dapat digunakan untuk mempresentasikan ketidaktepatan atau nilai samar, (Lotfi A. Zadeh, 1965). *Fuzzy* adalah istilah yang dipakai oleh untuk menyatakan kelompok atau himpunan yang dapat dibedakan dengan himpunan lain berdasarkan derajat keanggotaan dengan batasan yang tidak begitu jelas (samar), tidak seperti himpunan klasik yang membedakan keanggotaan himpunan menjadi dua, himpunan anggota atau bukan anggota. Kendali logika *fuzzy* dilakukan dalam tiga tahap, yaitu fuzzifikasi, evaluasi aturan dan defuzzifikasi. Komponen fuzzifikasi berfungsi untuk memetakan

masukan data tegas ke dalam himpunan *fuzzy* menjadi nilai *fuzzy* dari beberapa variabel linguistik masukan.



Gambar 2.4 Mekanisme kendali logika *fuzzy*

Basis pengetahuan berisi pengetahuan sistem kendali sebagai pedoman evaluasi keadaan sistem untuk mendapatkan keluaran sesuai yang diinginkan perancang. Basis pengetahuan terdiri dari basis data dan basis aturan *fuzzy*.

1. Basis data

Basis data merupakan komponen untuk mendefinisikan himpunan *fuzzy* dari masukan dan keluaran.

2. Basis aturan *fuzzy*

Basis Aturan *fuzzy* merupakan kumpulan pernyataan aturan 'IF-THEN' yang didasarkan kepada pengetahuan pakar. Metode yang digunakan dalam penentuan basis aturan *fuzzy* adalah menggunakan metode *trial and error*. Logika pengambilan keputusan disusun dengan cara menuliskan aturan yang menghubungkan antara masukan dan keluaran sistem *fuzzy*. Aturan ini diekspresikan dalam kalimat: 'jika <masukan> maka <keluaran>'. Metode ini mempunyai bentuk aturan seperti Persamaan 2.1.

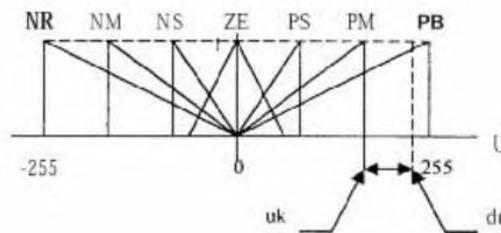
$$\text{IF } x \text{ is } A \text{ and } y \text{ is } B \text{ then } z = k \quad (2.1)$$

2.5.1 Kendali logika *Fuzzy*

Metode Mamdani merupakan teknik inferensi *fuzzy* yang dapat digunakan untuk pengambilan keputusan pada logika kontroler. Pertama kali diusulkan oleh Mamdani dan Assilian sebagai upaya untuk mengendalikan kontrol sintesis mesin uap dan kombinasi *boiler* (Mamdani dkk, 1975). Ada berbagai jenis metode *adaptive fuzzy logic*, pada penelitian ini digunakan metode *tuning output*, dimana metode

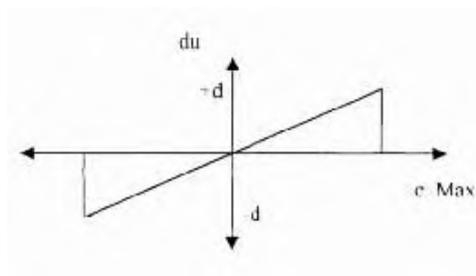
tuning output yang digunakan merupakan modifikasi dari metode *tuning membership function* pada variabel *output*. Pada metode *tuning membership function* pada variabel *output*, apabila sejumlah *fuzzy set* pada *input* kesalahan (*error*) dan selisih kesalahan (Δ *error*) didefinisikan dengan *membership function* bentuk segitiga tertentu, maka jumlah *fuzzy set* yang sama akan didefinisikan untuk *variable output* ($U(t)$), tetapi *membership function* dari variabel *output* tersebut dapat disesuaikan dalam semesta pembicaraan *output*. Untuk *membership function* $\mu_k(\mathbf{u})$ dari *fuzzy set* k dinyatakan dengan bentuk garis lurus, untuk itu persamaan pengaturan dapat diberikan sebagai berikut:

$$\mu_k(\mathbf{u}) = \frac{1}{uk + du} u \quad (2.2)$$



Gambar 2.5 *Fuzzy set* untuk *output* dengan variabel *membership*

Disini $\mu_k(\mathbf{u}) = 1$ sebelum dilakukan penyesuaian terhadap *membership function*. *Membership function* dari *fuzzy set* dinyatakan dalam semesta pembicaraan *output* dapat dengan mudah di-*tuning* dengan mengubah nilai du , salah satunya dengan mengamati perubahan kesalahan (*error*), dimana perubahan nilai du sebanding dengan besarnya perubahan nilai *error*.



Gambar 2.6 Grafik perubahan du terhadap *error*

Perubahan nilai du akan meningkatkan kontribusi dari tiap *rule* untuk *error* yang besar atau memperkecil *output* kontrol dengan mengurangi kontribusi dari tiap

rule untuk *error* yang kecil. Pada proses adaptif dengan metode *tuning output*, tidak dilakukan penggeseran *membership function output* yang digeser lebih besar atau lebih kecil dibanding dengan *error* yang terjadi untuk proses adaptifnya, tetapi nilai *output*-nya yang digeser dengan mengamati *error* yang terjadi, sehingga seolah-olah *membership function output*-nya juga tergeser (Christianto dkk, 2011).

2.6 Android

Android merupakan sistem operasi yang dikembangkan untuk perangkat *mobile* berbasis Linux. Pada awalnya sistem operasi ini dikembangkan oleh Android *Inc*, yang kemudian dibeli oleh Google pada tahun 2005. Android merupakan OS *mobile* yang tumbuh ditengah OS lainnya yang berkembang dewasa ini, (R Meier, 2012) OS lainnya seperti *Windows Mobile*, *iPhone iOS*, *Symbian* dan masih banyak lagi yang juga menawarkan kekayaan isi dan keoptimalan berjalan di atas perangkat *hardware* yang ada. Android juga dilengkapi dengan *Software Development Kit* (SDK) yang lengkap, emulator yang membantu untuk menguji coba aplikasi yang dibuat serta dokumentasi yang lengkap serta tidak membutuhkan lisensi dan biaya untuk memperolehnya. Maka Android merupakan pilihan yang tepat untuk pengembang.

Berikut ini merupakan beberapa versi peluncuran Android pada perkiraan tahun 2009 - 2013 yaitu:

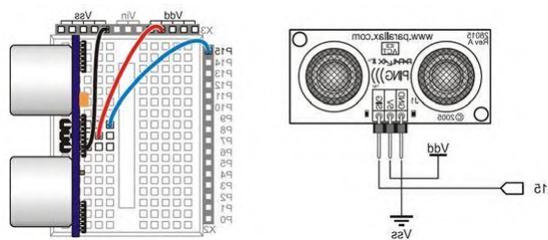
1. Android 1.5 Cupcake (Mei 2009)
2. Android 1.6 Donut (September 2009)
3. Android 2.0 - 2.1 Eclair (Desember 2009)
4. Android 2.2 - 2.2.3 Froyo (Mei 2010)
5. Android 2.3 - 2.4 Gingerbread (Desember 2010)
6. Android 3.0 - 3.2 Honeycomb (Februari 2011)
7. Android 4.0 Ice Cream Sandwich (Oktober 2011)
8. Android 4.1 - 4.2 Jelly Bean (Juli 2012)
9. Android 4.4.2 Kitkat (Oktober 2013).
10. Android 5.0 Lollipop
11. Android 6.0 Marshmallow

2.7 Sensor Ultrasonik

Sensor ultrasonik adalah sensor 40 kHz produksi *Parallax* yang banyak digunakan untuk aplikasi atau kontes robot cerdas. Kelebihan sensor ini adalah hanya membutuhkan 1 sinyal (SIG) selain jalur 5 V dan *ground* seperti pada Gambar 2.7.



Gambar 2.7 Sensor jarak ultrasonik



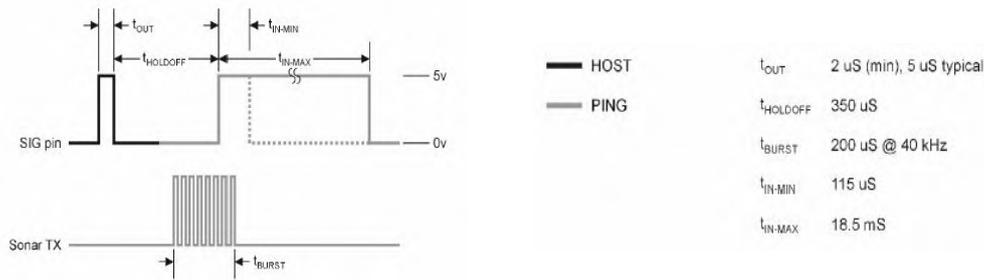
Gambar 2.8 Instalasi sensor ultrasonik

Sumber: www.parallax.com

Sensor ultrasonik mendeteksi jarak objek dengan cara memancarkan gelombang ultrasonik (40 KHz) selama $t = 200 \mu\text{s}$ kemudian mendeteksi pantulannya. Sensor ultrasonik memancarkan gelombang ultrasonik sesuai dengan kontrol dari mikrokontroler pengendali (pulsa *trigger* dengan *tout* min 2 μs).

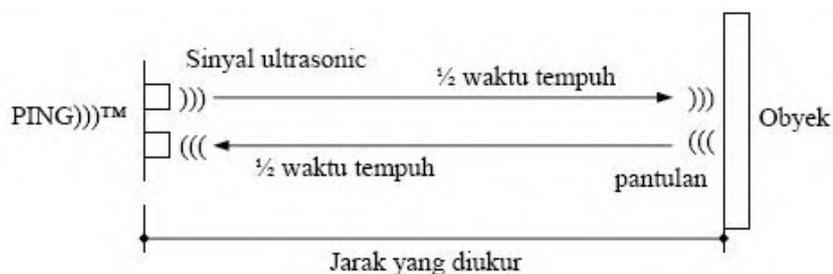
Spesifikasi sensor ultrasonik:

- Kisaran pengukuran 3cm-3m.
- Input trigger* –positive TTL pulse, 2 μs min., 5 μs tipikal.
- Echo hold off* 750 μs dari *fall of trigger pulse*.
- Delay before next measurement* 200 μs .
- Burst indicator* LED menampilkan aktifitas sensor.



Gambar 2.9 Diagram waktu sensor ultrasonik

Sensor ultrasonik mendeteksi jarak objek dengan cara memancarkan gelombang ultrasonik (40 kHz) selama t_{BURST} (200 μ s) kemudian mendeteksi pantulannya. Sensor ultrasonik memancarkan gelombang ultrasonik sesuai dengan kontrol dari mikrokontroler (Lim et al. 2014) pengendali (pulsa *trigger* dengan t_{OUT} min. 2 μ s). Gelombang ultrasonik ini melalui udara dengan kecepatan 344 meter per detik, mengenai objek dan memantul kembali ke sensor. Sensor ultrasonik adalah salah satu sensor yang paling akurat diantara beberapa sensor yang digunakan untuk mengukur jarak. Sensor ini mampu mendeteksi adanya penghalang dan mengirim *output* yang berupa jarak ke modul *display* sehingga dapat diketahui jarak antara penghalang dengan mobil agar benturan dapat dihindari (Susanto et al. 2007). Ultrasonik mengeluarkan pulsa *output high* pada pin SIG memancarkan gelombang ultrasonik dan setelah gelombang pantulan terdeteksi ultrasonik akan membuat *output low* pada pin SIG. Lebar pulsa *high* (t_{IN}) akan sesuai dengan lama waktu tempuh gelombang ultrasonik untuk 2x jarak ukur dengan objek. Maka jarak yang diukur adalah $[(t_{IN} \text{ s} \times 344 \text{ m/s}) \div 2]$ meter.



Gambar 2.10 Jarak ukur sensor ultrasonik

Sistem minimal mikrokontroler ATmega 8535 dan *software basic stamp editor* diperlukan untuk memprogram mikrokontroler dan mencoba sensor ini. Keluaran

dari pin SIG ini yang dihubungkan ke salah satu port di kit mikrokontroler. Contoh aplikasi sensor ultrasonik pada mikrokontroler BS2, dimana pin SIG terhubung ke pin 7, dan memberikan catu daya 5V dan *ground*. Fungsi *Sigout* untuk men-*trigger* ultrasonik, sedangkan fungsi *Sigin* digunakan untuk mengukur pulsa yang sesuai dengan jarak dari objek target (Lee et al. 2014).

[Halaman sengaja dikosongkan]

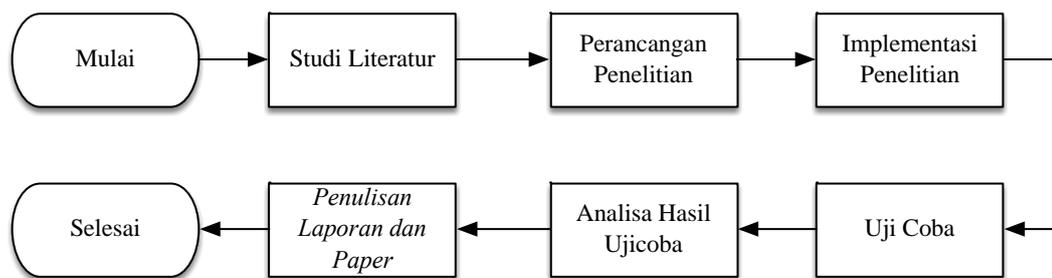
BAB 3

METODOLOGI PENELITIAN

Bab ini akan memaparkan tentang metodologi penelitian yang akan digunakan, terdiri dari studi literatur, perancangan metode dan algoritma, implementasi metode, pengujian metode dan analisa hasil ujicoba, penulisan laporan dan makalah. Selanjutnya jadwal kegiatan penelitian memuat garis waktu dari semua langkah penelitian.

3.1 Tahapan Penelitian

Berikut ini merupakan tahapan yang akan dilakukan dalam pengerjaan penelitian ini, Gambar 3.1 adalah diagram metodologi penelitian.



Gambar 3.1 Tahapan Metodologi Penelitian

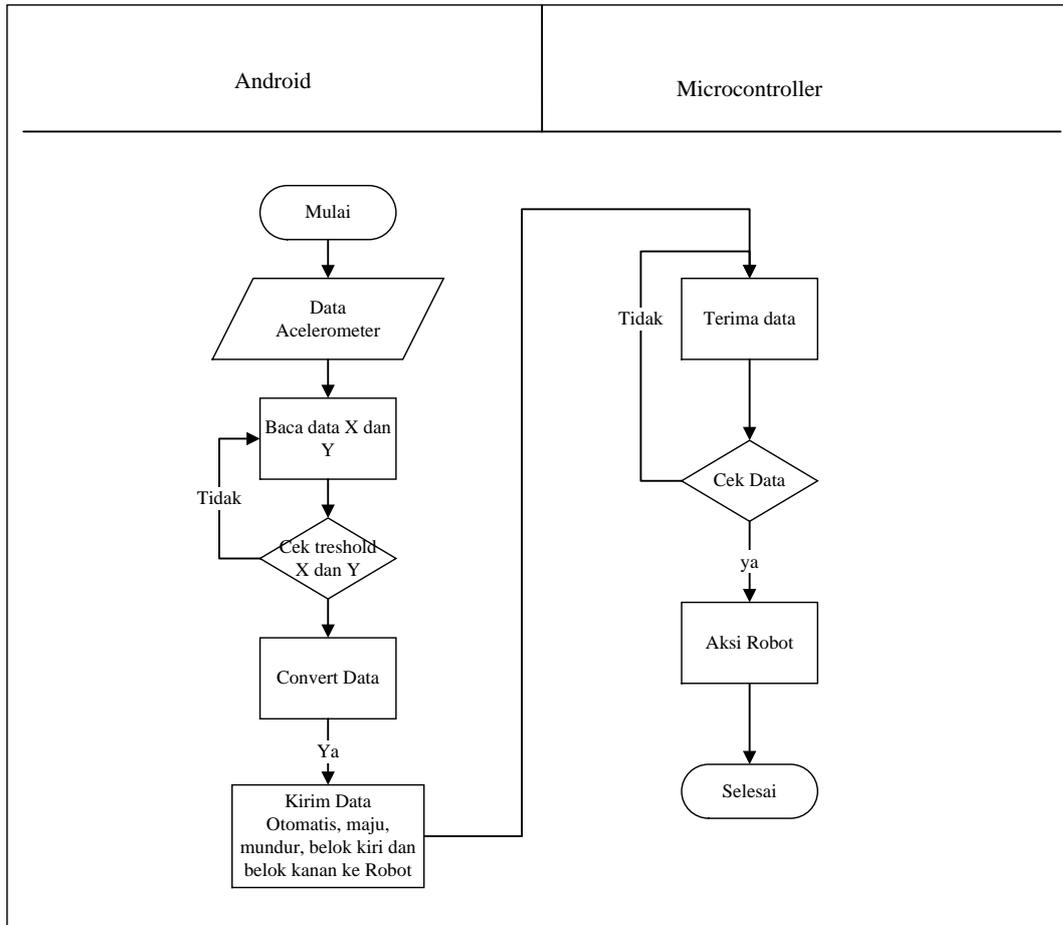
3.2 Studi Literatur

Penelitian ini diawali dengan proses pengkajian literatur yang berkaitan dengan topik penelitian. Proses ini meliputi pencarian referensi pendukung yang sesuai dan dapat dipertanggung jawabkan. Referensi yang digunakan dapat berupa buku, jurnal maupun artikel yang berkaitan dengan pengontrolan robot menggunakan *smartphone* dengan memanfaatkan *bluetooth* sebagai media komunikasi antara robot dan *smartphone* adapun studi literatur yang dilakukan meliputi:

1. *Literature review* tentang navigasi robot menggunakan *accelerometer*.
2. Konsep komunikasi robot menggunakan *bluetooth*.
3. Logika *fuzzy* untuk mengontrol kecepatan robot ketika dikontrol menggunakan gerakan tangan.

3.3 Rancangan Penelitian

Dalam pembuatan robot navigasi yang digerakkan melalui media *smartphone* ini terdapat beberapa tahap perancangan *system* yaitu tahap pembuatan desain antarmuka, dengan harus mengetahui prinsip kerja robot beroda, serta pembuatan algoritma dan program. Berikut merupakan gambaran umum aliran sistem Android dan mikrokontroler yang akan dibangun seperti pada Gambar 3.2.



Gambar 3.2 Diagram alir sistem komunikasi Android dan mikrokontroler

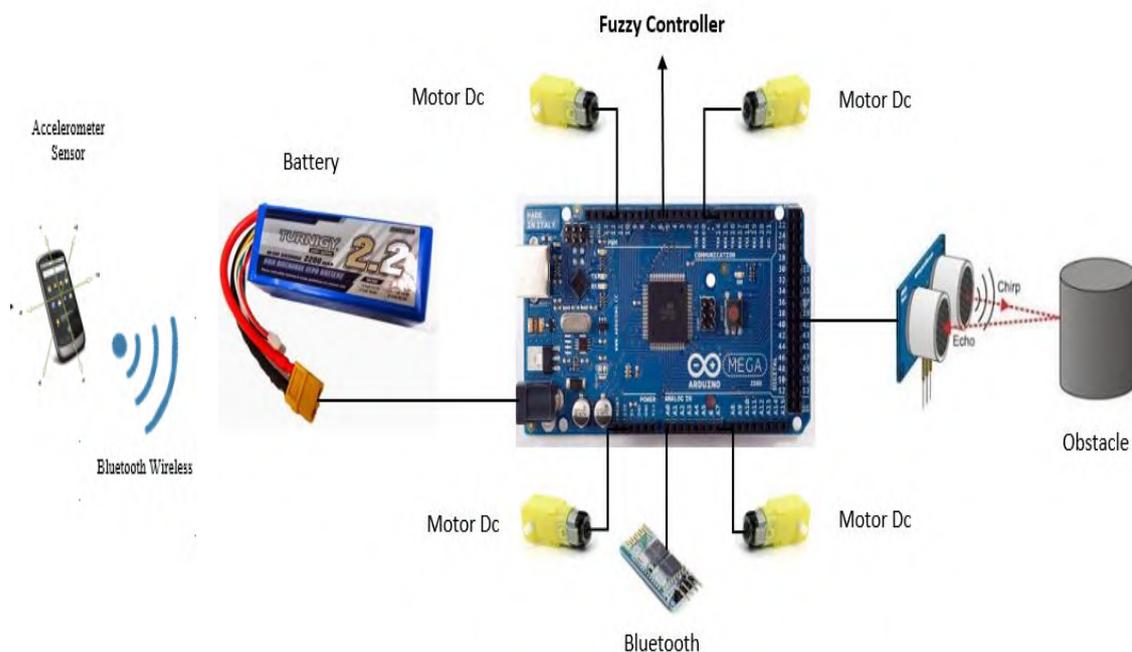
Berdasarkan Gambar 3.2 dapat dilihat bahwa pada penelitian ini terdapat data *input* yang dikirim melalui *bluetooth* ke mikrokontroler yaitu berupa data axis x dan y pada *accelerometer* sehingga menghasilkan *output* yaitu berupa gerakan robot. Pada penelitian ini terdiri dari dua kendali yaitu kendali otomatis yaitu. Robot menghindari halangan yang dilalui. Sedangkan kendali manual berupa gerakan maju, mundur, berhenti, belok kiri dan belok kanan yang dikontrol langsung

menggunakan pergerakan tangan pada *smartphone* dengan cara memiringkan *smartphone* seperti yang terlihat pada Gambar 3.12.

3.3.1 Perancangan Sistem

Pada penelitian ini dirancang sebuah robot yang dapat mempermudah dalam mengontrol robot sehingga pengguna tidak perlu melakukan pelatihan yang khusus untuk menggerakkan robot. Karena pergerakan tangan yang dilakukan merupakan pergerakan tangan yang sering dilakukan dalam kehidupan sehari-hari.

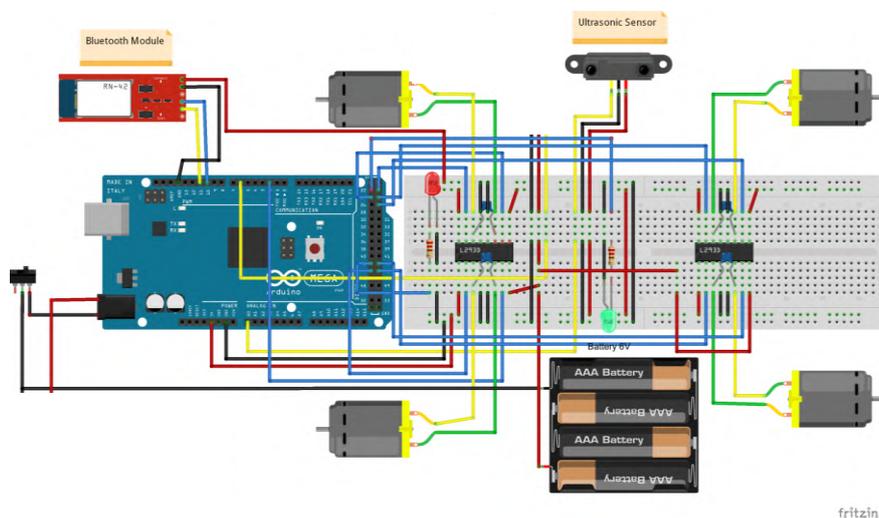
Prinsip kerja dari robot penunjuk arah ini ditentukan oleh serangkaian komponen elektronik yang terdiri dari perangkat mikrokontroler, sensor *ultrasonik*, motor DC dan komunikasi *Bluetooth*. Rancangan arsitektur robot beserta komponen yang digunakan dapat dilihat pada Gambar 3.3.



Gambar 3.3 Rancangan Arsitektur robot

Mikrokontroler digunakan sebagai pengontrol sekaligus sebagai otak dari robot untuk mengatur pergerakan motor untuk berjalan maju, mundur dan berbelok sesuai dengan intruksi yang diberikan oleh pengguna melalui perangkat *smartphone* yang terintegrasi dengan robot. Robot akan berinteraksi dengan lingkungan dengan memanfaatkan sensor ultrasonik sebagai pendeteksi halangan yang ada di depan.

Untuk lebih jelas mengenai sistem yang dirancang. Rangkaian skematik robot dapat dilihat pada Gambar 3.4.



Gambar 3.4 Rangkaian skematik robot

Dari Gambar 3.4 merupakan rangkaian robot yang akan dibangun pada sebuah rangka *acrylic* robot beroda 4 terdiri dari dua tingkat dengan panjang 20 cm, lebar 15 cm dan tinggi 14.5 cm. Untuk lebih jelasnya tentang rangkaian pin yang digunakan pada Arduino untuk dihubungkan ke beberapa komponen seperti motor *shield*, *bluetooth* sensor ultrasonik, berikut adalah penjelasannya.

3.3.2 Inisialisasi Penggunaan Pin pada Motor DC

Penelitian ini menggunakan 4 motor DC yang dikendalikan oleh dua buah motor *shield* dimana fungsinya sebagai pengatur kecepatan dua buah motor DC bagian depan dan dua buah motor DC pada bagian belakang. *Driver* ini menggunakan IC L298D dengan kontrol tegangan sebesar 5 Volt dari Arduino. Tentunya terdapat pin pada Arduino yang digunakan oleh *shield* tersebut untuk berkomunikasi. Adapun pin digital yang digunakan untuk masing-masing motor adalah sebagai berikut:

1. Motor kanan bagian depan

Pin digital untuk motor kanan depan menggunakan pin 22 dan pin 23. Sedangkan untuk mengatur kecepatan menggunakan pin 2 dimana rentang sinyal yang diberikan untuk mengatur kecepatan gerakan motor adalah dari 0 sampai 255 pwm.

2. Motor kiri bagian depan

Pin digital untuk motor kiri depan menggunakan pin 24 dan pin 25. Sedangkan untuk mengatur kecepatan menggunakan pin 3 dimana rentang sinyal yang diberikan untuk mengatur kecepatan gerakan motor adalah dari 0 sampai 255 pwm.

3. Motor kanan bagian belakang

Pin digital untuk motor kanan bagian belakang menggunakan pin 28 dan pin 29. Sedangkan untuk mengatur kecepatan menggunakan pin 4 dimana rentang sinyal yang diberikan untuk mengatur kecepatan gerakan motor adalah dari 0 sampai 255 pwm.

4. Motor kiri bagian belakang

Pin digital untuk motor kiri bagian depan menggunakan pin 30 dan pin 31. Sedangkan untuk mengatur kecepatan menggunakan pin 5 dimana rentang sinyal yang diberikan untuk mengatur kecepatan gerakan motor adalah dari 0 sampai 255 pwm.

3.3.3 Inisialisasi Penggunaan Pin pada Sensor Ultrasonik

Sensor ultrasonik akan bekerja jika mendapat *supply* tegangan sebesar 5 Volt. dimana tegangan 5 Volt dihubungkan dengan konektor Vcc dan *Ground* pada sensor. Untuk konektor SIG dapat dihubungkan dengan mikrokontroler. Konektor SIG adalah sebagai kontrol sensor dalam pendeteksian objek sekaligus pembacaan jarak objek dengan sensor ini. Adapun pin yang digunakan untuk konektor SIG pada sensor adalah sebagai berikut:

1. Sensor bagian kiri menggunakan pin 34 dan 35.
2. Sensor bagian kanan menggunakan pin 36 dan 37.
3. Sensor bagian depan menggunakan pin 38 dan 39.

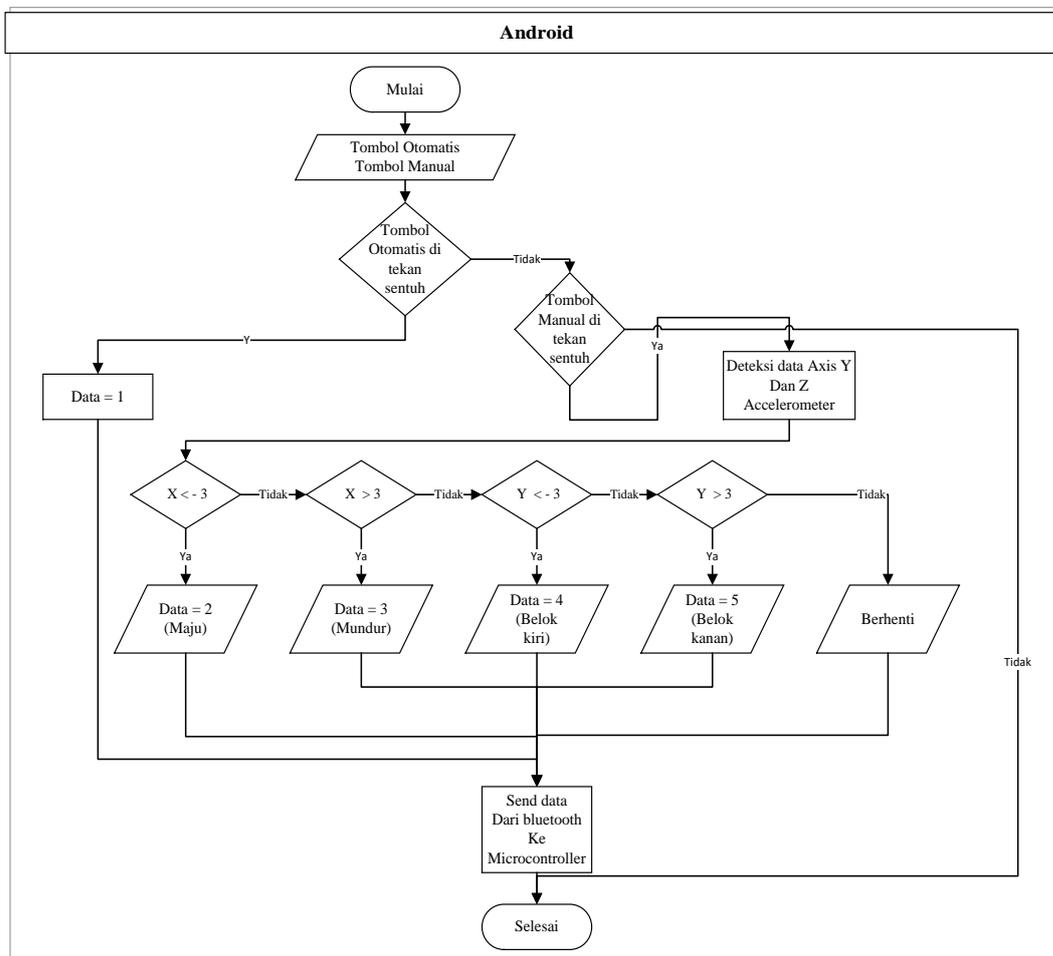
3.3.4 Inisialisasi Penggunaan Pin untuk *Bluetooth*

Bluetooth module HC-06 merupakan modul komunikasi nirkabel pada frekuensi 2.4GHz dengan *default* koneksi hanya sebagai *SLAVE*. *Interface* yang digunakan adalah serial RXD, TXD, VCC dan GND. *Built in* LED sebagai indikator koneksi *bluetooth*. Berikut adalah pin yang digunakan untuk komunikasi TX dan RX pada *bluetooth*.

1. TX menggunakan pin 10 pada Arduino merupakan pin yang berfungsi untuk mengirimkan data dari modul ke perangkat lain (mikrokontroler).
2. RX menggunakan pin 11 pada Arduino merupakan pin yang berfungsi untuk menerima data yang dikirim ke modul HC-06. Tegangan sinyal pada pin sama dengan tegangan sinyal pada pin TX.

3.4 Desain Komunikasi dan Navigasi Robot

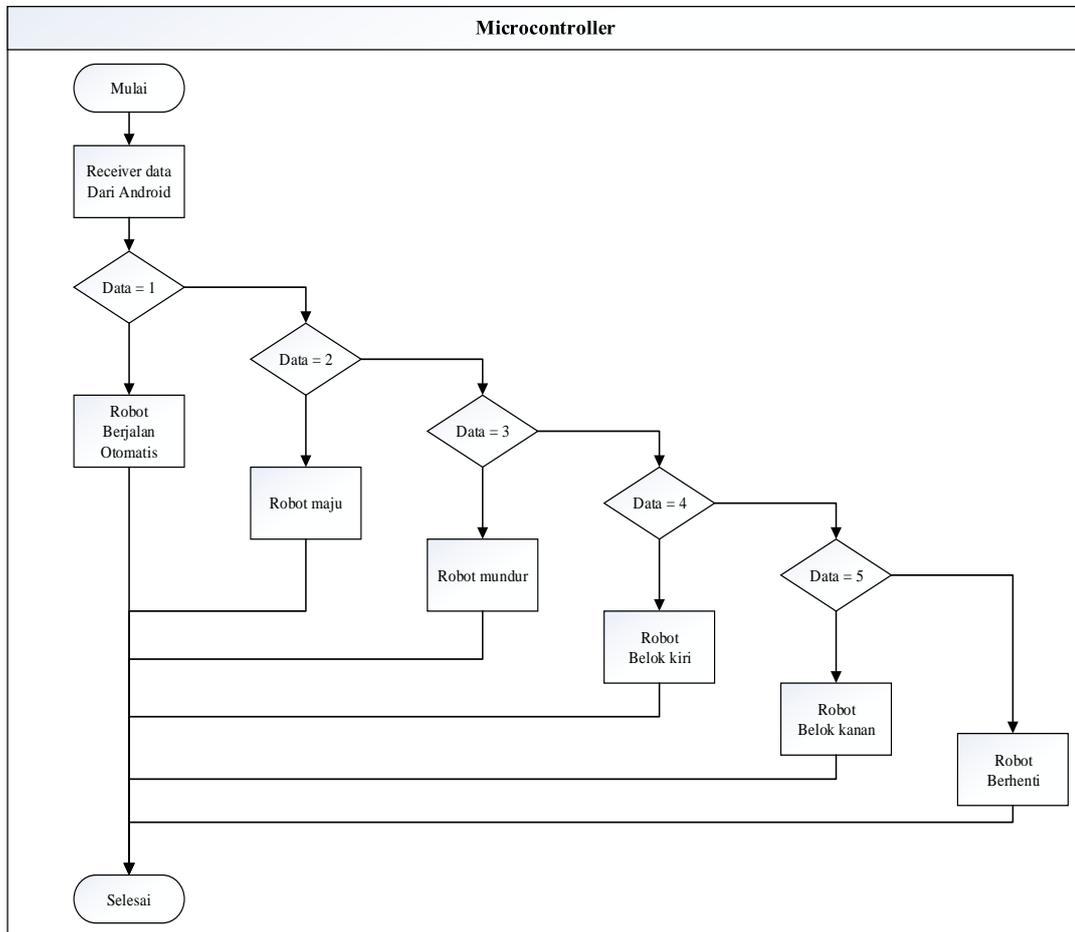
Setelah mendesain rangkaian sistem robot sesuai dengan skematik di atas langkah selanjutnya ialah mendesain sistem kerja pergerakan robot dan membangun sebuah aplikasi pada *smartphone* Android. Pergerakan robot dilakukan berdasarkan nilai *threshold* axis x dan y yang diberikan pada sensor *accelerometer*.



Gambar 3.5 Diagram alir data axis Y dan Z pada *smartphone*

Dalam melakukan navigasi robot terdapat dua tugas utama robot yaitu membaca data sensor *accelerometer* yang diterima dari perangkat *smartphone*

Android dan melakukan aktivitas *scanning* sensor *ultrasonic* pada robot untuk merekam data jarak pada lingkungan. Berikut adalah proses penerimaan data dari perangkat *smartphone* ke mikrokontroler seperti yang ditunjukkan pada Gambar 3.6



Gambar 3.6 Proses penerimaan data mikrokontroler

Dari Gambar 3.6 dapat dilihat adalah proses penerimaan data yang dikirim dari *smartphone* melalui media komunikasi *bluetooth* berupa data axis Y dan Z yang di inialisasikan menjadi dua jenis pengontrolan yaitu kontrol otomatis jika pengguna menekan tombol otomatis pada tampilan atarmuka *smartphone* yang dibaca sebagai data 1 pada mikrokontroler. Jika pengguna menekan tombol manual maka data akan diproses pada mikrokontroler berupa data 2 untuk aksi maju, data 3 untuk aksi mundur, data 4 untuk belok kiri dan data 5 untuk belok kanan, jika pengguna tidak melakukan aksi dengan memiringkan *smartphone* ke posisi depan maupun ke posisi samping maka aksi robot tidak akan melakukan aksi.

1. Pembacaan data sensor *accelerometer*

Setelah mendapat kiriman data dari *smartphone* Android yang berupa data pergerakan robot yang diinisialisasikan dengan satu karakter untuk setiap gerakan robot. Langkah selanjutnya ialah merepresentasikan data kiriman tersebut menjadi instruksi untuk robot. Untuk memberikan gambaran sistem mengenai ekstraksi data yang diterima dari perangkat *smartphone* ke mikrokontroler dapat dilihat pada Gambar 3.7.

```
Start
  Read (receive)
    IF receive == ' f ' THEN
      action = "forward"
    ELSE IF receive == 'b' THEN
      action = "backward"
    ELSE IF receive == 'l' THEN
      action = "turn_left"
    ELSE IF receive == 'r' THEN
      action = "turn_right"
    ELSE
      action = "stop"
    END IF
  Print action
End
```

Gambar 3.7 Pengolahan data yang diterima

2. *Scanning* Sensor ultrasonik

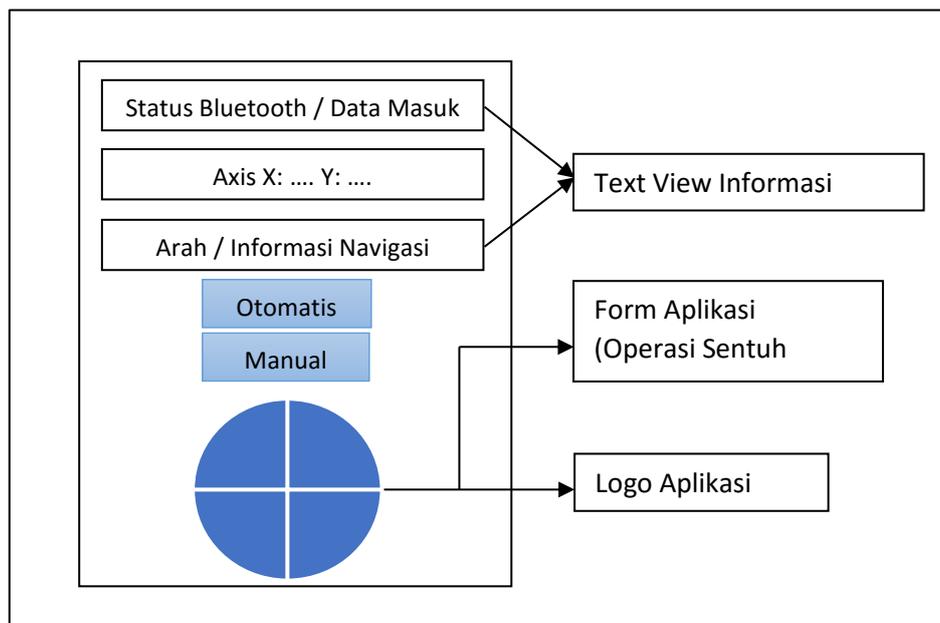
Proses *scanning* jarak dilakukan dengan membaca data pada sensor *ultrasonic* yang dipasang pada bagian depan robot. Hal ini bertujuan untuk mendeteksi jarak antara robot ke objek yang menghalang.

```
Start
  Read (front_distance)
    IF distance < 50 THEN
      action = "stop"
      Read (left_distance)
      Read (right_distance)
      IF left_distance > 100 AND
      right_distance > 100 THEN
        Print "choice_left_or_right"
      ELSE IF left_distance > 100 THEN
        Print "choice_left"
      ELSE IF right_distance > 100 THEN
        Print "choice_right"
      ELSE
        Print "choice_backward"
      END IF
    END IF
  END IF
```

Gambar 3.8 *Scanning* jarak robot

Proses *scanning* hanya berjalan pada saat kondisi robot berjalan maju. *Pseudocode scanning* jarak robot terhadap objek atau benda yang menghalanginya dapat dilihat pada Gambar 3.8.

Antarmuka pengguna yang dibangun merupakan antarmuka yang didesain sederhana, sehingga pengguna mudah untuk mengontrol robot dan tidak memiliki banyak fitur yang menyulitkan dalam menggunakan aplikasi. Pengoperasian aplikasi dilakukan dengan menggunakan sentuhan pada layar *smartphone* dengan 2 pilihan kontrol yaitu kontrol otomatis dan kontrol secara manual dengan gerakan memiringkan posisi *handphone* seperti yang terlihat pada Gambar 3.12 untuk menggerakkan robot maju, mundur, belok kiri dan belok kanan. Bentuk tampilan aplikasi kontrol robot pada *smartphone* seperti yang ditunjukkan pada Gambar 3.9.

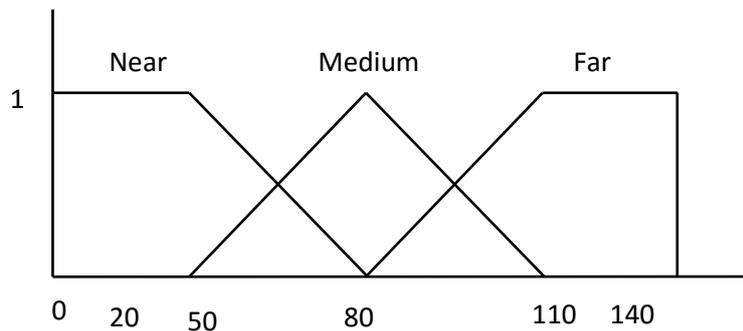


Gambar 3.9 Desain antarmuka pada *smartphone*

3.5 Desain Metode *Fuzzy*

Pada penelitian ini akan dirancang sebuah metode logika *fuzzy* (Hee Nam, Boo; Jung Tae 2000), dalam (Mohammad et al. 2013) yang berguna untuk mengatur kecepatan motor berdasarkan informasi dari sensor ultrasonik yang telah dipasang sebanyak tiga buah pada sisi depan, sisi kanan dan sisi kiri robot yang memberikan informasi jarak berupa digit–digit biner 6 bit dengan rentang pengukuran 3 cm hingga

1 m. Informasi dari jarak sensor ultrasonik tersebut akan diteruskan ke logika *fuzzy* dengan *membership function* secara heuristik seperti yang ditunjukkan pada Gambar 3.10. Hal ini bertujuan untuk meningkatkan ketepatan robot berhenti ketika menemukan halangan. Apabila jarak robot jauh dari halangan yang ada di depan maka robot akan bergerak semakin cepat, namun ketika halangan berada semakin dekat maka robot akan bergerak perlahan (mengurangi kecepatan secara adaptif) mendekati *threshold* halangan yang diatur dan kemudian robot akan berhenti.



Gambar 3.10 *Membership function* untuk *input* jarak dari sensor ultrasonik

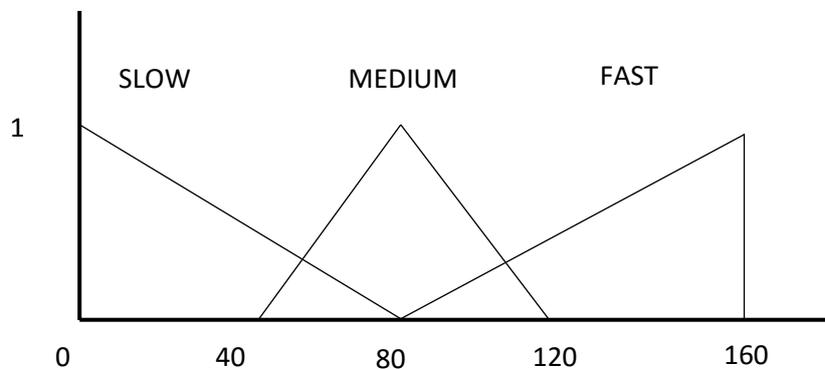
Model *membership function* yang dipakai adalah model kurva bahu yang terdiri dari kurva linear turun, kurva segitiga dan kurva linear naik. Jika dibuat batas-batas keanggotaan seperti Gambar 3.10 maka didapatkan kelandaian dari trapesium dan segitiga adalah sebesar 45° . Dengan demikian persamaan garis dari sisi miring dari trapesium dan segitiga. Sehingga didapatkan tabel *input* jarak seperti yang ditunjukkan pada Tabel 3.1.

Tabel 3.1 *Membership function* untuk *input* kecepatan dan jarak

Label	Nilai	Rentang masukan
Inrange	1	for $X \leq 20$
	$50 - X$	for $20 < X < 80$
	0	for $X \geq 80$
Midrange	$X - 50$	for $50 < X < 80$
	$110 - X$	for $80 < X < 110$
	0	for $X \leq 50$ or $X \geq 110$
	1	for $X = 50$

Outrange	$X - 80$	for $80 < X < 140$
	1	for $X \geq 140$
	0	for $X \leq 80$

Pada penelitian (Indar, 2008) digunakan model *singleton* pada *membership function* pada bagian *output* dengan tiga label. *Slow*, *medium* dan *fast* konsekuensinya pergerakan robot menjadi kurang halus maka pada penelitian ini akan dilakukan perubahan pada *membership function output* menjadi seperti Gambar 3.11.



Gambar 3.11 *Membership function* untuk kecepatan

Pada bagian *output* dirancang dengan pendekatan integer seperti pada *membership function* bagian input. PWM yang diatur untuk mengatur kecepatan motor yang memiliki resolusi 6 bit. Dengan demikian akan didapatkan nilai bit 1 untuk *duty cycle* 25%, 50 untuk *duty cycle* 50%, untuk proses berikutnya adalah mendapatkan *fuzzy output* dari *fuzzy variable* di atas dengan evaluasi aturan (*rule evaluation*) seperti yang terlihat pada Tabel 3.2.

Tabel 3.2 *Rule evaluation* untuk sistem *fuzzy* (Martin 2009)

Input			Output	
Sensor kiri	Sensor depan	Sensor kanan	Kecepatan	Manuver
Dekat	Dekat	Dekat	Pelan	Right big
Sedang	Sedang	Jauh	Menengah	Right small
Dekat	Sedang	Dekat	Pelan	No turn
Dekat	Jauh	Jauh	Cepat	Right small
Jauh	Dekat	Jauh	Menengah	Left small

Jauh	Dekat	Jauh	Pelan	Right big
Jauh	Jauh	Dekat	Cepat	Left small
Jauh	Jauh	Jauh	Cepat	No turn

3.6 Implementasi Penelitian

Pada tahapan ini dilakukan implementasi dari desain sistem kedalam bentuk *mobile* robot. Adapun implementasi penelitian terdiri atas dua lingkungan pengembangan yaitu pengembangan *mobile* robot dan pengembangan aplikasi Android sebagai unit kontrol navigasi robot bagi pengguna. Lingkungan penelitian untuk pengembangan sistem dan proses uji coba adalah sebagai berikut:

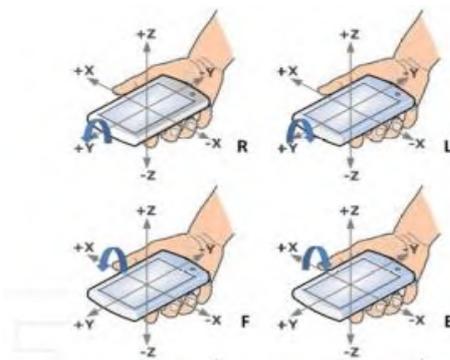
1. Spesifikasi perangkat keras yang digunakan
 - a. *CPU : Intel Core i5 – 2.5GHz*
 - b. *RAM: 8GB*
 - c. *Arduino Mega 2560*
 - d. *Sensor Ultrasonik (HCSR04)*
 - e. *Bluetooth Module (HC06)*
 - f. *Motor DC + Roda*
 - g. *Motor Driver*
 - h. *Battery*
 - i. *Step Down Regulator*
 - j. *Kabel Jumper*
 - k. *Rangka Acrylic Robot Beroda*
 - l. *Smartphone Android*
2. Spesifikasi perangkat lunak yang digunakan
 - a. *Sistem operasi Windows 10, 64 bit*
 - b. *Eclipse Luna*
 - c. *Android SDK + ADT*
 - d. *Arduino IDE*
 - e. *Photoshop*

3.7 Uji Coba dan Analisis

Tahap ini menjelaskan tentang skenario uji coba tentang pengujian metode usulan dengan metode perbandingan antara pengontrolan robot menggunakan logika *fuzzy* dan pengontrolan pergerakan tangan seperti yang dilakukan oleh (Wu dkk, 2013) yaitu kontrol robot menggunakan gerakan tangan pada *smartphone*. Pengujian dilakukan pada beberapa kecepatan dan jarak pada setiap kecepatan dan jarak akan dilakukan sebanyak 10 kali percobaan.

3.7.1 Skenario dan Uji Coba

Uji coba penelitian ini menggunakan pola gerak tangan yang umum digunakan dalam kehidupan sehari-hari. Seorang pengguna dapat dengan mudah mengulang semua gerakan tanpa melakukan pelatihan khusus. Gerak tangan untuk mengendalikan robot dengan menggenggam *smartphone* di tangan kanan pengguna dalam sebagai contoh. R (L) merupakan perintah untuk mengubah robot kanan (kiri) ketika pengguna memiringkan kontroler ke kanan (kiri). F (B) merupakan perintah untuk memindahkan robot (maju) ketika pengguna memiringkan *smartphone* ke depan (belakang). Merupakan perintah untuk membuat robot (mundur) ketika pengguna mengembalikan posisi *smartphone* dalam keadaan datar maka robot akan berhenti (S). merupakan perintah untuk berhenti. Pada penelitian ini merancang 5 gerakan tangan yang sederhana dari sumbu X dan Y pada *accelerometer* untuk pengontrolan robot, seperti yang ditunjukkan pada Gambar 3.12.



Gambar 3.12 Gerakan tangan sumbu x dan y *accelerometer*

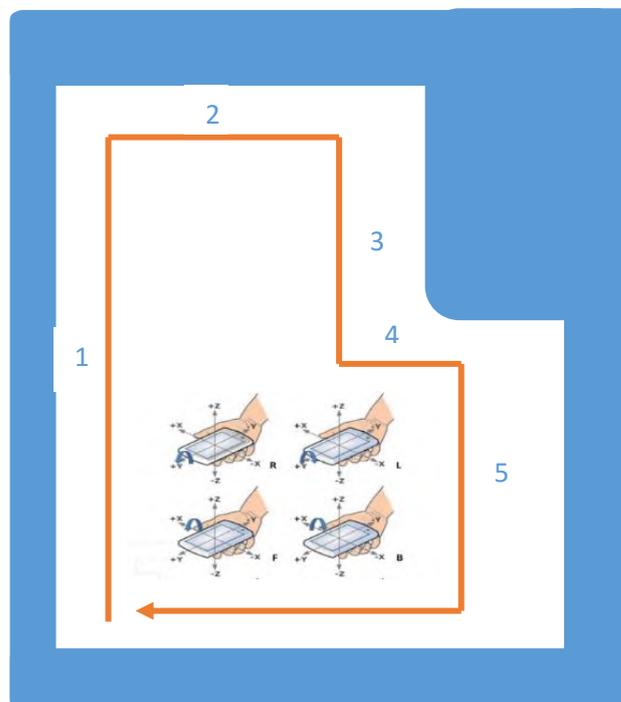
Ada beberapa skenario yang dilakukan pada setiap pengujian untuk mengetahui performa robot yaitu pengujian berdasarkan jarak dan kecepatan supaya mengetahui tingkat akurasi pada masing-masing metode ada pun jarak dan kecepatan

yang digunakan adalah dimulai dari jarak 20 cm sampai dengan 70 cm sedangkan untuk kecepatan menggunakan tiga kecepatan yaitu 160, 200 dan 240 pwm.

Dari setiap percobaan akan dilakukan pengukuran secara manual jarak sensor dengan halangan secara manual pada saat robot berhenti. Dari data pengukuran jarak yang dilakukan maka dilakukan kalkulasi untuk mendapatkan nilai rata-rata demi mengetahui nilai persentase *error* ketika robot berhenti sesuai dengan jarak yang telah ditanam dalam mikrokontroler.

3.7.2 Ruang Uji Coba

Untuk menguji kinerja kontrol robot dilakukan pada sebuah ruangan dengan panjang 4 meter dan lebar 3 meter. Robot dijalankan mengikuti garis merah muda yang diberi angka dari 1 sampai 5 sehingga robot kembali pada posisi awal, dengan kontroler genggam menggunakan *smartphone* untuk menggerakkan robot mendeteksi rintangan yang telah dirancang. Sehingga apabila robot menemukan halangan maka robot akan berhenti secara otomatis berdasarkan jarak yang ditentukan.



Gambar 3.13 Ruang uji coba

Pada uji coba ini pengguna diminta untuk menggunakan kontrol pergerakan tangan robot menggunakan logika *fuzzy* dan tanpa menggunakan metode logika

fuzzy. Kedua metode dikontrol pada ruangan yang sama, kecepatan yang sama dan pada gerakan tangan yang sama. Navigasi diulang 10 kali untuk kontrol manual agar mengetahui ketepatan kontrol pada setiap gerakan. Jarak robot akan diukur secara manual pada saat berhenti dengan halangan dan diambil nilai rata-rata dari 10 kali hasil pengujian pada masing jarak dan kecepatan sehingga mendapatkan nilai *error* dari setiap kontrol.

[Halaman ini sengaja dikosongkan]

BAB 4

HASIL DAN PEMBAHASAN

Berdasarkan perancangan yang telah dilakukan pada Bab 3, selanjutnya adalah melakukan pengujian dan analisis data dari beberapa percobaan. Pengujian ini dilakukan untuk mengetahui apakah sistem yang telah ditanamkan pada robot berjalan sesuai dengan harapan. Pada bab ini, akan dijelaskan mengenai performa kesesuaian robot berhenti pada jarak yang telah ditanam pada mikrokontroler dan kesesuaian robot bermanuver berdasarkan halangan yang dilalui.

4.1 Lingkungan Uji Coba

Dalam membangun robot ini digunakan beberapa perangkat yang mendukung baik perangkat keras maupun perangkat lunak. Adapun lingkungan penelitian untuk pengembangan sistem adalah sebagai berikut:

1. Spesifikasi perangkat keras yang digunakan seperti yang ditunjukkan pada Tabel 4.1.

Tabel 4.1 Spesifikasi perangkat keras yang digunakan.

No	Perangkat Keras yang digunakan
1	CPU: Intel Core i5 – 2.5 GHz RAM: 4GB
2	Arduino Mega (2650)
3	Sensor Ultrasonik (HCSR04)
4	Bluetooth Modul (HC06)
5	Motor DC + Roda
6	Motor Driver
7	Battery
8	Step Down Regulator
9	Kabel Jumper Male to Male
10	Kabel Jumper Male to Female
11	Kabel tunggal
12	Rangka Acrylic Robot Beroda
13	Smartphone Android
14	Spicer

2. Spesifikasi perangkat lunak yang digunakan.

Tabel 4.2 Spesifikasi perangkat lunak yang digunakan.

No	Perangkat lunak yang digunakan
1	Sistem Operasi Windows 10, 64 bit
2	Eclipse Luna
3	Android SDK + ADT
4	Arduino IDE
5	Photoshop

4.2 Pengujian Robot Otomatis Pada Ruangan Bebas

Pengujian ini bertujuan mengetahui kinerja dari sistem navigasi secara keseluruhan dengan menggunakan robot *mobile*. Kinerja yang dimaksud di sini adalah kesesuaian manuver terhadap instruksi yang ditanamkan pada kontroler. Pengujian dilakukan dengan membuat sebuah ruangan tes yang terdapat beberapa objek halangan yang dilalui oleh robot. Sehingga robot menghindari dari halangan tersebut dan mencari alternatif jalan yang bisa dilewatinya. Ada beberapa kombinasi halangan yang diberikan antara lain adalah sebagai berikut.

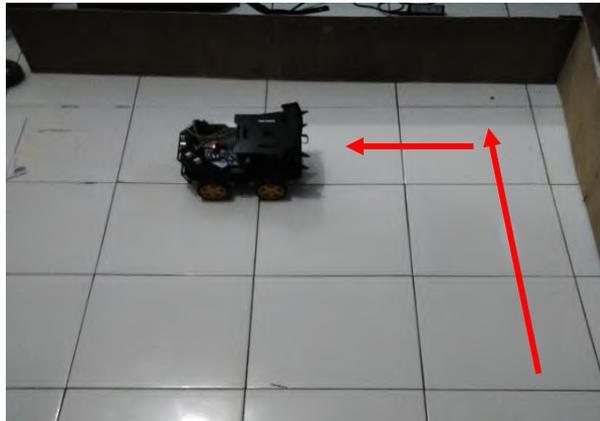
4.2.1 Halangan Berada di Depan dan Samping Kanan Robot

Untuk pengujian kondisi ini, halangan diatur berada di depan dan samping kiri dari robot dengan jaraknya ± 50 cm sehingga sensor jarak akan membaca halangan di sisi depan dan kanan. Untuk lebih jelasnya bentuk pengujian robot ini dapat dilihat pada Gambar 4.1.



Gambar 4.1 Robot ketika menemukan halangan disisi kanan dan di depan

Dari Gambar 4.1 dapat diketahui bahwa robot akan melakukan manuver ke kiri karena sensor jarak kanan mendeteksi adanya halangan di sisi kanan robot. Manuver ini dilakukan dengan membandingkan antara nilai pembacaan sensor jarak sisi kiri robot lebih besar daripada sisi kanan.



Gambar 4.2 Robot bermanuver kekiri

Untuk memastikan bahwa pergerakan robot sesuai dengan pembacaan sensor maka diambil data sensor pada saat berhenti setelah mendeteksi halangan. Ada tiga sensor yang dipasang yaitu berada di depan, sisi kanan dan kiri robot. Dilakukan lima kali percobaan dengan kondisi yang sama sehingga didapatkan hasil seperti pada Tabel 4.3.

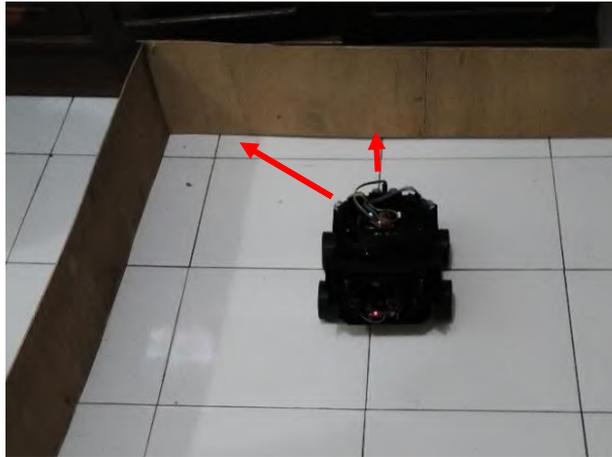
Tabel 4.3 Hasil pembacaan sensor untuk halangan di depan dan di kanan

Sensor Kiri (cm)	Sensor Depan (cm)	Sensor Kanan (cm)
51	23	31
73	43	29
56	43	45
76	44	26
71	44	31

Dari Tabel 4.3 dapat dilihat bahwa pergerakan robot sesuai dengan pembacaan sensor yang dipasang pada robot *mobile* yaitu robot bermanuver ke kiri sesuai dengan informasi jarak yang diterima dari tiga buah sensor ultrasonik.

4.2.2 Halangan Berada di Depan dan Samping Kiri Robot

Untuk pengujian kondisi ini, halangan diatur berada di depan dan kiri dari robot dengan jaraknya ± 50 cm sehingga sensor jarak akan membaca halangan di ketiga sisi sesuai posisi sensor yang terpasang pada robot. Untuk lebih jelasnya bentuk pengujian pada halangan ini dapat dilihat pada Gambar 4.3 dan Gambar 4.4.



Gambar 4.3 Robot ketika menemukan halangan di sisi kiri dan di depan

Dari Gambar 4.3 dapat diketahui bahwa robot akan melakukan manuver kanan karena sensor jarak kiri mendeteksi adanya halangan di sisi kiri robot. Manuver ini dilakukan dengan membandingkan antara nilai pembacaan sensor jarak sisi kiri robot lebih kecil daripada sisi kanan.



Gambar 4.4 Robot bermanuver kekanan

Untuk memastikan bahwa pergerakan robot sesuai dengan pembacaan sensor maka diambil data sensor pada saat berhenti setelah mendeteksi halangan. Hasil pembacaan sensor dilakukan dengan melakukan lima kali percobaan dengan kondisi yang sama sehingga didapatkan hasil seperti pada Tabel 4.4.

Tabel 4.4 Hasil Pembacaan sensor untuk halangan di depan dan kiri

Sensor Kiri (cm)	Sensor Depan (cm)	Sensor Kanan (cm)
23	42	70
31	44	62
42	41	52
38	46	56
37	46	56

Dari Tabel 4.4 Dapat dilihat bahwa pergerakan robot sesuai dengan intruksi sistem yang ditanam pada mikrokontroler robot bermanuver ke kanan berdasarkan pembacaan nilai jarak dari ketiga sensor ultrasonik yang terpasang.

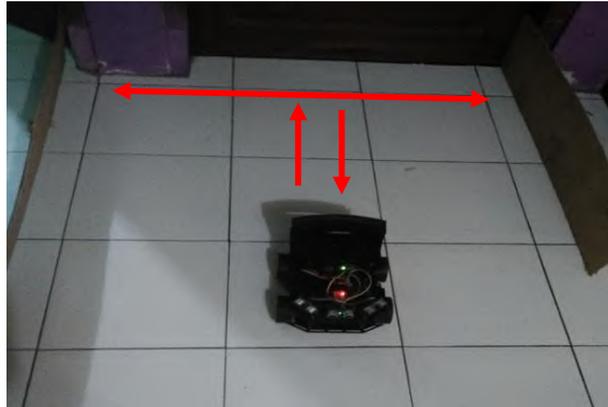
4.2.3 Halangan Berada di Depan, Kiri dan Kanan Robot

Untuk pengujian kondisi ini, halangan diatur berada di depan, samping kiri dan kanan dari robot dengan jaraknya ± 50 cm sehingga sensor jarak akan membaca halangan di ketiga sisi. Untuk lebih jelasnya bentuk pengujian robot ini dapat dilihat pada Gambar 4.5 dan aksi robot setelah mendeteksi halangan dapat dilihat pada Gambar 4.6.



Gambar 4.5 Robot ketika diberikan halangan di sisi kiri, kanan dan depan

Karena kesemua sensor jarak mendeteksi adanya halangan di sisi depan, sisi kiri dan sisi kanan robot. Manuver ini dilakukan dengan membandingkan antara nilai pembacaan ketiga sensor jarak.



Gambar 4.6 Robot bermanuver ke belakang

Dari Gambar 4.6 dapat diketahui bahwa robot akan melakukan manuver berbalik arah karena semua sensor mendeteksi adanya halangan di ketiga sisi robot. Untuk memastikan bahwa pergerakan robot sesuai dengan pembacaan sensor maka diambil data jarak robot dan halangan pada saat robot berhenti. Sehingga didapatkan hasil seperti pada Tabel 4.5.

Tabel 4.5 Tabel hasil percobaan robot bermanuver kebelakang

Ultrasonik Kiri (cm)	Ultrasonik Depan (cm)	Ultrasonik Kanan (cm)
18	44	26
23	49	21
18	46	26
20	49	24
16	50	30

Dari Tabel 4.5 dapat dilihat bahwa pergerakan robot sesuai dengan pembacaan sensor yang dipasang pada robot *mobile* yaitu robot berbalik arah ketika semua sensor ultrasonik mendeteksi adanya halangan.

4.2.4 Halangan Berada di Depan Robot, Melewati Persimpangan

Untuk pengujian kondisi ini, halangan diatur berada di depan dari robot dengan jaraknya ± 50 cm sehingga sensor jarak akan membaca halangan dari sisi depan terlebih dahulu dan kemudian mengambil keputusan belok kiri atau belok kanan sesuai dari data pembacaan sensor. Untuk lebih jelasnya bentuk pengujian robot ini dapat dilihat pada Gambar 4.7.



Gambar 4.7 Robot melewati persimpangan

Karena kesemua sensor jarak mendeteksi adanya halangan di sisi depan. Manuver ini dilakukan dengan membandingkan antara nilai pembacaan sensor kiri dan kanan. Berikut adalah posisi robot ketika mendekati halangan dan mendeteksi jarak di sekitar mendapati bahwa jarak sensor kanan lebih dekat dibandingkan sensor kiri maka robot akan bermanuver ke kanan. Seperti yang terlihat pada Gambar 4.8.



Gambar 4.8 Robot ketika halangan dari semua sisi

Dari Gambar 4.9 dapat diketahui bahwa robot akan melakukan manuver ke kiri karena sensor jarak kanan mendeteksi adanya halangan di sisi kanan robot. Manuver ini dilakukan dengan membandingkan antara nilai pembacaan sensor jarak sisi kanan robot lebih kecil daripada sisi kiri.



Gambar 4.9 Robot bermanuver ke kiri

Untuk memastikan bahwa pergerakan robot sesuai dengan pembacaan sensor maka diambil data jarak pada saat robot berhenti. Dilakukan dengan sebanyak lima kali percobaan dengan kondisi yang sama sehingga didapatkan hasil seperti pada Tabel 4.6.

Tabel 4.6 Percobaan robot bermanuver ke kiri dan kanan

Sensor Kiri (cm)	Sensor Depan (cm)	Sensor Kanan (cm)
17	12	24
25	17	18
12	16	20
25	21.5	18.5
15	21	30

4.2.5 Halangan Berada di Depan Robot Berbentuk Segi Tiga

Untuk pengujian kondisi ini, halangan diatur berada di depan dari robot dengan jaraknya ± 50 cm sehingga sensor jarak akan membaca halangan dari sisi depan terlebih dahulu. Namun dari beberapa pengujian robot menabrak halangan, ini disebabkan sensor ultrasonik tidak bisa mengirim dan memantulkan gelombang pada halangan yang berbentuk miring seperti yang terlihat pada Gambar 4.10.



Gambar 4.10 Robot melewati halangan di sisi depan berbentuk kerucut

Dari semua pengujian otomatis yang telah dilakukan robot bermanuver sesuai dengan harapan berdasarkan informasi jarak yang didapatkan dari sensor ultrasonik. Kecuali pada halangan tertentu robot menabrak halangan seperti yang terlihat pada Gambar 4.10. Pembacaan pada halangan ini bukan karena kegagalan metode yang diusulkan tapi karena kelemahan dari sensor ultrasonik itu sendiri. Berikut adalah *input* dan *output* dari hasil pengujian secara otomatis yang dilakukan seperti yang terlihat pada Tabel 4.7.

Tabel 4.7 *Input* dan *output* yang dihasilkan

Input			Output	
Sensor kiri	Sensor depan	Sensor kanan	Kecepatan	Manuver
Sedang	Dekat	Dekat	Pelan	Belok kanan
Dekat	Dekat	Sedang	Pelan	Belok kiri
Dekat	Dekat	Dekat	Pelan	Berbalik arah
Jauh	Jauh	Jauh	Cepat	Tidak belok
Jauh	Dekat	Jauh	Pelan	Tidak belok
Jauh	jauh	Dekat	Cepat	Belok kiri
Jauh	Sedang	Dekat	Menengah	Belok kiri
Dekat	Jauh	Jauh	Cepat	Belok kanan

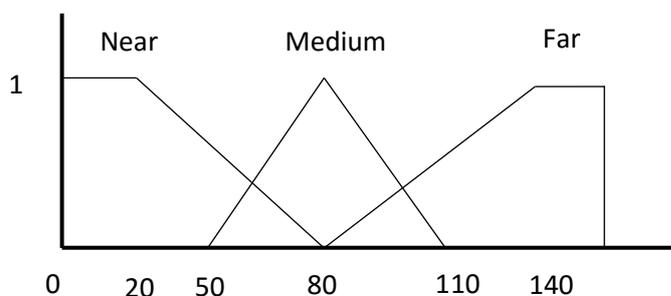
4.3 Pengujian Kontrol Manual Menggunakan Logika Fuzzy

Pengujian ini bertujuan mengetahui kinerja dari sistem navigasi dengan menggunakan logika *fuzzy* dan non *fuzzy*. Kinerja yang dimaksud di sini adalah

kesesuaian terhadap instruksi yang ditanamkan pada mikrokontroler pada pengujian ini akan dilakukan beberapa skenario yaitu pada jarak 20 cm, 30 cm, 40 cm, 50 cm, 60 cm dan 70 cm. Sedangkan untuk kecepatan menggunakan tiga buah kecepatan yaitu 160 pwm, 200 pwm, dan 240 pwm pada masing-masing jarak dan kecepatan dilakukan sebanyak sepuluh kali percobaan dengan tujuan agar bisa mengetahui performa kesesuaian robot berhenti ketika menemukan halangan.

4.3.1 *Membership function jarak yang digunakan*

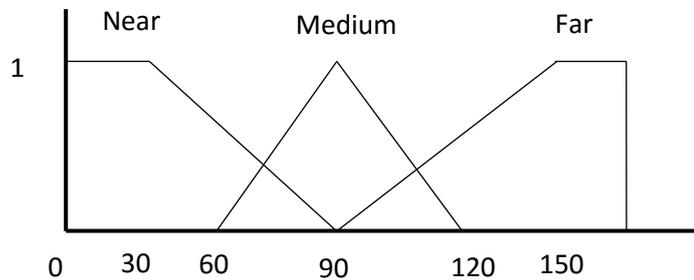
Gambar 4.11 merupakan *membership function* untuk pembacaan sensor ultrasonik terhadap halangan yang ada di depan kurva yang digunakan pada pengujian ini adalah kurva bahu yang terdiri dari kurva linear turun merupakan jarak dekat antara robot dan halangan yaitu dari 20 sampai dengan 80 cm. Jadi ketika robot mencapai pada nilai 20 cm maka robot akan berhenti karena pada jarak itu telah diset kecepatan motor hingga 0 pwm. Sedangkan pada kurva segitiga merupakan jarak menengah antara robot dan halangan yaitu pada nilai 50, 80 dan 100 cm. Pada kurva linear naik merupakan jarak jauh robot dengan halangan yaitu antara 80 hingga 140 cm.



Gambar 4.11 *Membership function* untuk jarak 20 cm

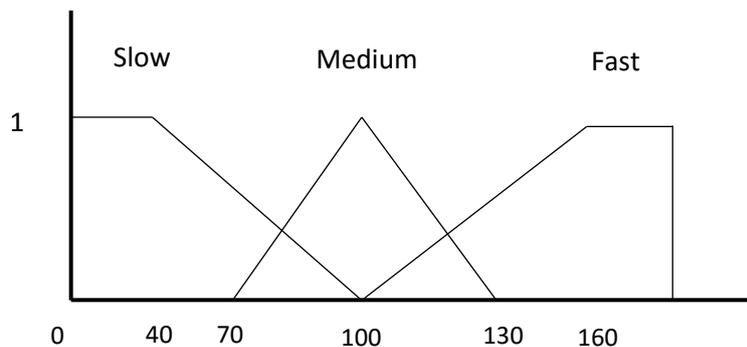
Gambar 4.12 merupakan *membership function* untuk pembacaan sensor ultrasonik terhadap halangan yang ada di depan kurva yang digunakan pada pengujian ini adalah kurva bahu yang terdiri dari kurva linear turun merupakan jarak dekat antara robot dan halangan yaitu dari 30 sampai dengan 90 cm. Ketika robot mencapai pada nilai 30 cm maka robot akan berhenti karena pada jarak itu telah diset kecepatan motor hingga 0 pwm. Sedangkan pada kurva segitiga merupakan jarak menengah antara robot dan halangan yaitu pada nilai 60, 90 dan 120 cm. Pada kurva

linear naik merupakan jarak jauh robot dengan halangan yaitu berkisar antara 80 hingga 150 cm pembacaan sensor ultrasonik dimulai pada jarak 150 cm.



Gambar 4.12 *Membership function* untuk jarak 30 cm

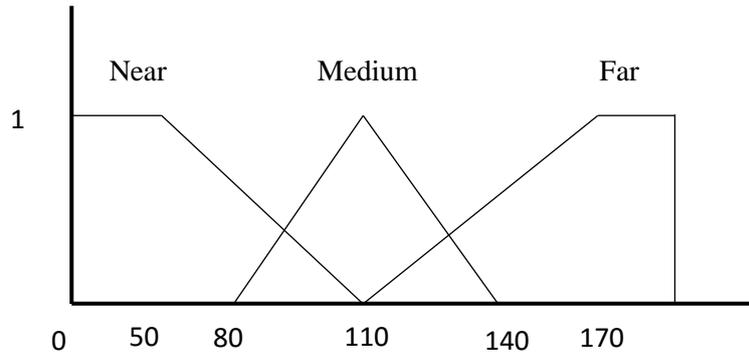
Gambar 4.13 merupakan *membership function* untuk pembacaan sensor ultrasonik terhadap halangan yang ada di depan kurva yang digunakan pada pengujian ini adalah kurva bahu yang terdiri dari kurva linear turun merupakan jarak dekat antara robot dan halangan yaitu dari 40 sampai dengan 100 cm. Jadi ketika robot mencapai pada nilai 40 cm maka robot akan berhenti. Karena pada jarak itu telah diset kecepatan motor hingga 0 pwm. Sedangkan pada kurva segitiga merupakan jarak menengah antara robot dan halangan yaitu pada nilai 70, 100 dan 130 cm. sedangkan pada kurva linear naik merupakan jarak jauh robot dengan halangan yaitu berkisar antara 100 hingga 160 cm. Pembacaan sensor ultrasonik dimulai pada jarak 160 cm.



Gambar 4.13 *Membership function* pada jarak 30 cm

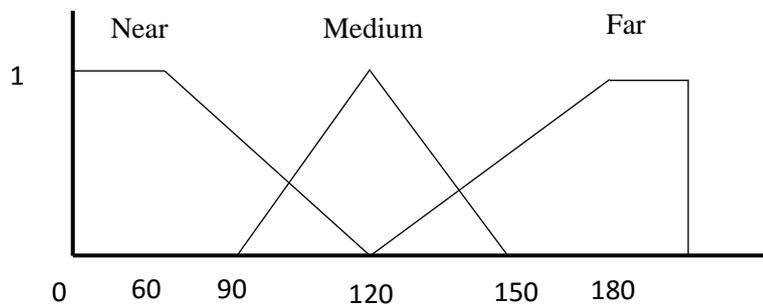
Gambar 4.14 Merupakan *membership function* untuk pembacaan sensor ultrasonik terhadap halangan yang ada di depan kurva yang digunakan pada pengujian ini adalah kurva bahu yang terdiri dari kurva linear turun merupakan jarak dekat antara robot dan halangan yaitu dari 50 sampai dengan 110 cm. Jadi ketika robot mencapai pada nilai 40 cm maka robot akan berhenti karena pada jarak itu telah

diset kecepatan motor hingga 0 pwm. Sedangkan pada kurva segitiga merupakan jarak menengah antara robot dan halangan yaitu pada nilai 80, 110 dan 140 cm. Dan pada kurva linear naik merupakan jarak jauh robot dengan halangan yaitu berkisar antara 110 hingga 170 cm. Pembacaan sensor ultrasonik dimulai pada jarak 170 cm.



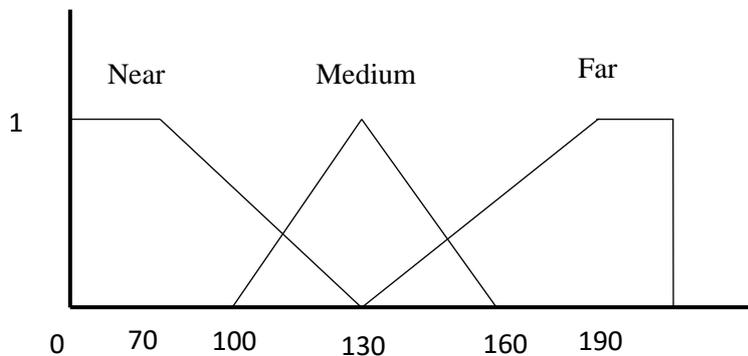
Gambar 4.14 *Membership function* untuk jarak 50 cm

Gambar 4.15 Merupakan *membership function* untuk pembacaan sensor ultrasonik terhadap halangan yang ada di depan kurva yang digunakan pada pengujian ini adalah kurva bahu yang terdiri dari kurva linear turun merupakan jarak dekat antara robot dan halangan yaitu dari 60 sampai dengan 120 cm. Jadi ketika robot mencapai pada nilai 60 cm maka robot akan berhenti karena pada jarak itu telah diset kecepatan motor hingga 0 pwm. Sedangkan pada kurva segitiga merupakan jarak menengah antara robot dan halangan yaitu pada nilai 90, 120 dan 150 cm. Sedangkan pada kurva linear naik merupakan jarak jauh robot dengan halangan yaitu pada rentang nilai antara 120 hingga 180 cm. Dan untuk pembacaan jarak dari sensor ultrasonik dimulai pada rentang nilai 180 cm hingga 60 cm.



Gambar 4.15 *Membership function* untuk jarak 60 cm

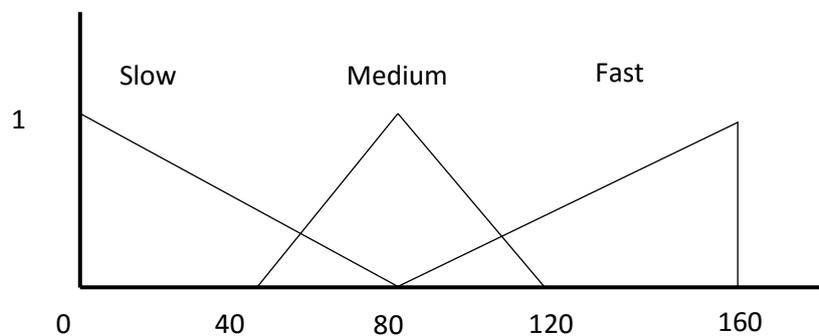
Gambar 4.16 Merupakan *membership function* untuk pembacaan sensor ultrasonik terhadap halangan yang ada di depan kurva yang digunakan pada pengujian ini adalah kurva bahu yang terdiri dari kurva linear turun merupakan jarak dekat antara robot dan halangan yaitu dari 70 sampai dengan 130 cm. Ketika robot mencapai pada nilai 70 cm maka robot akan berhenti karena pada jarak itu telah diset kecepatan motor hingga 0 pwm. Sedangkan pada kurva segitiga merupakan jarak menengah antara robot dan halangan yaitu pada nilai 100, 130 dan 160 cm. Dan pada kurva linear naik merupakan jarak jauh robot dengan halangan yaitu berkisar antara 130 hingga 190 cm. Jarak pembacaan sensor ultrasonik dimulai pada jarak 190 cm.



Gambar 4.16 *Membership function* untuk jarak 70 cm

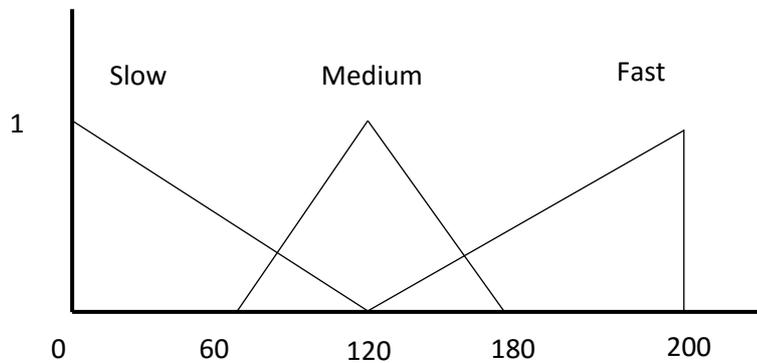
4.3.2 *Membership Function Untuk Kecepatan*

Pada tahap ini akan di gunakan tiga kecepatan maksimal dari 160, 200 dan 240 pwm ketika robot mulai bergerak hal ini bertujuan untuk menyamakan nilai kecepatan pengujian saat tidak menggunakan logika *fuzzy* agar mengetahui tingkat akurasi pada masing-masing metode. *Membership function* untuk kecepatan menggunakan logika *fuzzy* dapat dilihat pada Gambar 4.17.



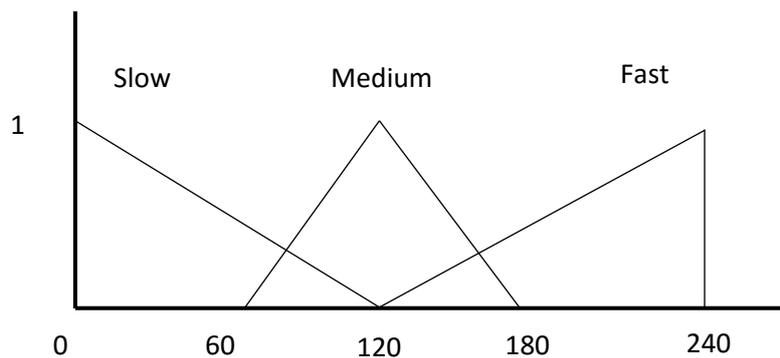
Gambar 4.17 *Membership function* pada kecepatan 160 pwm

Gambar 4.17 dapat dilihat bahwa kecepatan motor maksimal ketika digerakkan sebesar 160 pwm. Kecepatan motor akan semakin perlahan ketika mendekati nilai 0 sehingga motor akan berhenti.



Gambar 4.18 *Membership function* pada kecepatan 200 pwm

Gambar 4.18 dapat dilihat bahwa kecepatan motor maksimal ketika di gerakan sebesar 200 pwm. Kecepatan motor akan semakin perlahan ketika mendekati nilai 0 sehingga motor akan berhenti.



Gambar 4.19 *Membership function* pada kecepatan 240 pwm

Gambar 4.19 dapat dilihat bahwa kecepatan motor maksimal ketika digerakkan sebesar 240 pwm. Kecepatan motor akan semakin perlahan ketika mendekati nilai 0 sehingga motor akan berhenti.

4.3.3 Pengujian Dengan Jarak 20 Cm Menggunakan Logika *Fuzzy*

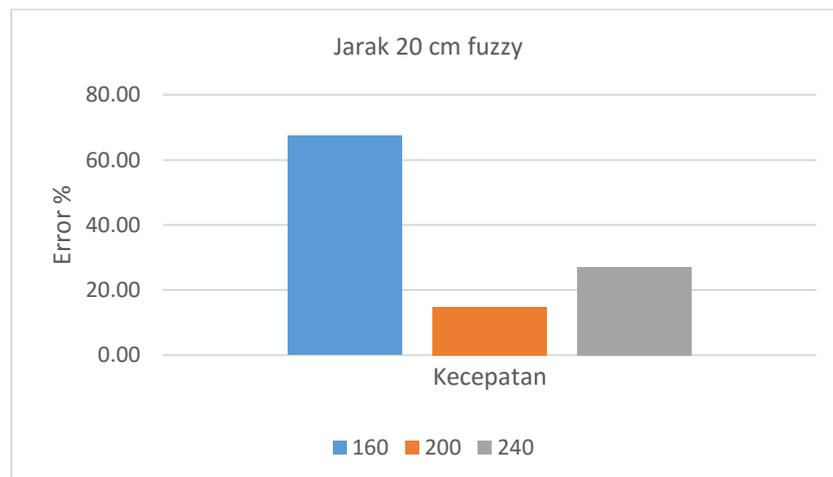
Pada pengujian ini dilakukan sebanyak 10 kali percobaan. Robot dikontrol maju di depan mendekati halangan dengan kecepatan awal 160, 200, dan 240 pwm. Pada setiap percobaan akan dilakukan pengukuran jarak pada saat robot berhenti dengan halangan yang ada di depan sehingga dapat diketahui berapa cm nilai *error* dari pergerakan robot. Kemudian akan diambil nilai rata-rata untuk mengetahui

persentase *error* nya dengan cara $\frac{\text{nilai rata-rata error}}{\text{jarak yang ditentukan}} \times 100\%$. Untuk lebih jelasnya hasil dari pengujian pada jarak 20 cm dapat dilihat pada Tabel 4.8.

Tabel 4.8 Hasil pengujian pada jarak 20 cm

MF 1 Pengujian menggunakan <i>Fuzzy</i>		Nilai error ketika robot menemukan halangan melewati threshold 20 cm										Rata rata (cm)	error %
Jarak	Kecepatan	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10		
20	160	14	14.4	15	12	12.5	14.5	13	13.5	12	14	13.49	67.45
20	200	3	2.5	4	5	2	3	2	2	3	3	2.95	14.75
20	240	5	5.5	6	5	4	5.5	6	6	6	5	5.4	27.00

Pada Tabel 4.8 dapat dilihat hasil pengujian bahwa pada kecepatan 160 pwm rata-rata *error* sebesar 13.49 cm dari jarak yang sebenarnya yaitu 20 cm. Sehingga persentase tingkat *error* sebesar 67.45%. Sedangkan pada kecepatan 200 pwm *error* nya hanya sebesar 14.75% dan pada kecepatan 240 pwm tingkat *error*-nya lebih tinggi dibandingkan dengan kecepatan 200 yaitu sebesar 27 %. Dengan itu bisa diambil kesimpulan bahwa pada kecepatan 200 pwm memiliki nilai *error* terkecil yaitu 14.75%. Untuk lebih jelas grafik perbandingan hasil dari pengujian. Seperti yang ditunjukkan pada Gambar 4.20.



Gambar 4.20 Grafik hasil pengujian pada jarak 20 cm

Dari Gambar 4.20 dapat dilihat perbedaan hasil nilai *error* robot melewati batas yang ditetapkan dimana telah ditanamkan perintah pada mikrokontroler robot berhenti pada jarak 20 cm.

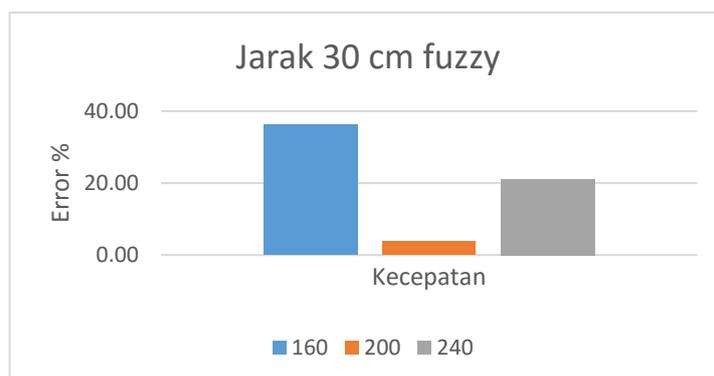
4.3.4 Pengujian Dengan Jarak 30 Cm Menggunakan Logika *Fuzzy*

Pada pengujian ini dilakukan sebanyak 10 kali percobaan pada jarak 30 cm robot dikontrol berjalan lurus mendekati halangan di depannya dengan kecepatan 160, 200, dan 240 pwm. Hasil dari pengujian pada jarak 30 cm ditunjukkan pada Tabel 4.9.

Tabel 4.9 Hasil Pengujian pada jarak 30 cm

MF 2 Pengujian menggunakan <i>Fuzzy</i>		Nilai error ketika robot menemukan halangan melewati threshold 30 cm										Rata rata	% error
Jarak	Kecepatan	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10		
30	160	13	10	10	10.5	14.5	11	10	11.5	9	9	10.85	36.17
30	200	1	0.5	0.5	1	1.5	0.5	2	1.5	1	2	1.15	3.83
30	240	4	5.5	5	6	8	7.5	7	7.5	7	6	6.35	21.17

Pada Tabel 4.9 dapat dilihat hasil pengujian bahwa pada kecepatan 160 pwm rata-rata *error* sebesar 10.85 cm dari jarak yang sebenarnya yaitu 30 cm. Sehingga persentase tingkat *error* sebesar 36.17%. Sedangkan pada kecepatan 200 pwm *error* nya sebesar 3.83% dan pada kecepatan 240 pwm tingkat *error*-nya lebih tinggi dibandingkan dengan kecepatan 200 pwm. Dengan itu bisa diambil kesimpulan bahwa pada kecepatan 200 pwm memiliki tingkat *error* terkecil yaitu 3.83%. Untuk lebih jelas grafik perbandingan hasil dari pengujian dapat dilihat pada Gambar 4.21.



Gambar 4.21 Grafik hasil pengujian pada jarak 30 cm

4.3.5 Pengujian dengan jarak 40 cm menggunakan logika *fuzzy*

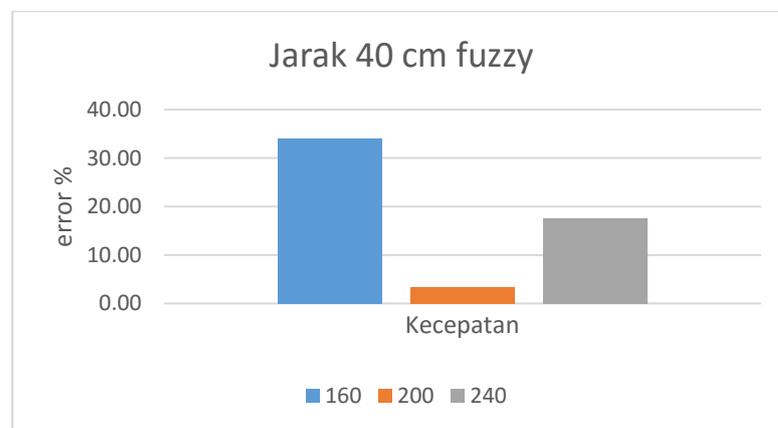
Pada pengujian ini dilakukan sebanyak 10 kali percobaan. Robot dikontrol pada sebuah ruangan maju lurus mendekati halangan di depannya dengan kecepatan awal 160, 200, dan 240 pwm. Pada setiap percobaan akan dilakukan pengukuran

jarak pada saat robot berhenti dengan halangan yang ada di depan sehingga dapat diketahui berapa cm nilai *error* pergerakan robot. Hasil dari percobaan dengan jarak 40 cm dapat dilihat pada Tabel 4.10.

Tabel 4.10 Hasil pengujian menggunakan *fuzzy* pada jarak 40 cm

MF 3 Pengujian menggunakan Fuzzy		Nilai error ketika robot menemukan halangan melewati threshold 40 cm										Rata-rata	Error %
Jarak	Kecepatan	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10		
40	160	14	15	11	15	11	12	14.5	17	15	11.5	13.6	34.00
40	200	1	1	1.5	2	1.5	1	1.5	0	2.5	1.5	1.35	3.38
40	240	6	6.5	5.5	8	8.5	5.5	6.5	8	8	8	7.05	17.63

Pada Tabel 4.8 Dapat dilihat hasil pengujian bahwa pada kecepatan 160 pwm rata-rata *error* sebesar 13.6 cm dari jarak yang sebenarnya yaitu 40 cm. Sehingga persentase tingkat *error* sebesar 34%. Sedangkan pada kecepatan 200 pwm *error*-nya hanya sebesar 3.38% dan pada kecepatan 240 pwm tingkat *error*-nya lebih tinggi dibandingkan dengan kecepatan 240 pwm. Dengan itu bisa diambil kesimpulan bahwa pada kecepatan 200 pwm memiliki tingkat *error* terkecil yaitu 3.38 %. Grafik perbandingan hasil dari pengujian dapat dilihat pada Gambar 4.22.



Gambar 4.22 Grafik hasil pengujian pada jarak 40 cm

4.3.6 Pengujian Dengan Jarak 50 Cm Menggunakan Logika Fuzzy

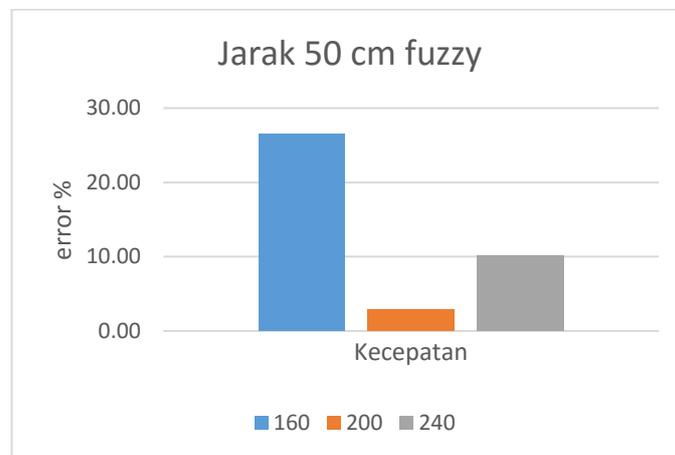
Pada pengujian ini dilakukan sebanyak 10 kali percobaan. Robot dikontrol pada sebuah ruangan maju lurus mendekati halangan di depannya dengan kecepatan awal 160, 200, dan 240 pwm. Pada setiap percobaan akan dilakukan pengukuran jarak pada saat robot berhenti dengan halangan yang ada di depan sehingga dapat

diketahui berapa cm nilai *error* pergerakan robot. Hasil dari percobaan dengan jarak 50 cm dapat dilihat pada Tabel 4.11.

Tabel 4.11 Hasil pengujian menggunakan *fuzzy* pada jarak 50 cm

MF 4 Fuzzy		Nilai error ketika robot menemukan halangan melewati threshold 50 cm										Rata-rata	Error %
Jarak	Kecepatan	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10		
50	160	13.5	13.5	13	13.5	13.5	14	15	12.5	12	12	13.25	26.50
50	200	0	1	1.5	2.5	1	2.5	1	2	1.5	1.5	1.45	2.90
50	240	5	7.5	5	4.5	5	3.5	5	6.5	4	5	5.1	10.20

Pada Tabel 4.11 dapat dilihat hasil pengujian bahwa pada kecepatan 160 pwm rata-rata *error* sebesar 13.25 cm dari jarak yang sebenarnya yaitu 50 cm. Sehingga persentase tingkat *error* sebesar 26.50%. Sedangkan pada kecepatan 200 pwm *error*nya hanya sebesar 2.90 % dan pada kecepatan 240 pwm tingkat *error*nya lebih tinggi yaitu 10.20% dibandingkan dengan kecepatan 200 pwm. Dengan itu bisa diambil kesimpulan bahwa pada kecepatan 200 pwm memiliki tingkat *error* terkecil 2.90 %. Grafik perbandingan hasil dari pengujian pada jarak 50 cm dapat dilihat pada Gambar 4.23.



Gambar 4.23 Grafik hasil pengujian pada jarak 50 cm

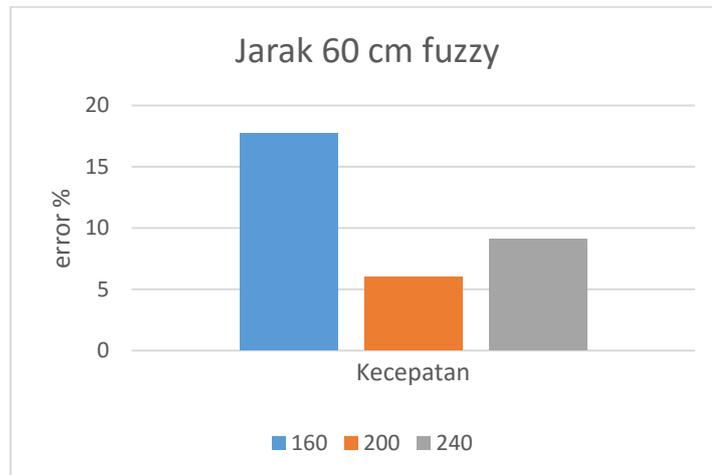
4.3.7 Pengujian Dengan Jarak 60 Cm Menggunakan Logika *Fuzzy*

Pada pengujian ini dilakukan sebanyak 10 kali percobaan pada jarak 60 cm robot dikontrol berjalan lurus mendekati halangan yang ada di depan robot dengan kecepatan 160, 200, dan 240 pwm. Hasil dari pengujian pada jarak 60 cm dapat dilihat pada Tabel 4.12.

Tabel 4.12 Hasil pengujian menggunakan *fuzzy* pada jarak 60 cm

MF 5 Peengujian Menggunakan fuzzy		Nilai error ketika robot menemukan halangan melewati threshold 60 cm										Rata-rata	Error %
Jarak	Kecepatan	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10		
60	160	17	19	17	18	18	17	17	19	18	17	17.7	29.50
60	200	3.5	4	3	4	3	3	3.4	2.5	4.5	5	3.59	5.98
60	240	5	5	5.5	4	7	5	6	6	5.5	5.5	5.45	9.08

Pada Tabel 4.12 dapat dilihat hasil pengujian pada kecepatan 160 pwm rata-rata *error* sebesar 17.5 cm dari jarak yang sebenarnya yaitu 60 cm. Sehingga persentase tingkat *error* sebesar 29.50%. Sedangkan pada kecepatan 200 pwm *error*-nya hanya sebesar 5.98% dan pada kecepatan 240 pwm tingkat *error*-nya lebih tinggi dibandingkan dengan kecepatan 200 pwm. Dengan itu bisa diambil kesimpulan bahwa pada kecepatan 200 pwm memiliki tingkat *error* terkecil 5.98 %. Grafik perbandingan hasil dari pengujian dapat dilihat pada Gambar 4.24.



Gambar 4.24 Grafik hasil pengujian pada jarak 60 cm

4.3.8 Pengujian Dengan Jarak 70 Cm Menggunakan Logika Fuzzy

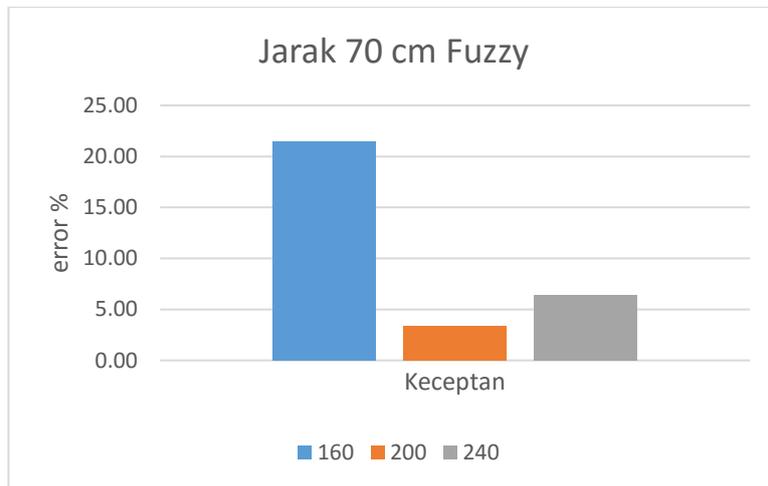
Pada pengujian ini dilakukan sebanyak 10 kali percobaan pada jarak 70 cm robot dikontrol maju lurus mendekati halangan dengan kecepatan 160, 200, dan 240 pwm. Hasil dari pengujian pada jarak 70 cm dapat dilihat pada Tabel 4.13.

Tabel 4.13 Hasil pengujian menggunakan *fuzzy* pada jarak 70 cm

MF 6 Peengujian Menggunakan fuzzy	Nilai error ketika robot menemukan halangan melewati threshold 70 cm	Rata-rata	Error %

Jarak	Kecepatan	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10		
70	160	16	17	18	15	13	14	14	14.5	14.5	14	15	21.43
70	200	2	1.5	1.5	4	4	2	1.5	2	3	2	2.35	3.36
70	240	4	4	4	4.5	4.5	3.5	4	6	6	4.5	4.5	6.43

Pada Tabel 4.13 dapat dilihat hasil pengujian bahwa pada kecepatan 160 pwm rata-rata *error* sebesar 15 cm dari jarak yang sebenarnya yaitu 70 cm. sehingga persentase tingkat *error* sebesar 21.43%. Sedangkan pada kecepatan 200 pwm *error*-nya hanya sebesar 3.36% dan pada kecepatan 240 pwm tingkat *error*-nya lebih tinggi dibandingkan dengan kecepatan 200 pwm. Dengan itu bisa diambil kesimpulan bahwa pada kecepatan 200 pwm memiliki tingkat *error* terkecil 3.36%. Grafik perbandingan hasil dari pengujian terlihat pada Gambar 4.35.



Gambar 4.25 Grafik hasil pengujian pada jarak 70 cm

4.4 Pengujian Kontrol Manual Tanpa *Fuzzy*

Pada pengujian ini dilakukan berdasarkan jarak yang telah digunakan pada pengujian sebelumnya menggunakan logika *fuzzy* pada kecepatan yang sama dan jarak yang sama juga. Diuji sebanyak 10 kali percobaan bertujuan agar mengetahui performa pada masing-masing metode.

4.4.1 Pengujian Dengan Kecepatan 160 Pwm

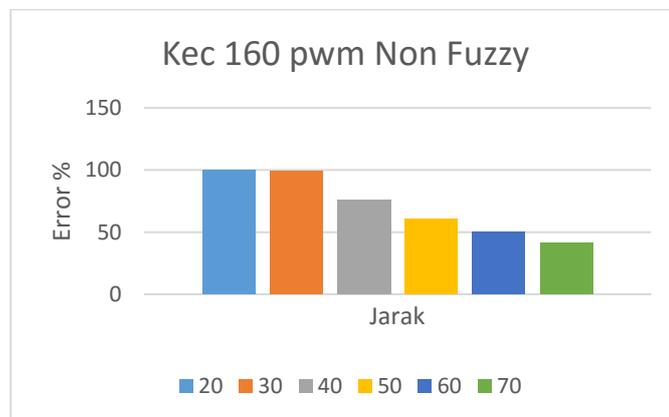
Pada pengujian ini robot dikontrol maju lurus mendekati halangan yang ada di depan dengan kecepatan 160, 200, dan 240 pwm dengan batas jarak robot akan berhenti secara otomatis pada jarak 30 cm. Pada masing-masing jarak dan kecepatan

akan dilakukan sebanyak 10 kali percobaan. Kemudian diambil data pengujian dengan cara mengukur jarak ketika robot berhenti dengan halangan diukur secara manual. Sehingga dapat diketahui berapa cm nilai *error* batas robot berhenti terhadap jarak yang telah ditentukan. Hasil dari pengujian pada kecepatan 160 pwm dapat dilihat pada Tabel 4.14.

Tabel 4.14 Hasil pengujian tanpa logika *fuzzy* dengan kecepatan 160 pwm

Non fuzzy kec 160 pwm	Nilai error (cm) ketika robot menemukan halangan melewati threshold dengan kecepatan 160 pwm											Rata-rata	Error %	
	Kecepatan	Jarak	p1	p2	p3	p4	p5	p6	p7	p8	p9			p10
160	20	20	20	20	20	20	20	20	20	20	20	20	20	100
160	30	29	29	30	30	30	30	30	30	30	30	29.8	99.33	
160	40	27.5	32	31	31	31.5	32	29	30	28.5	30	30.25	75.62	
160	50	30	29.5	29.5	29	29	32	29.5	30	32.5	33	30.4	60.8	
160	60	30.5	29.5	29	31	30.5	33	30.4	30	30	29	30.29	50.48	
160	70	28	29	31.5	29.2	30	29	28.5	29	30	27.5	29.17	41.67	

Pada Tabel 4.14 dapat dilihat hasil dari pengujian pada kecepatan 160 pwm. Pada jarak 20 cm tingkat *error* 100 %, jarak 30 cm 99.3 %, jarak 40 75%, jarak 50 sebesar 60.8%, jarak 60 sebesar 50.4 % dan pada jarak 70 cm sebesar 41.67 %. Dari hasil pengujian dapat disimpulkan bahwa semakin jauh jarak maka hasilnya semakin bagus. Untuk kecepatan gerak robot hasil yang terbaik dari beberapa pengujian adalah pada kecepatan 160 pwm, pada jarak 70 cm dengan persentase nilai *error* terkecil yaitu 41.67%. Grafik dari hasil pengujian dapat dilihat pada Gambar 4.26.



Gambar 4.26 Grafik hasil pengujian pada kecepatan 160 pwm

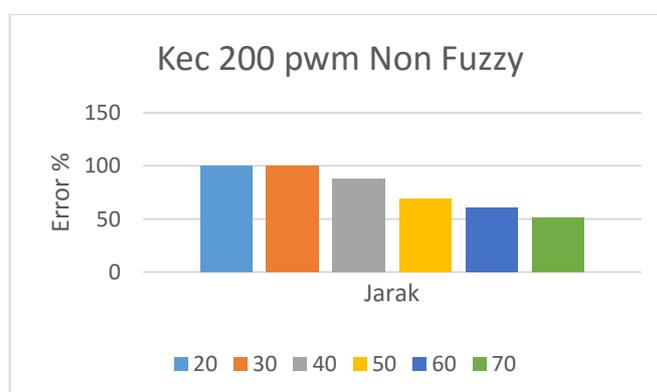
4.4.2 Pengujian Dengan Kecepatan 200 pwm

Pada pengujian ini dilakukan sebanyak 10 kali percobaan sama seperti yang dilakukan pada pengujian menggunakan logika *fuzzy*. Namun pada pengujian ini Berdasarkan 1 kecepatan dan 6 jarak yaitu dari jarak 20 sampai dengan 70 cm. Untuk lebih jelas dapat dilihat pada Tabel 4.15.

Tabel 4.15 Hasil pengujian dengan Kecepatan 200 pwm

Non fuzzy kec 200 pwm		Nilai error ketika robot menemukan halangan melewati threshold dengan kecepatan 200 pwm											
Kec	Jarak	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	Rata- rata	Error %
200	20	20	20	20	20	20	20	20	20	20	20	20	100
200	30	30	30	30	30	30	30	30	30	30	30	30	100
200	40	35	34	36	33	34	33.5	37.5	35.5	36	36	35.05	87.62
200	50	33.5	36	35.5	36	37	35	36	27	36	35	34.7	69.4
200	60	35.5	36.5	38	38.5	36	36	36	36.2	37.5	36.5	36.67	61.11
200	70	35.5	36.5	36	37	35.5	36	35.5	36	37	37	36.2	51.71

Pada Tabel 4.15 dapat dilihat hasil dari pengujian pada kecepatan 200 pwm. Pada jarak 20, 30 cm robot menabrak halangan yang ada di depan hingga nilai persentase *error* mencapai 100 %. Pada jarak 40 cm rata-rata *error* mencapai 35 cm. Jarak 50 cm rata-rata *error* 34.7 cm. Jarak 60 cm rata-rata *error* 36.67 cm. Sedangkan pada jarak 70 cm memiliki nilai *error* sekitar 36.2 cm. Sehingga dapat diambil kesimpulan bahwa hasil terbaik pengujian kontrol robot pada kecepatan 200 pwm yaitu pada jarak 70 cm dengan persentase *error* terkecil 51.71%. Hasil perbandingan pada pengujian pada kecepatan 200 pwm bisa dilihat pada Gambar 4.27.



Gambar 4.27 Grafik hasil pengujian dengan kecepatan 200 pwm

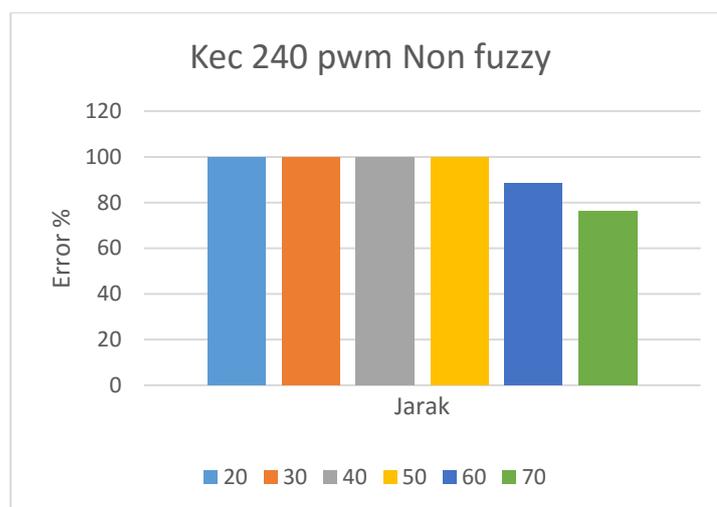
4.4.3 Pengujian Dengan Kecepatan 240 pwm

Pada pengujian ini dilakukan sebanyak 10 kali percobaan sama seperti yang dilakukan pada pengujian sebelumnya. Robot dikontrol maju lurus mendekati halangan yang ada di depan robot dengan kecepatan 240 pwm. Hasil dari pengujian dapat dilihat pada Tabel 4.16.

Tabel 4.16 Hasil Pengujian dengan kecepatan 240 pwm

Non fuzzy kec 200 pwm		Nilai error ketika robot menemukan halangan melewati threshold dengan kecepatan 200 pwm											
Kecepatan	Jarak	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	Rata-rata	Error %
240	20	20	20	20	20	20	20	20	20	20	20	20	100
240	30	30	30	30	30	30	30	30	30	30	30	30	100
240	40	40	40	40	40	40	40	40	40	40	40	40	100
240	50	50	50	50	50	50	50	50	50	50	50	50	100
240	60	51	54	55	50	54	53.5	53.5	52	53	55	53.1	88.5
240	70	52	54.5	55	51	54	53	53	55.5	53	53.5	53.45	76.3

Pada Tabel 4.16 dapat dilihat dari hasil pengujian bahwa pada jarak 20, 30, 40, dan 50 cm robot menabrak halangan yang ada di depan dengan nilai persentase *error* 100%. Ini karena kecepatan motor yang konstan apabila dipacu dengan kecepatan penuh maka robot menabrak halangan. Pada jarak 60 cm rata-rata *error* 53.1 cm sedangkan pada jarak 70 cm rata-rata 53.45 cm. Seperti yang terlihat pada Gambar 4.28.

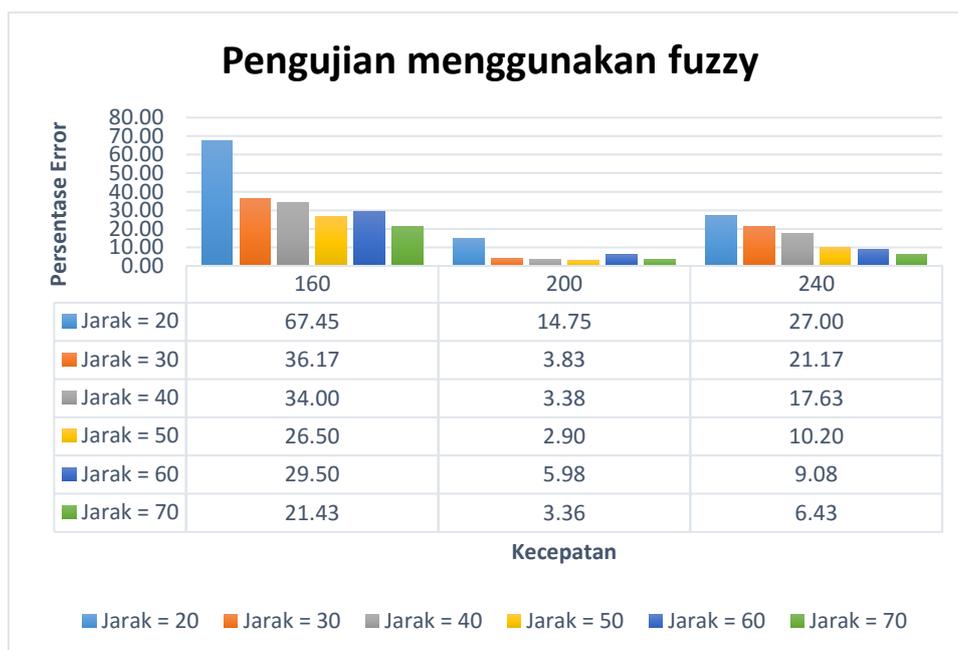


4.28 Grafik pengujian tanpa *fuzzy* kecepatan 240 pwm

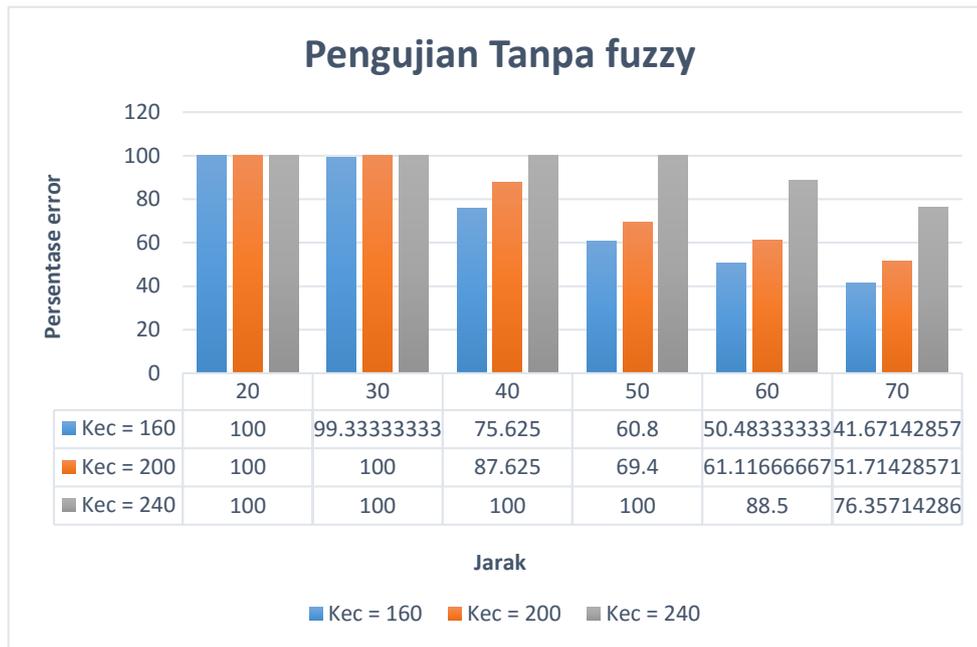
4.5 Analisis Hasil dan Pembahasan

Dari beberapa hasil pengujian kontrol manual maka dapat diambil kesimpulan performa dari masing- masing metode dari hasil pengujian bahwa menggunakan logika *fuzzy* lebih efektif dari segi akurasi. Dari hasil pengujian tanpa menggunakan logika *fuzzy* didapatkan nilai *error* 100 % pada pengujian dengan jarak 20, 30, 40, dan 50 dengan kecepatan 160, 200, dan 240 pwm. Maka dari data grafik yang ditunjukkan pada Gambar 4.29 adalah semakin kecil kecepatan semakin jauh halangan maka nilai *error* semakin kecil yaitu pada kecepatan 160 pwm dengan jarak 70 didapatkan nilai *error* terkecil dari hasil pengujian tanpa menggunakan logika *fuzzy* yaitu sebesar 58.4 %. Jadi dapat dipastikan bahwa menggunakan logika *fuzzy* lebih efisien dari segi akurasi pergerakan robot dibandingkan tanpa menggunakan *fuzzy*. Grafik hasil dari keseluruhan pengujian dapat dilihat pada Gambar 4.29 dan Gambar 4.30.

Dari hasil pengujian otomatis didapatkan beberapa hasil yaitu robot bermanuver sesuai dengan intruksi yang telah diberikan. Namun pada halangan yang miring atau berbentuk segitiga robot menabrak objek seperti pada Gambar 4.10. Karena sensor ultrasonik tidak bisa memantulkan dan mengirim gelombang pada halangan yang berbentuk miring dan objek yang kecil.



Gambar 4.29 Hasil keseluruhan pengujian menggunakan logika *fuzzy*



Gambar 4.30 Hasil keseluruhan penguujian tanpa logika *fuzzy*

[Halaman ini sengaja dikosongkan]

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisis data yang telah dilaksanakan pada bab 4, maka akan dapat diperoleh kesimpulan dan saran sebagai berikut:

1. Aplikasi dapat berkomunikasi dengan robot melalui jaringan *bluetooth* sebagai media komunikasi pengirim perintah kendali pada robot.
2. Dari beberapa pengujian kontrol robot tanpa menggunakan logika *fuzzy* bahwa semakin rendah pwm yang ditanam ke dalam mikrokontroler maka hasil akurasi robot berhenti pada jarak yang telah ditentukan semakin baik. Dan jika pwm nya tinggi maka hasil akurasinya semakin buruk.
3. Dari beberapa kecepatan dan halangan pada saat pengujian maka didapatkan hasil menggunakan logika *fuzzy* yang baik yaitu pada kecepatan 200 pwm dengan jarak 50 cm dengan *error* terkecil sekitar 2.9 %. Sedangkan tanpa logika *fuzzy* hasil terbaik pada kecepatan 160 pwm dengan jarak 70 cm dengan nilai akurasi 58.4 %.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, penulis memberikan saran supaya untuk kedepannya menggunakan sensor kamera sebagai pendeteksi halangan yang dilakukan secara *realtime* atau *streaming*. Sehingga tidak memerlukan sensor jarak dalam mendeteksi objek tetapi langsung menggunakan kamera.

Untuk pengembangan ke depan pengembangan sistem kendali dari robot diharapkan mampu menggunakan GPS (*Global Positioning System*) sehingga dapat mengendalikan pergerakan robot dengan cara memberikan titik koordinat kemana robot tersebut harus bergerak.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- Abdulsada, Hazim. "Underneath Vehicle Inspection Using Fuzzy Logic, Subsumption and OpenCV Library." *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering* 9.1: 102-107.
- Chauhan, Anamika, Sunil Semwal, and Rajneesh Chawhan. "Artificial neural network-based forest fire detection system using wireless sensor network." *2013 Annual IEEE India Conference (INDICON)*. IEEE, 2013.
- Christianto, Peter, Darjat, and Iwan Setiawan. *Pengaturan Kecepatan Motor DC Dengan Adaptive Fuzzy Logic Controller Metode Tuning Output*. Diss. Jurusan Teknik Elektro Fakultas Teknik, Universitas Diponegoro 2011.
- Cho, Jung Tae, and Boo Hee Nam. "A study on the fuzzy control navigation and the obstacle avoidance of mobile robot using camera." *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*. Vol. 4. IEEE, 2000.
- Gibb, Alicia M. *New media art, design, and the Arduino microcontroller: A malleable tool*. Diss. Pratt Institute, 2010.
- Indar Sugiarto, Lauw Lim Un Tung & Mohammad Ismail Rahman, 2008. Implementation of Fuzzy Logic in FPGA for Maze Tracking of a Mobile Robot Based on Ultrasonic Distance Measurement. *Jurnal Teknik Elektro*, 8(2), pp.96–102. Available at: <http://puslit2.petra.ac.id/ejournal/index.php/elk/article/view/17917>.
- Krofitsch, Christoph, et al. "Smartphone driven control of robots for education and research." *Robotics, Biomimetics, and Intelligent Computational Systems (ROBIONETICS), 2013 IEEE International Conference on*. IEEE, 2013.
- Lee, Seok Ju, et al. "Autonomous tour guide robot by using ultrasonic range sensors and QR code recognition in indoor environment." *IEEE International Conference on Electro/Information Technology*. IEEE, 2014.
- Lim, Jongil, et al. "Ultrasonic-sensor deployment strategies and use of smartphone sensors for mobile robot navigation in indoor environment." *IEEE International Conference on Electro/Information Technology*. IEEE, 2014.
- Mamdani, Ebrahim H., and Sedrak Assilian. "An experiment in linguistic synthesis with a fuzzy logic controller." *International journal of man-machine studies* 7.1 (1975): 1-13.
- Meier, Reto. *Professional Android 4 Application Development*. John Wiley & Sons, 2012.

Martin, W., 2009. Autonomous robot obstacle avoidance using a fuzzy logic control scheme. Available at:
http://www.cs.cmu.edu/~wmartin/files/fuzzy_control_robot.pdf

Mohammad, S.H.A., Jeffril, M.A. & Sariff, N., 2013. Mobile robot obstacle avoidance by using Fuzzy Logic technique. *2013 IEEE 3rd International Conference on System Engineering and Technology*, pp.331–335. Available at:
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6650194>.

Susanto, R., Kristanto, Y., Ridwanto, S., & Hisnuaji, D. (2007). Perancangan dan implementasi sensor parkir pada mobil menggunakan sensor ultrasonik. *CommIT (Communication and Information Technology) Journal*, 1(1), 18-29.

Wikipedia, (2016). *About ATI*. [Online] Available at:
https://en.wikipedia.org/wiki/Mobile_robot
[Diakses 22 juli 2016].

Wu, J. et al., 2013. Hand motion-based remote control interface with vibrotactile feedback for home robots. *International Journal of Advanced Robotic Systems*, 10.

Zadeh, Lotfi A. "Fuzzy sets." *Information and control* 8.3 (1965): 338-353.

RIWAYAT HIDUP PENULIS



Tengku Musri lahir pada tanggal 08 Maret 1985 di bengkalis, Riau. Merupakan anak kelima dari delapan bersaudara pasangan Bapak Tengku Ilyas dan Ibu Badariah. Penulis memulai pendidikan di SD Negeri 023 Meskom, lalu melanjutkan di SMP Negeri 08 Teluk Latak. Penulis menempuh jenjang pendidikan di SMAN 03 Bengkalis. Pada tahun 2006 penulis diterima di Program D3 Jurusan Teknik Informasi di Politeknik Bengkalis dan melanjutkan studi di program Lanjut Jenjang (LJ) di Institut Teknologi Bandung (ITB) hingga menyelesaikan studi tahap Diploma IV pada tahun 2013. Setelah menyelesaikan studi di tahap sarjana, penulis mengajukan beasiswa Pra S2-S2 Saintek Dikti dan akhirnya meneruskan pendidikan di Program Magister Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember (ITS) dengan memilih bidang keahlian Komputasi Cerdas dan Visualisasi.

e-mail: tengku.musry@gmail.com