

**SIMULASI MANIPULATOR ROBOT BERGERAK DENGAN
MENGUNAKAN KONTROLER LOGIKA *FUZZY* PADA PERANGKAT
LUNAK COPPELIASIM**

(Skripsi)

**Oleh
Mahesa Anggoro Jati**



**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

ABSTRAK

SIMULASI MANIPULATOR ROBOT BERGERAK DENGAN MENGUNAKAN KONTROLER LOGIKA *FUZZY* PADA PERANGKAT LUNAK COPPELIASIM

Oleh

MAHESA ANGGORO JATI

Dalam industri modern, penggunaan robot semakin berkembang untuk meningkatkan efisiensi dan akurasi dalam berbagai tugas. Salah satu jenis robot yang banyak digunakan adalah robot manipulator bergerak, yang menggabungkan kemampuan *mobile* robot dan manipulator untuk memindahkan objek dengan presisi tinggi. Namun, tantangan utama dalam pengoperasian robot ini adalah memastikan pergerakan yang akurat dalam lingkungan yang dinamis. Salah satu pendekatan yang dapat digunakan untuk mengatasi tantangan ini adalah penerapan kontroler logika *fuzzy*, yang mampu menangani ketidakpastian dalam pergerakan robot. Penelitian ini bertujuan untuk mensimulasikan pengendalian robot manipulator bergerak menggunakan metode *fuzzy* Sugeno yang mempertimbangkan jarak dan kecepatan sebagai variabel input. Metode ini digunakan untuk mengkategorikan akurasi pergerakan robot ke dalam tiga tingkat: rendah, sedang, dan tinggi. Simulasi dilakukan menggunakan perangkat lunak *CoppeliaSim*, yang diintegrasikan dengan *Python*, untuk menguji orientasi posisi robot serta keefektifan kontroler *fuzzy* Sugeno dalam menjalankan tugas pemindahan objek. Hasil pengujian menunjukkan bahwa robot dapat mengidentifikasi dan mengikuti koordinat posisi dengan tingkat akurasi yang baik. Dari 155 sampel titik koordinat, sistem berhasil mencapai akurasi sebesar 84,6% dalam memindahkan enam objek berbentuk persegi panjang dari titik awal ke titik akhir dengan bantuan titik sementara. Dengan hasil ini, metode *fuzzy logic* Sugeno terbukti efektif dalam meningkatkan presisi robot manipulator bergerak dalam simulasi.

Kata Kunci: Robot Manipulator Bergerak, Fuzzy Logic Sugeno, CoppeliaSim, Python

ABSTRACT

SIMULATION OF A MOBILE ROBOT MANIPULATOR USING A FUZZY LOGIC CONTROLLER IN COPPELIASIM SOFTWARE

By

MAHESA ANGGORO JATI

In modern industries, the use of robots is continuously evolving to enhance efficiency and accuracy in various tasks. One of the widely utilized types is the mobile manipulator robot, which combines the mobility of a mobile robot with the precision of a manipulator arm. However, a significant challenge in operating such robots lies in ensuring accurate movement in dynamic environments. One approach to addressing this challenge is the application of fuzzy logic controllers, which effectively handle uncertainties in robotic motion. This study aims to simulate the control of a mobile manipulator robot using the Sugeno fuzzy method, considering distance and speed as input variables. The method categorizes movement accuracy into three levels: low, medium, and high. The simulation is conducted using CoppeliaSim software, integrated with Python, to evaluate the robot's position orientation and the effectiveness of the Sugeno fuzzy controller in executing object transfer tasks. The results indicate that the robot successfully identifies and follows position coordinates with high accuracy. Out of 155 coordinate sample points, the system achieved an accuracy rate of 84.6% in transferring six rectangular objects from the starting point to the endpoint using intermediate points. These findings demonstrate that the Sugeno fuzzy logic method is effective in improving the precision of mobile manipulator robots in simulation.

Keywords: Mobile Manipulator Robot, Fuzzy Logic Sugeno, CoppeliaSim, Python

**SIMULASI MANIPULATOR ROBOT BERGERAK DENGAN MENGGUNAKAN
KONTROLER LOGIKA *FUZZY* PADA PERANGKAT LUNAK COPPELIASIM**

**Oleh
Mahesa Anggoro Jati
Skripsi**

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA TEKNIK**

Pada

**Jurusan Teknik Elektro
Fakultas Teknik
Universitas Lampung**



**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

Judul Skripsi : **SIMULASI MANIPULATOR ROBOT BERGERAK DENGAN
MENGUNAKAN KONTROLER LOGIKA FUZZY PADA
PERANGKAT LUNAK COPPELIASIM**

Nama Mahasiswa : **Mahesa Anggoro Jati**

No. Pokok Mahasiswa : 2015031035

Jurusan : Teknik Elektro

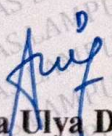
Fakultas : Teknik

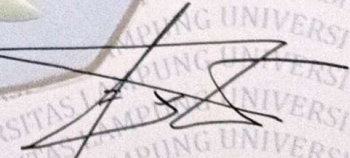
LEMBAR PERSETUJUAN

1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping

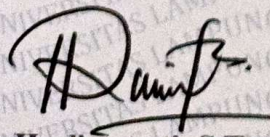

Ir. Anisa Ulya Darajat, S.T., M.T
NIP. 199106102019032024

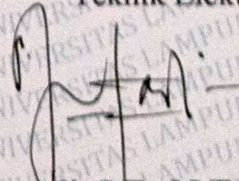

Dr. Eng. FX Arinto Setyawan, S.T., M.T
NIP. 196912191999031002

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Ketua Program Studi
Teknik Elektro


Herlinawati, S.T., M.T.
NIP. 197103141999032001


Sumadi, S.T., M.T.
NIP. 197311042000031001

LEMBAR PENGESAHAN

1. Tim Penguji

Ketua

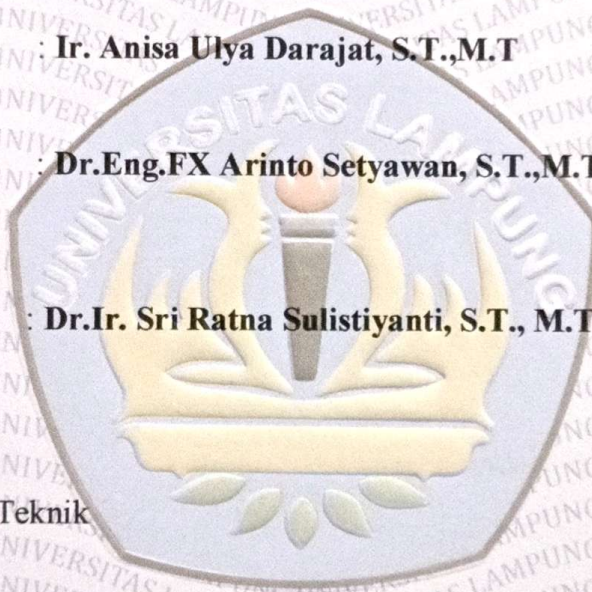
: **Ir. Anisa Ulya Darajat, S.T., M.T**

Sekretaris

: **Dr.Eng.FX Arinto Setyawan, S.T., M.T**

Penguji Bukan
Pembimbing

: **Dr.Ir. Sri Ratna Sulistiyan, S.T., M.T**



2. Dekan Fakultas Teknik

Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc.

NIP 197509282001121002

Tanggal Lulus Ujian Skripsi : 15 Januari 2025

SURAT PERNYATAAN

Dengan ini, saya menyatakan bahwa skripsi ini tidak mengandung karya yang telah dibuat oleh orang lain. Sepanjang pengetahuan saya, tidak ada pendapat atau karya yang ditulis maupun diterbitkan oleh pihak lain, kecuali yang telah dicantumkan secara tertulis dan dirujuk dalam naskah ini sesuai dengan daftar pustaka. Selain itu, saya juga menegaskan bahwa skripsi ini merupakan hasil karya saya sendiri.

Apabila pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan ketentuan hukum yang berlaku.

Bandar Lampung, 10 Februari 2025



Mahesa Anggoro Jati

NPM. 2015031035

RIWAYAT HIDUP



Saya lahir di Bandar Lampung pada 22 Januari 2002 sebagai anak pertama dari empat bersaudara, putra dari Bapak Andi Istanto dan Ibu Mei Eka Susanti. Pendidikan dasar saya tempuh di SDN 2 Sawah Lama dan lulus pada tahun 2014. Kemudian, saya melanjutkan ke SMP Negeri 16 Bandar Lampung dan menyelesaikan pendidikan pada tahun 2017. Selanjutnya, saya bersekolah di SMK 2 Mei Bandar Lampung dan berhasil lulus pada tahun 2020. Pada tahun yang sama, saya diterima sebagai mahasiswa di Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung melalui jalur SBMPTN. Selama masa perkuliahan, saya aktif dalam berbagai kegiatan akademik dan organisasi. Pada tahun 2021, saya bergabung dengan Himpunan Mahasiswa Teknik Elektro (Himatro) Universitas Lampung dan menjabat sebagai Sekretaris Departemen Sosial dan Kewirausahaan. Selain itu, saya melaksanakan Kerja Praktik di PT. Juang Jaya Abdi Alam Lampung Selatan pada Tahun 2023. Pencapaian ini menggambarkan dedikasi saya untuk terus berkembang, memberikan kontribusi di bidang teknologi, serta berperan aktif dalam kehidupan kampus. Saya berharap bisa terus memberikan kontribusi dalam meningkatkan kualitas dan pengembangan di dunia teknologi melalui perjalanan akademis dan keterlibatan dalam berbagai kegiatan organisasi yang saya ikuti. Di semester lima, saya memilih fokus pada bidang Elektronika dan Kendali. Selain itu, saya memiliki pengalaman mengikuti Program Merdeka Belajar Kampus Merdeka (MBKM) dalam skema Pertukaran Mahasiswa Merdeka (PMM) pada tahun 2022. Pada tahun 2023, saya menjalani Kerja Praktik di PT. Juang Jaya Abdi Alam, Lampung Selatan.

PERSEMBAHAN



Atas Rahmat Allah SWT
Dengan shalawat kepada Baginda Nabi Besar

Muhammad SAW

Karya Tulis ini ku persembahkan untuk:

Ayah dan Ibuku Tercinta

Andi Istanto

Mei Eka Susanti

Serta Adik-adik ku Tersayang

Galang Bayu Aji

Tri Bagoes Pamoengkas

Damar Arzaquna Wicaksono

Terima kasih untuk dukungan serta doa yang diberikan
selama ini. Sehingga saya berhasil menyelesaikan karya
ini.



MOTTO

“ Allah tidak akan memberikan beban kepada seseorang melebihi kemampuannya.. ”

(QS. Al-Baqarah 2:286)

“ Allah tidak menjanjikan bahwa hidup akan mudah, tetapi Dia memastikan bahwa di setiap kesulitan terdapat kemudahan. ”

(QS. Al-In syirah, 5-6)

Segala hal yang nyata dan kau inginkan tetapi belum sempat kau raih hanyalah fiksi,
tetapi semua mimpi yang fiksi dan kau usahakan adalah nyata

-Fredrik Ornata

*Always remain humble and kind. When the dreams you've been striving for come true
and your hard work pays off, allow yourself to feel proud, but never lose your
humility and kindness.*

-McGraw

SANWACANA

Alhamdulillah, segala puji dan syukur saya panjatkan ke hadirat Allah SWT atas limpahan rahmat, hidayah, serta pertolongan-Nya, sehingga penulis dapat menyelesaikan laporan skripsi yang berjudul **“Simulasi Manipulator Robot Bergerak dengan Menggunakan Kontroler Logika *Fuzzy* pada Perangkat Lunak Coppeliasim”** tepat waktu. Skripsi ini disusun sebagai salah satu persyaratan untuk menyelesaikan studi dan memperoleh gelar Sarjana Teknik di Program Studi Teknik Elektro, Fakultas Teknik, Universitas Lampung. Shalawat serta salam senantiasa tercurah kepada Nabi Muhammad SAW, beserta para sahabat dan para pengikutnya. Semoga kita semua memperoleh syafaat beliau pada hari akhir nanti. Dalam proses penyusunan skripsi ini, penulis menerima berbagai bentuk dukungan, baik dalam bentuk kontribusi pemikiran maupun dorongan moral dari berbagai pihak. Oleh karena itu, pada kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang telah berperan dalam penyelesaian karya ini.

Oleh karena itu, pada kesempatan ini, penulis ingin menyampaikan rasa terima kasih kepada:

1. Ibu Prof. Dr. Ir. Lusmeilia Afriani, D.E.A., I.P.M., selaku Rektor dari Universitas Lampung.
2. Bapak Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung.
3. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung.
4. Bapak Sumadi, S.T., M.T. selaku Ketua Program Studi Teknik Elektro Fakultas Teknik Universitas Lampung.
5. Ibu Anisa Ulya Darajat, S.T., M.T., selaku Pembimbing Utama tugas akhir, yang telah memberikan bimbingan, bantuan, serta dukungan penuh kepada penulis.
6. Bapak Dr.Eng.FX Arinto Setyawan,S.T.,M.T sebagai Dosen Pembimbing

Pendamping, yang telah banyak membimbing dan membantu penulis selama menjalani kuliah.

7. Bapak Dr. Ing. Ardian Ulvan, S.T., M.Sc., selaku Dosen Pembimbing Pembimbing akademik, yang telah memberikan bimbingan serta banyak membantu penulis selama masa perkuliahan..
8. Seluruh dosen dan staf Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung, atas ilmu, bimbingan, dan dedikasi yang telah diberikan kepada penulis selama menempuh masa studi.
9. Kedua Orangtua dan ketiga adik serta keluarga yang selalu mendoakan dan memberikan semangat serta dukungan sehingga penulis dapat menyelesaikan skripsi ini dengan lancar.
10. Auza Aulia yang selalu memberi dukungan dan semangat serta memberikan pertolongan, canda tawa, dalam setiap proses apapun selama menjadi mahasiswa.
11. Keluarga Besar Asrama Gedung D PMM ITS yang selalu memberi dukungan dan semangat serta memberikan pertolongan, canda tawa, dalam setiap proses apapun selama menjadi mahasiswa.
12. Keluarga Saponta 20 yang telah mendampingi penulis sejak awal perkuliahan hingga selesai. Terima kasih telah menjadi bagian berharga dalam hidup yang akan selalu dikenang dan diceritakan di masa tua.
13. Anna Zakia Latifah sebagai sahabat yang selalu memberi dukungan dan semangat serta memberikan pertolongan, canda tawa, dalam setiap proses apapun selama menjadi mahasiswa.
14. Keluarga besar Kobar 20 yang selalu memberi dukungan dan semangat serta memberikan pertolongan, canda tawa, dalam setiap proses apapun selama menjadi mahasiswa.
15. Yehezkiel Nainggolan, M.Ryandi Fatriq dan M. Faris Abdinantara yang selalu membersamai penulis dalam mengerjakan skripsi ini.
16. Muhammad Arif Rahmanto yang telah memberikan pinjaman laptop untuk mengerjakan skripsi.
17. Keluarga besar Hellios 2020, yang telah memberikan banyak motivasi, nilai-nilai sosial, dan bantuan dalam berbagai hal.
18. Kakak- Kakak Senior 2015-2019 yang telah memberikan motivasi, serta pelajaran hidup.

19. Keluarga Besar Bunker 21 yang telah kebersamai penulis dalam mengerjakan karya tulis ini.
20. Lintang, Tartila , Vicky MM, Peri Pernanda yang telah menemani dan memberikan motivasi.
21. Seluruh pihak yang tidak dapat disebutkan satu per satu, baik yang terlibat secara langsung maupun tidak langsung, yang telah memberikan bantuan dan dukungan kepada penulis dalam penyusunan skripsi ini.
22. Haiqal, Krisna dan Riski Aryun yang telah kebersamai penulis dalam menjalani hidup.

Semoga Allah SWT membalas segala kebaikan dan bantuan yang telah diberikan kepada penulis hingga skripsi ini dapat diselesaikan. Penulis menyadari bahwa laporan ini masih memiliki kekurangan, baik dalam penyusunan maupun dalam pemilihan kata. Oleh karena itu, dengan penuh keterbukaan, penulis menerima kritik dan saran yang membangun untuk perbaikan di masa mendatang. Harapan penulis, semoga skripsi ini dapat memberikan manfaat bagi para pembaca.

Bandar Lampung, 10 Februari 2025
Penulis,

Mahesa Anggoro Jati

DAFTAR ISI

ABSTRAK.....	x
<i>ABSTRACT</i>	xi
LEMBAR PERSETUJUAN	xiii
LEMBAR PENGESAHAN	xiv
SURAT PERNYATAAN	xv
RIWAYAT HIDUP	xvi
PERSEMBAHAN.....	xvii
SANWACANA	ii
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	viii
DAFTAR TABEL.....	i
BAB 1	1
PENDAHULUAN	1
1.1Latar Belakang	1
1.2 Tujuan Penelitian	3
1.3 Manfaat Penelitian	3
1.4 Rumusan Masalah	3
1.5 Batasan Masalah.....	3
1.6 Hipotesis.....	4
1.7 Sistematika Penulisan.....	4
BAB II.....	6
TINJAUAN PUSTAKA	6

2.1 Penelitian Terkait	6
2.2 Robot.....	7
2.2.1 <i>Mobile Robot</i>	8
2.2.2 Manipulator Robot	9
2.3 <i>Fuzzy logic</i> (Logika <i>Fuzzy</i>).....	10
2.4 Coppelia SIM	12
2.5 MATLAB	13
2.6 KUKA Youbot Manipulator	14
2.6.1 Perangkat Keras KUKA Youbot	15
2.6.2 Pemograman KUKA Youbot API.....	19
2.7 <i>Autonomous Robot</i>	20
2.8 <i>Quartenion</i>	22
BAB III	23
METODOLOGI PENELITIAN	23
3.1 Waktu dan Tempat	23
3.2 Alat dan Bahan.....	23
3.3 Tahapan Penelitian	24
3.4 Perancangan Metode <i>Fuzzy logic</i>	27
3.4.1 Menentukan Nilai Variabel	27
3.4.2 Fuzzifikasi.....	28
3.4.3 Pembentukan Aturan <i>Fuzzy</i>	34
3.4.4 Defuzzifikasi	35
3.5 Implementasi <i>Fuzzy logic</i> Controller Sugeno	36
3.5.1 Proses Fuzzifikasi.....	36
3.5.2 Proses Aturan <i>Fuzzy logic</i>	37
3.5.3 Defuzzifikasi	47
BAB IV	60
HASIL DAN PEMBAHASAN	60
4.1 Orientasi Posisi.....	60
4.1.1 Posisi Awal.....	60

4.1.2 Posisi <i>Path Planning</i> (Titik tujuan).....	61
4.2 <i>Task Planning</i>	64
4.3 Hasil Pengujian Orientasi Posisi	65
4.4 Pengujian Sistem Robot Bergerak Manipulator Menggunakan Metode <i>Fuzzy Logic Controller Sugeno</i>	69
BAB V	76
PENUTUP	76
5.1 Kesimpulan	76
5.2 Saran.....	76
DAFTAR PUSTAKA	77
LAMPIRAN.....	81

DAFTAR GAMBAR

Gambar 2. 1 Himpunan <i>fuzzy</i> segitiga	11
Gambar 2. 2 Diagram FLC (<i>Fuzzy logic Controller</i>)	11
Gambar 2. 3 Diagram alir Metode <i>Fuzzy</i>	12
Gambar 2. 4 Penggunaan Software Coppelia SIM	13
Gambar 2. 5 Titik Koordinat Robot KUKA Youbot Manipulator	16
Gambar 2. 6 Lengan KUKA Youbot Manipulator	16
Gambar 2. 7 Pergerakan Lengan Robot menuju posisi awal.	18
Gambar 2. 9 Elemen utama KUKA Youbot API yaitu YouBot Manipulator dan YouBot <i>Base</i>	19
Gambar 2. 9 <i>Road Map</i> Simulasi KUKA Youbot	21
Gambar 3. 1 Tahapan Penelitian.....	24
Gambar 3. 2 Perancangan Sistem Menggunakan <i>Fuzzy Logic</i>	25
Gambar 3. 3 Diagram Alir Metode <i>Fuzzy Logic Controller</i>	26
Gambar 3. 4 Grafik Keanggotaan Kecepatan.....	29
Gambar 3. 5 Grafik Keanggotaan Jarak.....	31
Gambar 3. 6 Grafik Keanggotaan Akurasi.....	32
Gambar 4. 1 Orientasi Posisi Robot KUKA Manipulator Bergerak.....	60
Gambar 4. 2 <i>Path Planning</i> Pada Coppeliasim	61
Gambar 4. 3 Koordinat <i>Path Planning</i> Pada Coppeliasim	62
Gambar 4. 4 Source Code Orientasi Posisi <i>Path Planning</i>	62
Gambar 4. 5 <i>Ocuppancy Grid Map</i>	63
Gambar 4. 6 <i>Task Planning</i> Robot Manipulator Bergerak	64
Gambar 4. 7 Grafik Pengujian Orientasi Posisi	69
Gambar 4. 8 Grafik Pengujian Kecepatan	74
Gambar 4. 9 Grafik Pengujian Jarak.....	75

DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait.....	6
Tabel 2. 2 Main characteristics onboard PC KUKA Youbot Manipulator.....	17
Tabel 3. 1 Alat dan Bahan	23
Tabel 3. 2 Pengujian Sistem	27
Tabel 3. 3 Representasi Data	28
Tabel 3.4 Aturan <i>Fuzzy</i>	34
Tabel 3.5 Perhitungan Aturan <i>Fuzzy</i> Logic Sugeno	37
Tabel 3.6 Hasil Perhitungan Aturan <i>Fuzzy</i> Logic Sugeno.....	40
Tabel 4. 1 Hasil Pengujian Orientasi Posisi.....	65
Tabel 4. 2 Pengujian <i>Fuzzy</i> Logic Controller Sugeno	70
Tabel 4.3 Pembuktian Pengujian Kontroller Logika <i>Fuzzy</i> Sugeno.....	73

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan ilmu pengetahuan dan juga teknologi, khususnya di bidang elektronik, telah berkembang di era modern ini. Perkembangan tersebut berpengaruh terhadap berbagai pekerjaan manusia, di mana semakin banyak tugas yang digantikan oleh teknologi kendali otomatis, terutama dalam sektor industri. Robot adalah perangkat membantu manusia menyelesaikan tugas-tugas yang sulit dijangkau serta mempercepat proses pengerjaannya. Salah satu robot yang dapat dimanfaatkan adalah Manipulator Robot Bergerak [1]. Robot ini merupakan kombinasi mobile robot dan robot manipulator. Mobile robot memiliki kemampuan berpindah dari satu lokasi ke lokasi lain, sementara robot manipulator atau arm robot dapat mengambil serta memindahkan objek, tetapi tidak dapat bergerak ke tempat lain [1]. Maka penelitian ini dibuat untuk merancang simulasi robot manipulator yang digabungkan dengan *mobile* robot. Dalam perancangan robot, sensor merupakan salah satu komponen yang sering digunakan. Sensor ini berfungsi sebagai input bagi robot, seperti sensor jarak dan kamera. Sensor jarak digunakan pada mobile robot dan end effector untuk mendeteksi jarak antara robot dengan objek di sekitarnya. Sementara itu, kamera berperan dalam mendeteksi warna pada suatu objek [2]. Pergerakan robot memerlukan metode perhitungan tertentu guna mengontrol dan mengoordinasikan beberapa motor servo pada lengan robot agar dapat bergerak menuju posisi yang diinginkan [4]. Tujuan utama dari sistem ini adalah memastikan ujung lengan robot (*end-point*) dapat mencapai posisi target yang telah ditetapkan. Robot manipulator ialah salah satu jenis robot yang paling umum digunakan dalam industri dan sering disebut sebagai robot lengan.

Robot ini terdiri dari rangka (*link*) dan sendi (*joint*) [2]. Robot manipulator yang meniru pergerakan lengan manusia dikenal sebagai lengan yang diartikulasikan. Semua sendi pada robot manipulator bersifat rotasi, sehingga gerakannya berbeda dari lengan manusia, karena robot memiliki kebebasan bergerak yang lebih luas dibandingkan dengan gerakan alami manusia [3]. *Autonomous* robot merupakan robot yang dapat beroperasi secara mandiri tanpa memerlukan kendali langsung dari manusia [1]. Robot ini berfungsi berdasarkan sensor yang dimilikinya dan mampu mengambil keputusan sendiri untuk menyelesaikan tugas dalam lingkungan kerjanya. Di dunia nyata, lingkungan kerja robot bersifat dinamis, terus berubah, dan tidak terstruktur. Membuat model lingkungan yang tidak terstruktur menjadi tantangan yang kompleks, bahkan memperoleh model matematis yang akurat untuk kondisi seperti ini hampir tidak mungkin dilakukan. Oleh karena itu, untuk mengembangkan *autonomous mobile robot* yang dapat beroperasi dalam lingkungan yang dinamis dan tidak terstruktur, diperlukan metode yang adaptif serta mampu belajar secara mandiri [5].

Tujuan dari dilakukannya penelitian ini yaitu merancang sebuah *mobile robot* yang dipadukan dengan manipulator robot menggunakan *kontroller logika fuzzy*. Jenis robot *Mobile Manipulator* ini bergerak secara otonom atau *autonomous* menjadikannya robot yang sangat membantu pekerjaan manusia terlebih lagi di dunia industri. Robot ini beroperasi secara *autonomous* untuk memindahkan objek dari posisi awal ke posisi akhir, dengan menggunakan pengaturan *Task Planning* dan juga perhitungan menggunakan *quaternion* pada model matematika yang akan dijalankan oleh Manipulator Robot Bergerak.

1.2 Tujuan Penelitian

Adapun tujuan penelitian ini adalah sebagai berikut :

1. Merancang system kendali Manipulator Robot Bergerak dengan menggunakan kontroler logika *fuzzy* pada perangkat lunak CoppeliaSIM berdasarkan *Task Planning*.
2. Merancang sistem pelacakan koordinat posisi dan orientasi robot pada suatu ruangan simulasi.

1.3 Manfaat Penelitian

Adapun manfaat penelitian ini adalah merancang sistem kendali manipulator robot bergerak dengan menggunakan software CoppeliaSIM dan juga dibuat dengan sistem yang lebih optimal dan stabil menggunakan metode logika *fuzzy* yang dapat menyesuaikan lingkungan sekitar yang berubah-ubah.

1.4 Rumusan Masalah

Berdasarkan permasalahan yang telah diidentifikasi, penelitian ini merumuskan masalah dalam merancang kinematika Manipulator Robot Bergerak serta menentukan Task Planning untuk mensimulasikan pergerakan robot menggunakan kontroler logika fuzzy pada perangkat lunak CoppeliaSim dan Matlab, dengan tujuan memastikan bahwa robot dapat beroperasi secara stabil dan mengikuti Task Planning yang telah dibuat.

1.5 Batasan Masalah

Adapun pembatasan masalah dalam penelitian ini adalah seperti diuraikan dibawah ini:

1. Tidak membahas dinamika robot dan hanya membahas kinematika robot.
2. Mengembangkan *Task Planning* untuk Manipulator Robot Bergerak dalam memindahkan 8 buah objek dari posisi awal ke posisi akhir dengan bantuan titik sementara.

1.6 Hipotesis

Mendapatkan hasil output dari Manipulator Robot Bergerak menggunakan kontroller logika *fuzzy* dapat dinyatakan hasilnya stabil dan berjalan sesuai dengan *Task Planning* yang telah dibuat.

1.7 Sistematika Penulisan

Agar memudahkan melakukan penulisan serta pemahaman tentang tugas akhir yang di buat, maka tulisan akan dibagi pada lima bab, yaitu :

BAB I PENDAHULUAN

Pada bab pendahuluan dilengkapi dengan latar belakang, tujuan , perumusan masalah, batasan masalah, manfaat, hipotesis, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini berisi tentang teori singkat tentang robot manipulator, metode *Fuzzy logic Controller* , Software Coppelia SIM dan kinematika Manipulator Robot Bergerak . Dalam bab tinjauan pustaka dijelaskan mengenai penelitian sebelumnya mengenai Manipulator Robot Bergerak dan Penerapan Kontroler Logika *Fuzzy*.

BAB III METODE PENELITIAN

Berisi rancangan model dan *Task Planning* Manipulator Robot Bergerak serta mensimulasikannya pada software Coppelia SIM, membuat model dengan menggunakan kontroler logika *fuzzy* pada software matlab, langkah yang akan dilaksanakan, penentuan dari spesifikasi rangkaian, blok diagram sebuah rangkaian, cara kerjanya, dan penjelasan dari bagian blok diagram.

BAB IV HASIL DAN PEMBAHASAN

Pada bab hasil serta pembahasan ada hasil penelitian serta analisis hasil penelitian yang didapat.

BAB V PENUTUP

Pada bab ini berisi kesimpulan yang diperoleh pada hasil serta saran pada pengembangan penelitian ini lebih lanjut.

DAFTAR PUSTAKA

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Adapun penelitian sebelumnya yang mendasari penelitian ini ialah sebagai berikut.

Tabel 2. 1 Penelitian Terkait.

No	Judul	Nama Penulis	Kelebihan/Kekurangan
1	Implementasi Metode <i>Fuzzy Logic</i> Untuk Sistem Pengereman Robot Mobile Berdasarkan Jarak dan Kecepatan	Hardin Syah Nasution, Akhmad Jayadi, Rickendry (2022).	Menggunakan sensor ultrasonik HC-SR04 dan sensor encoder rotary yang memiliki keterbatasan dalam mendeteksi jarak dan G8 sehingga dapat terjadi kesalahan dalam pengukuran
2	Rancang Bangun Kendali Kecepatan Berdasarkan Jarak Pada Mobile Robot Menggunakan Metode <i>Fuzzy Logic</i>	Erwin Susanto, Porman Pangaribuan, Ronaldo G P Hasibuan Nur (2015).	Sensor ultrasonic HC-SR04 yang digunakan tidak terlalu presisi. Hasil deteksi sensor pada jarak hambatan lebih dari 100 cm menunjukkan tingkat ketidakpresisian yang tinggi.
3	Implementasi Logika <i>Fuzzy</i> Pada Robot Penjajak Dinding	Nurul Esti Khomariah, Imaru'ul Qais (2019).	Penelitian ini hanya menggunakan sensor Sonar Range Finder (SRF) sebagai sensor utama, yang memiliki keterbatasan dalam mendeteksi jarak dan kondisi lingkungan yang tidak pasti
4	Desain Robot Pengikut Manusia Sederhana Dengan <i>Fuzzy Logic Controller</i>	Tresna Dewi, Yurni Oktarina, Pola Risma, Sari Kartini (2019).	Penggunaan sensor akustik sebagai bagian dari sistem. Sensor akustik untuk mendeteksi suara atau sinyal akustik yang berasal dari manusia atau lingkungan. Data dari sensor ini kemudian diproses oleh <i>Fuzzy Logic Controller</i> untuk menentukan arah gerakan robot, memastikan robot tetap mengikuti manusia secara optimal.

Beberapa tahun terakhir, berbagai penelitian membahas mengenai perancangan mengenai robot manipulator maupun robot navigasi (*mobile robot*) menggunakan beberapa metode. dalam rentang waktu 2003-Sekarang. Pada Tabel 2.1 adalah beberapa artikel tersebut digunakan sebagai referensi.

2.2 Robot

Secara umum, robot dapat didefinisikan sebagai sebuah sistem yang terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*) yang mampu menjalankan tugas tertentu yang biasanya dilakukan oleh manusia. Robot dikembangkan oleh manusia untuk membantu, bahkan menggantikan, tugas yang memerlukan tingkat ketelitian tinggi atau melibatkan risiko besar. Salah satu jenis robot yang dikembangkan adalah robot pendeteksi objek berbasis sensor kamera, yang dirancang dengan kemampuan visual untuk mendeteksi warna objek [8]. Robot ini mengamati objek berupa benda, kemudian mengolah data yang diperoleh, khususnya informasi mengenai warna. Berdasarkan data tersebut, robot akan melakukan tindakan tertentu sesuai dengan program yang telah ditentukan sebelumnya oleh manusia [7]. Robot dengan kemampuan seperti ini lebih dikenal sebagai *robot vision* [9].

Permasalahan utama dalam pengembangan *robot vision* adalah *computer vision*. *Computer vision* bertujuan untuk menganalisis dan mengambil keputusan yang berguna mengenai objek fisik serta lingkungan sekitar berdasarkan gambar yang diperoleh dari sensor. Seperti yang dikemukakan dalam referensi [9], "Tujuan dari penglihatan komputer (*computer vision*) adalah mengembangkan algoritma yang dapat menerima gambar pada masukan yang menghasilkan interpretasi yang menjelaskan objek apa saja yang ada, dalam posisi atau orientasi seperti apa, serta memberikan informasi mengenai hubungan spasial tiga dimensi antara objek-objek tersebut." Dengan kata lain, tujuan utama dari *computer vision* adalah mengembangkan algoritma yang dapat mengolah gambar sebagai masukan, kemudian menghasilkan interpretasi mengenai objek yang terdeteksi, termasuk posisi serta hubungan spasial tiga dimensi antar objek.

Robot pendeteksi objek berbasis sensor kamera adalah sistem otomatis yang menggabungkan berbagai tahapan dalam persepsi visual, seperti pengambilan

gambar, pengolahan citra, identifikasi objek, dan proses pengambilan keputusan. Robot ini terdiri dari komponen perangkat keras, termasuk struktur mekanik dan aktuator berupa motor servo, serta perangkat lunak yang bertugas mengolah data dari sensor kamera untuk mengendalikan aktuator. Sensor kamera berfungsi dalam mendeteksi warna objek dan mengubah data visual menjadi format digital. Selanjutnya, data tersebut diproses oleh unit pemrosesan pusat (CPU) robot untuk mengatur pergerakan aktuator [10]. Saat sensor kamera menangkap gambar objek, citra tersebut diubah menjadi format digital. Dalam konsep digital, gambar terdiri dari kumpulan titik-titik (piksel) yang tersusun dalam bentuk matriks, di mana setiap nilai piksel merepresentasikan tingkat kecerahan atau derajat keabuan (*gray-scale*). Gambar dengan kedalaman warna 8-bit memiliki 256 tingkat kecerahan. Pada gambar berwarna, setiap piksel memiliki nilai derajat keabuan yang terpisah untuk komponen warna merah (R), hijau (G), dan biru (B) [11]. Jika masing-masing komponen warna R, G, dan B memiliki kedalaman 8-bit, maka satu piksel dalam gambar berwarna dinyatakan dalam $(8+8+8) = 24$ bit, atau setara dengan 2^{24} tingkat variasi warna. Dalam proses deteksi atau pengenalan objek, diperlukan pemisahan area tertentu dalam sebuah citra dengan tingkat ketepatan yang tinggi. Proses ini dikenal sebagai segmentasi dan memiliki peran krusial dalam meningkatkan akurasi hasil deteksi maupun pengenalan objek. Dengan segmentasi, citra dibagi menjadi beberapa bagian atau segmen yang lebih sederhana dan bermakna, sehingga dapat dianalisis lebih lanjut. Teknik ini digunakan untuk mengekstrak informasi dari citra, seperti identifikasi komponen mesin, deteksi manusia, serta pencarian citra dengan karakteristik serupa. Secara umum, terdapat beberapa pendekatan yang sering digunakan dalam segmentasi citra, di antaranya berbasis intensitas, warna, dan bentuk. Segmentasi gambar merupakan proses pemisahan objek-objek dalam suatu gambar yang telah dipilih, sehingga memungkinkan analisis yang lebih spesifik dan terarah.

2.2.1 Mobile Robot

Mobile Robot adalah robot yang memiliki kemampuan untuk bergerak dan berpindah tempat. Struktur dari Mobile Robot dirancang agar dapat melakukan

perpindahan posisi secara dinamis. Adapun sistem yang membentuk robot mobile terdiri dari beberapa komponen utama, yaitu sebagai berikut.

- a. Sistem pengendali merupakan kombinasi antara algoritma pemrograman dan komponen mekanik yang dirancang untuk memberikan perintah langsung kepada robot agar bergerak sesuai kondisi input serta umpan balik yang diterima.
- b. Sensor berfungsi untuk mendeteksi kondisi lingkungan sekitar dan memberikan informasi sebagai umpan balik bagi sistem pengendali robot.
- c. Aktuator, yang terhubung dengan bagian mekanik robot, berperan sebagai penggerak utama yang membentuk struktur fisik robot.

Mobile robot dapat dikategorikan menjadi dua jenis utama, yaitu robot beroda dan robot berkaki. Perbedaan utama di antara keduanya terletak pada sistem pergerakannya, terutama pada bagian mekanik yang terhubung langsung dengan aktuator sebagai komponen pengendali utama. Pada robot beroda, aktuator berperan dalam menggerakkan roda sebagai sistem utama mobilitasnya, sedangkan pada robot berkaki, aktuator bekerja bersama mekanisme pergerakan yang dirancang menyerupai langkah kaki. Dalam perancangan mobile robot beroda, bentuk fisik dan struktur robot disesuaikan dengan sistem gerakannya yang berbasis roda [13]. Perubahan posisi yang terjadi pada robot beroda tergolong dalam kategori gerak *non-holonomic*.

2.2.2 Manipulator Robot

Manipulator Robot merupakan robot yang dirancang khusus untuk melaksanakan tugas-tugas tertentu, memungkinkan robot tersebut untuk memanipulasi objek. Robot jenis ini banyak diterapkan dalam berbagai bidang, seperti industri, pertanian, dan kesehatan. Struktur manipulator robot terdiri dari beberapa komponen utama, yaitu:

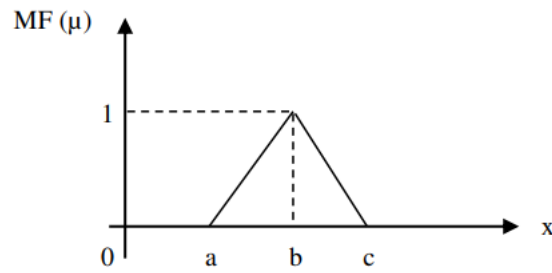
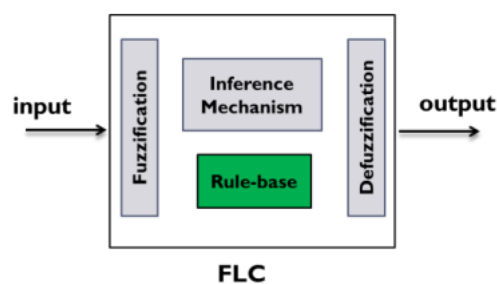
- a. Sistem pengontrol, yang berfungsi sebagai pengendali pergerakan robot. Komponen ini merupakan perpaduan antara algoritma pemrograman dan rangkaian elektronik berbasis mikroprosesor. Fungsinya adalah untuk

mengirimkan perintah langsung kepada robot berdasarkan masukan yang diterima agar dapat menjalankan tugas tertentu.

- b. Manipulator, yaitu bagian mekanik yang bertugas untuk memindahkan, mengangkat, serta melakukan manipulasi terhadap objek atau benda kerja..
- c. Sistem sensor, yang berperan dalam mendeteksi dan memberikan informasi mengenai kondisi manipulator guna meningkatkan ketepatan gerakan robot.
- d. Sistem aktuator berperan sebagai komponen yang menggerakkan robot secara fisik.

2.3 Fuzzy logic (Logika Fuzzy)

Fuzzy Logic Controller (FLC) atau pengendali logika *fuzzy* merupakan salah satu penerapan logika *fuzzy* dalam sistem kendali. FLC telah banyak diterapkan pada berbagai sistem dinamis, baik yang sederhana maupun yang kompleks. Salah satu keunggulan utama dari FLC adalah kemampuannya untuk beroperasi tanpa memerlukan model matematis dari plant yang dikendalikan. Proses pengambilan keputusan dalam FLC (Fuzzy Logic Controller) didasarkan pada aturan-aturan yang telah ditetapkan sebelumnya dalam sistem pengendali selama proses kontrol berlangsung. FLC, yang merupakan singkatan dari pengendali logika fuzzy, adalah sistem kendali yang mengintegrasikan konsep himpunan fuzzy dalam perancangannya. FLC terdiri dari tiga tahap utama, yaitu fuzzifikasi, inferensi, adalah tahap pertama yang bertujuan untuk mengubah nilai tegas (crisp) variabel menjadi bentuk fuzzy [18]. Nilai fuzzy yang dihasilkan kemudian digunakan sebagai input dalam proses inferensi. Pada tahap inferensi, keputusan dibuat berdasarkan input yang ada dengan mengacu pada aturan logika yang telah ditentukan. Terakhir, nilai keluaran dari inferensi yang masih dalam bentuk fuzzy akan diubah kembali menjadi nilai tegas melalui proses defuzzifikasi

Gambar 2. 1 Himpunan *fuzzy* segitigaGambar 2. 2 Diagram FLC (*Fuzzy logic Controller*)

Dalam proses defuzzifikasi, terdapat berbagai metode yang dapat diterapkan, salah satunya adalah metode Takagi-Sugeno-Kang (TSK). Metode ini memiliki kemudahan dalam penerapannya dan dapat diimplementasikan dalam bahasa pemrograman karena menggunakan himpunan singleton pada variabel keluaran. Persamaan fuzzifikasi dapat dijelaskan melalui Persamaan 2.1:

$$\mu_{\text{Sedang}}(X) = \begin{cases} 0 & \text{jika } x < a \text{ atau } x > c, \\ \frac{x-a}{b-a} & \text{jika } a \leq x < b \\ \frac{c-x}{c-b} & \text{jika } b \leq x \leq c \end{cases} \quad (2.1)$$

Keterangan:

a,b,c = Koefisien fungsi keanggotaan

Nilai keluaran crisp diperoleh dengan mengalikan derajat keanggotaan nilai ke-i dengan nilai keluaran dari variabel ke-i. Simulasi dilakukan untuk mengevaluasi kinerja sistem yang telah dirancang sebelum diimplementasikan pada

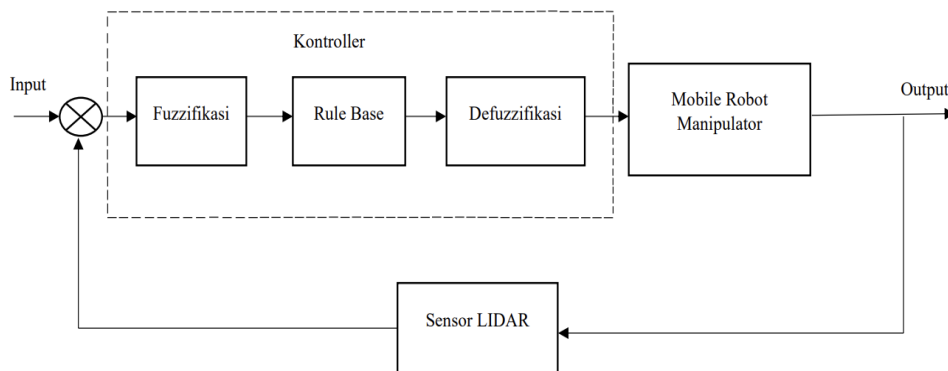
perangkat keras. Adapun persamaan untuk penentuan titik pusat dari daerah fuzzy dilakukan dengan Persamaan 2.2

$$Crisp = \frac{Total\ Tertimbang}{Total\ Bobot} \quad (2.2)$$

$$Crisp = \frac{\sum_{i=1}^{25} (Z_i \times a_i)}{\sum_{i=1}^{25} a_i}$$

Hasil dari simulasi ini juga berfungsi sebagai referensi serta bahan perbandingan dengan hasil pengujian pada perangkat fisik. Dalam sistem ini, pengguna dapat merancang fungsi keanggotaan untuk variabel input dan output sesuai dengan metode defuzzifikasi yang dipilih. Kemudian, aturan dasar dapat ditambahkan melalui menu *Edit Rules*, sementara hasil simulasi dapat dilihat melalui menu *View Rules*.

Adapun diagram alir dari penggunaan *fuzzy* terlihat pada Gambar 2.3

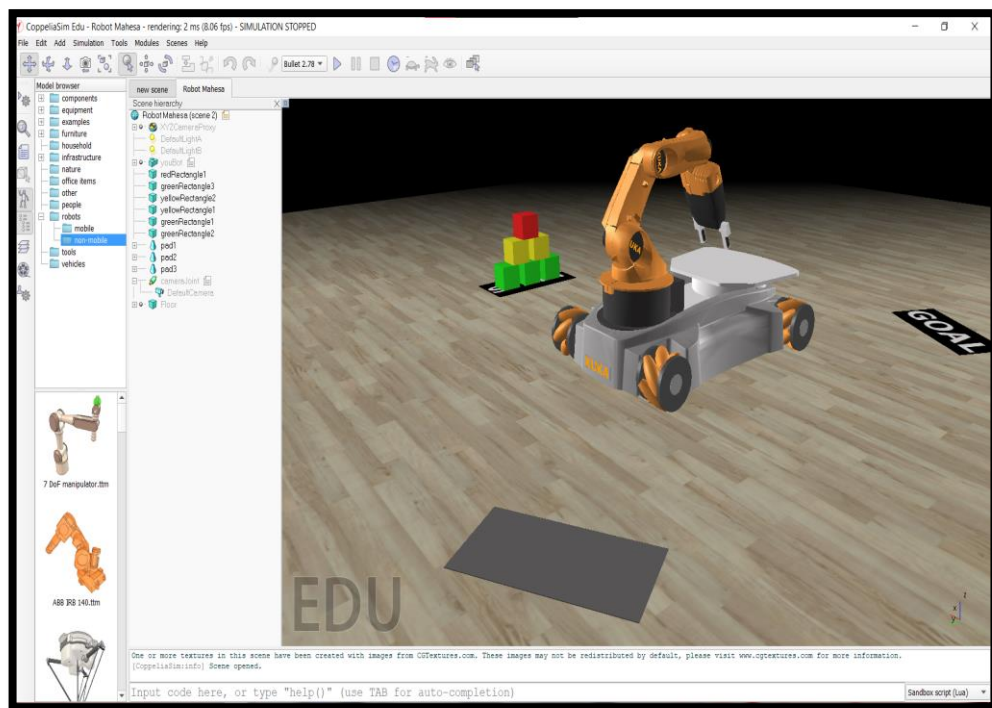


Gambar 2.3 Diagram alir Metode *Fuzzy*

2.4 Coppelia SIM

CoppeliaSim, yang sebelumnya dikenal sebagai V-REP, adalah simulator robotika yang dilengkapi lingkungan yang terintegrasi menggunakan arsitektur kontrol yang terdistribusi. Setiap pada objek atau model dalam simulator ini dapat dikendalikan secara independen melalui tertanam, plugin, node ROS, klien API jarak jauh. Fitur ini membuat CoppeliaSim sangat fleksibel dan cocok untuk

berbagai aplikasi multi-robot. Pengontrol dapat dikembangkan menggunakan berbagai bahasa pemrograman, seperti C, C++, Python, Java, Lua, Matlab, atau Octave. CoppeliaSim banyak digunakan dalam pengembangan algoritma cepat, simulasi otomatisasi industri, pembuatan prototipe dan verifikasi sistem, pendidikan di bidang robotika, pemantauan jarak jauh, analisis keselamatan, penerapan sebagai kembar digital, serta berbagai aplikasi lainnya. Tampilan dari software Coppelia SIM terlihat pada Gambar 2.4



Gambar 2. 4 Penggunaan *Software Coppelia SIM*

2.5 MATLAB

MATLAB, yang merupakan kepanjangan MATrix LABoratory, meningkatkan efisiensi penggunaan perangkat lunak matriks yang sebelumnya dikembangkan oleh LINPACK (sistem paket linear) dan EISPACK. MATLAB memiliki berbagai kegunaan, seperti untuk Pemodelan dan simulasi, pengembangan algoritma, akuisisi serta analisis data, eksplorasi, pembuatan prototipe, komputasi dan matematika, visualisasi, serta pengembangan aplikasi, termasuk antarmuka pengguna (GUI). MATLAB dilengkapi dengan berbagai fungsi matematika umum yang sering digunakan dalam bidang matematika, yang sebagian besar dirancang untuk menyerupai penulisan matematis langsung.

MATLAB merupakan bahasa pemrograman yang digunakan dalam sistem interaktif, di mana elemen data utamanya berupa *array*. Tanpa perlu mendefinisikan dimensi secara manual, MATLAB memungkinkan Solusi untuk berbagai permasalahan dalam komputasi teknik, terutama yang berhubungan dengan operasi matriks dan vektor. Selain itu, program dapat ditulis dalam bahasa lain seperti C dan Fortran jika diperlukan.

Berikut ini merupakan jendela yang tersedia pada *software* MATLAB:

- a. *Window Utama* ialah jendela utama yang menyatukan keseluruhan lingkungan kerja yang tersedia di MATLAB, berfungsi untuk tempat *docking form* lainnya.
- b. *Launch Pad window* berfungsi untuk panduan bagi pengguna untuk memilih opsi dari fungsi atau toolbox yang ada di MATLAB.
- c. *Workspace window* berfungsi untuk navigasi bagi pengguna untuk menampilkan informasi variabel aktif di workspace saat itu. Hal ini bertindak sebagai ruang penyimpanan yang menyimpan semua variabel serta perintah digunakan selama sesi penggunaan.
- d. *Current Directory window* berfungsi untuk *browser* direktori aktif, mirip dengan fungsi *window explorer*.

2.6 KUKA Youbot Manipulator

KUKA youBot di desain dengan antarmuka terbuka penuh dan memungkinkan pengembang mengakses sistem di hampir semua tingkat kontrol perangkat keras. Kemudahan dilengkapi dengan antarmuka pemrograman aplikasi (KUKA youBot API), dengan antarmuka dan pembungkus untuk kerangka kerja robot terkini seperti ROS (*Robot Operating System*) atau OROCOS (*Open Robot Control Software*), dengan simulasi sumber terbuka di Gazebo dan dengan beberapa contoh kode yang menunjukkan cara memprogram KUKA youBot.

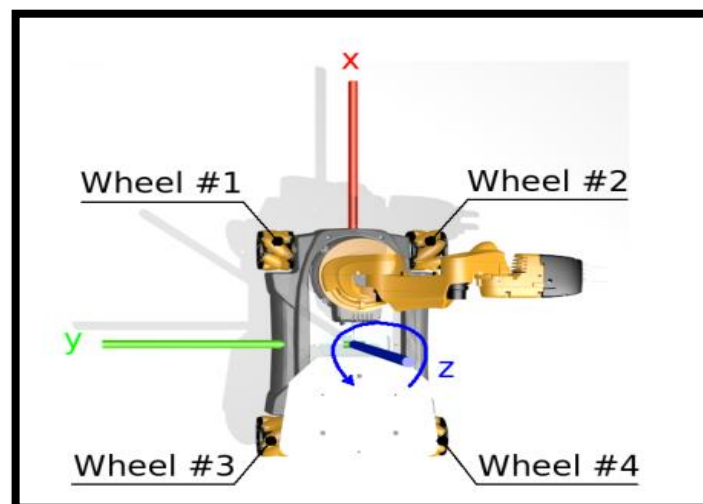
KUKA youBot *omni-directional mobile* platform terdiri dari sasis robot, empat roda mecanum, motor, tenaga dan papan PC onboard. Pengguna dapat menjalankan program di board ini, atau mengendalikannya dari komputer jarak jauh. Platform ini dilengkapi dengan stik *Live-USB* dengan Ubuntu Linux yang sudah diinstal sebelumnya dan driver untuk perangkat kerasnya. Setiap platform

mobile robots dilengkapi dengan bodi aluminium yang kokoh, sistem penggerak yang seimbang (diferensial dua roda dengan kastor atau skid-steer empat roda), motor DC *reversibel*, kontrol motor dan elektronik drive, *encoder* gerak resolusi tinggi, dan daya baterai, semuanya dikelola oleh onboard mikrokontroler dan perangkat lunak server. Pengembangan perangkat lunak mencakup fondasi *Advanced Robotics Interface for Applications* (ARIA) dan *ARNetworking*, dirilis di bawah GNU *Public License*, dan lengkap dengan pustaka dan kode sumber C++, Java, dan Python yang didokumentasikan sepenuhnya. Beberapa lingkungan pengembangan aplikasi robotika pihak ketiga juga telah muncul dari penelitian, termasuk Saphira dari SRI International, Ayllu dari *Brandeis University*, Pyro dari Bryn Mawr dan *Swarthmore Colleges*. KUKA youBot memiliki 5 lengan *degrees of freedoms* (DOF) dan dua jari gripper.

2.6.1 Perangkat Keras KUKA Youbot

a. KUKA youBot omni-directional mobile platform

Basis KUKA youBot adalah omni-directional mobile platform segala arah dengan empat roda mecanum. Gambar 2.6 mengilustrasikan bingkai dasar (koordinat) yang terlampir, seperti yang akan digunakan dalam KUKA youBot API. Letaknya di tengah odometri. Nilai positif untuk rotasi terhadap sumbu z menghasilkan gerakan berlawanan arah jarum jam, seperti yang ditunjukkan pada Gambar 2.5

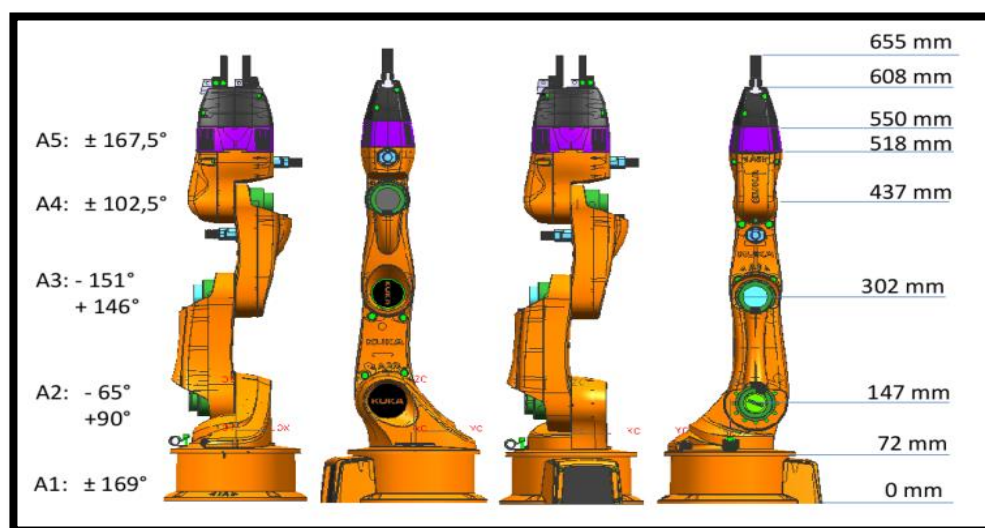


Gambar 2.5 Titik Koordinat Robot KUKA Youbot Manipulator

Pengontrol motor dari empat roda *omni-directional* dapat diakses melalui *Ethernet* dan *EtherCAT*. Kabel *Ethernet* dapat dicolokkan ke PC atau ke komputer eksternal. Stekernya adalah colokan kabel jaringan standar yang cocok dengan setiap port *Ethernet* standar. Di sisi atas alas KUKA youBot dilengkapi panel dengan colokan listrik (*output 24V*) dan port *Ethernet* (diberi label sebagai *EtherCAT 1* dan *EtherCAT 2*) untuk dua lengan KUKA youBot. Steker listrik ketiga (*input 24V*) dapat digunakan untuk memberi daya pada KUKA youBot dan mengisi ulang baterai (secara bersamaan). Ada satu slot *Ethernet* untuk menghubungkan PC onboard ke LAN melalui kabel jaringan. Layar LCD akan menunjukkan status baterai. Untuk menghubungkan pengontrol *EtherCAT* internal ke PC eksternal yaitu membuka penutup alas di sisi kanan, cabut kabel *Ethernet* dan sambungkan kembali ke kabel ekstensi yang masuk ke komputer eksternal.

b. KUKA Youbot Arm (Lengan)

Lengan KUKA youBot adalah rantai serial dengan lima sumbu berputar. End Effector dari KUKA Youbot adalah gripper dua jari, yang dapat dilepas. Gambar 2.6 mengilustrasikan struktur kinematik dasar dengan basis, dan driver motor untuk masing-masing sambungan dapat diakses melalui *Ethernet* dan *EtherCAT*.



Gambar 2.6 Lengan KUKA Youbot Manipulator

c. Onboard PC (Personal Computer)

Setiap KUKA youBot memiliki Mini-ITX board sebagai komputer terpasang atau onboard PC (*Personal Computer*). Karakteristik utamanya yaitu berdasarkan data Advantechs AIMB 212 .

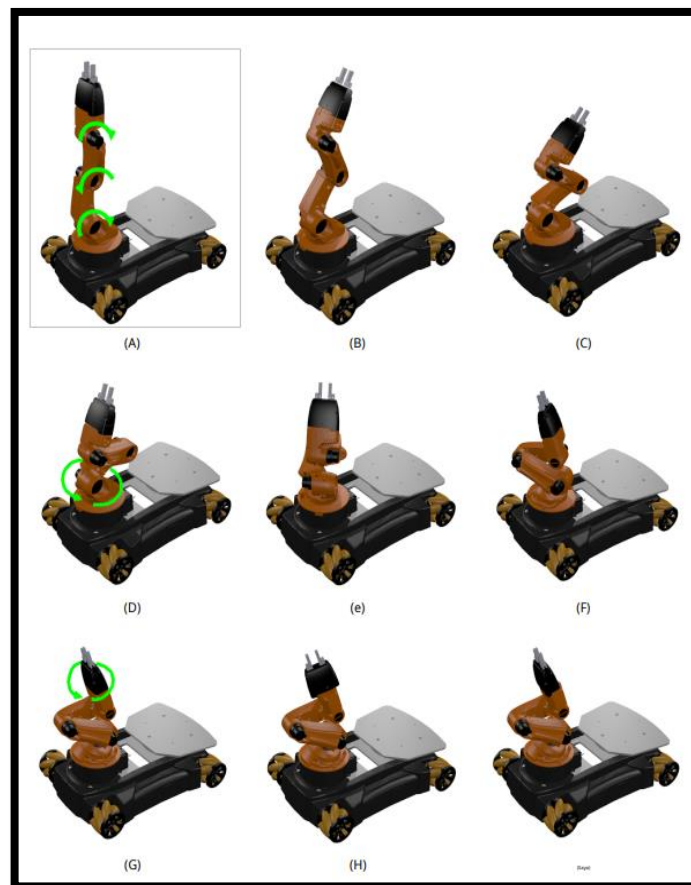
Tabel 2. 2 *Main characteristics onboard PC KUKA Youbot Manipulator*

Processory	Intel Atom™ Dual Core D510 (1M Cache, 2 x 1.66 GHz)
Memory	2GB single-channel DDR2 667 MHz
Graphics	Embedded Gen3.5 + GFX Core, 400-MHz render clock Frequency, up to 224 MB shared memory
Harddrive	32GB SSD Drive

Tabel 2.2 menampilkan spesifikasi perangkat keras dari sebuah komputer atau perangkat yang terpasang di dalam robot KUKA YouBot Manipulator. Perangkat ini menggunakan prosesor Intel Atom™ Dual Core D510, yang memiliki dua inti dengan kecepatan 1.66 GHz dan cache sebesar 1MB. Prosesor ini dikenal hemat daya dan biasanya digunakan pada perangkat dengan kebutuhan komputasi ringan. Perangkat ini disertai dengan RAM DDR2 berkapasitas 2GB dan berkecepatan 667 MHz. Kapasitas ini cukup untuk menjalankan tugas-tugas dasar seperti browsing internet, mengetik dokumen, dan aplikasi ringan lainnya. Untuk grafis, sistem menggunakan kartu grafis terintegrasi Gen3.5 + GFX Core dengan kecepatan pemrosesan 400 MHz dan memori berbagi hingga 224MB dari RAM utama, yang berarti perangkat ini tidak memiliki kartu grafis khusus dan mengandalkan memori sistem untuk kebutuhan visual. Penyimpanan pada perangkat ini menggunakan SSD berkapasitas 32GB. Meskipun kapasitasnya tergolong kecil, penggunaan SSD dapat kecepatan akses data yang baik dibandingkan hard drive konvensional. Dengan spesifikasi seperti ini, perangkat lebih cocok digunakan untuk tugas-tugas ringan seperti mengetik, menjelajah internet, atau menjalankan aplikasi dengan kebutuhan sumber daya rendah.

Sambungan lengan KUKA youBot dilengkapi dengan *encoder* posisi untuk mengukur sudut sambungan. *Encoder* yang digunakan adalah *encoder* posisi relatif, yang berarti, *encoder* tersebut tidak menghasilkan sudut sambungan absolut tetapi

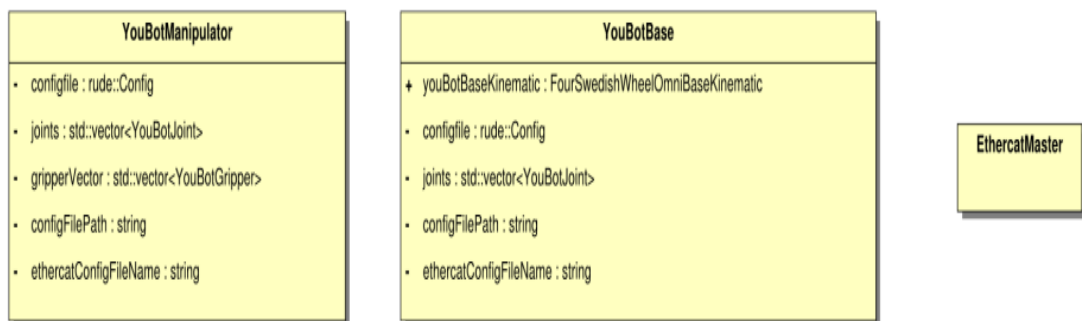
perpindahan sudut relatif terhadap posisi referensi atau posisi ini posisi rumah atau posisi istirahat untuk menunjukkan lengan harus memulai gerakannya dari posisi ini. Sebagai posisi acuan alami, menggunakan penghentian mekanis pada batas sambungan. Begitu lengan mendekati batas sendi ini, maka lengan harus digerakkan dengan sangat perlahan dan lembut, tidak peduli apakah digerakkan secara manual atau dengan program untuk menghindari kerusakan pada lengan. Pada posisi awal, lengan terlihat seperti terlipat menjadi satu. Prosedur homing dapat dijalankan selama motor sambungan tidak diberi daya (yaitu tidak ada arus yang dialirkan). Hal ini selalu terjadi saat robot dimatikan. Hal ini juga harus terjadi setelah suatu program menyelesaikan operasinya dengan benar. Namun, program-program mempunyai kecenderungan untuk gagal sewaktu-waktu. Mungkin saja motor sambungan tetap diberi daya dan tidak dapat digerakkan. Dalam kasus seperti ini, disarankan untuk mematikan dan mem-boot ulang seluruh robot untuk menghindari kerusakan.



Gambar 2.7 Pergerakan Lengan Robot menuju posisi awal.

2.6.2 Pemograman KUKA Youbot API

KUKA youBot API adalah antarmuka pemrograman yang dapat mengakses dan mengontrol perangkat keras KUKA youBot. Menggunakan KUKA youBot API dalam jangka pendek dapat digantikan dengan KUKA youBot OODL API yang berjangka lebih panjang. Pengertian OODL yaitu lapisan Perangkat berorientasi objek dan menekankan desain perangkat lunak berorientasi objek di balik API. Perangkat lunak yang bersifat open source ini dapat dipandang sebagai driver tingkat tinggi untuk robot yang terdiri dari sekumpulan sub-sistem fungsional yang dipisahkan. Dalam KUKA youBot API, lengan robot direpresentasikan sebagai rantai kinematik 5 DOF, dan platform seluler omnidirection diperlakukan sebagai kumpulan sambungan putar, yang setiap sambungan terdiri dari motor dan kotak roda gigi.



Gambar 2. 8 Elemen utama KUKA Youbot API yaitu YouBotManipulator dan YouBotBase

KUKA youBot API membedakan berbagai jenis data yang disediakan oleh gabungan. Ini adalah *set point* dan *sense data*. Setiap kuantitas fisik diwakili oleh kelasnya sendiri untuk *setPoint* atau *sense data*, yang juga memperhitungkan unit fisiknya. Prinsip yang sama berlaku untuk parameter gabungan, di mana setiap besaran fisik (misalnya arus, hambatan kumparan motor) diwakili oleh kelasnya sendiri. Jumlah dan karakter parameter yang didukung oleh API ditentukan oleh firmware yang dijalankan pada pengontrol motor. Untuk informasi lebih rinci tentang berbagai parameter.

2.7 Autonomous Robot

Untuk menjalankan fungsinya secara otonom pada ruangan, robot memiliki empat komponen utama, yaitu:

1. Persepsi

Robot mampu membuat peta ruangan beroperasi dan menganalisis rintangan serta jalan dilalui menuju tujuan dalam ruang tersebut. Peta ini dibentuk menggunakan kamera RGB-D, dimana kamera 3D yang dapat mengenali lingkungan pada membaca kedalaman objek-objek di sekitar. Data yang diperoleh kemudian diproses oleh komputer papan tunggal dan diubah menjadi baca laser *scan* untuk menyusun dalam ruangan (*gmapping*). Proses ini menggunakan *simultaneous localization and mapping* (SLAM), dengan penerapan versi *Rao-Blackwellized Particle Filter* yang mengonversi gambar dari grid map, yang mengurangi ukuran sampling dan secara efektif dapat mengurangi waktu komputasi [19].

2. Lokalisasi

Setelah bergerak, robot harus dapat menentukan posisinya relatif terhadap tujuan (odometri) dan menilai rintangan yang ada untuk menghindari tabrakan. Untuk mengetahui lokasi yang telah ditempuh, robot tidak hanya mengandalkan kamera RGB-D untuk mengenali lingkungan, tetapi juga menggunakan sensor *wheel encoder* dan *gyroscope* untuk mengumpulkan data odometri, seperti posisi, orientasi, dan kecepatan robot. Data ini diperoleh dari motor penggerak yang dikendalikan oleh driver motor pada komputer papan tunggal. Untuk mencegah tabrakan yang disebabkan oleh perubahan mendadak pada peta ruangan, robot dilengkapi dengan sensor bumper dari berfungsi melindungi robot dari benturan dengan objek yang muncul tiba-tiba atau yang tidak dapat terdeteksi oleh kamera RGB-D, seperti objek transparan atau objek yang terlalu rendah dan dianggap datar.[17].

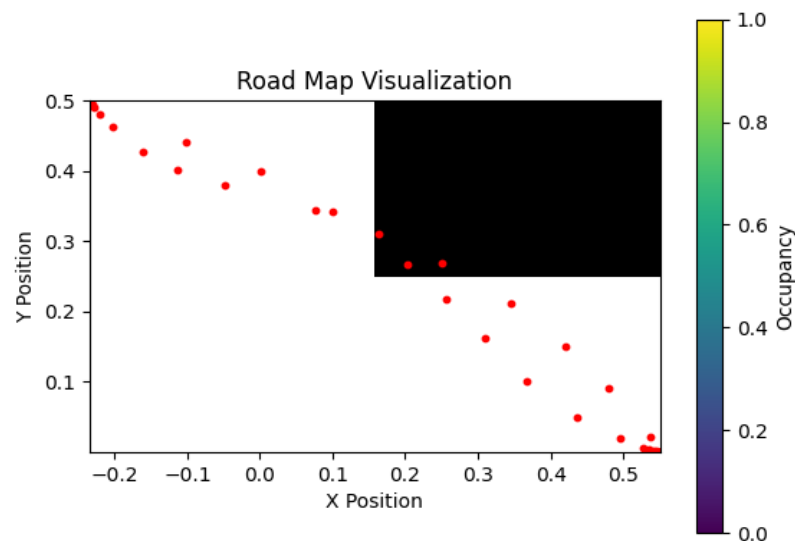
3. Perencanaan Jalur (*Path Planning*)

Agar robot mencapai tujuan yang diberikan, robot dapat menentukan jalur yang optimal dan layak dari posisinya sambil menghindari rintangan. Perencanaan jalur ini dilakukan menggunakan *gmapping* dan *Robot Operating System* (ROS) yang memanfaatkan pemrograman berbasis

objek (OOP). Proses ini melibatkan pengolahan objek atau data yang diperoleh untuk dibandingkan dengan data dari sistem sensor untuk mengestimasi kedalaman per piksel berdasarkan pola referensi yang paling sesuai dengan pola yang dihasilkan dari proyeksi [19]. ROS adalah *framework middleware* sumber terbuka berbasis Unix dengan arsitektur modular, fleksibel, dan tahan banting dalam aplikasi robotika. Struktur ROS mencakup tiga tingkat konsep yang berbeda, yaitu tingkat *filesystem*, tingkat *computational graph*, dan *tingkat community*, yang digunakan untuk membentuk perencanaan global sebagai bagian dari *path planning*. Sementara itu, setelah robot bergerak, untuk perencanaan jalur lokal digunakan algoritma *Monte Carlo Localization* (MCL). MCL ialah algoritma lokalisasi yang memiliki keyakinan $bel(x)$ pada partikel dalam *Adaptive Monte Carlo Localization* (AMCL), yang diterapkan dengan metode *Kullback-Leibler Divergence* (KLD)-*sampling*. Untuk menghindari rintangan dalam perencanaan lokal, digunakan pada *Dynamic Window Approach* (DWA).

4. Kontrol Gerakan (*Motion Control*)

Robot mampu melakukan pergerakan menuju tujuan berdasarkan peta yang sudah dibuat. Robotika merupakan ilmu yang melibatkan persepsi, pengamatan, dan manipulasi dunia fisik melalui perangkat penggerak yang dikendalikan oleh komputer [20].



Gambar 2.9 *Road Map* Simulasi KUKA Youbot

Gambar 2.9 merupakan visualisasi *road map* dari simulasi KUKA YouBot, yang menampilkan jalur pergerakan robot dalam lingkungan dengan peta okupansi. Titik-titik merah menunjukkan jalur navigasi yang direncanakan, sementara area hitam merepresentasikan rintangan yang tidak dapat dilewati oleh robot. Sumbu X dan Y menunjukkan koordinat posisi dalam lingkungan simulasi, sedangkan skala warna di sisi kanan menunjukkan tingkat okupansi, di mana warna lebih gelap menandakan area terhalang.

2.8 Quaternion

Secara pragmatis, *Quaternion* didefinisikan sebagai sebuah kuantitas yang terdiri dari empat angka dengan komponennya yaitu skalar dan vektor secara deskriptif dalam ilmu matematika *quaternion* merupakan eksistensi dari bilangan-bilangan kompleks yang komutatif dan diterapkan dalam mekanika dan rotasi tiga dimensi yang membentuk *normed division algebra* empat dimensi. *Quaternion* merupakan kombinasi linear dari *global quaternion* q ($1, \hat{I}, \hat{J}$ dan $\hat{K}$) yang direpresentasikan ke dalam persamaan sebagai berikut :

$$q = q_0 + \mathbf{q}$$

$$\mathbf{q} = q_1\hat{I} + q_2\hat{J} + q_3\hat{K}$$

$$q = q_0 + q_1\hat{I} + q_2\hat{J} + q_3\hat{K}$$

dan dapat juga direpresentasikan dalam bentuk flag (i, j dan k)

$$q = q_0 + q_1i + q_2j + q_3k$$

Permasalahan yang terjadi terkait dengan aplikasi interpolasi orientasi pada robot industri untuk mengontrol kecepatan dan percepatan robot dengan *Spherical Liner Interpolation* (SLERP) menggunakan unit quaternion hanya mampu menggenerate lintasan kontinyu C1, sedangkan dibutuhkan kombinasi *quaternion* lainnya agar dapat menjamin kontinuitas C3 dari lintasan orientasi. Secara umum orientasi interpolasi robot industri dapat menggunakan sumbu sudut euler, namun dapat menimbulkan *Gimbal Lock* yang dapat menyebabkan kerusakan pada robot. Hal ini disebabkan karena sudut euler sendiri dapat mengarah ke hasil yang tidak diinginkan yang disebabkan oleh nonlinearitas tiga sudut Euler sehingga area .

BAB III

METODOLOGI PENELITIAN

3.1 Waktu dan Tempat

Adapun penelitian ini dilakukan pada bulan Januari 2024 sampai dengan bulan Februari 2025. Penelitian ini dilaksanakan di Laboratorium Teknik Kendali, Laboratorium Terpadu Jurusan Teknik Elektro, Universitas Lampung.

3.2 Alat dan Bahan

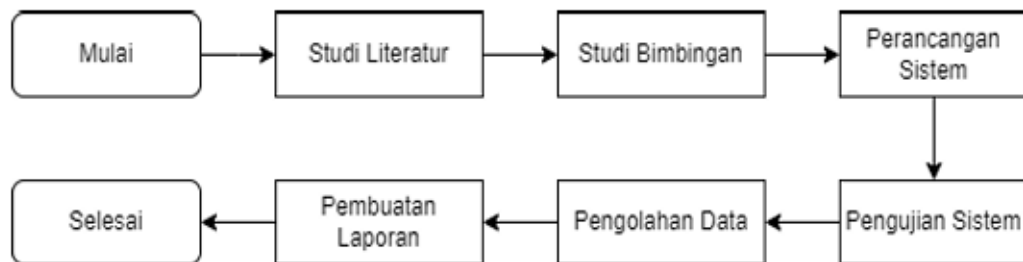
Adapun alat serta bahan yang digunakan pada penelitian ini antara lain :

Tabel 3.1 Alat dan Bahan

No	Alat dan Bahan	Keterangan penggunaan
1	Laptop Notebook MSI	Sebagai perangkat keras untuk melakukan dan membuat perancangan sistem
2	Coppelia SIM	Sebagai software untuk merancang Robot Pioneer 3-DX
3	Mathlab	Sebagai Software untuk merancang Plant <i>Fuzzy</i> dan PID
4	Python	Sebagai Software untuk menampilkan serta menginisiasi hasil simulasi.

3.3 Tahapan Penelitian

Adapun tahapan pada penelitian ini adalah sebagai berikut



Gambar 3. 1 Tahapan Penelitian

Pada Gambar 3.1 terlihat tahapan penelitian yang dimulai dengan studi literatur, diikuti dengan studi bimbingan, perancangan sistem, pengujian sistem, pengolahan data, dan akhirnya diakhiri dengan pembuatan laporan.

1. Studi Literatur

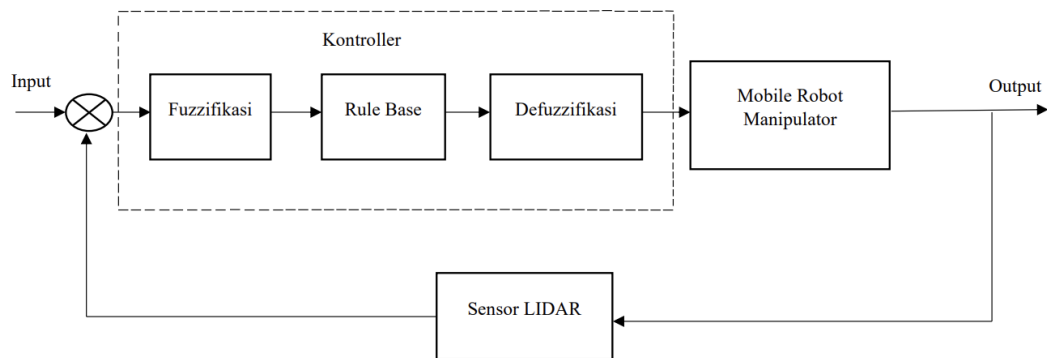
Studi literatur dilaksanakan untuk mempelajari topik-topik yang relevan dengan tugas akhir, seperti CoppeliaSIM, Python, pengaturan Task Planning, Autonomous Robot, dan kontroler logika fuzzy. Materi tersebut diperoleh dari berbagai sumber yang dapat dipertanggungjawabkan, seperti buku, jurnal, dan referensi dari situs web yang terpercaya.

2. Studi Bimbingan

Proses bimbingan dilaksanakan melalui diskusi, sesi tanya jawab, serta pencarian solusi dengan dosen pembimbing mengenai alat yang dikembangkan. Tujuan dari bimbingan ini adalah untuk memperluas wawasan dan mendukung kelancaran dalam penyelesaian tugas akhir.

3. Perancangan Sistem

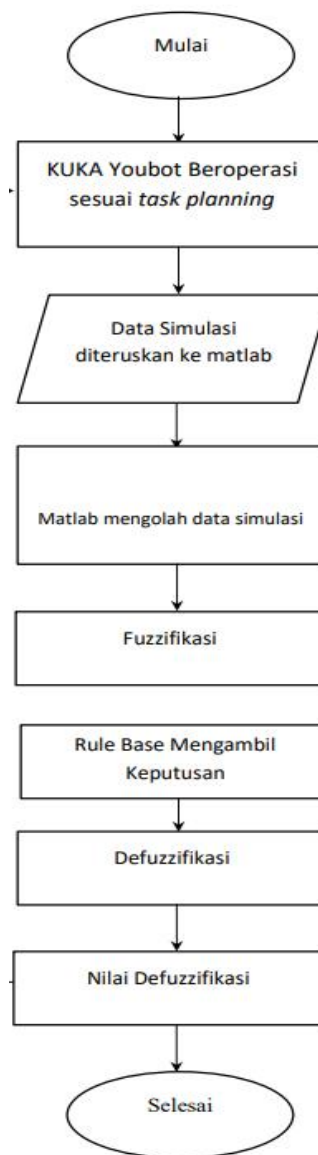
Perancangan sistem diawali dengan menentukan *Task Planning* yang akan digunakan sebagai dasar dalam penelitian. Setelah itu, langkah selanjutnya adalah membuat pemrograman python dan bahasa lua agar dapat mencoba simulasi pada *software* CoppeliaSIM dengan kontroler logika *fuzzy*.



Gambar 3. 2 Perancangan Sistem Menggunakan *Fuzzy Logic*

Pada tahap perancangan perangkat lunak, sistem kendali direpresentasikan dalam bentuk diagram alir (*flowchart*). Diagram ini menunjukkan urutan proses yang dijalankan oleh sistem secara sistematis. Di dalam *flowchart* tersebut, terdapat *rule base* yang berperan sebagai kumpulan aturan dasar yang dirancang melalui proses pemrograman sistem. Kemampuan sistem *fuzzy* dalam mengambil keputusan bergantung pada sekumpulan aturan yang telah ditetapkan. Umumnya, aturan-aturan ini bersifat intuitif dan disusun dalam format pernyataan kualitatif menggunakan pola "jika-maka", sehingga lebih mudah dipahami. Rule base ini mencakup berbagai kondisi yang diklasifikasikan dalam tahap fuzzifikasi sebagai berikut:

1. Input kecepatan dibedakan menjadi lima kategori: sangat lambat, lambat, sedang, cepat dan sangat cepat.
2. Input jarak dibedakan menjadi lima kategori: sangat dekat, dekat, sedang, jauh dan sangat jauh.
3. Dari masukan tersebut, data tersebut akan diproses melalui fuzzifikasi untuk menghasilkan output yang terdiri dari tiga kondisi nilai: akurasi tinggi, akurasi sedang, dan akurasi rendah.
4. Setelah itu dibentuk aturan *fuzzy* atau yang disebut *rule base*, pada penelitian ini terdiri dari 25 *rule base*.
5. Setelah diproses bagian rulebase tahap selanjutnya yaitu masuk bagian defuzzifikasi



Gambar 3. 3 Diagram Alir Metode *Fuzzy Logic Controller*

Pengimplementasian controller logika *fuzzy sugeno* pada sistem ini akan mengarah bagaimana tingkat keakurasian robot bergerak manipulator mengoperasikan untuk memindahkan 6 buah objek *rectangle* dari titik awal ke titik akhir dengan bantuan titik sementara berdasarkan jarak dan kecepatan. Ketika robot bergerak manipulator melakukan tindakan baik mengambil maupun menaruh objek akan berpengaruh pada jarak dan kecepatan untuk mendapatkan tingkat keakurasian. Maka diciptakan sistem robot bergerak manipulator menggunakan controller logika *fuzzy sugeno*.

4. Pengujian Sistem

Pada tahap pengujian sistem dilakukan beberapa pengujian dan lokasi pengujian sistem yang dilakukan di Laboratorium Teknik Kendali Universitas Lampung. Adapun tabel pengujian dapat dilihat pada Tabel 3.2 adalah sebagai berikut:

Tabel 3.2 Pengujian Sistem

Pengujian	Indikator keberhasilan	Keterangan
Pengujian <i>path planning</i>	Peletakkan <i>path planning</i> sesuai dengan koordinat yang ditentukan, penelitian ini memiliki 3 pad (<i>start</i> , sementara, <i>goal</i>)	Menguji keberhasilan <i>path planning</i>
Pengujian <i>Task Planning</i>	Robot akan memindahkan 6 Buah objek kotak dari titik <i>start</i> ke titik <i>goal</i> dengan bantuan titik sementara	Menguji keberhasilan <i>Task Planning</i>
Pengujian orientasi posisi	Mendapatkan posisi koordinat robot sepanjang simulasi yang ditampilkan pada software <i>python</i>	Menguji orientasi posisi robot
Pengujian metode <i>fuzzy logic controller</i> sugeno	Mendapatkan output akurasi robot yang optimal dalam menjalankan <i>Task Planning</i> berdasarkan kedua input yaitu jarak dan kecepatan	Menguji tingkat keakurasian robot berdasarkan jarak dan kecepatan

3.4 Perancangan Metode *Fuzzy logic*

3.4.1 Menentukan Nilai Variabel

Pada tahap bagian perancangan kontroller logika *fuzzy* untuk mengklasifikasikan tingkat keakurasian Manipulator Robot Bergerak. Dimulai dari masukan nilai berupa variabel input berupa nilai kecepatan dan jarak. Nilai variabel output berupa tingkat keakurasian Manipulator Robot Bergerak berdasarkan analisa pada penelitian ini. Tabel 3.3 merupakan representasi data pada penelitian ini.

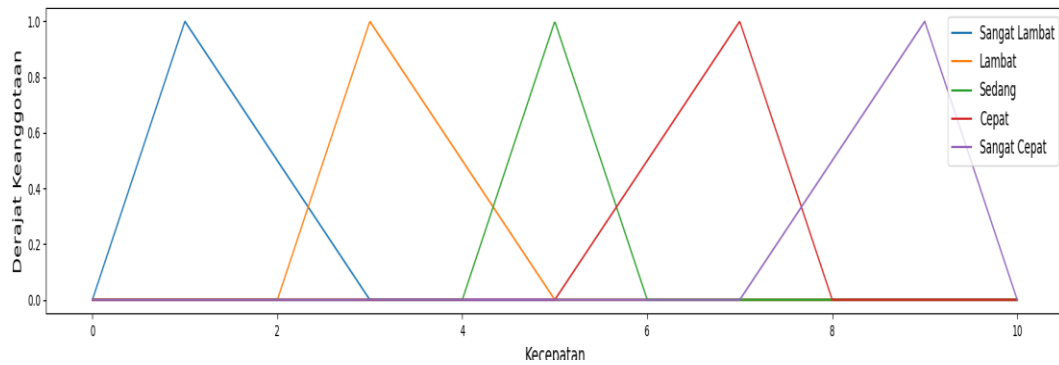
Tabel 3.3 Representasi Data

Fungsi	Variabel <i>Fuzzy</i> Logic	Himpunan <i>Fuzzy</i> Logic	Semesta Pembicara	Domain
Input	Kecepatan	Sangat Lambat	[0 - 10]	[0 1 3]
		Lambat		[2 3 5]
		Sedang		[4 5 6]
		Cepat		[5 7 8]
		Sangat Cepat		[7 9 10]
	Jarak	Sangat Dekat	[0 - 2]	[0 0,2 0,3]
		Dekat		[0,2 0,4 0,5]
		Sedang		[0,4 0,6 0,8]
		Jauh		[0,7 1,2 1,5]
		Sangat Jauh		[1,3 1,7 2]
Output	Akurasi	Rendah	[0 - 100]	[0 25 50]
		Sedang		[25 50 75]
		Tinggi		[50 75 100]

3.4.2 Fuzzifikasi

Pada tahap fuzzifikasi, terdapat dua nilai crisp dari Manipulator Robot Bergerak untuk mengklasifikasikan derajat keanggotaan dari setiap nilai input dan output. Nilai masukan digunakan untuk menentukan tingkat keakuratan dari Manipulator Robot Bergerak. Fungsi keanggotaan didapat dari grafik linear dan segitiga pada setiap fungsi keanggotaan. Proses fuzzifikasi ini menghasilkan nilai *fuzzy* input, yaitu proses mengubah sebuah nilai masukan menjadi nilai *fuzzy*. Nilai masukan tegas yang ada pada tahap ini dimasukkan ke dalam fungsi pengaburan yang telah ditentukan sebelumnya, menghasilkan nilai *fuzzy* masukan. Fuzzifikasi juga menentukan fungsi anggota untuk variabel masukan dan keluaran. Untuk penelitian ini, terdapat dua variabel masukan dan satu variabel keluaran. Variabel masukan terdiri dari nilai kecepatan dan jarak, di mana setiap variabel masukan memiliki himpunan *fuzzy* sendiri. Himpunan *fuzzy* untuk variabel masukan kecepatan meliputi lambat, sedang, dan cepat, sementara untuk variabel masukan jarak, himpunan *fuzzy* meliputi dekat, sedang, dan jauh. Variabel masukan kecepatan dan jarak masing-masing memiliki fungsi keanggotaan trapesium. Nilai variabel masukan kecepatan memiliki rentang nilai 0 – 11, nilai 0 merupakan nilai

minimum dan nilai 11 merupakan nilai maksimum, dan nilai variabel masukan arak memiliki rentang nilai 0 – 2, nilai 0 merupakan nilai minimum dan nilai 2 merupakan nilai maksimum.



Gambar 3. 4 Grafik Keanggotaan Kecepatan

Berikut merupakan persamaan derajat keanggotaan Kecepatan.

Derajat keanggotaan Kecepatan Sangat Lambat berdasarkan persamaan (2.1):

$$\mu_{\text{SangatLambat}}(X) = \begin{cases} 1 & \text{jika } x \leq 1 \\ \frac{3-x}{3-1} & \text{jika } 1 < x < 3 \\ 0 & \text{jika } x \geq 3 \end{cases}$$

Derajat keanggotaan Kecepatan Lambat berdasarkan persamaan (2.1):

$$\mu_{\text{Lambat}}(X) = \begin{cases} 1 & \text{jika } x \leq 5 \\ \frac{5-x}{5-3} & \text{jika } 3 < x < 5 \\ 0 & \text{jika } x \geq 5 \end{cases}$$

Derajat keanggotaan Kecepatan Sedang berdasarkan persamaan(2.1):

$$\mu_{\text{Sedang}}(X) = \begin{cases} 0 & \text{jika } x \leq 4 \text{ atau } \geq 6 \\ \frac{x-4}{6-4} & \text{jika } 4 < x < 6 \\ \frac{8-x}{6-5} & \text{jika } 5 < x < 6 \\ 1 & \text{jika } x = 6 \end{cases}$$

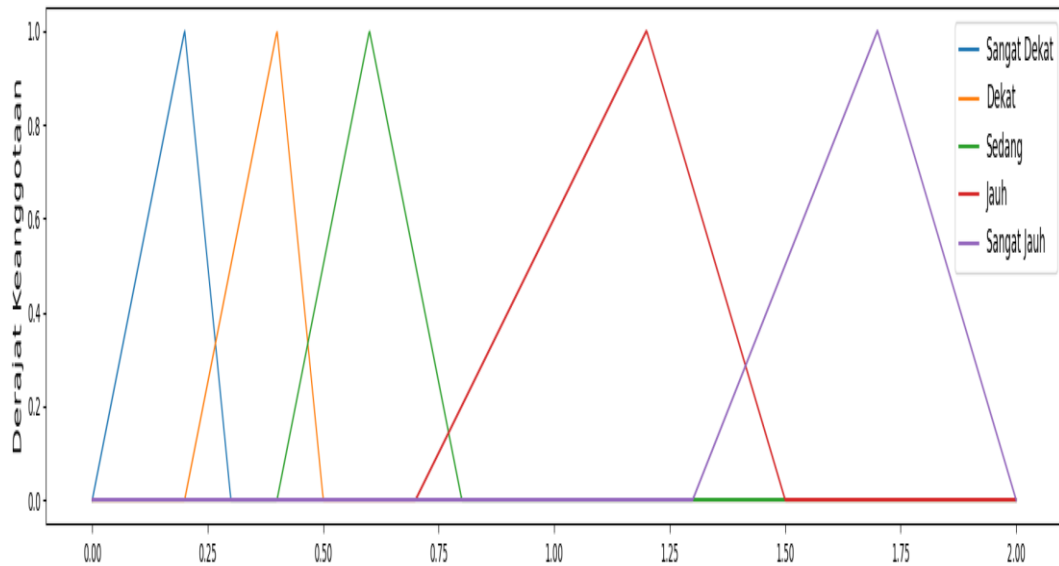
Derajat keanggotaan Kecepatan Cepat berdasarkan persamaan(2.1):

$$\mu_{\text{Cepat}}(X) = \begin{cases} 0 & \text{jika } x \leq 5 \\ \frac{x-5}{8-5} & \text{jika } 5 < x < 8 \\ 1 & \text{jika } x \geq 8 \end{cases}$$

Derajat keanggotaan Kecepatan Sangat Cepat berdasarkan persamaan(2.1):

$$\mu_{\text{SangatCepat}}(X) = \begin{cases} 0 & \text{jika } x \leq 7 \\ \frac{x-7}{10-7} & \text{jika } 7 < x < 10 \\ 1 & \text{jika } x \geq 10 \end{cases}$$

Berdasarkan Gambar 3.4 derajat keanggotaan kecepatan, untuk himpunan *fuzzy* kecepatan sangat lambat memiliki rentan minimal 0 dan nilai maksimal 3 dengan fungsi keanggotaan linear naik. Nilai kurang dari sama dengan 3 memiliki derajat keanggotaan 1 dan jika nilainya lebih dari sama dengan 3 memiliki derajat keanggotaan 0 pada himpunan *fuzzy* kecepatan sangat lambat. Untuk himpunan *fuzzy* kecepatan lambat memiliki rentan nilai 2 sampai 5, jika nilainya kurang dari sama dengan 2 maka atau lebih dari sama dengan 5 memiliki derajat keanggotaan 0, untuk himpunan *fuzzy* kecepatan sedang merupakan fungsi keanggotaan segitiga. Pada himpunan *fuzzy* kecepatan sedang memiliki rentan nilai 4 sampai 6, jika nilainya kurang dari sama dengan 4 dan lebih dari sama dengan 6, maka derajat keanggotaannya 0, untuk himpunan *fuzzy* kecepatan cepat merupakan fungsi keanggotaan segitiga. Pada himpunan *fuzzy* kecepatan cepat memiliki rentan nilai 5 sampai 8, jika nilainya kurang dari sama dengan 5 dan lebih dari sama dengan 8, maka derajat keanggotaannya 0, untuk himpunan *fuzzy* kecepatan cepat merupakan fungsi keanggotaan segitiga. Pada himpunan *fuzzy* kecepatan sangat cepat memiliki rentan nilai 7 sampai 10, jika nilainya kurang dari sama dengan 7 dan lebih dari sama dengan 10, maka derajat keanggotaannya 0, untuk himpunan *fuzzy* kecepatan sedang merupakan fungsi keanggotaan segitiga.



Gambar 3. 5 Grafik Keanggotaan Jarak

Berikut merupakan persamaan derajat keanggotaan Jarak

Derajat keanggotaan Jarak Sangat Dekat berdasarkan persamaan (2.1):

$$\mu_{\text{Sangat Dekat}}(X) = \begin{cases} 1 & \text{jika } x \leq 0,2 \\ \frac{0,3-x}{0,3-0,2} & \text{jika } 0,2 < x < 0,3 \\ 0 & \text{jika } x \geq 0,3 \end{cases}$$

Derajat keanggotaan Jarak Dekat berdasarkan persamaan (2.1):

$$\mu_{\text{Dekat}}(X) = \begin{cases} 1 & \text{jika } x \leq 0,5 \\ \frac{0,5-x}{0,5-0,2} & \text{jika } 0,2 < x < 0,5 \\ 0 & \text{jika } x \geq 0,5 \end{cases}$$

Derajat keanggotaan Jarak Sedang berdasarkan persamaan(2.1):

$$\mu_{\text{Sedang}}(X) = \begin{cases} 0 & \text{jika } x \leq 0,4 \text{ atau } \geq 0,8 \\ \frac{x-0,4}{0,6-0,4} & \text{jika } 0,4 < x < 0,6 \\ \frac{0,8-x}{0,8-0,6} & \text{jika } 0,6 < x < 0,8 \\ 1 & \text{jika } x = 0,8 \end{cases}$$

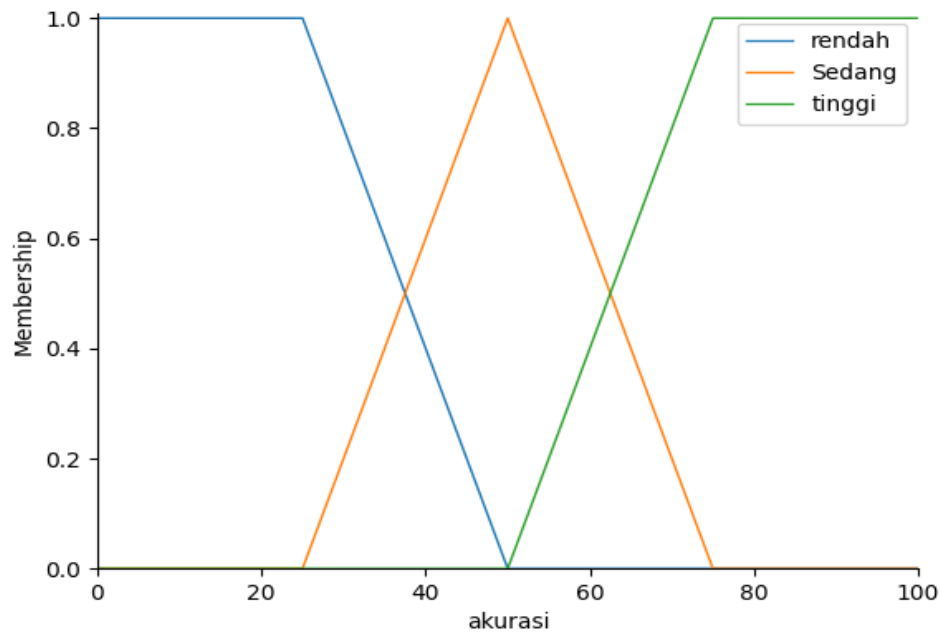
Derajat keanggotaan Jarak Jauh berdasarkan persamaan(2.1):

$$\mu_{\text{Jauh}}(X) = \begin{cases} 0 & \text{jika } x \leq 0,7 \\ \frac{x-0,7}{1,5-0,7} & \text{jika } 0,7 < x < 1,5 \\ 1 & \text{jika } x \geq 1,5 \end{cases}$$

Derajat keanggotaan Jarak Sangat Jauh berdasarkan persamaan(2.1):

$$\mu_{\text{SangatJauh}}(X) = \begin{cases} 0 & \text{jika } x \leq 1,3 \\ \frac{x-1,3}{2-1,3} & \text{jika } 1,3 < x < 2 \\ 1 & \text{jika } x \geq 2 \end{cases}$$

Berdasarkan Gambar 3.5 derajat keanggotaan jarak, untuk himpunan *fuzzy* jarak sangat dekat memiliki rentan minimal 0 dan nilai maksimal 0,3 dengan fungsi keanggotaan linear naik. Nilai kurang dari sama dengan 0,2 memiliki derajat keanggotaan 1 dan jika nilainya lebih dari sama dengan 0,3 memiliki derajat keanggotaan 0 pada himpunan *fuzzy* jarak sangat dekat. Untuk himpunan *fuzzy* jarak dekat memiliki rentan nilai 0,2 – 0,5, jika nilainya kurang dari sama dengan 0,2 maka atau lebih dari sama dengan 0,5 memiliki derajat keanggotaan 0, untuk himpunan *fuzzy* jarak sedang merupakan fungsi keanggotaan segitiga. Pada himpunan *fuzzy* jarak sedang memiliki rentan nilai 0,4 – 0,8, jika nilainya kurang dari sama dengan 0,4 maka derajat keanggotaannya 0, jika nilainya lebih dari sama dengan 0,6 maka derajat keanggotaannya 1, pada himpunan *fuzzy* jarak sedang merupakan fungsi keanggotaan linear naik. Pada himpunan *fuzzy* jarak jauh memiliki rentan nilai 0,7 sampai 1,5, jika nilainya kurang dari sama dengan 0,7 dan lebih dari sama dengan 1,5, maka derajat keanggotaannya 0, untuk himpunan *fuzzy* jarak jauh merupakan fungsi keanggotaan segitiga. Pada himpunan *fuzzy* jarak sangat jauh memiliki rentan nilai 1,3 sampai 2, jika nilainya kurang dari sama dengan 1,3 dan lebih dari sama dengan 2, maka derajat keanggotaannya 0, untuk himpunan *fuzzy* jarak sangat jauh merupakan fungsi keanggotaan segitiga.



Gambar 3. 6 Grafik Keanggotaan Tingkat Akurasi

Berikut merupakan persamaan derajat keanggotaan Akurasi

Derajat keanggotaan Akurasi Kecil berdasarkan persamaan (2.1):

$$\mu_{\text{Rendah}}(X) = \begin{cases} 1 & \text{jika } x \leq 25 \\ \frac{50-x}{50-25} & \text{jika } 25 < x < 50 \\ 0 & \text{jika } x \geq 50 \end{cases}$$

Derajat keanggotaan Akurasi Sedang berdasarkan persamaan(2.1):

$$\mu_{\text{Sedang}}(X) = \begin{cases} 0 & ; \text{jika } x \leq 25 \text{ atau } \geq 75 \\ \frac{x-25}{50-25} & ; \text{jika } 25 < x < 50 \\ \frac{75-x}{75-50} & ; \text{jika } 50 < x < 75 \\ 1 & ; x = 50 < x < 75 \end{cases}$$

Derajat keanggotaan Akurasi tinggi berdasarkan persamaan(2.1):

$$\mu_{\text{Tinggi}}(X) = \begin{cases} 0 & \text{jika } x \leq 50 \\ \frac{x-50}{100-50} & \text{jika } 50 < x < 100 \\ 1 & \text{jika } x \geq 100 \end{cases}$$

Berdasarkan Gambar 3.7 derajat keanggotaan tingkat akurasi Manipulator Robot Bergerak untuk himpunan *fuzzy* rendah memiliki rentan minimal 0 dan nilai maksimal 50 dengan fungsi keanggotaan linear naik. Nilai sama dengan 25 memiliki derajat keanggotaan 1 dan jika nilainya lebih dari sama dengan 50 memiliki derajat keanggotaan 0 pada himpunan *fuzzy* rendah. Untuk himpunan *fuzzy* sedang memiliki rentan nilai 25– 75, jika nilainya kurang dari sama dengan 25 maka atau lebih dari sama dengan 75 memiliki derajat keanggotaan 0, untuk himpunan *fuzzy* sedang merupakan fungsi keanggotaan segitiga. Pada himpunan *fuzzy* tinggi memiliki rentan nilai 50 – 100, jika nilainya kurang dari sama dengan 50 maka derajat keanggotaannya 0, jika nilainya lebih besar sama dengan 75 maka derajat keanggotaannya 1, pada himpunan *fuzzy* tinggi merupakan fungsi keanggotaan linear naik

3.4.3 Pembentukan Aturan *Fuzzy*

Dalam proses pembentukan aturan *fuzzy*, nilai masukan *fuzzy* diubah menjadi keluaran *fuzzy* dengan mengikuti aturan *IF-THEN*, yang kemudian menghasilkan nilai masukan *fuzzy*. Pada tahap ini, nilai masukan tegas dimasukkan ke dalam fungsi pengaburan yang telah ditetapkan sehingga diperoleh nilai masukan *fuzzy*. Dalam penelitian ini, terdapat dua variabel masukan dan satu variabel keluaran. Variabel masukan yang digunakan adalah kecepatan dan jarak, di mana masing-masing memiliki himpunan *fuzzy*. Himpunan *fuzzy* untuk variabel masukan kecepatan terdiri dari sangat lambat, lambat, sedang, cepat, dan sangat cepat. Sementara itu, himpunan *fuzzy* untuk variabel masukan jarak mencakup sangat dekat, dekat, sedang, jauh, dan sangat jauh. Baik variabel masukan kecepatan maupun jarak menggunakan fungsi keanggotaan berbentuk segitiga.

Tabel 3.4 Tabel Aturan *Fuzzy*

No	Jika		Maka
	Kecepatan	Jarak	
1	Sangat Lambat	Sangat Dekat	Tinggi
2	Sangat Lambat	Dekat	Tinggi
3	Sangat Lambat	Sedang	Sedang
4	Sangat Lambat	Jauh	Rendah
5	Sangat Lambat	Sangat Jauh	Rendah
6	Lambat	Sangat Dekat	Tinggi
7	Lambat	Dekat	Tinggi
8	Lambat	Sedang	Sedang
9	Lambat	Jauh	Rendah
10	Lambat	Sangat Jauh	Rendah
11	Sedang	Sangat Dekat	Tinggi
12	Sedang	Dekat	Tinggi
13	Sedang	Sedang	Sedang
14	Sedang	Jauh	Rendah
15	Sedang	Sangat Jauh	Rendah
16	Cepat	Sangat Dekat	Sedang
17	Cepat	Dekat	Sedang
18	Cepat	Sedang	Rendah
19	Cepat	Jauh	Rendah
20	Cepat	Sangat Jauh	Rendah
21	Sangat Cepat	Sangat Dekat	Sedang
22	Sangat Cepat	Dekat	Rendah
23	Sangat Cepat	Sedang	Rendah
24	Sangat Cepat	Jauh	Rendah
25	Sangat Cepat	Sangat Jauh	Rendah

3.4.4 Defuzzifikasi

Tahap terakhir dalam logika *fuzzy* adalah defuzzifikasi dimana variabel *fuzzy* yang diolah pada tahap mekanisme inferensi diubah kembali menjadi nilai tegas dengan cara mengalikan dengan nilai *output* (z). Kemudian setelah ditentukannya predikat α dan Nilai *output* (z) maka nilai tegas (*Crisp*) dapat dihitung menggunakan persamaan berikut. Proses defuzzifikasi digunakan untuk mengubah nilai keanggotaan *fuzzy* menjadi keputusan yang lebih spesifik atau bilangan real, yang mana mengembalikan nilai *fuzzy* menjadi nilai *crisp* (bilangan real), serta

mengubah *output fuzzy* menjadi nilai *crisp* berdasarkan fungsi keanggotaan yang telah ditetapkan. Defuzzifikasi diperlukan karena output dari sistem *fuzzy* adalah variabel linguistik, yang perlu dikonversi menjadi variabel *crisp*. *Input* dari proses defuzzifikasi adalah himpunan *fuzzy* yang diperoleh melalui komposisi aturan *fuzzy*, sementara *output* adalah sebuah bilangan dalam domain himpunan *fuzzy* tersebut. Oleh karena itu, jika suatu himpunan *fuzzy* diketahui dalam rentang tertentu, maka perlu dihasilkan nilai *crisp* (bilangan real) sebagai *output* atau keputusan. Salah satu metode yang digunakan dalam proses defuzzifikasi ini adalah metode *Centroid* (titik pusat). Metode ini mempertimbangkan kondisi setiap area *fuzzy*, sehingga memberikan hasil yang lebih akurat. Penentuan titik pusat dari daerah *fuzzy* dilakukan dengan menggunakan Persamaan 2.2. Apabila nilai tegas (*Crisp*) sudah ditentukan maka dikembalikan pada klasifikasi nilai output.

3.5 Implementasi *Fuzzy logic Controller* Sugeno

Pada implementasi sistem controller logika *fuzzy* sugeno ini merupakan pengolahan data untuk mengukur tingkat akurasi robot bergerak manipulator berdasarkan jarak dan kecepatan. Untuk variabel input berupa nilai kecepatan dan jarak, sebagai contoh dengan input kecepatan = 2ms dan jarak = 0,4m.

3.5.1 Proses Fuzzifikasi

Langkah awal pada proses controller logika *fuzzy* Sugeno ini yaitu fuzzifikasi untuk menentukan derajat keanggotaan dari nilai input tersebut.

- a. Untuk Proses fuzzifikasi pada kecepatan dapat dilihat pada Gambar 4.9 yang merupakan kurva linear naik, turun dan segitiga.

Fuzzifikasi Kecepatan (2 ms berada pada derajat keanggotaan Lambat)

$x = 2\text{ms}$; $a = 1$ dan $b = 2$

menghitung kurva linear segitiga dari derajat keanggotaan Kecepatan Sangat Lambat dan Lambat.

$$\mu(X) = \frac{b - x}{b - a} = \frac{2 - 0}{2 - 1} = \frac{2}{1} = 2 (\text{Kecepatan Sangat Lambat})$$

$$\mu(X) = \frac{x - a}{b - a} = \frac{2 - 1}{2 - 1} = \frac{1}{1} = 1 (\text{Kecepatan Lambat})$$

- b. Untuk proses fuzzifikasi pada jarak dapat dilihat pada grafik Gambar 4.10 yang merupakan kurva linear naik dan segitiga.

Fuzzifikasi Jarak (0,4m berada pada derajat keanggotaan Dekat)

$x = 0,4$; $a = 0,2$ dan $b = 0,5$

Menghitung kurva linear segitiga dari derajat keanggotaan Dekat.

$$\mu(X) = \frac{b - x}{b - a} = \frac{0,5 - 0,4}{0,5 - 0,3} = \frac{0,1}{0,2} = 0,5 \text{ (Dekat)}$$

3.5.2 Proses Aturan *Fuzzy logic*

Langkah kedua dalam proses logika *fuzzy* yaitu proses aturan *fuzzy*, pada langkah ini menggunakan logika *fuzzy* Sugeno. Hasil langkah proses logika *fuzzy* ini berdasarkan aturan yang sudah dijelaskan pada Tabel 3.5

Tabel 3.5 Perhitungan Aturan *Fuzzy Logic* Sugeno

Aturan	Variabel Input	Variabel Output
[R1]	Jika Kecepatan Sangat Lambat Dan Jarak Sangat Dekat $\alpha \text{ Predikat1} = \min (\mu_{\text{SangatLambat}}(X) \cap \mu_{\text{SangatDekat}}(X))$ $= \min(2;0)$ $= 0$	Maka Akurasi Tinggi $\alpha \text{ Predikat1} = \frac{z-50}{100-25}$ $0 = \frac{z-50}{100-25}$ $0 = \frac{z-50}{75}$ $0 \cdot 50 = z - 75$ $0 = z - 75$ $0 + 75 = z$ $z = 75$ $z1 = 75$
[R2]	Jika Kecepatan Sangat Lambat Dan Jarak Dekat $\alpha \text{ Predikat1} = \min (\mu_{\text{SangatLambat}}(X) \cap \mu_{\text{Dekat}}(X))$ $= \min(2;0,5)$ $= 0,5$	Maka Akurasi Tinggi $\alpha \text{ Predikat1} = \frac{z-50}{100-25}$ $0 = \frac{z-50}{100-25}$ $0 \cdot 50 = z - 75$ $0 = z - 75$ $0 + 75 = z$ $z = 75$ $z2 = 75$

Aturan	Variabel Input	Variabel Output
[R3]	Jika Kecepatan Sangat Lambat Dan Jarak Sedang $\alpha \text{ Predikat1} = \min (\mu_{\text{SangatLambat}}(X) \cap \mu_{\text{Sedang}}(X))$ $= \min(2;0)$ $= 0$	Maka Akurasi Sedang $\alpha \text{ Predikat1} = \frac{z-50}{75-25}$ $0 = \frac{z-50}{75-25}$ $0 = \frac{z-50}{50}$ $0 \cdot 50 = z - 50$ $0 = z - 50$ $0 + 50 = z$ $z = 50$ $z3 = 50$
[R4]	Jika Kecepatan Sangat Lambat Dan Jarak Jauh $\alpha \text{ Predikat1} = \min (\mu_{\text{SangatLambat}}(X) \cap \mu_{\text{Jauh}}(X))$ $= \min(2;0)$ $= 0$	Maka Akurasi Rendah $\alpha \text{ Predikat1} = \frac{z-25}{50-25}$ $0 = \frac{z-25}{50-25}$ $0 = \frac{z-25}{25}$ $0 \cdot 25 = z - 25$ $0 = z - 25$ $0 + 25 = z$ $z = 25$ $z3 = 25$
[R5]	Jika Kecepatan Sangat Lambat Dan Jarak Sangat Jauh $\alpha \text{ Predikat1} = \min (\mu_{\text{SangatLambat}}(X) \cap \mu_{\text{SangatJauh}}(X))$ $= \min(2;0)$ $= 0$	Maka Akurasi Rendah $\alpha \text{ Predikat1} = \frac{z-25}{50-25}$ $0 = \frac{z-25}{50-25}$ $0 = \frac{z-25}{25}$ $0 \cdot 25 = z - 25$ $0 = z - 25$ $0 + 25 = z$ $z = 25$ $z3 = 25$

Aturan	Variabel Input	Variabel Output
[R6]	Jika Kecepatan Lambat Dan Jarak Sangat Dekat $\alpha \text{ Predikat1} = \min (\mu_{\text{Lambat}}(X) \cap \mu_{\text{SangatDekat}}(X))$ $= \min(1;0)$ $= 0$	Maka Akurasi Tinggi $\alpha \text{ Predikat1} = \frac{z-50}{100-25}$ $0 = \frac{z-50}{100-25}$ $0 = \frac{z-50}{75}$ $0 \cdot 50 = z - 75$ $0 = z - 75$ $0 + 75 = z$ $z = 75$ $z_2 = 75$
[R7]	Jika Kecepatan Lambat Dan Jarak Dekat $\alpha \text{ Predikat1} = \min (\mu_{\text{Lambat}}(X) \cap \mu_{\text{Dekat}}(X))$ $= \min(1;0,5)$ $= 0$	Maka Akurasi Tinggi $\alpha \text{ Predikat1} = \frac{z-50}{100-25}$ $0 = \frac{z-50}{100-25}$ $0 = \frac{z-50}{75}$ $0 \cdot 50 = z - 75$ $0 = z - 75$ $0 + 75 = z$ $z = 75$ $z_2 = 75$
[R8]	Jika Kecepatan Lambat Dan Jarak Sedang $\alpha \text{ Predikat1} = \min (\mu_{\text{Lambat}}(X) \cap \mu_{\text{Sedang}}(X))$ $= \min(1;0)$ $= 0$	Maka Akurasi Sedang $\alpha \text{ Predikat1} = \frac{z-50}{75-25}$ $0 = \frac{z-50}{75-25}$ $0 = \frac{z-50}{50}$ $0 \cdot 50 = z - 50$ $0 = z - 50$ $0 + 50 = z$ $z = 50$ $z_3 = 50$

Aturan	Variabel Input	Variabel Output
[R9]	Jika Kecepatan Lambat Dan Jarak Jauh $\alpha \text{ Predikat1} = \min (\mu_{\text{Lambat}}(X) \cap \mu_{\text{Jauh}}(X))$ $= \min(1; 0)$ $= 0$	Maka Akurasi Rendah $\alpha \text{ Predikat1} = \frac{z-25}{50-25}$ $0 = \frac{z-25}{50-25}$ $0 = \frac{z-25}{25}$ $0 \cdot 25 = z - 25$ $0 = z - 25$ $0 + 25 = z$ $z = 25$ $z3 = 25$
[R10]	Jika Kecepatan Lambat Dan Jarak Sangat Jauh $\alpha \text{ Predikat1} = \min (\mu_{\text{Lambat}}(X) \cap \mu_{\text{SangatJauh}}(X))$ $= \min(1; 0)$ $= 0$	Maka Akurasi Rendah $\alpha \text{ Predikat1} = \frac{z-25}{50-25}$ $0 = \frac{z-25}{50-25}$ $0 = \frac{z-25}{25}$ $0 \cdot 25 = z - 25$ $0 = z - 25$ $0 + 25 = z$ $z = 25$ $z3 = 25$
[R11]	Jika Kecepatan Sedang Dan Jarak Sangat Dekat $\alpha \text{ Predikat1} = \min (\mu_{\text{Sedang}}(X) \cap \mu_{\text{SangatDekat}}(X))$ $= \min(0; 0)$ $= 0$	Maka Akurasi Tinggi $\alpha \text{ Predikat1} = \frac{z-50}{100-25}$ $0 = \frac{z-50}{100-25}$ $0 = \frac{z-50}{75}$ $0 \cdot 50 = z - 75$ $0 = z - 75$ $0 + 75 = z$ $z = 75$ $z2 = 75$

Aturan	Variabel Input	Variabel Output
[R12]	Jika Kecepatan Sedang Dan Jarak Dekat $\alpha \text{ Predikat1} = \min (\mu_{\text{Sedang}}(X) \cap \mu_{\text{Dekat}}(X))$ $= \min(0; 0,5)$ $= 0$	Maka Akurasi Tinggi $\alpha \text{ Predikat1} = \frac{z-50}{100-25}$ $0 = \frac{z-50}{100-25}$ $0 = \frac{z-50}{75}$ $0 \cdot 50 = z - 75$ $0 = z - 75$ $0 + 75 = z$ $z = 75$ $z2 = 75$
[R13]	Jika Kecepatan Sedang Dan Jarak Sedang $\alpha \text{ Predikat1} = \min (\mu_{\text{Sedang}}(X) \cap \mu_{\text{SangatSedang}}(X))$ $= \min(0; 0)$ $= 0$	Maka Akurasi Sedang $\alpha \text{ Predikat1} = \frac{z-50}{75-25}$ $0 = \frac{z-50}{75-25}$ $0 = \frac{z-50}{50}$ $0 \cdot 50 = z - 50$ $0 = z - 50$ $0 + 50 = z$ $z = 50$ $z3 = 50$
[R14]	Jika Kecepatan Sedang Dan Jarak Jauh $\alpha \text{ Predikat1} = \min (\mu_{\text{Sedang}}(X) \cap \mu_{\text{Jauh}}(X))$ $= \min(0; 0)$ $= 0$	Maka Akurasi Rendah $\alpha \text{ Predikat1} = \frac{z-25}{50-25}$ $0 = \frac{z-25}{50-25}$ $0 = \frac{z-25}{25}$ $0 \cdot 25 = z - 25$ $0 = z - 25$ $0 + 25 = z$ $z = 25$ $z3 = 25$

Aturan	Variabel Input	Variabel Output
[R15]	Jika Kecepatan Sedang Dan Jarak Sangat Jauh $\alpha \text{ Predikat1} = \min (\mu_{\text{Sedang}}(X) \cap \mu_{\text{SangatJauh}}(X))$ $= \min(0;0)$ $= 0$	Maka Akurasi Rendah $\alpha \text{ Predikat1} = \frac{z-25}{50-25}$ $0 = \frac{z-25}{50-25}$ $0 = \frac{z-25}{25}$ $0 \cdot 25 = z - 25$ $0 = z - 25$ $0 + 25 = z$ $z = 25$ $z3 = 25$
[R16]	Jika Kecepatan Cepat Dan Jarak Sangat Dekat $\alpha \text{ Predikat1} = \min (\mu_{\text{Cepat}}(X) \cap \mu_{\text{SangatDekat}}(X))$ $= \min(0;0)$ $= 0$	Maka Akurasi Sedang $\alpha \text{ Predikat1} = \frac{z-50}{75-25}$ $0 = \frac{z-50}{75-25}$ $0 = \frac{z-50}{50}$ $0 \cdot 50 = z - 50$ $0 = z - 50$ $0 + 50 = z$ $z = 50$ $z3 = 50$
[R17]	Jika Kecepatan Cepat Dan Jarak Dekat $\alpha \text{ Predikat1} = \min (\mu_{\text{Cepat}}(X) \cap \mu_{\text{Dekat}}(X))$ $= \min(0;0,5)$ $= 0$	Maka Akurasi Sedang $\alpha \text{ Predikat1} = \frac{z-50}{75-25}$ $0 = \frac{z-50}{75-25}$ $0 = \frac{z-50}{50}$ $0 \cdot 50 = z - 50$ $0 = z - 50$ $0 + 50 = z$ $z = 50$ $z3 = 50$

Aturan	Variabel Input	Variabel Output
[R18]	Jika Kecepatan Cepat Dan Jarak Sedang $\alpha \text{ Predikat1} = \min (\mu_{\text{Cepat}}(X) \cap \mu_{\text{Sedang}}(X))$ $= \min(0; 0)$ $= 0$	Maka Akurasi Rendah $\alpha \text{ Predikat1} = \frac{z-25}{50-25}$ $0 = \frac{z-25}{50-25}$ $0 = \frac{z-25}{25}$ $0 \cdot 25 = z - 25$ $0 = z - 25$ $0 + 25 = z$ $z = 25$ $z3 = 25$
[R19]	Jika Kecepatan Cepat Dan Jarak Jauh $\alpha \text{ Predikat1} = \min (\mu_{\text{Cepat}}(X) \cap \mu_{\text{Jauh}}(X))$ $= \min(0; 0)$ $= 0$	Maka Akurasi Rendah $\alpha \text{ Predikat1} = \frac{z-25}{50-25}$ $0 = \frac{z-25}{50-25}$ $0 = \frac{z-25}{25}$ $0 \cdot 25 = z - 25$ $0 = z - 25$ $0 + 25 = z$ $z = 25$ $z3 = 25$
[R20]	Jika Kecepatan Cepat Dan Jarak Sangat Jauh $\alpha \text{ Predikat1} = \min (\mu_{\text{Cepat}}(X) \cap \mu_{\text{SangatJauh}}(X))$ $= \min(0; 0)$ $= 0$	Maka Akurasi Rendah $\alpha \text{ Predikat1} = \frac{z-25}{50-25}$ $0 = \frac{z-25}{50-25}$ $0 = \frac{z-25}{25}$ $0 \cdot 25 = z - 25$ $0 = z - 25$ $0 + 25 = z$ $z = 25$ $z3 = 25$

Aturan	Variabel Input	Variabel Output
[R21]	Jika Kecepatan Sangat Cepat Dan Sangat Dekat $\alpha \text{ Predikat1} = \min (\mu_{\text{SangatCepat}}(X) \cap \mu_{\text{SangatDekat}}(X))$ $= \min(0;0)$ $= 0$	Maka Akurasi Sedang $\alpha \text{ Predikat1} = \frac{z-50}{75-25}$ $0 = \frac{z-50}{75-25}$ $0 = \frac{z-50}{50}$ $0 \cdot 50 = z - 50$ $0 = z - 50$ $0 + 50 = z$ $z = 50$ $z3 = 50$
[R22]	Jika Kecepatan Sangat Cepat Dan Dekat $\alpha \text{ Predikat1} = \min (\mu_{\text{SangatCepat}}(X) \cap \mu_{\text{Dekat}}(X))$ $= \min(0;0,5)$ $= 0$	Maka Akurasi Rendah $\alpha \text{ Predikat1} = \frac{z-25}{50-25}$ $0 = \frac{z-25}{50-25}$ $0 = \frac{z-25}{25}$ $0 \cdot 25 = z - 25$ $0 = z - 25$ $0 + 25 = z$ $z = 25$ $z3 = 25$
[R23]	Jika Kecepatan Sangat Cepat Dan Sedang $\alpha \text{ Predikat1} = \min (\mu_{\text{SangatCepat}}(X) \cap \mu_{\text{Sedang}}(X))$ $= \min(0;0)$ $= 0$	Maka Akurasi Rendah $\alpha \text{ Predikat1} = \frac{z-25}{50-25}$ $0 = \frac{z-25}{50-25}$ $0 = \frac{z-25}{25}$ $0 \cdot 25 = z - 25$ $0 = z - 25$ $0 + 25 = z$ $z = 25$ $z3 = 25$

Aturan	Variabel Input	Variabel Output
[R24]	Jika Kecepatan SangatCepat Dan Jarak Jauh $\alpha \text{ Predikat1} = \min (\mu_{\text{Lambat}}(X) \cap \mu_{\text{Jauh}}(X))$ $= \min(0;0)$ $= 0$	Maka Akurasi Rendah $\alpha \text{ Predikat1} = \frac{z-25}{50-25}$ $0 = \frac{z-25}{50-25}$ $0 = \frac{z-25}{25}$ $0 \cdot 25 = z - 25$ $0 = z - 25$ $0 + 25 = z$ $z = 25$ $z3 = 25$
[R25]	Jika Kecepatan SangatCepat Dan Jarak Sangat Jauh $\alpha \text{ Predikat1} = \min (\mu_{\text{Lambat}}(X) \cap \mu_{\text{SangatJauh}}(X))$ $= \min(0;0)$ $= 0$	Maka Akurasi Rendah $\alpha \text{ Predikat1} = \frac{z-25}{50-25}$ $0 = \frac{z-25}{50-25}$ $0 = \frac{z-25}{25}$ $0 \cdot 25 = z - 25$ $0 = z - 25$ $0 + 25 = z$ $z = 25$ $z3 = 25$

Berdasarkan Tabel 3.6 merupakan perhitungan aturan logika *fuzzy* Sugeno dengan mencari nilai minimum α -predikat dari setiap aturan yang sudah ada. Nilai z merupakan nilai tingkat akurasi berdasarkan Tabel 3.5. Berikut merupakan Hasil Perhitungan Aturan logika *fuzzy* Sugeno.

Tabel 3.6 Hasil Perhitungan Aturan *Fuzzy Logic* Sugeno

No	Jika		Maka
	Kecepatan	Jarak	
1	Sangat Lambat (1)	Sangat Dekat (0)	Tinggi ($z_1=75$)
2	Sangat Lambat (1)	Dekat (0,5)	Tinggi ($z_2=75$)
3	Sangat Lambat (1)	Sedang (0)	Sedang ($z_3=50$)
4	Sangat Lambat (1)	Jauh (0)	Rendah ($z_4=25$)
5	Sangat Lambat (1)	Sangat Jauh (0)	Rendah ($z_5=25$)
6	Lambat (0)	Sangat Dekat (0)	Tinggi ($z_6=75$)
7	Lambat (0)	Dekat (0,5)	Tinggi ($z_7=75$)
8	Lambat (0)	Sedang (0)	Sedang ($z_8=50$)
9	Lambat (0)	Jauh (0)	Rendah ($z_9=25$)
10	Lambat (0)	Sangat Jauh (0)	Rendah ($z_{10}=25$)
11	Sedang (0)	Sangat Dekat (0)	Tinggi ($z_{11}=75$)
12	Sedang (0)	Dekat (0,5)	Tinggi ($z_{12}=75$)
13	Sedang (0)	Sedang (0)	Sedang ($z_{13}=50$)
14	Sedang (0)	Jauh (0)	Rendah ($z_{14}=25$)
15	Sedang (0)	Sangat Jauh (0)	Rendah ($z_{15}=25$)
16	Cepat (0)	Sangat Dekat (0)	Sedang ($z_{16}=50$)
17	Cepat (0)	Dekat (0,5)	Sedang ($z_{17}=50$)
18	Cepat (0)	Sedang (0)	Rendah ($z_{18}=25$)
19	Cepat (0)	Jauh (0)	Rendah ($z_{19}=25$)
20	Cepat (0)	Sangat Jauh (0)	Rendah ($z_{20}=25$)
21	Sangat Cepat (0)	Sangat Dekat (0)	Sedang ($z_{21}=50$)
22	Sangat Cepat (0)	Dekat (0,5)	Rendah ($z_{22}=25$)
23	Sangat Cepat (0)	Sedang (0)	Rendah ($z_{23}=25$)
24	Sangat Cepat (0)	Jauh (0)	Rendah ($z_{24}=25$)
25	Sangat Cepat (0)	Sangat Jauh (0)	Rendah ($z_{25}=25$)

3.5.3 Defuzzifikasi

Langkah selanjutnya yaitu defuzzifikasi menggunakan metode rata-rata tertimbang (Weight of average). Adapun perhitungannya dapat dilakukan menggunakan Persamaan 2.2

$$\begin{aligned}
 Crisp &= \frac{\text{Total Tertimbang}}{\text{Total Bobot}} \\
 &= \frac{\sum_{i=1}^{25} (Z_i \times a_i)}{\sum_{i=1}^{25} a_i} \\
 &= \frac{(Z_1 \times a_1) + (Z_2 \times a_2)(Z_3 \times a_3) (Z_4 \times a_4)(Z_5 \times a_5)(Z_6 \times a_6)(Z_7 \times a_7) \\
 &\quad (Z_8 \times a_8)(Z_9 \times a_9)(Z_{10} \times a_{10})(Z_{11} \times a_{11})(Z_{12} \times a_{12})(Z_{13} \times a_{13})(Z_{14} \times a_{14}) \\
 &\quad (Z_{15} \times a_{15})(Z_{16} \times a_{16})(Z_{17} \times a_{17})(Z_{18} \times a_{18})(Z_{19} \times a_{19})(Z_{20} \times a_{20}) \\
 &\quad (Z_{21} \times a_{21})(Z_{22} \times a_{22})(Z_{23} \times a_{23})(Z_{24} \times a_{24}) \\
 &\quad (Z_{25} \times a_{25})}{(a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8 + \dots + a_{25})} \\
 &= \frac{(75 \times 0) + (75 \times 0,5) + (50 \times 0) + (25 \times 0) + (25 \times 0) + (75 \times 0) + (75 \times 0,5) \\
 &\quad + (50 \times 0) + (25 \times 0) + (25 \times 0) + (75 \times 0) + (75 \times 0) + (50 \times 0) + (25 \times 0) \\
 &\quad + (25 \times 0) + (50 \times 0) + (50 \times 0) + (25 \times 0) + (25 \times 0) + (25 \times 0) + (50 \times 0) \\
 &\quad + (25 \times 0) + (25 \times 0) + (25 \times 0) + (25 \times 0)}{(0 + 0,5 + 0 + 0 + 0 + 0 + 0 + 0,5 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0)} \\
 &= 75
 \end{aligned}$$

Pada nilai akurasi ini menunjukan berada pada nilai akurasi sebesar 75%. Untuk kurva tingkat akurasi dapat dilihat pada Gambar 3.6 yang merupakan derajat keanggotaan tingkat akurasi robot.

BAB V

PENUTUP

5.1 Kesimpulan

Adapun kesimpulan yang diperoleh berdasarkan penelitian ini ialah sebagai berikut:

1. Telah terealisasikan perancangan simulasi robot bergerak manipulator menggunakan kontroler logika *fuzzy* Sugeno, tingkat akurasi robot bergerak manipulator diengaruhi oleh jarak dan kecepatan ketika akan melakukan tindakan baik meletakkan atau mengambil objek. Tingkat keseluruhan akurasi dari pengujian didapat sebesar 84,6% dari 26 data uji yang diperoleh.
2. Telah terealisasikan pelacakan titik koordinat posisi robot bergerak manipulator selama berlangsungnya simulasi secara *realtime*, dengan durasi simulasi 2 menit 57 detik dan dihasilkan sebanyak 257 data orientasi posisi robot bergerak manipulator.

5.2 Saran

Adapun saran yang dapat diajukan untuk penelitian selanjutnya pada penelitian ini ialah sebagai berikut:

1. Melakukan pengembangan implementasi alat untuk melihat tingkat akurasi robot manipulator bergerak menggunakan metode logika *fuzzy* sugeno, apakah metode serta *Task Planning* yang telah dibuat berfungsi sebagaimana mestinya serta untuk memastikan nilai *error* yang lebih real.
2. Melakukan kontrol penuh dari setiap *link* dan *joint* pada robot *Degree Of Freedom* (DOF) pada lengan robot agar dapat beroperasi dengan baik,

DAFTAR PUSTAKA

- [1] Tresna Dewi, Yumi Oktarina, Pola Risma dan Sari Kartini “Desain Robot Pengikut Manusia Sederhana dengan *Fuzzy logic Controller*” Politeknik Sriwijaya Palembang, 2019.
- [2] G. Dewantoro, D. Susilo, and P. Pideksa Adi, “Implementasi Pengendali Logika *Fuzzy* pada Navigasi Robot Penjejak Dinding,” *Teknol. Elektro*, vol. 16, no. 02, 2017.
- [3] A. Priswanto and J. Prasajo, “Sistem Kendali Hybrid PID-FLC Pada Studi Kasus Robot Penjejak Dinding Berkaki,” 2020.
- [4] Kusumadewi, S., dan Purnomo, H., “Aplikasi Logika Fuzzzy untuk Pendukung Keputusan”, Graha Ilmu, Yogyakarta, 2010.
- [5] Hisyam M. Fuad AL Azka, Muhammad Syariffudien Zuhrie , dkk. “Rancang Bangun Sistem Potitioning Mobile Robot Omnidirectional wheel Menggunakan STM32 Berbasis *Fuzzy logic Controller*” *Teknol. Elektro*, vol 10, no. 03, 2021.
- [6] Qian, Jia, and Chao chang. 2019. *Motion Control Of Omnidirectional Mobile Robot Based On Fuzzy Pid*. Chinese Control And Decision Conference (Ccdc) : 5149-5154
- [7] Putri Repina Kesuma, Tresna Dewi, dkk. “Desain *Fuzzy logic Controller* Untuk Pengendali Pergerakan Mobile Manipulator” Politeknik Negeri Sriwijaya, 2020.
- [8] Hendrix R., Owan P., Garbini J., and Devasia S., 2019. Context-Specific Separable Gesture Selection for Control of a Robotic Manufacturing Assistant. *IFAC PapersOnline*, 51(34), pp. 89-96.

- [9] Dewi T., Nurmaini S., Risma P., Oktarina Y., and Roriz M., 2019, Inverse Kinematic Analysis of 4 DOF Pick and Place Arm Robot Manipulator using *Fuzzy logic* Controller, IJECE, 10(2), pp. 1376-1386. doi:10.11591/ijece.v10i2.pp1376-1386.
- [10] Dewi T., Amperawan, Risma P., Oktarina Y., and Yudha D. A., 2020, Finger Cue for Mobile Robot Motion Control, Computer Engineering and Application Journal, 9(1), pp. 39-48. doi: 10.18495/COMENGAPP.V9I1.319.
- [11] Oktarina Y., Dewi T., and Risma T., 2020, The Concept of Automatic Transport System Utilizing Weight Sensor, Computer Engineering and Application Journal, 9(2), pp. 155-163. doi:10.18495/COMENGAPP.V0I0.339
- [12] Dewi T., Risma P., Taqwa A., Rusdianasari, and Renaldi H., 2020, Experimental analysis on solar powered mobile robot as the prototype for environmentally friendly automated transportation, Proc. iCAST on Engineering Science, 24-25 Oct 2019, Bali: Indonesia, doi:10.1088/1742-6596/1450/1/012034.
- [13] Dewi T., Risma P., and Oktarina Y., 2020, Fruit Sorting Robot based on Color and Size for an Agricultural Product Packaging System, Bulletin of Electrical Engineering, and Informatics (BEEI), 9(4), pp. 1438-1445. doi: 10.11591/eei.v9i4.2353
- [14] Oktarina Y., Dewi T., Risma P., and Nawawi M., 2020, Tomato Harvesting Arm Robot Manipulator; a Pilot Project, Journal of Physics: Conference Series, 1500, p 012003, Proc. 3rd FIRST, Palembang: Indonesia.
- [15] Dewi T., Oktarina Y., Risma P., and Kartini S., 2019, Desain Robot Pengikut Manusia Sederhana dengan *Fuzzy logic* Controller, Proc. Annual Research Seminar (ARS), 5(1), pp. 12-16, 16 Nov 2019, Palembang: Indonesia.
- [16] Dewi T., Sitompul C., Risma P., Oktarina Y., Jelista R., Mulyati M., 2019, Simulation Analysis of Formation Control Design of Leader-Follower

- Robot Using *Fuzzy logic* Controller, Proc 2019 ICECOS, 2-3 Oct. 2019, Batam Island: Indonesia. doi:10.1109/ICECOS47637.2019.8984433.
- [17] Yudha H. M., Dewi T., Hasana N., Risma P., Oktarina, Y. Kartini S., 2019, Performance Comparison of *Fuzzy logic* and Neural Network Design for Mobile Robot Navigation, Proc. 2019 ICECOS, 2-3 Oct. 2019, Batam Island: Indonesia. doi:10.1109/ICECOS47637.2019.8984577.
- [18] Risma P., Dewi T., Oktarina Y., and Wijanarko Y., 2019. Neural Network Controller Application on a Visual based Object Tracking and Following Robot. Computer Engineering and Application Journal, 10.18495/COMENGAPP.V8I1.280.
- [19] Kuswad S., Natasya A., Tamara M.N., and Adji I., 2018. Optimasi Sistem Navigasi Robot Bencana Dengan Algoritma Bug Dan Jaringan Syaraf Tiruan. JTIK, . 5(5), pp. 635-642.
- [20] Hardin S. Nasution, Akhmad Jayadi, dan Rikendry., 2022, Implementasi Metode *Fuzzy logic* Untuk Sistem Pengereman Robot Mobile Berdasarkan Jarak dan Kecepatan. JTIKOM vol 3, no. 1, 2022.
- [21] Jayadi, A., Susanto, T. and Adhinata, F.D., 2021. Sistem Kendali Proporsional pada Robot Penghindar Halangan (Avoider) Pioneer P3-DX. Majalah Ilmiah Teknologi Elektro, 20(1), pp.47-52.
- [22] Rahmat, B.F., Basjaruddin, N.C. and Saefudin, D., 2019, August. Sistem Penghindar Tabrakan Depan-Belakang Kooperatif Berbasis Logika *Fuzzy*. In Prosiding Industrial Research Workshop and National Seminar (Vol. 10, No. 1, pp. 155-161).
- [23] Sulistiani, H., Muludi, K. and Syarif, A., 2021. Implementation Of Various Artificial Intelligence Approach For Prediction And Recommendation Of Personality Disorder Patient. In Journal of Physics: Conference Series (Vol. 1751, No. 1, p. 012040). IOP Publishing.
- [24] Rossi, F., Sembiring, J.P., Jayadi, A., Putri, N.U. and Nugroho, P., 2021, October. Implementation of *Fuzzy logic* in PLC for Three-Story Elevator

Control System. In 2021 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE) (pp. 179-185). IEEE.

- [25] Muludi, K., Syarif, A. and Wantoro, A., 2021. Implementation of *Fuzzy*-based Model for Prediction of Prostate Cancer. In Journal of Physics: Conference Series (Vol. 1751, No. 1, p. 012041). IOP Publishing.