PERANCANHAN SISTEM KENDALI KINEMATIKA GERAK BERBASIS FUZZY LOGIC DAN FUSI SENSOR (ODOMETRY & IMU) UNTUK ADAPTASI MEDAN PADA ROBOT PEMOTONG RUMOUT OTONOM

Prototipe

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana.

Oleh :

Nicodemus Oloan Simanjuntak

2305110016

PROGRAM STUDI TEKNIK KOMPUTER

FAKULTAS TEKNIK

UNIVERSITAS NEGERI SEMARANG

SEMARANG, TAHUN 2026

PERSETUJUAN PEMBIMBING

Prototipe berjudul "**PERANCANGAN SISTEM KENDALI KINEMATIKA GERAK BERBASIS FUZZY LOGIC DAN FUSI SENSOR (ODOMETRY & IMU) UNTUK ADAPTASI MEDAN PADA ROBOT PEMOTONG RUMPUT OTONOM**" yang disusun oleh

Nama	: Nicodemus Oloan Simanjuntak
NIM	: 2305110016
Prodi/Fakultas	: Teknik Komputer/Teknik

telah disetujui untuk diajukan ke penilaian proposal prototipe.

Semarang, 25 September 2026
Pembimbing

Dr. Feddy Setio Pribadi, S.Pd., M.T.
197808222003121002

DAFTAR ISI

PERSETUJUAN PEMBIMBING ii
DAFTAR ISI iii
DAFTAR TABEL iv
DAFTAR GAMBAR vi
DAFTAR PERSAMAAN viii
BAB I 1
PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Tujuan Prototipe 4
1.4 Manfaat Prototipe 4
1.4.1 Bagi Penulis 4
1.4.2 Bagi Universitas Negeri Semarang 5
1.4.3 Bagi Pengelola Ruang Terbuka Hijau dan Masyarakat 5
1.4.4 Bagi Lingkungan 5
1.4.5 Bagi Industri 5
1.5 Potensi Dampak Fungsional/Komersial 6
BAB II 9
KAJIAN PUSTAKA 9
2.1 Tinjauan Pustaka 9
2.1.1 Identifikasi Kesenjangan Penelitian *(Research Gap)* dan Kebaruan Prototipe 12
2.2 Landasan Teoritik 13
2.2.1 Sistem Gerak Sabuk Rantai dan Kinematika *Skid-Steering* 13
2.2.2 Odometri Berbasis *Rotary Encoder* 16
2.2.3 Inertial Measurement Unit MPU-9250 18
2.2.4 Fusi Odometri dan IMU untuk Estimasi Gerak Lokal 20
2.2.5 Sistem Penentuan Posisi Global GNSS dan Protokol NMEA 0183 22
2.2.6 Logika Penghindaran Halangan Dinamis (Smart Dodge) dan State Machine Navigasi 24
2.2.7 Kendali Kinematika Berbasis Fuzzy Sugeno 25
BAB III 29
METODE PELAKSANAAN 29

3.1 Pendekatan Perancangan 29
3.2 Lokasi dan Waktu 30
3.2.1 Lokasi Perancangan dan Pengujian 30
3.2.2 Area dan Kondisi Pengujian Lapangan 30
3.2.3 Waktu dan Jadwal Pelaksanaan 32
3.3 Prosedur Perancangan 33
3.3.1 Analisis Kebutuhan 33
3.3.2 Perancangan Sistem 34
3.3.3 Implementasi 36
3.3.4 Pengujian Sistem 37
3.3.5 Pemeliharaan 38
3.4 Teknik Perancangan 39
3.4.1 Pemilihan Komponen dan Arsitektur Sistem 39
3.4.2 Desain Fisik 43
3.4.3 Desain Pengawatan Sistem 48
3.4.4 Perancangan Model Fuzzy Logic 50
3.5 Teknik Uji Kelayakan Hasil 72
3.5.1 Pengujian Sensor dan Karateristik Sistem Penggerak 73
3.5.2 Pengujian Ketelitian Estimasi Odometri-IMU 74
3.5.3 Pengujian Pengendali Fuzzy Sugeno dan Pelacakan Lintasan 76
3.5.4 Matriks Indikator Capaian Kuantitatif 78
DAFTAR PUSTAKA 80
LAMPIRAN 83

DAFTAR TABEL

Tabel 1.1 Perbandingan robot pemotong rumput komersial dengan prototipe. 8
Tabel 2.1. Perbandingan penelitian terdahulu dengan rancangan subsistem kendali gerak 11
Tabel 1.2 Matriks Komparasi Penelitian Terdahulu dan Posisi Prototipe 12
Tabel 2.2 Hubungan kecepatan kedua sisi penggerak dan arah Gerak robot.....14
Tabel 2.3 Peran sensor MPU-9250 pada subsistem kendali Gerak 18
Tabel 2.4 Peran data dalam estimasi Gerak lokal 20
Tabel 2.5 Data posisi global yang diperlukan subsistem navigasi.....23
Tabel 2.6 Status navigasi dan respons subsistem kendali gerak 25
Tabel 2.7 Variabel kendali kinematika FIS 2.....26
Tabel 2.8 Contoh arah koreksi pada aturan FIS 2 27
Tabel 3.1 Penerapan pendekatan perancangan pada subsistem kendali gerak.....30
Tabel 3.2 Lokasi dan fungsi kegiatan perancangan 30
Tabel 3.3 Data kondisi Lokasi yang dicatat sebelum pengujian.....31
Tabel 3.4 Rencana waktu perancangan subsistem kendali gerak 33
Tabel 3.5 Kebutuhan perangkat keras dan perangkat lunak subsistem kendali gerak 34
Tabel 3.6. Spesifikasi ESP32 Dev Board.....40
Tabel 3.7 Spesifikasi dan pengaturan driver BTS7960.....41
Tabel 3.8 Spesifikasi motor wiper depan Canter MK461340.....42
Tabel 3.9 Spesifikasi IMU MPU-9250 42
Tabel 3.10 Spesifikasi rotary encoder 1000 PPR.....43
Tabel 3.11. Material rancangan fisik 47
Tabel 3.12 Rancangan alokasi sambungan dan pin subsistem penggerak 50
Tabel 3.13 Hubungan input-output FIS 1, FIS 2, FSM.....51
Tabel 3.14 Variabel masukan dan keluaran pengendali fuzzy.....53
Tabel 3.15 Hubungan kondisi keluaran dan makna perintah gerak 55
Tabel 3.16 Hubungan status FSM dengan pengendali fuzzy dan penggerak robot 56
Tabel 3.17 Masukan dan keluaran FIS 1.....58
Tabel 3.18 Parameter fungsi keanggotaan masukan FIS 1 58
Tabel 3.19 Konsekuen dan aturan FIS 1 59
Tabel 3.20 Parameter fungsi keanggotaan galat arah.....60

Tabel 3.21 Parameter fungsi keanggotaan galat lateral 60
Tabel 3.22 Konsekuen keluaran PWM FIS 2.....61
Tabel 3.23 Logika FSM subsistem kendali gerak.....62
Tabel 3.24 Basis aturan FIS 1 untuk perlambatan 63
Tabel 3.25 Basis aturan FIS 2 untuk pelacakan lintasan.....66
Tabel 3.26 Rencana pemeriksaan model fuzzy pada MATLAB 67
Tabel 3.27 Hasil simulasi FIS 1 pada seluruh domain masukan 68
Tabel 3.28 Hasil simulasi FIS 2 dengan kondisi galat nol.....69
Tabel 3.29 Hasil simulasi FIS 2 dengan variasi galat arah dan galat lateral nol....69
Tabel 3.30 Hasil simulasi FIS 2 dengan galat arah nol dan variasi galat lateral....70
Tabel 3.31 Hasil simulasi FIS 2 pada kombinasi galat arah dan galat lateral.....71
Tabel 3.32 Rencana pengujian rotary encoder.....73
Tabel 3.33 Rencana pengujian MPU-9250 74
Tabel 3.34 Format pencatatan karakterisasi sistem penggerak.....74
Tabel 3.35 Skenario pengujian odometri-IMU 75
Tabel 3.36 Kondisi dan parameter pencatatan pengujian 75
Tabel 3.37 Indikator evaluasi estimasi gerak local 76
Tabel 3.38 Format rekapitulasi hasil pengujian odometri-IMU.....76
Tabel 3.39 Acuan pengujian keluaran FIS 1 76
Tabel 3.40 Acuan pengujian kombinasi masukan FIS 2.....77
Tabel 3.41 Rencana pemeriksaan keseuaian implementasi pengendali.....77
Tabel 3.42 Skenario pengujian pelacakan lintasan 78
Tabel 3.43 Indikator evaluasi pelacakan lintasan 78
Tabel 3.44 Matriks indicator capaian kuantitatif subsistem kendali gerak.....79
Tabel 3.45 Format rekapitulasi penilaian kalayakan hasil 79

DAFTAR GAMBAR

Gambar 3.1 Alur pendekatan perancangan subsistem kendali gerak.....29
Gambar 3.2 Tampak atas Lapangan Woodball UNNES.....32
Gambar 3.3 Digram alir prosedur perancangan subsistem kendali Gerak.....39
Gambar 3.4 ESP32 Dev Board berbasis ESP-WROOM-32.....40
Gambar 3.5 Modul driver motor BTS7960.....40
Gambar 3.6 Motor Driver depan Canter MK461340.....41
Gambar 3.7 Modul IMU MPU-9250.....42
Gambar 3.8 Rotary Encdoder 100 PPR.....43
Gambar 3.9 Tampak atas desain robot.....44
Gambar 3.10 Tampak bawah desain robot.....45
Gambar 3.11 Tampak samping desain robot.....45
Gambar 3.12 Tampak depan desain robot.....46
Gambar 3.13 Tampak belakang desain robot.....46
Gambar 3.14 Tampak keseluruhan desain robot.....47
Gambar 3.15 Desain wiring subsistem penggerak.....48
Gambar 3.16 Arsitektur kendali gerak menggunakan FIS 1, FIS 2, dan Fusi encoder-IMU.....52
Gambar 3.17 Control surface simulasi FIS 1 : Perlambatan.....68
Gambar 3.18 Control surface PWM_Kiri.....71
Gambar 3.19 Control surface PWM_Kanan.....72

DAFTAR PERSAMAAN

Persamaan 2.1 14
Persamaan 2.2 15
Persamaan 2.3 15
Persamaan 2.4 15
Persamaan 2.5 15
Persamaan 2.6 16
Persamaan 2.7 17
Persamaan 2.8 17
Persamaan 2.9 17
Persamaan 2.10 18
Persamaan 2.11 19
Persamaan 2.12 19
Persamaan 2.13 19
Persamaan 2.14 19
Persamaan 2.15 20
Persamaan 2.16 21
Persamaan 2.17 21
Persamaan 2.18 22
Persamaan 2.19 23
Persamaan 2.20 23
Persamaan 2.21 24
Persamaan 2.22 24
Persamaan 2.23 26
Persamaan 2.24 26
Persamaan 2.25 27
Persamaan 2.26 27
Persamaan 2.27 27
Persamaan 2.28 28
Persamaan 3.1 54
Persamaan 3.2 54
Persamaan 3.3 55

Persamaan 3.4 57
Persamaan 3.5 57
Persamaan 3.6 64
Persamaan 3.7 66
Persamaan 3.8 66

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pemeliharaan ruang terbuka berumput merupakan kegiatan yang diperlukan untuk menjaga kerapian, kenyamanan, dan fungsi suatu kawasan. Area seperti lapangan olahraga, taman, lingkungan kampus, dan kawasan perumahan memerlukan pemotongan rumput secara berkala. Penggunaan mesin pemotong rumput konvensional masih membutuhkan keterlibatan operator untuk mengarahkan mesin, mengatur lintasan, menghindari halangan, dan memastikan seluruh area telah dikerjakan. Pada area yang luas, kegiatan tersebut membutuhkan waktu dan tenaga serta berpotensi menghasilkan pengulangan lintasan maupun bagian area yang belum terpotong. Kondisi ini mendorong pengembangan teknologi otomasi yang dapat membantu pelaksanaan pemeliharaan rumput secara lebih teratur dan terukur.

Salah satu lokasi yang menjadi perhatian dalam pengembangan teknologi tersebut adalah Lapangan Woodball Universitas Negeri Semarang. Lapangan ini ditetapkan sebagai lokasi studi kasus dan rencana pengujian robot pemotong rumput otonom. Berdasarkan pengukuran awal menggunakan Google Earth, luas area yang ditinjau diperkirakan sekitar 3.058 m² dengan bentuk batas yang tidak beraturan. Pengukuran tersebut merupakan estimasi awal yang perlu diverifikasi di lapangan. Untuk menyesuaikan pengujian dengan kemampuan prototipe, tahap awal pengujian direncanakan pada petak seluas 100 m² di dalam area tersebut. Karakteristik lokasi menjadi pertimbangan dalam penentuan batas operasi, penyusunan lintasan, dan evaluasi kemampuan robot.

Robot pemotong rumput otonom memerlukan kemampuan untuk mengetahui posisi, mengenali lingkungan, dan menentukan arah pergerakan. Informasi posisi saja belum cukup untuk menggambarkan keberadaan halangan di sekitar robot. Inoue dkk. (2022) mengembangkan sistem navigasi dan penghindaran halangan pada robot pemotong rumput menggunakan dua kamera RGB-D dan IMU. Penelitian tersebut menggunakan jaringan saraf konvolusional untuk mengenali halangan serta *visual simultaneous localization and mapping* untuk memperkirakan posisi robot tanpa mengandalkan sinyal satelit. Dalam pengujian pada lingkungan kebun yang disimulasikan, selisih antara posisi estimasi dan posisi sebenarnya sebesar 0,30 m, dan robot dapat bergerak tanpa bertabrakan dengan halangan. Pendekatan ini menjadi salah satu dasar pengembangan sistem navigasi yang menggabungkan informasi posisi dengan persepsi lingkungan.

Dalam penelitian ini, navigasi dirancang dengan memanfaatkan Global Positioning System (GPS) sebagai referensi posisi global, deteksi halangan berbasis YOLOv8n, serta coverage path planning untuk menyusun lintasan cakupan area. Perencanaan lintasan diperlukan agar robot dapat mengerjakan area secara sistematis dan mengurangi pengulangan lintasan yang tidak diperlukan. Sementara itu, informasi dari kamera digunakan untuk mengenali halangan pada jalur kerja sebagai dasar pengambilan keputusan untuk melanjutkan pergerakan, melakukan manuver, atau menghentikan operasi. Kemampuan tersebut perlu dihubungkan dengan sistem kendali gerak agar keputusan navigasi dapat diterjemahkan menjadi gerakan fisik robot.

Pengoperasian robot pada permukaan berumput juga menghadapi variasi kondisi tanah, kemiringan, kelembapan, dan traksi. Perbedaan kondisi tersebut dapat menimbulkan slip dan penyimpangan lintasan, terutama pada robot dengan penggerak sabuk rantai yang berbelok melalui perbedaan kecepatan sisi kiri dan kanan. Sathyamoorthy dkk. (2022), melalui pengembangan TerraPN, memanfaatkan informasi interaksi robot dengan medan, termasuk getaran IMU dan galat odometri, untuk mendukung navigasi pada permukaan yang memiliki karakteristik berbeda. Temuan tersebut menjadi pertimbangan bahwa kondisi medan perlu diperhatikan dalam perancangan kendali robot luar ruangan. Pada prototipe yang dikembangkan, data odometri dan Inertial Measurement Unit (IMU) dipadukan untuk membantu estimasi perpindahan serta orientasi, sedangkan kendali fuzzy Sugeno digunakan untuk mengatur respons gerak terhadap kesalahan arah dan lintasan.

Selain navigasi dan pergerakan, mekanisme pemotongan merupakan bagian utama yang menentukan kemampuan robot dalam menjalankan tugasnya. Perbedaan tinggi dan kepadatan rumput dapat menyebabkan perubahan beban pada motor pemotong. Oleh karena itu, rancangan robot perlu menyediakan mekanisme pengaturan elevasi dek pemotong agar posisi pemotongan dapat disesuaikan dengan kondisi operasi. Penelitian ini merencanakan penggunaan sensor arus sebagai umpan balik beban motor, kendali Proportional–Integral–Derivative (PID), dan aktuator linear untuk mengatur elevasi dek. Mekanisme self-locking pada aktuator dimanfaatkan untuk mempertahankan posisi dek tanpa pengaktifan motor aktuator secara terus-menerus. Kinerja pengaturan tersebut perlu dievaluasi melalui pengukuran perubahan ketinggian, respons terhadap peningkatan beban, dan konsumsi energi.

Robot juga dilengkapi fungsi penyemprotan berbasis Variable Rate Application (VRA) sebagai bagian dari pengembangan pemeliharaan vegetasi. Pendekatan ini diarahkan untuk menyesuaikan keluaran penyemprotan berdasarkan informasi kondisi area sasaran. Rahimi Azghadi dkk. (2024) menunjukkan bahwa penyemprotan robotik berbasis visi komputer pada lahan tebu dapat mengurangi

penggunaan herbisida rata-rata sebesar 35% dibandingkan penyemprotan menyeluruh. Hasil tersebut memberikan dasar mengenai potensi penyemprotan terarah, meskipun kinerjanya tidak dapat langsung dianggap sama pada prototipe dan lingkungan pengujian yang berbeda. Dalam penelitian ini, indeks VRA dari pengolahan citra diproses menggunakan fuzzy Mamdani untuk menghasilkan perintah kendali pompa melalui sinyal Pulse Width Modulation (PWM).

Penggabungan seluruh fungsi tersebut membutuhkan koordinasi antarsubsistem. Perubahan lintasan akibat halangan harus diteruskan kepada pengendali gerak, sedangkan pengoperasian pemotong dan penyemprot perlu mengikuti kondisi serta status misi robot. Pengaturan elevasi dek juga harus berlangsung tanpa menghambat pembacaan sensor maupun komunikasi antarpengendali. Dengan demikian, evaluasi robot perlu mencakup kemampuan setiap subsistem sekaligus keberhasilan integrasinya dalam menjalankan suatu misi. Pengujian dilakukan secara bertahap agar hubungan antara keputusan perangkat lunak dan respons perangkat keras dapat diperiksa.

Kontribusi utama pengembangan ini terletak pada integrasi visi komputer dan perencanaan jalur, kendali gerak dan fusi sensor, mekanisme pemotongan dan pengelolaan daya, serta penyemprotan VRA dalam satu platform robot. Perancangan juga mempertimbangkan ketersediaan komponen, kemudahan perawatan, dan batas biaya pengembangan. Kesesuaian rancangan terhadap kebutuhan pengguna perlu dibuktikan melalui pengujian pada lokasi studi kasus, sehingga penilaian terhadap prototipe didasarkan pada data kinerja yang diperoleh.

Berdasarkan permasalahan tersebut, dilakukan penelitian berjudul "Perancangan dan Implementasi Robot Pemotong Rumput Otonom dengan Sistem Navigasi Cerdas". Penelitian diarahkan untuk menghasilkan prototipe yang mampu menjalankan pemotongan rumput melalui lintasan yang direncanakan, merespons halangan, menyesuaikan kendali gerak terhadap kondisi medan, mengatur elevasi dek, dan mengendalikan penyemprotan sesuai masukan. Prototipe beserta hasil pengujiannya diharapkan menjadi dasar pengembangan teknologi otomasi pemeliharaan area berumput yang sesuai dengan kebutuhan operasional pengguna.

1.2 Rumusan Masalah

Permasalahan utama dalam penelitian ini adalah bagaimana pekerjaan pemeliharaan rumput dapat diselesaikan oleh robot sementara operator cukup memantau jalannya pekerjaan. Permasalahan tersebut dirinci sebagai berikut.

1. Bagaimana area berumput dapat dipotong secara merata tanpa operator terus-menerus mengarahkan robot, sekaligus tetap merespons halangan pada jalur kerja?

2. Bagaimana robot dapat mengikuti lintasan yang dituju ketika kondisi permukaan dan traksi berubah, sehingga area yang direncanakan tetap terjangkau?
3. Bagaimana tinggi pemotongan dapat disesuaikan dengan kondisi rumput tanpa menyebabkan motor pemotong terhenti akibat beban berlebih atau menggunakan daya pengatur dek secara terus-menerus?
4. Bagaimana jumlah herbisida yang disemprotkan dapat disesuaikan dengan kondisi area sasaran agar penggunaan cairan lebih terukur?
5. Bagaimana seluruh fungsi tersebut dapat bekerja bersama untuk menyelesaikan misi pada area pengujian dengan pengawasan operator dan biaya pengembangan yang terjangkau?

1.3 Tujuan Prototipe

Tujuan penelitian ini adalah menghasilkan prototipe robot pemotong rumput yang mampu membantu menyelesaikan pekerjaan pemeliharaan area berumput dengan pengawasan operator. Secara khusus, prototipe ditujukan untuk:

1. Mengerjakan area uji secara sistematis dengan target cakupan sekurang-kurangnya 80% serta keberhasilan merespons halangan statis sekurang-kurangnya 80% pada skenario pengujian yang ditetapkan.
2. Mengikuti lintasan acuan pada permukaan berumput dan meningkatkan ketelitian estimasi posisi serta orientasi lokal, dengan target penurunan galat sekurang-kurangnya 20% dibandingkan penggunaan odometri saja.
3. Menyediakan sekurang-kurangnya tiga tingkat ketinggian pemotongan, mempertahankan posisi dek yang dipilih, dan merespons peningkatan beban motor agar proses pemotongan dapat berlangsung.
4. Menyesuaikan penyemprotan herbisida dengan kondisi area sasaran pada sekurang-kurangnya tiga tingkat debit, dengan galat debit aktual terhadap debit acuan tidak lebih dari 10% pada kondisi pengujian yang ditetapkan.
5. Menjalankan fungsi-fungsi tersebut sebagai satu misi dengan target keberhasilan sekurang-kurangnya 80% dari skenario uji tanpa intervensi pengendalian gerak oleh operator selama misi, serta menjaga biaya komponen dalam batas anggaran pengembangan Rp10.000.000.

1.4 Manfaat Prototipe

1.4.1 Bagi Penulis

1. Menambah pengetahuan dan keterampilan dalam merancang serta mengintegrasikan sistem mekanik, elektronika, kendali, dan *computer vision* pada robot otonom.
2. Meningkatkan kemampuan dalam melakukan pengujian, menganalisis kinerja, dan menyelesaikan permasalahan teknis pada sistem robotika.

1.4.2 Bagi Universitas Negeri Semarang

1. Memberikan kontribusi rujukan ilmiah dan pengayaan riset terapan di bidang robotika cerdas, sistem navigasi otonom, dan visi komputer bagi civitas akademika Universitas Negeri Semarang, khususnya di lingkungan Program Studi Teknik Komputer dan Fakultas Teknik.
2. Mendukung program hilirisasi produk inovasi mahasiswa serta menyediakan prototipe teknologi otomasi yang dapat diimplementasikan untuk efisiensi pemeliharaan fasilitas ruang terbuka hijau kampus, khususnya Lapangan Woodball UNNES.

1.4.3 Bagi Pengelola Ruang Terbuka Hijau dan Masyarakat

1. Memberikan alternatif teknologi untuk membantu pekerjaan pemotongan rumput dan penyemprotan secara otomatis.
2. Mengurangi keterlibatan langsung operator sehingga pekerjaan pemeliharaan lahan dapat dilakukan dengan lebih aman dan efisien.

1.4.4 Bagi Lingkungan

1. Mendukung penyemprotan cairan secara lebih terukur berdasarkan kondisi area yang terdeteksi oleh sistem.
2. Mengurangi potensi penggunaan herbisida secara berlebihan melalui penerapan sistem penyemprotan VRA.

1.4.5 Bagi Industri

1. Mendukung pengembangan teknologi robotika dalam negeri untuk membantu pemeliharaan ruang terbuka hijau secara lebih efisien.
2. Menjadi langkah awal menuju kemandirian pengembangan robot pemotong rumput otonom yang dapat disesuaikan dengan kebutuhan dan kondisi lahan di Indonesia.

1.4.6 Bagi Perkembangan Ilmu Pengetahuan dan Teknologi

1. Memberikan kontribusi terhadap penerapan GPS, *coverage path planning*, YOLOv8n, fuzzy logic, PID, dan fusi sensor pada robot pemeliharaan lahan.
2. Menjadi dasar bagi penelitian selanjutnya dalam meningkatkan akurasi navigasi, efisiensi energi, kualitas pemotongan, serta ketepatan penyemprotan.

1.4.7 Bagi Negara

1. Mendukung pengembangan teknologi oleh mahasiswa Indonesia.
2. Menjadi dasar pengembangan sistem yang sesuai dengan kebutuhan pemeliharaan area berumput di dalam negeri.

1.5 Potensi Dampak Fungsional/Komersial

1. Potensi dampak fungsional: Robot berpotensi mengurangi pekerjaan manual dalam pemotongan rumput dan penyemprotan pada area pengujian. GPS dan coverage path planning membantu robot mengikuti lintasan yang terencana, sedangkan deteksi halangan dan kendali gerak mendukung penyesuaian pergerakan saat terdapat halangan statis. Elevasi dek dapat diatur sesuai tinggi pemotongan yang dibutuhkan, sementara pemantauan arus motor dan kendali PID membantu sistem merespons perubahan beban. Sistem variable rate application (VRA) berpotensi menyesuaikan debit penyemprotan berdasarkan kondisi area yang terdeteksi kamera. Dampak tersebut akan dinilai melalui pengujian cakupan area, penghindaran halangan, kinerja pemotongan, ketepatan debit, dan keberhasilan misi tanpa intervensi manual.
2. Potensi dampak komersial: Prototipe ini berpotensi dikembangkan menjadi produk pemeliharaan rumput yang menggabungkan navigasi otonom, pemotongan, dan penyemprotan sesuai kebutuhan. Calon penggunanya meliputi pengelola fasilitas kampus, lapangan olahraga, kawasan perumahan, dan penyedia jasa pemeliharaan taman. Sebagai karya yang dikembangkan dan dirakit oleh tim mahasiswa Indonesia, robot ditargetkan memiliki harga jual maksimal Rp15.000.000 per unit sehingga berpotensi menjadi alternatif yang lebih terjangkau dibandingkan beberapa produk impor pada Tabel 1.1. Kelayakan produk dan keunggulan harganya masih perlu dibuktikan melalui pengujian kinerja, keselamatan, ketahanan, serta perhitungan biaya produksi dan layanan purnajual.

1.5 Potensi Dampak Fungsional/Komersial

1. Potensi dampak fungsional: Robot berpotensi mengurangi pekerjaan manual dalam pemotongan rumput dan penyemprotan pada area pengujian. GPS dan coverage path planning membantu robot mengikuti lintasan yang terencana, sedangkan deteksi halangan dan kendali gerak mendukung penyesuaian pergerakan saat terdapat halangan statis. Elevasi dek dapat diatur sesuai tinggi pemotongan yang dibutuhkan, sementara pemantauan arus motor dan kendali PID membantu sistem merespons perubahan beban. Sistem variable rate application (VRA) berpotensi menyesuaikan debit penyemprotan berdasarkan kondisi area yang terdeteksi kamera. Dampak tersebut akan dinilai melalui pengujian cakupan area, penghindaran halangan, kinerja pemotongan, ketepatan debit, dan keberhasilan misi tanpa intervensi manual.
2. Potensi dampak komersial: Prototipe ini berpotensi dikembangkan menjadi produk pemeliharaan rumput yang menggabungkan navigasi otonom, pemotongan, dan penyemprotan sesuai kebutuhan. Calon penggunanya meliputi pengelola fasilitas kampus, lapangan olahraga, kawasan

perumahan, dan penyedia jasa pemeliharaan taman. Sebagai karya yang dikembangkan dan dirakit oleh tim mahasiswa Indonesia, robot ditargetkan memiliki harga jual maksimal Rp15.000.000 per unit sehingga berpotensi menjadi alternatif yang lebih terjangkau dibandingkan beberapa produk impor pada Tabel 1.1. Kelayakan produk dan keunggulan harganya masih perlu dibuktikan melalui pengujian kinerja, keselamatan, ketahanan, serta perhitungan biaya produksi dan layanan purnajual.

Nama Produk	Gambar Produk	Harga Produk	Kelebihan	Kekurangan
Husqvarna Automower 450X NERA		Rp 98.480.000	a. Kapasitas dan jangkauan pemotongan jauh lebih luas hingga 5.000 m². b. Memiliki deteksi halangan dan pengaturan zona kerja.	a. Harga lebih mahal dibandingkan dengan prototipe. b. Tidak dilengkapi penyemprotan. c. Perangkat EPOS tidak termasuk dalam harga varian dasar. d. Alat jauh lebih besar dan berat dibanding prototipe sehingga tidak fleksibel dipindahkan.
Mammotion LUBA 3 AWD 1500		Rp 40.595.654	a. Menggunakan LIDAR 360° b. Sudah dilengkapi asisten berbasis AI	a. Harga lebih mahal. b. Sparepart susah dicari karena barang luar negeri. c. Maintenance sulit karena kita harus maintenance sendiri.
Segway Navimow i105		Rp 14.325.771	a. Memiliki sistem navigasi EFLS 2.0 b. Memiliki sistem Mapping dan deteksi halangan. c. Harga sedikit lebih murah dibanding target harga jual prototipe	a. Pengaturan tinggi pemotongan rumput masih manual dengan sistem bongkar pasang. b. Tidak dilengkapi penyemprotan. c. Konfigurasi mapping agak lama.

Tabel 1.1 Perbandingan robot pemotong rumput komersial dengan prototipe.

BAB II
KAJIAN PUSTAKA

2.1 Tinjauan Pustaka

Perancangan subsistem kendali gerak robot pemotong rumput otonom ini mengacu pada penelitian terdahulu dalam tiga bidang yang saling berkaitan: (1) estimasi posisi dan orientasi robot pemotong rumput, (2) kalibrasi kinematika serta odometri robot berpenggerak sabuk rantai, dan (3) kendali fuzzy untuk mengikuti lintasan. Tinjauan ini digunakan untuk menentukan hubungan antara putaran penggerak, pembacaan sensor, dan respons gerak robot. Hasil penelitian terdahulu dipertimbangkan sesuai jenis penggerak dan lingkungan pengujiannya, karena kinerja robot beroda pada permukaan dalam ruangan tidak dapat langsung dianggap sama dengan kinerja robot bersabuk rantai di lapangan berumput.

Inoue dkk. (2022) mengembangkan navigasi robot pemotong rumput menggunakan dua kamera RGB-D dan *inertial measurement unit* (IMU). Posisi robot diperkirakan melalui *visual odometry* yang dipadukan dengan pengukuran kecepatan sudut IMU, kemudian dikoreksi saat kamera mengenali penanda lokasi. Dalam percobaan navigasi pada area terbuka yang disusun menyerupai kebun, *root mean square error* (RMSE) posisi tercatat sebesar 0,874 m tanpa koreksi penanda dan 0,297 m ketika koreksi penanda digunakan. Hasil tersebut menunjukkan pentingnya pembaruan estimasi posisi ketika galat gerak terakumulasi. Namun, sistem itu mengandalkan kamera kedalaman dan penanda lokasi; rancangan subsistem pada penelitian ini menggunakan encoder dan MPU-9250 untuk estimasi gerak lokal, sedangkan GPS menjadi referensi posisi global pada subsistem navigasi.

Pada robot berpenggerak sabuk rantai, Cantelli dkk. (2016) meneliti kalibrasi parameter kinematika dan offset magnetometer untuk mendukung lokalisasi robot luar ruangan. Penelitian tersebut menggunakan *extended Kalman filter* untuk memperkirakan parameter penggerak, termasuk jari-jari efektif dan jarak antarsisi penggerak, serta menguji pengaruh perubahan kualitas informasi posisi. Penulis menjelaskan bahwa putaran penggerak yang terbaca encoder tidak selalu mewakili perpindahan robot ketika terjadi slip, terutama saat robot *skid-steering* berbelok. Temuan ini menjadi dasar untuk mengkalibrasi parameter gerak pada prototipe fisik, alih-alih menentukan seluruh parameter hanya dari ukuran rancangan mekanik.

Penelitian Cantelli dkk. (2016) juga memperlihatkan bahwa informasi orientasi dari sensor inersia dan magnetometer memerlukan pemeriksaan bias serta gangguan medan magnet sebelum digunakan untuk memperbaiki estimasi gerak.

Hal tersebut relevan dengan pemasangan MPU-9250 pada robot yang membawa motor penggerak dan struktur logam. Akan tetapi, konfigurasi sensor serta algoritma pada penelitian tersebut tidak langsung menetapkan metode fusi bagi prototipe ini. Pengaruh penambahan IMU terhadap ketelitian estimasi posisi dan orientasi tetap harus dibuktikan melalui perbandingan dengan odometri encoder saja pada lintasan dan kondisi uji yang sama.

Untuk kendali mengikuti lintasan, Nguyen dan Vu (2022) menggabungkan model kinematika penggerak diferensial dengan pengendali *fuzzy logic*. Pengujian dilakukan pada lintasan sepanjang 60 m yang mencakup segmen lurus, melingkar, dan miring, dengan perbandingan hasil simulasi serta percobaan fisik. Pada kecepatan acuan 0,6 m/s, tabel hasil dari 30 percobaan melaporkan RMSE posisi eksperimen sebesar 0,015 m pada sumbu (x) dan 0,013 m pada sumbu (y); galat kecepatan linear yang dilaporkan sebesar 0,007 m/s. Penelitian tersebut mendukung penggunaan keluaran pengendali fuzzy bersama model kinematika untuk menghasilkan gerak kiri dan kanan. Meskipun demikian, nilai galatnya berasal dari robot beroda dan tidak dapat dijadikan target kinerja langsung bagi prototipe bersabuk rantai pada rumput.

Omrane dkk. (2016) menerapkan pengendali fuzzy Sugeno orde nol pada robot bergerak dengan dua motor, dua encoder optik, dan sensor inframerah untuk navigasi serta penghindaran halangan. Penelitian tersebut menyusun 35 aturan untuk mencapai posisi tujuan dan 62 aturan pada konfigurasi yang juga menangani halangan. Pengendali diuji melalui simulasi MATLAB dan SIMIAM serta percobaan pada robot. Studi ini memperlihatkan bahwa aturan fuzzy dapat diterjemahkan menjadi perintah kecepatan sisi kiri dan kanan. Namun, masukan sensor, jumlah aturan, dan kondisi pengujian dalam ruangan berbeda dari kebutuhan prototipe ini, sehingga basis aturan untuk galat arah dan galat lintasan harus dirancang serta diuji tersendiri.

Keempat penelitian tersebut memberikan acuan yang berbeda bagi subsistem yang dirancang. Studi Inoue dkk. menunjukkan perlunya mengendalikan akumulasi galat estimasi pada robot pemotong rumput; Cantelli dkk. menjadi acuan kalibrasi gerak robot bersabuk rantai; Nguyen dan Vu menunjukkan penerapan model kinematika bersama kendali fuzzy; sedangkan Omrane dkk. memberi contoh penerapan Sugeno pada perintah gerak robot. Berdasarkan acuan tersebut, penelitian ini memusatkan perancangan pada integrasi odometri encoder dan IMU MPU-9250 sebagai umpan balik gerak lokal, lalu menggunakan fuzzy Sugeno untuk mengoreksi gerak dua sisi sabuk rantai saat mengikuti lintasan dari subsistem navigasi. Kelayakannya akan dinilai dari galat estimasi posisi dan orientasi, galat mengikuti lintasan, serta respons gerak pada kondisi permukaan yang diuji di Lapangan Woodball UNNES.

Tabel 2.1. Perbandingan penelitian terdahulu dengan rancangan subsistem kendali gerak

No	Peneliti (tahun)	Platform robot dan komputasi	Sensor dan estimasi gerak	Metode kendali	Hasil yang dilaporkan	Kesenjangan terhadap penelitian ini
1	Inoue dkk. (2022)	Robot pemotong rumput beroda; Jetson AGX Xavier dan laptop	Dua kamera RGB-D, IMU, *visual odometry*, dan penanda lokasi	Perencanaan lintasan serta penghindaran halangan	RMSE posisi pada uji navigasi turun dari 0,874 m tanpa penanda menjadi 0,297 m dengan koreksi penanda	Koreksi posisi bergantung pada kamera kedalaman dan penanda lokasi; penggeraknya bukan sabuk rantai
2	Cantelli dkk. (2016)	Robot sabuk rantai; pengendali sbRIO 9626	Encoder, AHRS, dan DGPS; kalibrasi parameter kinematika serta offset magnetometer menggunakan EKF	Estimasi posisi dan kalibrasi parameter gerak	Diuji pada lintasan dengan kehilangan sinyal dan penurunan kualitas GPS; penelitian menunjukkan pentingnya kalibrasi dan koreksi orientasi saat odometri mengalami galat	Susunan sensor dan metode EKF melibatkan DGPS; belum menjawab kinerja fusi lokal encoder–MPU-9250 pada prototipe ini
3	Nguyen dan Vu (2022)	Robot beroda diferensial; laptop dan STM32F407VET6	Encoder, kompas digital/IMU, dan pemindai laser	Model kinematika dan pengendali fuzzy untuk mengikuti lintasan	Pada acuan 0,6 m/s, uji 30 kali melaporkan RMSE posisi 0,015 m pada sumbu (x), 0,013 m pada sumbu (y), serta galat kecepatan linear 0,007 m/s	Hasil diperoleh pada robot beroda; pengaruh slip sabuk rantai di permukaan berumput belum dievaluasi

4	Omrane dkk. (2016)	Robot beroda diferensial QuickBot; simulasi MATLAB dan SIMIAM	Dua encoder optik dan sensor jarak inframerah	Fuzzy Sugeno orde nol untuk perintah kecepatan kiri dan kanan	Menggunakan 35 aturan untuk menuju tujuan dan 62 aturan pada konfigurasi penghindaran halangan; diuji melalui simulasi dan percobaan robot	Masukan sensor dan lingkungan uji berbeda; tidak mengevaluasi fusi encoder–IMU pada robot sabuk rantai
5	Prototipe Nicodemus (2026)	Robot sabuk rantai *skid-steering*; ESP32, dua motor Canter MK461340, dan dua BTS7960	Dua encoder dan MPU-9250 untuk estimasi posisi serta orientasi lokal; GPS ditangani subsistem navigasi	Model kinematika terkalibrasi dan fuzzy Sugeno; simulasi MATLAB, implementasi C++ pada ESP32	Target: RMSE posisi dan orientasi masing-masing berkurang ≥20% dibanding odometri saja; pengujian awal pada petak 100 m²	Menguji integrasi estimasi gerak lokal dan koreksi lintasan pada platform fisik serta kondisi permukaan Lapangan Woodball UNNES

Tabel 1.2 Matriks Komparasi Penelitian Terdahulu dan Posisi Prototipe

2.1.1 Identifikasi Kesenjangan Penelitian *(Research Gap)* dan Kebaruan Prototipe

Berdasarkan Tabel 2.1, penelitian terdahulu telah menunjukkan keberhasilan estimasi gerak robot pemotong rumput, kalibrasi robot bersabuk rantai, dan kendali lintasan berbasis fuzzy. Namun, penerapan temuan tersebut pada subsistem kendali gerak prototipe ini masih menyisakan tiga kesenjangan berikut.

1. Ketidakpastian model gerak pada sabuk rantai. Perbedaan putaran sisi kiri dan kanan dapat digunakan untuk memperkirakan arah gerak robot, tetapi slip saat berbelok menyebabkan perpindahan aktual menyimpang dari hasil perhitungan encoder. Cantelli dkk. (2016) menunjukkan pentingnya kalibrasi parameter kinematika robot bersabuk rantai. Pada prototipe ini, parameter seperti jarak efektif antarsisi penggerak masih perlu ditentukan dan diperiksa melalui uji gerak pada permukaan berumput.

2. Ketelitian fusi encoder–IMU pada konfigurasi prototipe belum terbukti. Inoue dkk. (2022) menunjukkan manfaat informasi IMU dan koreksi penanda lokasi dalam estimasi gerak robot pemotong rumput, sedangkan

Cantelli dkk. (2016) menggunakan susunan sensor dan metode fusi yang berbeda. Kedua hasil tersebut belum menunjukkan seberapa besar perubahan galat ketika odometri dua encoder dipadukan dengan MPU-9250 pada robot sabuk rantai ini. Karena itu, estimasi hasil fusi perlu dibandingkan dengan odometri saja menggunakan acuan pengukuran yang independen.

3. Kendali lintasan perlu diuji bersama estimasi gerak pada medan berumput. Nguyen dan Vu (2022) menunjukkan bahwa model kinematika dan kendali fuzzy dapat digunakan untuk mengikuti lintasan pada robot beroda. Omrane dkk. (2016) juga menerapkan fuzzy Sugeno untuk menghasilkan perintah kecepatan sisi kiri dan kanan. Akan tetapi, hasil kedua penelitian tersebut belum dapat langsung digunakan untuk menilai respons robot bersabuk rantai ketika traksi berubah. Pengujian perlu menghubungkan galat estimasi posisi dan orientasi dengan galat lintasan, galat arah, serta respons kecepatan penggerak.

Kebaruan prototipe ini terletak pada penerapan dan evaluasi terpadu model kinematika *skid-steering* yang dikalibrasi, estimasi gerak lokal berbasis encoder–MPU-9250, dan kendali fuzzy Sugeno pada satu platform robot pemotong rumput bersabuk rantai. Evaluasi direncanakan pada lintasan lurus dan berbelok di Lapangan Woodball UNNES. Target awalnya adalah menurunkan RMSE posisi dan RMSE orientasi masing-masing sekurang-kurangnya 20% dibanding odometri tanpa fusi, serta mengukur kemampuan robot mengikuti lintasan acuan. Angka tersebut merupakan kriteria keberhasilan yang akan diuji**,** bukan hasil yang telah diperoleh.

2.2 Landasan Teoritik

Landasan teoretik ini menjelaskan konsep yang digunakan untuk merancang subsistem kendali gerak robot pemotong rumput otonom. Pembahasan dimulai dari prinsip gerak sabuk rantai *skid-steering* dan model kinematikanya, kemudian dilanjutkan dengan odometri berbasis *rotary encoder*, pengukuran IMU MPU-9250, serta fusi kedua sumber data untuk memperkirakan gerak lokal robot. Estimasi tersebut menjadi umpan balik bagi kendali fuzzy Sugeno dalam mengatur gerak sisi kiri dan kanan. Bagian akhir menjelaskan hubungan subsistem kendali gerak dengan informasi posisi global GNSS dan keputusan penghindaran halangan dari subsistem navigasi.

2.2.1 Sistem Gerak Sabuk Rantai dan Kinematika *Skid-Steering*

Prototipe robot menggunakan dua sabuk rantai yang digerakkan secara independen oleh motor sisi kiri dan kanan. Arah gerak ditentukan oleh arah putaran kedua penggerak, sedangkan manuver berbelok dihasilkan dari perbedaan

kecepatan kedua sisi. Pada sistem *skid-steering*, perubahan arah melibatkan pergeseran sabuk rantai terhadap permukaan. Akibatnya, hubungan antara putaran penggerak dan gerak robot dipengaruhi oleh traksi serta kondisi medan (Cantelli dkk., 2016).

No	Kondisi kecepatan sisi penggerak	Gerak yang diperikaran
1	$v_R = v_L > 0$	Maju lurus
2	$v_R = v_L < 0$	Mundur lurus
3	$v_R < v_L$	Berbelok ke kiri
4	$v_R < v_L$	Berbelok ke kanan
5	$v_R = -v_L > 0$	Berputar ke kiri di sekitar pusat poros
6	$v_R = -v_L < 0$	Berputar ke kanan di sekitar pusat poros

Tabel 2.2 Hubungan kecepatan kedua sisi penggerak dan arah Gerak robot

Pada tabel tersebut, (v_R) dan (v_L) adalah kecepatan linear sabuk rantai sisi kanan dan kiri. Tanda positif menyatakan gerak sisi penggerak ke arah maju robot. Arah belok pada tabel mengikuti konvensi sudut positif berlawanan arah jarum jam. Kondisi gerak lurus merupakan keadaan ideal; pada robot fisik, selisih traksi atau respons motor dapat menyebabkan robot menyimpang meskipun kedua sisi diberi perintah kecepatan yang sama.

Sebagai model awal, kecepatan linear pusat robot (v) dan kecepatan sudut (ω) dihitung dari kecepatan kedua sisi sebagai berikut:

$$v = \frac{v_R + v_L}{2}$$

Persamaan 2.1

$$\omega = \frac{v_R - v_L}{B_{\text{ef}}}$$

Persamaan 2.2

(B_{ef}) menyatakan jarak efektif antarsisi penggerak dalam meter. Parameter ini digunakan karena titik putar robot saat melakukan *skid-steering* dapat berbeda dari perkiraan berdasarkan jarak geometris kedua sabuk rantai. Nilainya diperlakukan sebagai hasil kalibrasi pada robot fisik dan dapat berubah menurut kondisi permukaan. Oleh sebab itu, Persamaan (2.1) – (2.2) merupakan pendekatan kinematika yang kesesuaiannya perlu diperiksa melalui pengukuran gerak aktual.

Untuk memperjelas kedua arah putaran di tempat, misalkan besar kecepatan setiap sisi adalah $(u > 0)$. Dengan memasukkan kecepatan sisi kanan dan kiri ke Persamaan (2.1)–(2.2), diperoleh:

Putar kiri di sekitar poros :

$$v_R = +u, \qquad v_L = -u$$

$$\Longrightarrow$$

$$v = 0, \qquad \omega = \frac{2u}{B_{\text{ef}}} > 0.$$

Persamaan 2.3

Putar kanan di sekitar poros :

$$v_R = -u, \qquad v_L = +u$$

$$\Longrightarrow$$

$$v = 0, \qquad \omega = -\frac{2u}{B_{\text{ef}}} < 0.$$

Persamaan 2.4

Sebaliknya, jika pengendali menghasilkan kecepatan linear acuan (v_{ref}) dan kecepatan sudut acuan (ω_{ref}), perintah kecepatan setiap sisi diperoleh dari:

$$v_{R,\text{ref}} = v_{\text{ref}} + \frac{B_{\text{ef}}}{2}\omega_{\text{ref}}$$

$$v_{L,\text{ref}} = v_{\text{ref}} - \frac{B_{\text{ef}}}{2}\omega_{\text{ref}}$$

Persamaan 2.5

Persamaan (2.5) menjadi dasar penerjemahan koreksi arah menjadi kecepatan penggerak kiri dan kanan. Putaran aktual kedua sisi nantinya dibaca melalui *rotary encoder*, sedangkan perubahan orientasi robot diamati menggunakan IMU MPU-9250. Perbandingan keduanya diperlukan karena putaran sabuk rantai saat slip tidak selalu menghasilkan perpindahan atau perubahan arah sebesar yang diperkirakan oleh model (Cantelli dkk., 2016).

2.2.2 Odometri Berbasis *Rotary Encoder*

Odometri digunakan memperkirakan perubahan posisi dan orientasi robot dari putaran penggerak. Pada prototipe ini, *rotary encoder* dipasang pada sisi penggerak kiri dan kanan untuk menghasilkan hitungan putaran selama robot bergerak. Hitungan tersebut diubah menjadi perkiraan perpindahan kedua sabuk rantai, kemudian digunakan untuk memperbarui posisi lokal robot. Cantelli dkk. (2016) menggunakan data encoder dalam estimasi gerak robot berpenggerak sabuk rantai dan menunjukkan pentingnya kalibrasi parameter kinematika. Namun, putaran penggerak tidak selalu sama dengan perpindahan robot di atas tanah karena sabuk rantai dapat mengalami slip, terutama saat berbelok (Yamauchi dkk., 2017).

Hubungan antara resolusi encoder dan jumlah hitungan untuk satu putaran sproket penggerak dinyatakan pada Persamaan (2.6).

$$\boldsymbol{N_{\mathrm{rev},i} = P_i\, k_i\, G_i,} \qquad \boldsymbol{i \in \{L, R\}}$$

Persamaan 2.6

$(N_{\mathrm{rev},i})$ adalah jumlah hitungan efektif untuk satu putaran sproket sisi (i); (P_i)adalah resolusi encoder dalam *pulse per revolution* (PPR); (k_i) adalah faktor pembacaan sinyal; dan (G_i) adalah jumlah putaran poros tempat encoder dipasang untuk menghasilkan satu putaran sproket. Apabila encoder dipasang langsung pada poros sproket, $(G_i = 1)$. Pada encoder dengan keluaran kuadratur A dan B, faktor (k_i) bergantung pada penerapan pembacaan ×1, ×2, atau ×4. Nilai 1000 PPR yang tercantum dalam rancangan merupakan spesifikasi nominal; nilai $(N_{\mathrm{rev},i})$ yang digunakan dalam program harus ditetapkan sesuai encoder, pemasangan, dan konfigurasi pembacaan yang akhirnya dipakai.

Selisih hitungan encoder bertanda dalam satu interval pengamatan diubah menjadi perpindahan masing-masing sisi melalui Persamaan (2.7). Tanda positif menyatakan gerak maju, sedangkan tanda negatif menyatakan gerak mundur menurut konvensi arah yang ditetapkan pada robot.

$$\Delta s_i = \frac{2\pi r_{\text{ef},i} \Delta N_i}{N_{\text{rev},i}} \qquad i \in LR$$

Persamaan 2.7

(ΔN_i) adalah perubahan hitungan encoder pada sisi (i), sedangkan $\left(r_{\text{ef},i}\right)$ adalah jari-jari efektif penggerak yang diperoleh melalui kalibrasi perpindahan. Parameter efektif diperlukan karena ukuran geometris sproket saja belum tentu menghasilkan perkiraan jarak yang sesuai dengan gerak robot. Fazekas dkk. (2021) menunjukkan bahwa kalibrasi model odometri berpengaruh terhadap ketelitian estimasi gerak. Dalam prototipe ini, kalibrasi dilakukan dengan membandingkan jarak yang dihitung dari encoder terhadap jarak tempuh yang diukur secara langsung.

Perpindahan pusat robot dan perubahan orientasi dari odometri dihitung berdasarkan selisih perpindahan sisi kanan dan kiri, sebagaimana dinyatakan pada Persamaan (2.8).

$$\Delta s = \frac{\Delta s_R + \Delta s_L}{2}, \qquad \Delta \theta_{\text{odo}} = \frac{\Delta s_R - \Delta s_L}{B_{\text{ef}}}$$

Persamaan 2.8

(Δs_R) dan (Δs_L)masing-masing merupakan perkiraan perpindahan sisi kanan dan kiri, (Δs) merupakan perpindahan pusat robot, dan ($\Delta \theta_{\text{odo}}$)merupakan perubahan orientasi menurut odometri. (B_{ef}) adalah jarak efektif antarsisi penggerak yang ditentukan melalui pengujian gerak berbelok. Nilai tersebut tidak langsung disamakan dengan jarak geometris antarsabuk karena putaran robot *skid-steering* melibatkan pergeseran sabuk terhadap permukaan (Cantelli dkk., 2016).

Dengan posisi awal sebagai acuan, estimasi posisi lokal diperbarui pada setiap interval pembacaan. Pendekatan orientasi titik tengah digunakan untuk menghitung perubahan koordinat pada Persamaan (2.9).

$$x_k = x_{k-1} + \Delta s \cos\left(\theta_{k-1} + \frac{\Delta \theta_{\text{odo}}}{2}\right)$$

$$y_k = y_{k-1} + \Delta s \sin\left(\theta_{k-1} + \frac{\Delta \theta_{\text{odo}}}{2}\right)$$

$$\theta_k = wrap(\theta_{k-1} + \Delta \theta_{\text{odo}})$$

Persamaan 2.9

(x_k), (y_k), dan (θ_k) adalah estimasi posisi serta orientasi pada langkah ke-(k). Fungsi (\operatorname{wrap}) mengembalikan sudut ke rentang representasi yang digunakan secara konsisten dalam program. Koordinat yang diperoleh merupakan estimasi relatif terhadap posisi awal robot, bukan koordinat geografis dari GPS.

Hitungan encoder juga digunakan untuk memperkirakan kecepatan linear masing-masing sisi sebagai umpan balik kendali. Untuk interval pembacaan (Δt), kecepatannya dihitung dengan Persamaan (2.10).

$$\hat{v}_i = \frac{\Delta s_i}{\Delta t}, \quad i \in \{L, R\}$$

Persamaan 2.10

Model pada Persamaan (2.6)–(2.10) merupakan odometri dasar yang menghubungkan putaran penggerak dengan perpindahan robot. Kalibrasi dapat mengurangi galat parameter yang relatif tetap, tetapi tidak menjamin hilangnya galat akibat slip yang berubah selama operasi. Yamauchi dkk. (2017) memperlihatkan pengaruh slip memanjang dan menyamping terhadap odometri robot sabuk rantai pada medan miring berpasir. Oleh karena itu, hasil odometri prototipe perlu diperiksa melalui uji gerak lurus, berbelok, dan berputar dengan membandingkannya terhadap jarak serta orientasi yang diukur secara langsung.

2.2.3 Inertial Measurement Unit MPU-9250

Robot menggunakan *Inertial Measurement Unit* (IMU) MPU-9250 untuk memperoleh informasi gerak sudut dan kondisi orientasi badan robot. MPU-9250 memiliki giroskop, akselerometer, dan magnetometer tiga sumbu. Ketiganya menghasilkan data pengukuran, sedangkan orientasi robot diperoleh setelah data tersebut dikalibrasi dan diolah (TDK InvenSense, 2016). Peran setiap sensor diringkas pada Tabel 2.3.

Sensor	Besaran yang diukur	Pemanfaatan pada robot	Keterbatasan utama
Giroskop	Keceptan sudut pada sumbu (x), (y), dan (z)	Mengamati putaran robot dan memperkirakan perubahan arah	Bias dan derau menyebabkan galat bertambah saat data diintegrasikan
Akselerometer	Percepatan spesifik pada tiga sumbu termasuk proyeksi gravitasi	Membantu menilai kemiringan badan robot dan memeriksa kondisi gerak	Getaran dan percepatan saat robot bergerak memengaruhi perkiraan kemiringan
Magnetometer	Komponen medan magnet pada tiga sumbu	Menyediakan informasi acuan arah setelah kalibrasi dan kompensasi kemiringan	Pembacaanya dapat terganggu motor, kabel arus besar, dan bagian logam robot

Tabel 2.3 Peran sensor MPU-9250 pada subsistem kendali Gerak

Giroskop menjadi sumber utama untuk mengamati manuver berbelok. Dalam pendekatan gerak dua dimensi, perubahan arah robot diperkirakan dari kecepatan sudut terhadap sumbu vertikal badan robot. Sebelum digunakan, bias giroskop diestimasi dari (M) sampel ketika robot diam, sebagaimana Persamaan (2.11).

$$\widehat{b_z} = \frac{1}{M}\sum_{j=1}^{M} \omega_{z,j}^{\text{diam}}$$

Persamaan 2.11

$(\widehat{b_z})$ adalah estimasi bias giroskop sumbu (z), sedangkan $(\omega_{z,j}^{\text{diam}})$ adalah sampel kecepatan sudut saat robot diam yang telah dikonversi ke satuan rad/s. Kecepatan sudut terkoreksi diperoleh dengan mengurangkan bias tersebut dari hasil pembacaan, seperti pada Persamaan (2.12).

$$\widehat{\omega_{z,k}} = \omega_{z,k}^{\text{ukur}} - \widehat{b_z}$$

Persamaan 2.12

Dengan interval pembacaan (Δt_k), orientasi menurut giroskop diperbarui melalui Persamaan (2.13).

$$\widehat{\theta_{\text{gyro},k}} = wrap\left(\widehat{\theta_{\text{gyro},k-1}} + \widehat{\omega_{z,k}}\Delta t_k\right)$$

Persamaan 2.13

Persamaan (2.13) merupakan pendekatan gerak bidang datar. Ketika badan robot mengalami kemiringan yang berarti, putaran terhadap sumbu (z) sensor tidak dapat langsung dianggap sama dengan perubahan arah pada bidang horizontal; orientasi tiga dimensi perlu diperhitungkan. Selain itu, kesalahan bias yang masih tersisa akan terintegrasi bersama waktu. Untuk bias sisa yang mendekati konstan sebesar (δb_z), besar galat orientasi akibat bias diperkirakan melalui Persamaan (2.14).

$$\left|e_{\theta,\text{bias}}(T)\right| \approx |\delta b_z| T$$

Persamaan 2.14

(T) adalah lama pengukuran. Hubungan tersebut menjelaskan mengapa kalibrasi saat diam perlu dilakukan, tetapi juga mengapa hasil integrasi giroskop saja tidak boleh dianggap sebagai orientasi yang selalu akurat. “Wondosen dkk. (2021) meneliti estimasi orientasi menggunakan data sensor inersia dan magnetik dengan mempertimbangkan gangguan medan magnet.”

Akselerometer dapat membantu mengenali kemiringan melalui arah gravitasi ketika pengaruh percepatan gerak relatif kecil. Sensor ini tidak memberikan pengukuran arah hadap (*yaw*) secara langsung. Magnetometer dapat menyediakan acuan arah terhadap medan magnet, tetapi memerlukan kalibrasi dan pemeriksaan gangguan sebelum dipakai pada robot. Costanzi dkk. (2016) menunjukkan bahwa gangguan magnet perlu diperhitungkan dalam estimasi orientasi robot bergerak. Pada prototipe ini, pembacaan magnetometer akan diperiksa ketika motor penggerak mati dan menyala untuk menilai pengaruh sistem kelistrikan dan struktur robot terhadap data arah.

Data IMU dibaca oleh ESP32 dengan pencatatan waktu pengambilan sampel. Arah sumbu sensor terhadap sasis, satuan kecepatan sudut, bias saat diam, dan perubahan pembacaan ketika robot bergerak perlu ditetapkan secara konsisten sebelum data orientasi digunakan sebagai umpan balik kendali. Hasil pengukuran IMU kemudian dievaluasi terhadap perubahan sudut acuan pada uji gerak berbelok dan berputar.

2.2.4 Fusi Odometri dan IMU untuk Estimasi Gerak Lokal

Fusi odometri dan *Inertial Measurement Unit* (IMU) menggabungkan informasi putaran penggerak dengan pengukuran gerak badan robot. Pada prototipe ini, dua encoder menghasilkan perkiraan perpindahan sisi kiri dan kanan, sedangkan giroskop MPU-9250 mengukur kecepatan sudut badan robot. Penggabungan keduanya diperlukan karena encoder dapat salah merepresentasikan gerak ketika sabuk rantai mengalami slip, sementara orientasi yang diperoleh dari integrasi giroskop dapat mengalami *drift* akibat bias (Reitbauer dan Schmied, 2021; Zuo dkk., 2020)

Data	Sumber	Peran dalam estimasi	Galat yang perlu diperhatikan
$(\Delta s_R, \Delta s_L)$	Encoder kanan dan kiri	Menghitung perpindahan pusat robot dan perubahan arah menurut odometri	Slip sabuk, hitungan pulsa, dan parameter kinematika
$(\widehat{\omega_z})$	Giroskop MPU-9250	Mengamati putaran badan robot	Bias, derau, dan kesalahan orientasi pemasangan sensor
Data akselerometer	MPU-9250	Membantu menilai kemiringan dan kondisi gerak	Getaran serta percepatan robot
Data magnetometer	MPU-9250	Calon acuan arah apabila hasil kalibrasi dan pemeriksaan gangguan memadai	Pengaruh motor, arus listrik, dan komponen logam

Tabel 2.4 Peran data dalam estimasi Gerak lokal

Perubahan Orientasi giroskop selama interval (Δt_k) dihitung dari kecepatan sudut yang telah dikoreksi terhadap bias, sebagaimana persamaan (2.15).

$$\boldsymbol{\Delta\theta_{\mathrm{gyro},k} = \widehat{\omega_{z,k}}\Delta t_k}$$

Persamaan 2.15

Nilai tersebut dibandingkan dengan perubahan orientasi menurut odometri pada interval yang sama. Selisih kedua perkiraan dinyatakan sebagai residu pada Persamaan (2.16).

$$r_{\theta,k} = wrap\left(\Delta\theta_{\text{odo},k} - \Delta\theta_{\text{gyro},k}\right)$$

Persamaan 2.16

Residu $(r_{\theta,k})$ menunjukkan ketidaksesuaian antara putaran yang diperkirakan dari encoder dan putaran yang terukur pada badan robot. Residu yang besar dapat muncul ketika sabuk rantai slip, tetapi juga dapat disebabkan bias giroskop, gangguan pembacaan, atau ketidaksamaan waktu pengambilan data. Oleh karena itu, residu digunakan sebagai indikator ketidaksesuaian gerak, bukan langsung dinyatakan sebagai persentase slip. "Reitbauer dan Schmied (2021) memadukan data encoder, IMU, dan GNSS untuk mengestimasi gerak kendaraan pertanian berpenggerak sabuk rantai. Zuo dkk. (2020) menunjukkan bahwa slip perlu diperhitungkan dalam estimasi gerak robot *skid-steering*. Pada prototipe ini, fusi untuk estimasi gerak lokal menggunakan encoder dan IMU; GPS digunakan sebagai referensi navigasi global."

Sebagai model awal yang dapat diuji, perubahan orientasi lokal dinyatakan sebagai gabungan berbobot dari kedua sumber data pada Persamaan (2.17).

$$\Delta\theta_{\text{f},k} = wrap\left[(1-\alpha_k)\Delta\theta_{\text{odo},k} + \alpha_k\Delta\theta_{\text{gyro},k}\right], \qquad 0 \le \alpha_k \le 1$$

Persamaan 2.17

(α_k) adalah bobot giroskop: semakin besar nilainya, semakin besar kontribusi pengukuran putaran badan robot terhadap perubahan orientasi. Nilai $(\alpha_k = 1)$ merepresentasikan rancangan dasar yang menggunakan encoder untuk perpindahan linear dan giroskop untuk perubahan arah. Bobot lain hanya dapat ditetapkan setelah hasil encoder dan IMU dibandingkan dengan pengukuran sudut acuan. Pembobotan tidak dengan sendirinya menghilangkan *drift*, karena kedua sumber tersebut sama-sama memberikan informasi gerak relatif.

Perpindahan pusat robot (Δs_k) tetap diperoleh dari encoder. Perubahan orientasi hasil fusi kemudian digunakan untuk memperbarui posisi lokal robot dengan pendekatan titik tengah, sebagaimana Persamaan (2.18).

$$x_k = x_{k-1} + \Delta s_k \cos\left(\theta_{\text{f},k-1} + \frac{\Delta\theta_{\text{f},k}}{2}\right)$$

$$y_k = y_{k-1} + \Delta s_k \sin\left(\theta_{f,k-1} + \frac{\Delta\theta_{f,k}}{2}\right)$$

$$\theta_{f,k} = wrap\left(\theta_{f,k-1} + \Delta\theta_{f,k}\right)$$

Persamaan 2.18

Sebelum Persamaan (2.15)–(2.18) diterapkan pada ESP32, pembacaan encoder dan IMU harus menggunakan interval waktu, konvensi arah putaran, dan orientasi sumbu yang konsisten. Estimasi ini masih dapat mengalami galat posisi saat sabuk rantai slip dalam arah maju atau menyamping, karena giroskop tidak mengukur perpindahan linear robot. GPS menyediakan referensi posisi global bagi subsistem navigasi; dalam rancangan ini, GPS belum ditetapkan sebagai masukan perhitungan fusi lokal. Kinerja fusi akan dinilai dengan membandingkan estimasi posisi dan orientasinya terhadap pengukuran acuan serta terhadap hasil odometri encoder saja.

2.2.5 Sistem Penentuan Posisi Global GNSS dan Protokol NMEA 0183

GNSS menyediakan koordinat global untuk menentukan area kerja dan lintasan acuan robot, sedangkan subsistem kendali gerak menggunakan encoder dan IMU untuk memperkirakan gerak lokal; GPS belum menjadi masukan fusi lokal. Ketelitian GNSS dapat berubah akibat kondisi penerimaan sinyal, termasuk tutupan pohon, sehingga posisi di Lapangan Woodball UNNES perlu diperiksa terhadap titik acuan sebelum digunakan sebagai batas operasi (Tomaštík dan Everett, 2023; Radočaj dkk., 2022). Jika penerima GPS mengirim data berformat NMEA 0183, pesan GGA dan RMC dapat digunakan untuk membaca koordinat, waktu, serta status validitas posisi. Jika posisi bersumber dari perangkat Android, format data yang diterima harus disesuaikan dengan antarmuka perangkat tersebut (u-blox, 2025).

Data	Sumber atau contoh pesan	Fungsi pada robot	Pemeriksaan sebelum digunakan
Lintang dan bujur	GGA atau RMC	Menentukan lokasi global dan titik batas area	Nilai tersedia dan status posisi valid
Waktu pengukuran	GGA atau RMC	Menilai usia data posisi	Waktu sesuai dengan urutan pembacaan
Kualitas *position fix*	GGA	Menandai kondisi solusi posisi	Kualitas memenuhi kriteria yang ditetapkan saat pengujian

Status navigasi	RMC	Membedakan data posisi valid dan tidak valid	Status valid sebelum koordinat diterima
Kecepatan dan arah lintasan terhadap tanah	RMC, bila tersedia	Informasi pendukung pemantauan perpindahan	Tidak digunakan sebagai pengganti orientasi IMU ketika robot diam atau bergerak sangat lambat

Tabel 2.5 Data posisi global yang diperlukan subsistem navigasi

Koordinat lintang dan bujur pada pesan NMEA umumnya ditulis sebagai derajat dan menit desimal. Untuk keperluan perhitungan, nilai tersebut dikonversi menjadi derajat desimal menurut Persamaan (2.19).

$$q_{\text{desimal}} = \sigma\left(d + \frac{m}{60}\right)$$

Persamaan 2.19

Pada persamaan tersebut, (d) adalah bagian derajat, (m) adalah bagian menit desimal, dan $(sigma)$ bernilai $(+1)$ untuk lintang utara atau bujur timur serta (-1) untuk lintang selatan atau bujur barat. Bagian derajat pada lintang dan bujur harus dipisahkan sesuai format masing-masing sebelum konversi dilakukan.

Agar koordinat dapat dibandingkan dengan posisi dan lintasan robot dalam satuan meter, titik geografis di sekitar lapangan dapat diproyeksikan secara lokal terhadap titik acuan (ϕ_0, λ_0). Untuk area pengujian yang kecil, pendekatan bidang singgung sederhana dinyatakan pada Persamaan (2.20).

$$x_G \approx R\cos(\phi_0)\,(\lambda - \lambda_0)$$

$$y_G \approx R(\phi - \phi_0)$$

Persamaan 2.20

(x_G) dan (y_G) masing-masing menyatakan perpindahan ke arah timur dan utara dalam meter; (R) adalah jari-jari bumi yang digunakan dalam pendekatan; sedangkan lintang (ϕ) dan bujur (λ) harus dinyatakan dalam radian. Persamaan ini merupakan pendekatan untuk area lokal, sehingga hasilnya tetap perlu dibandingkan dengan pengukuran lapangan.

Apabila sejumlah titik batas (x_i, y_i) telah diperoleh dan diurutkan mengikuti keliling area, luas poligon tertutup dapat dihitung menggunakan Persamaan (2.21). Titik ke$-n + 1)$ didefinisikan sama dengan titik pertama.

$$A = \frac{1}{2}\left|\sum_{i=1}^{n}(x_i y_{i+1} - x_{i+1} y_i)\right|, \qquad (x_{n+1}, y_{n+1}) = (x_1, y_1)$$

Persamaan 2.21

Luas sekitar 3.058 m² yang diperoleh sebelumnya dari Google Earth tetap merupakan estimasi awal area Lapangan Woodball UNNES. Perhitungan dari titik GNSS dapat digunakan sebagai pemeriksaan tambahan, tetapi titik yang mengalami galat posisi tidak boleh langsung diperlakukan sebagai batas aman robot.

Ketelitian posisi horizontal pada pengujian dapat dinyatakan dengan *root mean square error* (RMSE) terhadap (N) titik acuan, sebagaimana Persamaan (2.22).

$$\mathrm{RMSE}_{xy} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[\left(x_{G,i} - x_{\mathrm{ref},i}\right)^2 + \left(y_{G,i} - y_{\mathrm{ref},i}\right)^2\right]}$$

Persamaan 2.22

Pasangan $(x_{G,i}, y_{G,i})$ merupakan posisi hasil GNSS, sedangkan $(x_{\mathrm{ref},i}, y_{\mathrm{ref},i})$ merupakan posisi acuan yang diukur dengan metode yang ditetapkan pada pengujian. Nilai RMSE ini menilai posisi global, bukan ketelitian fusi encoder–IMU. Selain galat posisi, pengujian perlu mencatat ketersediaan *fix*, usia data, dan perubahan hasil pembacaan di area terbuka maupun dekat pepohonan. Hasilnya menjadi dasar untuk menetapkan kapan data posisi layak digunakan oleh navigasi.

2.2.6 Logika Penghindaran Halangan Dinamis (Smart Dodge) dan State Machine Navigasi

Pada rancangan tim, Orange Pi 4 Pro menjalankan app_yolov8.py untuk mendeteksi halangan dan menentukan status navigasi, sedangkan ESP32 pada subsistem kendali gerak mengeksekusi perintah melalui dua driver BTS7960 dan motor wiper MK461340. *Finite State Machine* (FSM) mengatur perpindahan antara mengikuti lajur, memperlambat gerak, menghindar, dan berhenti. Pemisahan keputusan navigasi dari pelaksanaan kendali motor memungkinkan encoder dan IMU tetap digunakan untuk memantau gerak robot selama manuver. Pendekatan FSM untuk mengatur perpindahan mode mengikuti lintasan dan menghindari halangan juga digunakan oleh Lucet dkk. (2024).

Status	Kondisi dari subsistem navigasi	Respons ESP32 dan penggerak
STATE_PATROL	Lajur Boustrophedon tersedia dan jalur di depan dinyatakan bebas	Mengikuti lajur dengan kecepatan acuan operasi; encoder dan IMU memberikan umpan balik gerak
STATE_SLOWDOWN	Halangan memasuki Zona Kuning, dengan rentang rancangan 1,5–2,5 m	Mengurangi kecepatan acuan menjadi 50% sambil mempertahankan kendali arah
STATE_EMERGENCY_STOP	Orang atau hewan memasuki Zona Merah, dengan batas rancangan <1,5 m; atau jalur tidak aman	Menerima perintah berhenti S dan meniadakan perintah gerak motor
STATE_SMART_DODGE	Halangan statis berada pada lajur dan navigasi telah menentukan sisi serta lintasan deviasi yang aman	Mengikuti lintasan deviasi menggunakan kendali motor kiri–kanan dan umpan balik odometri–IMU
STATE_REJOIN	Robot telah melewati halangan	Mengikuti *waypoint rejoin* hingga kembali sejajar dengan lajur sapuan, kemudian masuk ke STATE_PATROL

Tabel 2.6 Status navigasi dan respons subsistem kendali gerak

2.2.7 Kendali Kinematika Berbasis Fuzzy Sugeno

Kendali kinematika bertugas mengubah penyimpangan posisi dan arah robot terhadap lintasan acuan menjadi perintah bagi penggerak kiri dan kanan. Pada prototipe ini, lintasan acuan disediakan oleh subsistem navigasi, sedangkan posisi dan orientasi lokal diperkirakan dari fusi odometri–IMU. Kedua informasi tersebut menjadi masukan FIS 2 berbasis Sugeno orde nol yang dijalankan pada ESP32. Nguyen dan Vu (2022) menunjukkan penerapan kendali fuzzy bersama model kinematika untuk mengikuti lintasan robot *differential drive*. Aji dan Sutikno (2023) menerapkan keluaran Sugeno untuk mengatur kecepatan motor kiri dan kanan pada robot pengikut lintasan. Rancangan penelitian ini menerapkan prinsip tersebut pada robot bersabuk rantai yang parameter geraknya perlu dikalibrasi di lapangan.

Variabel	Simbol	Sumber	Peran
Arah lintasan acuan	(θ_{ref})	Subsistem navigasi	Menentukan arah yang perlu diikuti
Posisi dan arah lokal robot	(x_f, y_f, θ_f)	Fusi odometri–IMU	Menyatakan keadaan gerak robot
Galat arah	(e_θ)	Hasil perhitungan	Masukan fuzzy untuk koreksi orientasi

Galat lintasan melintang	(e_{ct})	Hasil perhitungan	Masukan fuzzy untuk koreksi posisi terhadap lajur
Koreksi gerak diferensial	(Δu)	Keluaran FIS 2	Mengubah perintah penggerak kiri dan kanan

Tabel 2.7 Variabel kendali kinematika FIS 2

Galat arah merupakan selisih antara arah lintasan acuan dan orientasi hasil fusi, dengan hasil dibatasi pada rentang sudut utama menggunakan fungsi $wrap$, sebagaimana Persamaan (2.23).

$$[\boldsymbol{e_{\theta,k} = wrap\left(\theta_{ref,k} - \theta_{f,k}\right)}$$

Persamaan 2.23

Galat lintasan melintang adalah jarak bertanda antara pusat robot dan garis acuan yang melalui titik $(A = (x_A, y_A))$. Nilainya dihitung dengan Persamaan (2.24).

$$[\boldsymbol{e_{ct,k} = -(x_{f,k} - x_A)\sin\theta_{ref,k} + (y_{f,k} - y_A)\cos\theta_{ref,k}}$$

Persamaan 2.24

Dengan konvensi sumbu (x) ke timur dan (y) ke utara, $(e_{ct} > 0)$menunjukkan robot berada di sebelah kiri arah lajur, sedangkan $(e_{ct} < 0)$menunjukkan robot berada di sebelah kanan. Koordinat lintasan acuan dan hasil estimasi robot harus dinyatakan dalam kerangka koordinat lokal yang sama; selisih koordinat GPS mentah dalam derajat tidak dapat langsung digunakan sebagai galat dalam meter. Akmal dkk. (2024) meneliti kendali pelacakan lintasan berbasis fuzzy dengan mempertimbangkan gangguan pada umpan balik arah. Hal tersebut menjadi alasan untuk memeriksa pengaruh galat sensor terhadap kedua masukan FIS pada prototipe ini, tanpa menganggap metode *interval type-2* dalam penelitian tersebut sama dengan Sugeno orde nol yang dirancang di sini.

Masukan $(e_\theta) dan (e_{ct})$ difuzifikasi ke dalam himpunan linguistik negatif besar (NB), negatif kecil (NK), nol (Z), positif kecil (PK), dan positif besar (PB). Lima himpunan per masukan menghasilkan rancangan awal paling banyak **25** kombinasi aturan. Batas numerik setiap himpunan ditetapkan setelah rentang galat robot diperoleh dari simulasi MATLAB dan pengujian awal; karena itu, nilai batasnya belum dinyatakan sebagai hasil kalibrasi. Fungsi keanggotaan segitiga untuk tiga titik $(a < b < c)$ dapat ditulis sebagai Persamaan (2.25), sedangkan himpunan pada ujung domain dapat menggunakan fungsi trapesium.

$$\boldsymbol{\mu(x,a,b,c)} = \begin{cases} \boldsymbol{0} \\ \boldsymbol{\frac{x-a}{b-a}} \\ \boldsymbol{\frac{c=x}{c-b}} \end{cases} \boldsymbol{, x \leq ataux \geq c, a < x \leq b, b < x < c}$$

Persamaan 2.25

Setiap aturan berbentuk JIKA galat arah berada pada himpunan tertentu DAN galat melintang berada pada himpunan tertentu, MAKA keluaran koreksi bernilai konstanta (z_i). Kekuatan aktivasi aturan ke$-(i)$ dihitung menggunakan operator minimum:

$$[\boldsymbol{w_i = \mathbf{mi\,n}\,!\left[\mu_{\theta,i}(e_\theta), \mu_{ct,i}(e_{ct})\right]}$$

Persamaan 2.26

Kondisi masukan	Kecenderungan keluaran (Δu)	Respons penggerak
$e_\theta \approx 0,\ e_{ct} \approx 0$	Mendekati nol	Kiri dan kanan mengikuti perintah dasar yang seimbang
$e_\theta > 0,\ e_{ct} \approx 0$	Positif	Penggerak kanan diperkuat relatif terhadap kiri agar robot berbelok ke kiri
$(e_\theta < 0,\ e_{ct} \approx 0$	Negatif	Penggerak kiri diperkuat relatif terhadap kanan agar robot berbelok ke kanan
$e_\theta \approx 0,\ e_{ct} > 0$	Negatif	Robot diarahkan kembali ke kanan menuju lajur
$e_\theta \approx 0,\ e_c t < 0$	Positif	Robot diarahkan kembali ke kiri menuju lajur

Tabel 2.8 Contoh arah koreksi pada aturan FIS 2

Tabel 2.8 menunjukkan arah respons, bukan nilai PWM akhir. Ketika $(e_\theta) dan (e_{ct})$ meminta koreksi yang berlawanan, besarnya keluaran ditentukan oleh aturan yang aktif dan bobot keanggotaannya. Karena konsekuen Sugeno orde nol berupa konstanta, keluaran tegas (Δu_k) diperoleh melalui rata-rata tertimbang pada Persamaan (2.27).

$$\boldsymbol{\Delta uk = \frac{\sum_{i=1}^{n} w_i z_i}{\sum_{i=1}^{n} w_i}\ ,\ \sum_{i=1}^{n} w_i > 0}$$

Persamaan 2.27

Jika tidak ada aturan yang aktif, pengendali harus menangani kondisi tersebut secara eksplisit dan tidak melakukan pembagian dengan nol. Keluaran (Δu_k) kemudian digabungkan dengan perintah gerak dasar $(u_{0,k})$ untuk menghasilkan perintah kiri dan kanan:

$$u_{L,k}^{F} = sat\left(u_{0,k} - \Delta u_k\right)$$

$$u_{R,k}^{F} = sat\left(u_{0,k} + \Delta u_k\right)$$

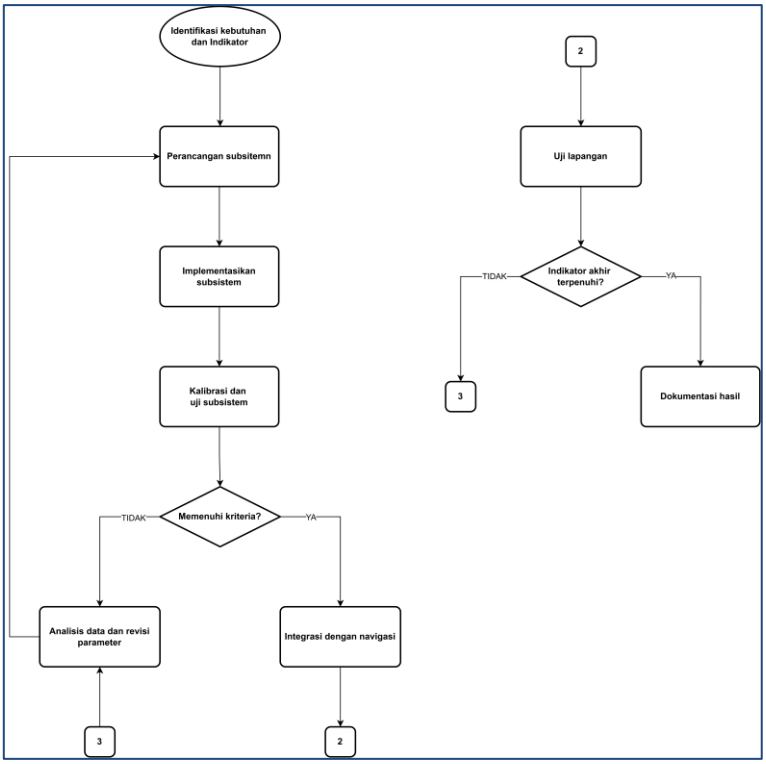
Persamaan 2.28

Fungsi (sat) membatasi perintah sesuai rentang PWM dan arah putaran yang telah diuji pada motor wiper MK461340 serta driver BTS7960. Tanda koreksi pada Persamaan (2.28) berlaku untuk konvensi penggerak kiri–kanan pada Tabel 2.8; arah fisik motor harus diperiksa saat perakitan. Pasangan $((u_L^F, u_R^F))$ menjadi keluaran FIS 2 yang selanjutnya diizinkan, diperlambat, atau dihentikan oleh FSM .Hasil perhitungan MATLAB akan dibandingkan dengan keluaran program pada ESP32 untuk kombinasi masukan galat yang sama, kemudian respons fisik robot diuji pada lintasan lurus, belokan, dan permukaan berumput.

BAB III
METODE PELAKSANAAN

3.1 Pendekatan Perancangan

Perancangan subsistem kendali gerak menggunakan pendekatan prototipe iteratif. Rancangan awal dimodelkan, diimplementasikan, dikalibrasi, dan diuji; data hasil pengujian digunakan untuk memperbaiki parameter sebelum pengujian berikutnya. Pendekatan ini diperlukan karena gerak robot bersabuk rantai dengan kemudi *skid-steering* dipengaruhi slip, karakteristik penggerak, dan kondisi permukaan. Dengan demikian, parameter kinematika dari desain robot perlu diperiksa pada prototipe fisik (Cantelli dkk., 2016; Zuo dkk., 2020).



Gambar 3.1 Alur pendekatan perancangan subsistem kendali gerak

Penerapan setiap tahap diringkas pada Tabel 3.1. Desain robot digunakan sebagai acuan penempatan komponen dan penentuan parameter geometri awal. Model gerak dan pengendali fuzzy Sugeno terlebih dahulu disimulasikan menggunakan MATLAB, kemudian perhitungan kendali diimplementasikan dalam C++ pada ESP32. Ukuran pemasangan, arah putaran motor, pembacaan sensor, serta respons gerak diperiksa kembali pada prototipe fisik.

Tahap	Fokus Pekerjaan	Keluaran yang direncanakan
Indetifikasi kebutuhan	Menetapkan fungsi penggerak, perintah lintasan dan status FSM yang diterima dari navigasi, data umpan balik, serta indikator kinerja	Spesifikasi fungsi dan kriteria pengujian

Perancangan	Menyusun model gerak *skid-steering*, estimasi lokal encoder–IMU, FIS 2 Sugeno untuk koreksi lintasan, dan hubungan antarkomponen	Model kinematika, diagram sistem, dan parameter awal
Simulasi	Menguji respons model gerak dan keluaran FIS 2 terhadap kombinasi galat arah serta galat lintasan menggunakan MATLAB	Hasil simulasi sebagai acuan implementasi
Implementasi	Memasang dua motor wiper Canter MK461340, dua driver BTS7960, dua rotary encoder, dan IMU MPU-9250; memprogram pembacaan sensor dan kendali motor pada ESP32	Subsistem penggerak dan program awal yang dapat diuji
Kalibrasi dan uji susistem	Memeriksa arah motor, pulsa efektif encoder, bias dan orientasi sumbu IMU, estimasi gerak, keluaran fuzzy, serta kemampuan mengikuti lintasan	Data kalibrasi dan hasil uji untuk menilai rancangan
Analisis dan revisi	Membandingkan hasil pengujian dengan indikator, menelusuri sumber penyimpangan, lalu memperbarui parameter terkait	Rancangan yang diperbarui dan catatan perubahan
Integrasi dan uji lapangan	Menghubungkan lintasan dan status navigasi dari Orange Pi 4 Pro dengan kendali motor pada ESP32 serta mengirimkan umpan balik gerak	Subsistem kendali gerak yang siap diuji bersama sistem robot
Dokumentasi	Mencatat konfigurasi akhit, hasil pengujian, dan keterbatasan prototipe	Bahan evaluasi dan laporan prototipe

Tabel 3.1 Penerapan pendekatan perancangan pada subsistem kendali gerak

Pada sistem terintegrasi, subsistem navigasi menyediakan lintasan acuan, referensi posisi global dari GPS, serta status operasi berdasarkan deteksi halangan. Subsistem kendali gerak memperkirakan posisi dan orientasi lokal dari encoder dan IMU, menghitung galat terhadap lintasan acuan, lalu menggunakan FIS 2 Sugeno untuk mengoreksi perintah motor kiri dan kanan. Status FSM menentukan apakah perintah tersebut dijalankan, diperlambat, atau dihentikan. GPS digunakan sebagai referensi navigasi global dan belum ditetapkan sebagai masukan fusi lokal. Keberhasilan pendekatan dinilai melalui ketelitian estimasi gerak, kesesuaian keluaran ESP32 terhadap simulasi MATLAB, kemampuan mengikuti lintasan, dan respons penggerak terhadap perubahan status

3.2 Lokasi dan Waktu

3.2.1 Lokasi Perancangan dan Pengujian

Lokasi	Kegiatan yang direncanakan	Peralatan atau fasilitas utama	Keluaran
Labroratorium E11	Perancangan model kinematika, diagram rangkaian, program odometri, fusi sensor, dan kendali fuzzy	Komputer, perangkat pengembangan program, dokumen rancangan mekanik robot	Model, diagram, program, dan konfigurasi awal
Labroratorium E11	Pemasangan komponen, pemeriksaan sambungan listrik, kalibrasi encoder dan IMU, serta uji putaran motor dan perintah berhenti	Peralatan perakitan, alat ukur listrik, alat ukur panjang dan sudut, penyangga robot	Rakitan subsistem dan data kalibrasi awal
Lapangan Woodball UNNES	Pengujian gerak lurus, berbelok, mengikuti lintasan, dan integrasi dengan navigasi pada permukaan berumput	Petak uji, penanda batas dan titik acuan, alat ukur jarak, perangkat pencatat data, fasilitas penghentian darurat	Data kinerja subsistem pada lokasi studi kasus

Tabel 3.2 Lokasi dan fungsi kegiatan perancangan

3.2.2 Area dan Kondisi Pengujian Lapangan

Pengujian awal direncanakan pada petak seluas 100 m² di dalam Lapangan Woodball UNNES. Bentuk, dimensi sisi, dan posisi petak ditentukan melalui pengukuran langsung sebelum pengujian. Batas petak, titik awal robot, lintasan

acuan, titik pengukuran, serta zona aman diberi penanda sehingga setiap percobaan dapat dilakukan pada konfigurasi yang tercatat.

Tampak atas lapangan yang telah diperoleh dari Google Earth digunakan sebagai acuan awal untuk memilih posisi petak. Ukuran petak dan titik acuannya tetap diverifikasi di lokasi. Data panjang sisi batas lapangan dari Google Earth tidak digunakan untuk menyatakan luas petak uji maupun luas seluruh lapangan tanpa pengukuran poligon yang sesuai.

Parameter	Cara pencatatan yang direncanakan	Kegunaan
Batas dan luas petak uji	Mengukur serta menandai batas petak hingga luasnya mencapai 100 m²	Menetapkan wilayah pengujian yang dapat diulang
Titik awal dan arah hadap robot	Menandai posisi dan orientasi awal pada setiap skenario	Menyamakan kondisi awal antarpercobaan
Bentuk dan panjang lintasan	Mengukur segmen lurus, titik belok, dan titik akhir	Menyediakan lintasan acuan bagi pengendali
Kondisi permukaan	Mencatat kondisi rumput, tanah, dan bagian permukaan yang tidak rata	Menjelaskan variasi respons gerak dan kemungkinan slip
Kemiringan pada area yang digunakan	Mengukur atau mencatat kemiringan pada segmen uji	Menentukan kondisi medan yang benar-benar diuji
Kondisi cuaca dan permukaan	Mencatat waktu, cuaca, serta keadaan permukaan saat percobaan	Memastikan hasil dibandingkan pada kondisi yang terdokumentasi
Keberadaan pengguna dan halangan	Memeriksa area sebelum robot dioperasikan	Menetapkan zona aman dan skenario pengujian

Tabel 3.3 Data kondisi Lokasi yang dicatat sebelum pengujian

Pengujian direncanakan pada siang hari, saat tidak hujan, tidak terdapat genangan pada petak, dan area kerja telah dikosongkan dari pengguna lapangan. Operator mengawasi robot selama percobaan dan dapat mengaktifkan penghentian darurat apabila diperlukan. Kondisi medan yang tidak tercakup dalam percobaan tidak dinyatakan telah teruji.

Tampak atas Lapangan Woodball UNNES beserta batas petak 100 m², titik awal, lintasan acuan, dan zona aman akan disajikan sebagai Gambar 3.2



Gambar 3.2 Tampak atas Lapangan Woodball UNNES

3.2.3 Waktu dan Jadwal Pelaksanaan

Pelaksanaan direncanakan selama 16 minggu dalam satu periode pengembangan prototipe. Jadwal pada Tabel 3.4 menunjukkan urutan relatif kegiatan; tanggal kalender disesuaikan dengan jadwal akademik, ketersediaan komponen, dan kesempatan penggunaan lapangan. Pengujian luar ruang dijadwalkan setelah pemeriksaan dasar penggerak dan fungsi penghentian robot selesai.

Kegiatan	M1-M2	M3-M4	M5-M6	M7-M8	M9-M10	M11-M12	M13-M14	M15-M16
Analisis kebutuhan dan antarmuka navigasi								
Pemeriksaan rancangan V08 dan komponen fisik								
Perakitan pebggerak encoder, IMU, dan driver								
Kalibrasi sensor dan karakterisasi motor								
Pemodelan kinematika dan pengembangan odometri–IMU								
Perancangan dan pengujian kendali fuzzy Sugeno								
Integrasi dengan subsistem navigasi								

Survei dan penandaan petak uji								
Pengujian lapangan dan perbaikan parameter								
Pengolahan data dan dokumentasi hasil								

Tabel 3.4 Rencana waktu perancangan subsistem kendali gerak

Keterangan: M = minggu pelaksanaan, kotak berwarna = kegiatan direncanakan berlangsung.

Kegiatan yang saling bergantung dapat bergeser apabila hasil pemeriksaan komponen atau kalibrasi memerlukan perbaikan. Setiap pergeseran jadwal dan perubahan rancangan dicatat agar urutan pelaksanaan serta konfigurasi yang digunakan pada pengujian akhir tetap dapat ditelusuri.

3.3 Prosedur Perancangan

Prosedur perancangan subsistem kendali gerak dilakukan melalui analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Model kinematika, fusi odometri–IMU, serta pengendali *fuzzy* Sugeno terlebih dahulu disusun dan diperiksa melalui simulasi MATLAB pada laptop. Hasil rancangan kemudian diimplementasikan sebagai program C++ pada ESP32 untuk membaca sensor dan mengendalikan motor penggerak robot. Data hasil pengujian digunakan untuk memperbaiki parameter rancangan sebelum pengujian diulang.

3.3.1 Analisis Kebutuhan

Analisis kebutuhan menetapkan komponen, fungsi, dan aliran data pada subsistem kendali gerak. Robot menggunakan dua motor wiper depan Canter MK461340 untuk menggerakkan sabuk rantai kiri dan kanan. Masing-masing motor dikendalikan melalui satu modul BTS7960. Dua *rotary encoder* nominal 1000 PPR menyediakan data putaran penggerak, sedangkan IMU MPU-9250 menyediakan data gerak inersial dan orientasi. ESP32 direncanakan sebagai pengendali yang membaca sensor, menjalankan perhitungan kendali, dan menghasilkan sinyal untuk kedua driver motor.

Subsistem navigasi pada Orange Pi 4 Pro menyediakan lintasan acuan dan keputusan terkait halangan. Data posisi GPS digunakan pada subsistem navigasi sebagai referensi global, sedangkan estimasi gerak lokal yang dirancang pada bagian ini menggunakan odometri dan IMU. Laptop digunakan untuk simulasi

MATLAB, penulisan program C++ melalui Arduino IDE, serta pencatatan dan analisis data pengujian.

No	Komponen	Jumlah	Fungsi dalam prosedur perancangan
1	Motor wiper depan Canter MK461340	2	Menggerakkan sabuk rantai kiri dan kanan
2	Driver motor BTS7960	2	Mengatur arah dan besaran penggerak masing-masing motor berdasarkan perintah ESP32
3	IMU MPU-9250	1	Menyediakan data giroskop, akselerometer, dan magnetometer untuk estimasi gerak serta pemeriksaan orientasi
4	ESP32	1	Membaca encoder dan IMU, menjalankan program C++, serta memberikan perintah ke dua driver BTS7960
5	Orange Pi 4 Pro	1	Menyediakan hasil navigasi berupa lintasan acuan dan status terkait halangan melalui antarmuka yang disepakati antarsubsistem
6	Laptop dengan MATLAB dan Fuzzy Logic Toolbox	1	Merancang serta mensimulasikan model kinematika, fusi sensor, dan fuzzy Sugeno sebelum implementasi fisik
7	Arduino IDE dan program C++	1	Menulis, mengunggah, dan memeriksa program kendali yang berjalan pada ESP32
8	Perangkat pengukuran posisi, jarak, dan sudut acuan	Sesuai kebutuhan uji	Menghasilkan data pembanding yang independen dari encoder dan MPU-9250
9	Rotary Encoder nominal 1000 PPR	2	Mengukur putaran sisi kiri dan kanan untuk odometri

Tabel 3.5 Kebutuhan perangkat keras dan perangkat lunak subsistem kendali gerak

Pada akhir tahap ini, spesifikasi listrik komponen, jenis antarmuka sensor, letak pemasangan encoder, orientasi sumbu IMU, dan format pertukaran data antara Orange Pi 4 Pro dan ESP32 harus tercatat.

3.3.2 Perancangan Sistem

3.3.2.1 Arsitektur Perangkat Keras

Desain perangkat keras menentukan letak dua motor wiper Canter MK461340, dua encoder, MPU-9250, ESP32, dan dua driver BTS7960 pada model

robot Trinautom. Kedua encoder dirancang untuk membaca sisi penggerak yang berbeda. IMU ditempatkan pada dudukan yang tidak mudah berubah posisi, dengan arah sumbu sensor terhadap arah depan robot ditentukan dan dicatat. Diagram pengawatan disusun untuk menunjukkan sambungan encoder dan MPU-9250 ke ESP32, sambungan sinyal kendali ESP32 ke masing-masing BTS7960, serta jalur daya kedua motor. Rancangan juga memisahkan kebutuhan daya motor dari kebutuhan daya rangkaian logika. Nomor pin, tegangan suplai, dan kapasitas arus ditetapkan berdasarkan papan serta komponen fisik yang digunakan sebelum rangkaian diberi daya.

3.3.2.2 Arsitektur Perangkat Lunak

Perangkat lunak dibagi menjadi dua lingkungan kerja. MATLAB pada laptop digunakan untuk menghitung respons model kinematika dan fusi odometri–IMU, menyusun sistem *fuzzy* Sugeno orde nol melalui Fuzzy Logic Designer, serta memeriksa keluaran aturan menggunakan *Rule Viewer*. Skenario simulasi mencakup gerak lurus, perubahan arah, dan kombinasi galat arah serta galat lintasan. Keluaran simulasi disimpan sebagai data acuan implementasi.

Program C++ pada ESP32 dirancang untuk menghitung pulsa dua encoder, membaca MPU-9250, memberi penanda waktu pada data, memperbarui estimasi gerak, dan menghitung keluaran *fuzzy*. Program kemudian menghasilkan perintah arah dan *Pulse Width Modulation* (PWM) untuk driver BTS7960 kiri dan kanan. Program C++ ditulis dan diunggah melalui Arduino IDE pada laptop; ketika robot beroperasi, program tersebut berjalan pada ESP32, bukan pada laptop MATLAB. ESP32 juga menerima lintasan acuan dan status keselamatan dari subsistem navigasi Orange Pi 4 Pro. Format pesan, laju pembaruan, dan jenis sambungan komunikasi ditetapkan bersama pada tahap integrasi. Status berhenti harus dapat mengesampingkan keluaran *fuzzy* sehingga kedua motor berhenti ketika navigasi menyatakan pergerakan tidak aman.

3.3.2.3 Perancangan Model Kinematika dan Fuzzy

Model kinematika dirancang untuk mengubah putaran penggerak kiri dan kanan menjadi perkiraan perpindahan serta perubahan arah robot. MATLAB digunakan untuk memeriksa hasil model dengan masukan putaran yang diketahui. Parameter jarak tempuh per hitungan encoder dan jarak efektif antarsisi penggerak diperlakukan sebagai parameter awal yang harus dikalibrasi pada robot fisik. Rancangan fusi menggunakan perpindahan dari odometri dan informasi gerak sudut dari MPU-9250 untuk menghasilkan estimasi posisi serta orientasi lokal. Data kedua sensor harus menggunakan acuan waktu dan konvensi sumbu yang sama.

Pengendali *fuzzy* Sugeno menggunakan galat arah (*heading error*) dan galat lintasan (*cross-track error*) sebagai dua masukan utama. Pada MATLAB Fuzzy Logic Designer disusun rentang masukan, fungsi keanggotaan, aturan, serta konsekuen untuk perintah gerak kiri dan kanan. Keluaran MATLAB diperiksa terlebih dahulu, lalu aturan dan parameter yang sama dipindahkan ke program C++ pada ESP32.

3.3.3 Implementasi

3.3.3.1 Impelmentasi Perangkat Keras

Dua motor wiper Canter MK461340 dipasang pada sisi penggerak kiri dan kanan. Masing-masing motor dihubungkan ke driver BTS7960 tersendiri. Dua encoder dipasang sesuai rancangan untuk membaca putaran penggerak, sedangkan MPU-9250 dipasang pada dudukan dengan orientasi sumbu yang telah ditentukan. ESP32, driver, sensor, dan sumber daya kemudian dihubungkan mengikuti diagram *wiring*. Sebelum robot dijalankan, sambungan listrik diperiksa. Motor kiri dan kanan diuji secara terpisah melalui perintah maju, mundur, dan berhenti. Arah hitungan encoder dibandingkan dengan arah putaran aktual. MPU-9250 diperiksa saat robot diam dan ketika orientasi robot diubah. Hasil pemeriksaan dicatat sebelum seluruh komponen dioperasikan bersama.

3.3.3.2 Implementasi Perangkat Lunak

Program dikembangkan dan diuji pada perangkat yang menjalankannya:

1. Pada laptop dengan MATLAB, dibuat model perhitungan odometri dan fusi odometri–IMU. Fuzzy Logic Designer digunakan untuk menyusun fuzzy inference system Sugeno, menampilkan fungsi keanggotaan, dan memeriksa pasangan masukan–keluaran melalui Rule Viewer. Keluaran untuk skenario uji disimpan sebagai acuan.
2. Pada laptop dengan Arduino IDE, ditulis program C++ untuk ESP32. Program memuat pembacaan dua encoder, pembacaan MPU-9250, perhitungan estimasi gerak, perhitungan galat terhadap lintasan, aturan fuzzy Sugeno, penerimaan status navigasi, serta pembentukan perintah PWM dan arah untuk dua BTS7960.
3. Pada ESP32 setelah program diunggah, pembacaan sensor dan perhitungan kendali berjalan berulang saat robot beroperasi. Hasil pembacaan encoder, data IMU, estimasi posisi–orientasi, galat lintasan, keluaran fuzzy, dan PWM kedua motor dikirim sebagai data pemantauan untuk pencatatan pengujian.

4. Pada antarmuka Orange Pi 4 Pro–ESP32, lintasan acuan dan status keselamatan dikirim dari subsistem navigasi ke pengendali gerak. Perintah berhenti diperiksa sebelum PWM diterapkan pada motor.

3.3.3.3 Implementasi Model

Parameter kinematika hasil kalibrasi dimasukkan ke program MATLAB dan C++ dengan satuan yang konsisten. Fungsi keanggotaan, konsekuen, dan aturan Sugeno pada ESP32 dibuat sesuai rancangan MATLAB. Keluaran kedua implementasi dibandingkan menggunakan kombinasi galat arah dan galat lintasan yang sama sebelum motor dijalankan menggunakan keluaran pengendali. Model fusi kemudian menerima rekaman encoder dan MPU-9250 yang memiliki penanda waktu. Estimasi hasil fusi digunakan untuk menghitung galat terhadap lintasan acuan. Hasil akhirnya diterjemahkan menjadi perintah bagi BTS7960 kiri dan kanan, dengan batas perintah dan keputusan berhenti tetap diberlakukan.

3.3.4 Pengujian Sistem

3.3.4.1 Pengujian Perangkat Keras

1. Menguji kedua motor wiper Canter MK461340 melalui BTS7960 secara terpisah untuk memeriksa arah putaran, respons terhadap perubahan PWM, dan kemampuan berhenti.
2. Memutar masing-masing penggerak dengan jumlah putaran yang diketahui, lalu membandingkannya dengan hitungan encoder untuk memperoleh hitungan efektif per putaran dan arah pencacahan.
3. Mengukur keluaran MPU-9250 ketika robot diam dan saat diubah ke sudut acuan tertentu untuk memeriksa bias giroskop serta orientasi pemasangan sensor.
4. Memeriksa tegangan, arus, dan suhu driver serta motor saat uji awal dan saat robot bergerak berbeban.

3.3.4.2 Pengujian Perangkat Lunak

1. Memberikan data putaran penggerak yang diketahui pada model odometri di MATLAB dan program C++, kemudian membandingkan posisi serta orientasi hasil perhitungannya.
2. Memberikan pasangan galat arah dan galat lintasan yang sama ke Rule Viewer MATLAB dan program C++ pada ESP32. Keluaran PWM kiri dan kanan dari keduanya dicatat untuk memeriksa kesesuaian implementasi aturan Sugeno.

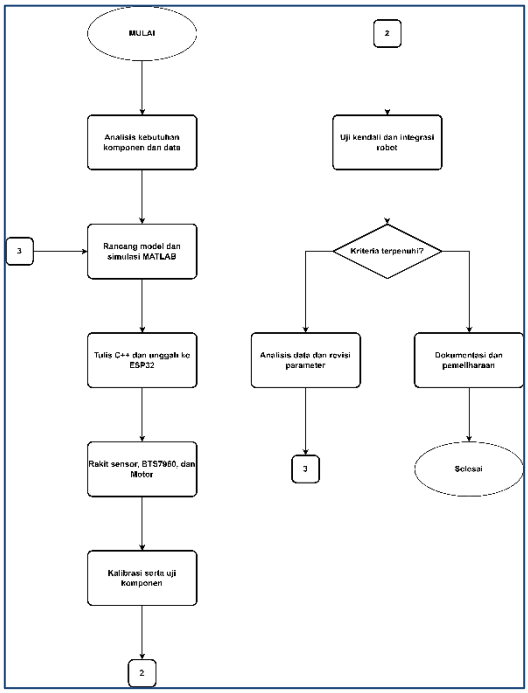
3. Memeriksa pembacaan encoder dan MPU-9250 pada ESP32 menggunakan penanda waktu, termasuk respons program terhadap data yang hilang atau tidak valid.
4. Mengirim status navigasi uji dari Orange Pi 4 Pro ke ESP32 untuk memastikan status berhenti mengesampingkan perintah motor yang dihasilkan pengendali fuzzy.
5. Memeriksa bahwa pembacaan sensor dan pembaruan kendali tetap berlangsung selama perubahan status gerak, termasuk ketika robot berbelok atau menjalankan manuver yang diperintahkan navigasi.

3.3.4.3 Pengujian Model dan Integrasi

1. Menjalankan robot pada lintasan lurus dan berbelok untuk menentukan hubungan hitungan encoder dengan jarak tempuh serta mengkalibrasi parameter jarak efektif antarsisi penggerak.
2. Merekam encoder dan MPU-9250 pada lintasan uji yang sama. Posisi dan orientasi hasil odometri saja serta fusi odometri–IMU dibandingkan dengan pengukuran acuan yang independen. Target rancangan adalah penurunan RMSE posisi dan RMSE orientasi masing-masing sekurang-kurangnya 20% dibandingkan odometri saja.
3. Menguji kemampuan mengikuti lintasan dengan mencatat galat arah, galat lintasan, kecepatan, serta PWM motor kiri dan kanan. Data tersebut digunakan untuk menilai hubungan antara keputusan fuzzy dan gerakan fisik robot.
4. Menguji integrasi dengan navigasi untuk memastikan lintasan acuan diterima, koreksi gerak dijalankan, dan motor berhenti ketika status keselamatan mengharuskannya.
5. Melaksanakan pengujian awal terintegrasi pada petak sekitar 100 m² di Lapangan Woodball UNNES. Batas petak, kondisi permukaan, lintasan, konfigurasi program, dan hasil setiap pengulangan dicatat.

3.3.5 Pemeliharaan

Setelah pengujian, kondisi sambungan ESP32–sensor, konektor kedua BTS7960, pemasangan encoder, dudukan MPU-9250, motor, dan sabuk rantai diperiksa. Program C++ pada ESP32, model MATLAB, parameter kalibrasi, aturan *fuzzy*, dan rekaman pengujian disimpan menurut versi. Jika encoder, transmisi, motor, atau posisi IMU berubah, parameter yang dipengaruhi perubahan tersebut diperiksa dan dikalibrasi kembali sebelum hasil uji berikutnya dibandingkan.



Gambar 3.3 Digram alir prosedur perancangan subsistem kendali Gerak

3.4 Teknik Perancangan

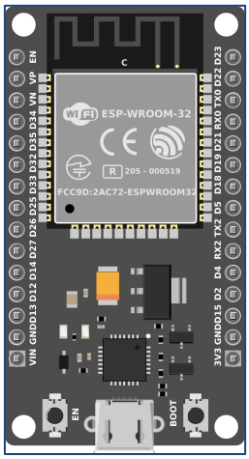
Teknik perancangan subsistem kendali gerak mencakup pemilihan komponen, penempatan fisik, pengawatan, pemodelan kinematika, fusi odometri–IMU, dan penyusunan pengendali *fuzzy* Sugeno. Model dihitung dan disimulasikan terlebih dahulu menggunakan MATLAB pada laptop. Parameter yang telah diperiksa kemudian diterapkan dalam program C++ pada ESP32 untuk mengendalikan dua motor penggerak sabuk rantai. Hubungan antara keluaran program dan gerak aktual robot ditetapkan melalui kalibrasi, karena putaran motor dan nilai PWM tidak dapat langsung dianggap setara dengan kecepatan robot di atas rumput.

3.4.1 Pemilihan Komponen dan Arsitektur Sistem

Komponen dipilih berdasarkan fungsi, kesesuaian tegangan, kemudahan pemasangan, dan kebutuhan pengaturan aliran. Spesifikasi komponen dibedakan dari pengaturan yang ditetapkan dalam rancangan.

3.4.1.1 ESP32 Dev Board

ESP32 digunakan sebagai pengendali keluaran penyemprotan. Program C++ yang ditulis melalui Arduino IDE menerima data serial, memeriksa perintah, dan menghasilkan PWM untuk driver pompa. Modul ESP-WROOM-32 menyediakan GPIO, UART, serta pengendali PWM yang mendukung kebutuhan tersebut.



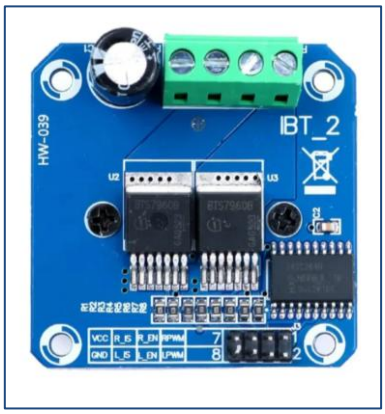
Gambar 3.4 ESP32 Dev Board berbasis ESP-WROOM-32

Spesifikasi	Detail
Modul utama	ESP-WROOM-32.
Prosesor	Xtensa dual-core 32 bit, frekuensi hingga 240 MHz.
Memori modul	SRAM 520 KB dan flash SPI 4 MB.
Tegangan logika	3,3V; rentang suplai modul 3,0–3,6V.
Antarmuka yang digunakan	USB serial melalui konverter pada board dan GPIO PWM.
Pengaturan rancangan	PWM 8 bit dengan nilai perintah 0–255.

Tabel 3.6. Spesifikasi ESP32 Dev Board

3.4.1.2 Driver Motor BTS7960

Modul BTS7960 tipe IBT-2 digunakan untuk menyalurkan daya ke pompa berdasarkan sinyal kendali ESP32. Dua IC half-bridge pada modul membentuk penggerak motor yang dapat dikendalikan melalui RPWM, LPWM, R_EN, dan L_EN. Pada prototipe ini, pompa dijalankan dalam satu arah sehingga hanya satu jalur PWM yang digunakan untuk mengatur keluaran.



Gambar 3.5 Modul driver motor BTS7960

Spesifikasi	Detail
Tipe modul	IBT-2 dengan dua IC BTS7960.
Rentang suplai daya IC	5,5–27,5 V menurut datasheet BTS7960.
Suplai daya pada rancangan	12 V dari step-down, masuk melalui terminal B+ dan B−.
Pin kendali	RPWM, LPWM, R_EN, dan L_EN.
Keluaran daya	Terminal M+ dan M− menuju pompa.
Kemampuan PWM IC	Hingga 25 kHz; frekuensi kerja pompa ditetapkan melalui pengujian.
Suplai logika rancangan	3,3 V untuk varian modul dengan buffer 74HC244 seperti pada gambar.

Tabel 3.7 Spesifikasi dan pengaturan driver BTS7960

Rentang tegangan dan kemampuan PWM mengacu pada spesifikasi IC BTS7960. Kelayakan modul untuk pompa ditentukan dari arus kerja dan suhu saat pengujian; angka arus pada datasheet IC tidak digunakan sebagai jaminan arus kontinu seluruh modul.

3.4.1.3 Motor Wiper Depan Canter MK461340

Dua motor wiper depan Canter MK461340 digunakan sebagai penggerak sabuk rantai kiri dan kanan robot. Setiap motor dikendalikan secara terpisah oleh driver BTS7960 melalui perintah arah dan PWM dari ESP32. Putaran, arus, serta respons motor saat berbeban akan diukur untuk menetapkan parameter penggerak dan memastikan kesesuaiannya dengan operasi robot.



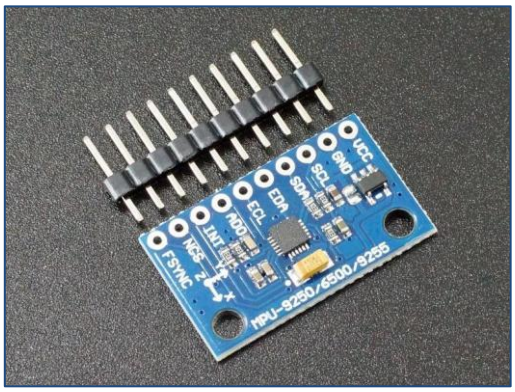
Gambar 3.6 Motor Driver depan Canter MK461340

Spesifikasi	Detail
Model	Motor wiper depan Canter MK461340
Tegangan nominal	24V
Jumlah PIN	5
Pengendalian	PWM dan arah melalui driver BTS7960

Tabel 3.8 Spesifikasi motor wiper depan Canter MK461340

3.4.1.4 IMU MPU-9250

IMU MPU-9250 digunakan untuk membaca gerak dan orientasi robot melalui giroskop, akselerometer, dan magnetometer tiga sumbu. Pada subsistem kendali gerak, pembacaan kecepatan sudut sumbu vertikal dari giroskop dipadukan dengan odometri encoder untuk memperkirakan orientasi lokal robot. Sensor dihubungkan ke ESP32 melalui I²C dan dikalibrasi setelah terpasang pada sasis.



Gambar 3.7 Modul IMU MPU-9250

Spesifikasi	Detail
Sensor utama	Giroskop, akselerometer, dan magnetometer; masing-masing tiga sumbu
Rentang giroskop	Dapat diatur pada ±250, ±500, ±1.000, atau ±2.000 °/s
Rentang akselerometer	Dapat diatur pada ±2, ±4, ±8, atau ±16 g
Antarmuka yang digunakan	I²C menuju ESP32
Tegangan catu IC sensor	Terminal M+ dan M− menuju pompa.
Kemampuan PWM IC	2,4–3,6 V; tegangan masukan modul diperiksa pada papan yang dibeli
Suplai logika rancangan	3,3 V untuk varian modul dengan buffer 74HC244 seperti pada gambar.
Pengaturan rancangan	Giroskop sumbu vertikal sebagai masukan utama estimasi yaw; bias dan orientasi pemasangan dikalibrasi

Tabel 3.9 Spesifikasi IMU MPU-9250

3.4.1.5 Rotary Encoder 1000 PPR

Dua *rotary encoder* 1000 PPR digunakan untuk membaca putaran penggerak sabuk rantai kiri dan kanan. ESP32 menghitung perubahan pulsa setiap sisi sebagai masukan odometri. Sebelum perhitungan jarak diterapkan, jumlah hitungan efektif per putaran penggerak ditentukan melalui kalibrasi sesuai posisi pemasangan encoder dan cara pembacaan sinyal.



Gambar 3.8 Rotary Encdoder 100 PPR

Spesifikasi	Detail
Resolution	1000 PPR
Output Phases	2-phase quadrature (Channel A and Channel B)
Operating Voltage	5 VDC to 24 VDC
Maximum Speed	Up to 5,000 to 6,000 RPM
Frequency Response	Up to 100 kHz max frequency capacity
Current Consumption	< 80 mA typical quiescent/load

Tabel 3.10 Spesifikasi rotary encoder 1000 PPR

3.4.2 Desain Fisik

3.4.2.1 Tata Letak Komponen

Tata letak komponen dirancang berdasarkan hubungan antara penggerak sabuk rantai, pembacaan gerak robot, dan ruang pemasangan perangkat kendali. Pembahasan pada subbab ini difokuskan pada posisi dua motor wiper Canter MK461340, dua rotary encoder, dua driver BTS7960, IMU MPU-9250, dan ESP32. Komponen subsistem lain ditampilkan pada gambar untuk memperlihatkan ruang yang tersedia dan hubungan fisiknya dengan subsistem kendali gerak. Posisi akhir, dimensi dudukan, dan jarak antarkomponen ditetapkan berdasarkan gambar rancangan serta pemeriksaan pada komponen fisik.

a. Tampak atas *(Top View)*

Tampak atas menunjukkan tata letak robot dari atas, dengan dua sabuk rantai di sisi kiri dan kanan sebagai penggerak *skid-steering*, kamera dan Orange Pi untuk persepsi serta navigasi, serta ESP32 dan MPU-9250 untuk kendali gerak dan pembacaan orientasi badan robot. Pada sistem ini, ESP32 menerima perintah navigasi, mengolah data encoder dan IMU, lalu mengatur gerak penggerak kiri dan

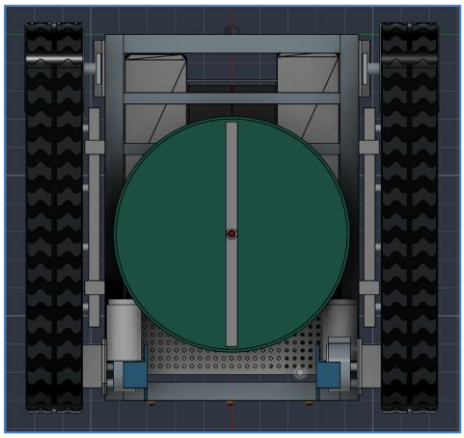
kanan; komponen subsistem pemotongan dan penyemprotan menempati sasis yang sama. Agar gambar dapat digunakan sebagai acuan perakitan dan perhitungan kinematika, arah depan robot, orientasi sumbu MPU-9250, posisi motor dan encoder pada tiap sisi, serta jarak antarsisi penggerak perlu diberi penanda. Posisi encoder dan jarak tersebut harus diperiksa pada robot fisik, sedangkan pembacaan IMU perlu diuji saat motor mati dan menyala untuk menilai pengaruh getaran serta kelistrikan.



Gambar 3.9 Tampak atas desain robot

b. Tampak bawah *(Bottom View)*

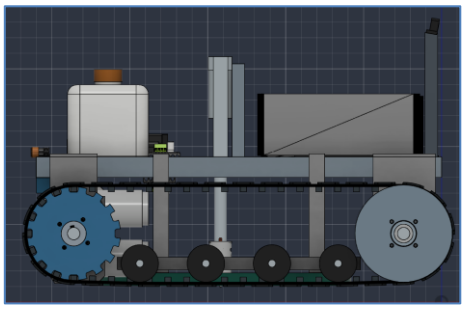
Tampak bawah memperlihatkan dua motor penggerak yang terhubung dengan sabuk rantai kiri dan kanan, pisau pemotong, serta struktur sasis bawah. Kedua motor menjadi aktuator gerak *skid-steering*; perbedaan putaran sisi kiri dan kanan menghasilkan perubahan arah robot. Dari tampak ini, pemasangan motor perlu diperiksa terhadap keselarasan poros penggerak dan ruang gerak sabuk rantai, sedangkan posisi pisau perlu dipastikan tidak mengganggu motor, transmisi, atau jalur kabel. Dudukan encoder belum ditampilkan dalam rancangan, sehingga poros yang akan diukur dan ruang pemasangannya masih harus ditetapkan sebelum perhitungan odometri difinalkan.



Gambar 3.10 Tampak bawah desain robot

c. Tampak samping *(Side View)*

Tampak samping memperlihatkan ketinggian sproket penggerak utama, sproket penopang, *idler*, dan jalur sabuk rantai terhadap sasis serta permukaan tanah. Susunan tersebut menentukan bagian sabuk yang bersentuhan dengan tanah, jarak bebas sasis, dan kestabilan gerak robot pada permukaan berumput. Sproket penggerak utama meneruskan putaran motor ke sabuk, sedangkan penopang dan *idler* menjaga jalur serta tegangan sabuk selama robot bergerak. Untuk perhitungan odometri, diameter sproket pada gambar digunakan sebagai parameter rancangan awal; hubungan antara putaran sproket dan perpindahan robot tetap perlu dikalibrasi karena selip sabuk dapat menyebabkan perpindahan aktual berbeda dari hasil perhitungan.

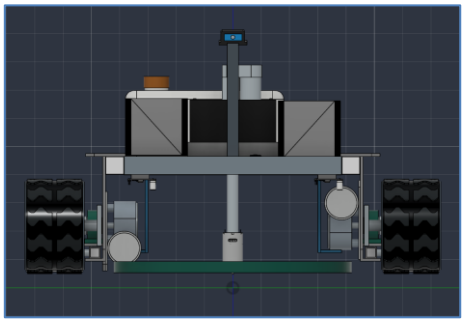


Gambar 3.11 Tampak samping desain robot

d. Tampak depan *(Front View)*

Tampak depan memperlihatkan kamera, dua sabuk rantai, motor penggerak yang tampak dari depan, dan mekanisme pemotong. Posisi kamera perlu memberikan bidang pandang ke arah lintasan robot agar informasi halangan dapat diteruskan ke sistem navigasi sebelum robot mendekatinya. Susunan sabuk rantai dan motor kiri–kanan diperiksa untuk memastikan keduanya tidak menghalangi

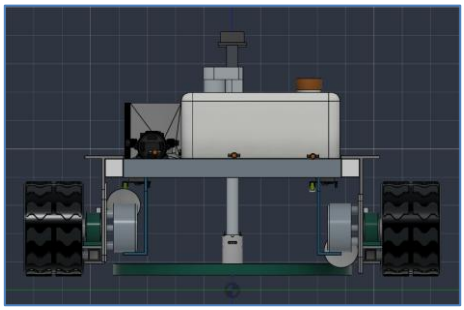
gerak mekanisme pemotong; kesesuaian posisi kedua sisi juga menjadi acuan awal model gerak *skid-steering*.



Gambar 3.12 Tampak depan desain robot

e. Tampak belakang *(Back View)*

Tampak belakang memperlihatkan susunan kedua sabuk rantai, motor penggerak, sasis, dan mekanisme pemotong dari arah belakang; kamera tidak tampak karena ditempatkan menghadap arah gerak robot. Tampak ini melengkapi pemeriksaan dari depan, terutama untuk menilai ruang pemasangan motor kiri dan kanan, akses ke sambungan penggerak, serta jarak antara komponen penggerak dan pemotong.

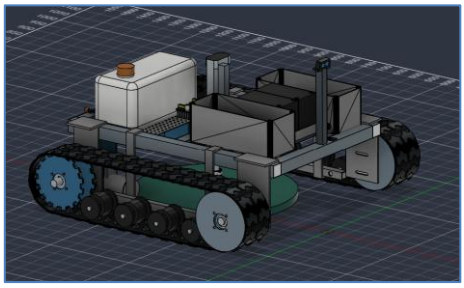


Gambar 3.13 Tampak belakang desain robot

f. Tampak keseluruhan *(Eagle View)*

Tampak keseluruhan memperlihatkan hubungan antara sasis, dua sabuk rantai, penggerak, kamera, perangkat kendali, mekanisme pemotong, dan komponen subsistem lain dalam satu rakitan. Tampak keseluruhan digunakan untuk menilai apakah setiap subsistem memiliki ruang pemasangan dan akses perawatan yang memadai tanpa mengganggu lintasan sabuk rantai maupun gerak dek pemotong. Bagi subsistem kendali gerak, gambar ini juga menjadi acuan untuk memeriksa orientasi kamera terhadap arah maju robot, posisi MPU-9250 terhadap badan robot, dan kemungkinan jalur kabel sensor berdekatan dengan motor serta

kabel daya. Kesesuaian tata letak pada gambar tetap akan diperiksa setelah komponen dipasang pada sasis fisik.



Gambar 3.14 Tampak keseluruhan desain robot

3.4.2.2 Material dan Sambungan

Material dipilih untuk menopang komponen, menyediakan tempat pengikat, dan memudahkan perawatan. Pilihan bahan struktural berikut merupakan rancangan material; ukuran penampang, ketebalan pelat, dan jenis filamen ditetapkan setelah penyesuaian terhadap rangka serta proses pembuatan yang digunakan.

Bagian	Material rancangan	Fungsi
Rangka utama	Baja hollow galvanis.	Menopang seluruh komponen pada bidang 65 × 45 cm.
Batang melintang	Baja hollow dengan pelapisan cat.	Menghubungkan sisi rangka dan menopang dudukan.
Pelat dasar	Pelat baja berlubang.	Menopang tangki dan pompa serta menyediakan titik pengikat.
Kotak mikrokontroler	Termoplastik hasil cetak 3D.	Menempatkan ESP32, BTS7960, dan jalur kabel.
Dudukan dan pengikat	Braket logam, baut, mur, dan pengikat tangki.	Mempertahankan posisi komponen serta memudahkan pembongkaran.
Penahan kabel dan selang	Klem serta cable tie.	Menjaga jalur tetap rapi dan tidak tertarik.

Tabel 3.11. Material rancangan fisik

3.4.2.3 Teknik Desain Menggunakan Autodesk Fusion

Autodesk Fusion digunakan untuk membuat model tiga dimensi secara parametrik. Panjang rangka, lebar rangka, ukuran dudukan, dan jarak antarkomponen dijadikan parameter agar perubahan ukuran dapat diterapkan tanpa menggambar ulang seluruh rancangan.

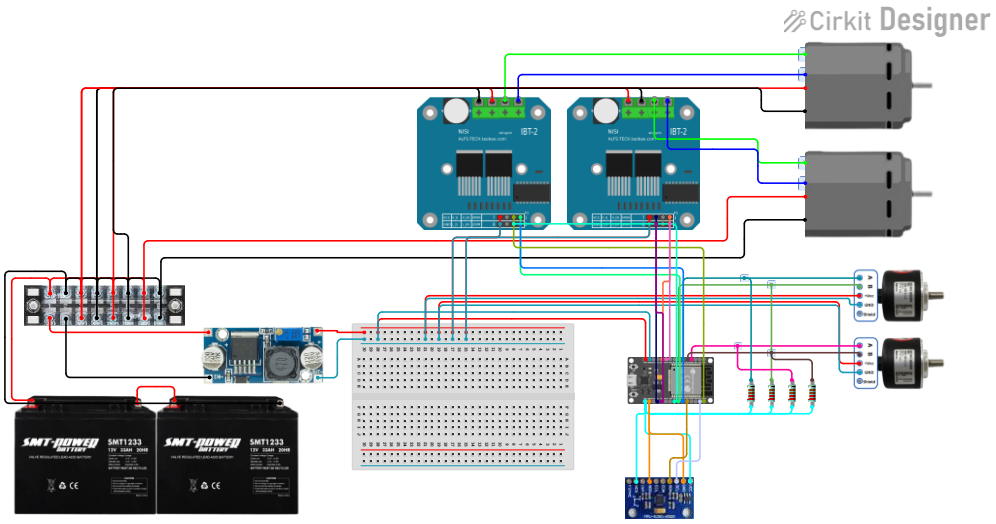
1. Membuat sketsa bidang rangka berukuran 650 × 450 mm, kemudian membentuk profil dan pelat menggunakan fitur pemodelan padat.
2. Membuat setiap komponen sebagai bagian terpisah dengan ukuran dari pengukuran aktual, lalu menyusun posisinya pada rakitan.

3. Menentukan lubang pengikat, ruang konektor, jalur selang, dan akses pengisian tangki serta pelepasan tutup kotak.
4. Memeriksa kemungkinan tabrakan antarkomponen dan ruang perawatan, kemudian menyiapkan gambar kerja untuk pembuatan dudukan dan perakitan.

3.4.3 Desain Pengawatan Sistem

3.4.3.1 Desain Wiring

Desain wiring menghubungkan dua baterai 12 V, terminal block, modul *step-down*, ESP32, dua driver BTS7960, dua motor wiper Canter MK461340, dua rotary encoder 1000 PPR, dan IMU MPU-9250. Kedua baterai disusun seri untuk menghasilkan tegangan nominal 24 V yang didistribusikan melalui terminal block ke jalur daya penggerak. Modul *step-down* menurunkan tegangan menjadi 5 V untuk catu rangkaian kendali. ESP32 mengirimkan sinyal kendali ke masing-masing BTS7960 sehingga motor sisi kiri dan kanan dapat diatur secara terpisah.



Gambar 3.15 Desain wiring subsistem penggerak

Kedua encoder mengirimkan pulsa putaran penggerak kiri dan kanan ke ESP32 sebagai masukan odometri, sedangkan MPU-9250 mengirimkan data gerak badan robot untuk estimasi orientasi. ESP32 mengolah kedua sumber data tersebut sebagai umpan balik kendali fuzzy Sugeno. *Breadboard* digunakan untuk sambungan sinyal dan distribusi catu kendali pada tahap prototipe; jalur arus motor disambungkan melalui terminal dan kabel daya. Ground ESP32, sensor, dan kedua driver dihubungkan sebagai referensi sinyal bersama.

3.4.3.2 Alokasi Sambungan dan Pin

Alokasi sambungan disusun untuk menghubungkan dua motor penggerak, dua encoder 1000 PPR, dan MPU-9250 ke ESP32. Setiap motor dikendalikan oleh satu BTS7960, sedangkan kanal A dan B dari masing-masing encoder dibaca secara terpisah untuk memperoleh jumlah serta arah putaran. MPU-9250 menggunakan komunikasi I²C.

Komponen	Terminal atau sinyal	Sambungan yang direncanakan	Fungsi
Baterai 12 V × 2	Susunan seri	Terminal block daya 24 V	Sumber daya penggerak
Terminal block	Keluaran 24 V dan GND	BTS7960 kiri, BTS7960 kanan, serta masukan *step-down*	Distribusi daya
Step-down	Keluaran 5 V dan GND	Suplai kendali melalui jalur 5 V	Menurunkan tegangan baterai untuk perangkat kendali
BTS7960 kiri	RPWM; LPWM	ESP32 GPIO25; GPIO26	Kendali arah dan PWM motor kiri
BTS7960 kiri	R_EN dan L_EN	ESP32 GPIO18	Mengaktifkan driver kiri
BTS7960 kiri	Keluaran motor	Motor wiper kiri	Menggerakkan sabuk rantai kiri
BTS7960 kanan	RPWM; LPWM	ESP32 GPIO27; GPIO14	Kendali arah dan PWM motor kanan
BTS7960 kanan	R_EN dan L_EN	ESP32 GPIO19	Mengaktifkan driver kanan
BTS7960 kanan	Keluaran motor	Motor wiper kanan	Menggerakkan sabuk rantai kanan
Encoder kiri	Kanal A; kanal B	ESP32 GPIO32; GPIO33	Pembacaan pulsa dan arah putaran kiri
Encoder kanan	Kanal A; kanal B	ESP32 GPIO34; GPIO35	Pembacaan pulsa dan arah putaran kanan

Kedua encoder	VCC; GND	Suplai sesuai spesifikasi encoder; GND bersama	Catu daya encoder
Setiap kanal encoder	Resistor *pull-up* 10 kΩ	Kanal A/B ke rel logika 3,3 V; empat resistor seluruhnya	Menetapkan level tinggi keluaran NPN *open collector*
MPU-9250	SDA; SCL	ESP32 GPIO21; GPIO22	Komunikasi I²C
MPU-9250	VCC; GND	Suplai sesuai spesifikasi modul; GND bersama	Catu daya IMU
MPU-9250	AD0; INT	AD0 ke GND; INT belum dialokasikan	Menetapkan alamat I²C; pembacaan awal dilakukan tanpa interupsi
ESP32, sensor, dan driver	GND	Referensi GND bersama	Acuan tegangan sinyal kendali

Tabel 3.12 Rancangan alokasi sambungan dan pin subsistem penggerak

3.4.4 Perancangan Model Fuzzy Logic

3.4.4.1 Arasitektur Kendali Fuzzy

Arsitektur kendali gerak menggunakan dua Fuzzy Inference System (FIS) Sugeno orde nol yang dijalankan pada ESP32 dan dikoordinasikan oleh *Finite State Machine* (FSM). FIS 1 mengatur perlambatan berdasarkan kedekatan halangan, sedangkan FIS 2 menghasilkan koreksi gerak berdasarkan galat arah dan galat lateral terhadap lintasan. Orange Pi menyediakan informasi halangan, lintasan acuan, dan status navigasi. Dua encoder 1000 PPR serta MPU-9250 menyediakan umpan balik untuk memperkirakan posisi dan orientasi lokal robot.

FIS 1 menerima masukan Jarak_Piksel, yaitu koordinat vertikal titik bawah *bounding box* objek hidup, seperti manusia atau hewan, yang terdeteksi melalui pengolahan citra pada Orange Pi. Peningkatan nilai piksel menunjukkan objek semakin dekat dengan robot. Rule base FIS 1 menghasilkan Pengali_Kecepatan pada rentang 0–1 untuk mengurangi PWM kedua motor secara bertahap hingga perintah gerak menjadi nol. FIS 2 mengatur pasangan PWM kiri dan kanan untuk pelacakan lintasan, sedangkan FIS 1 mengatur perlambatan ketika objek hidup terdeteksi. FSM menentukan mode operasi dan memberikan prioritas pada perintah berhenti darurat terhadap keluaran kedua FIS.

FIS 2 menerima galat arah (e_{θ}) dan galat lateral (e_{ct}). Kedua galat dihitung dengan membandingkan lintasan acuan terhadap estimasi posisi dan orientasi lokal hasil fusi encoder–IMU. Pulsa encoder kiri dan kanan digunakan untuk memperkirakan perpindahan, sedangkan giroskop MPU-9250 memberikan informasi putaran badan robot. Berdasarkan kedua masukan tersebut, rule base FIS 2 menghasilkan pasangan keluaran PWM_Kiri dan PWM_Kanan untuk mengarahkan robot kembali menuju lintasan acuan. Koreksi diferensial (Δu) merepresentasikan setengah selisih kedua keluaran PWM, sehingga menunjukkan kecenderungan arah belok yang dihasilkan oleh pasangan perintah motor.

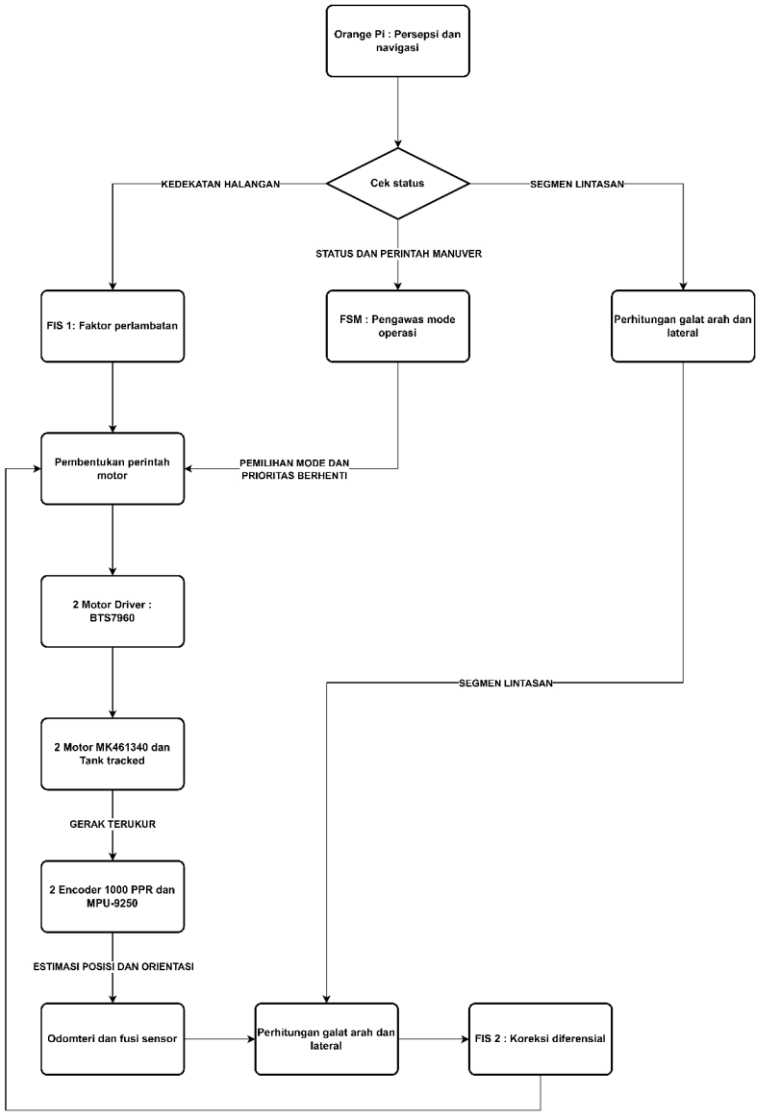
Pengendali	Masukan	Keluaran dan peran
FIS 1 Sugeno — perlambatan	Jarak_Piksel dari titik bawah *bounding box* objek hidup yang terdeteksi oleh CV	Pengali_Kecepatan pada rentang 0-1 untuk mengurangi PWM kedua motor secara bertahap
FIS 2 Sugeno — pelacakan lintasan	Galat arah (e_{θ}) dan galat lateral (e_{ct})	Pasangan PWM_Kiri dan PWM_Kanan untuk mengoreksi gerak menuju lintasan acuan, (Δu) menyatakan setengah selisih kedua keluaran
FSM — pengawas operasi	Kelas halangan, zona bahaya, dan status navigasi, serta posisi dan orientasi local robot	Menentukan pelacakan, perlambatan, penghindaran halangan, putar balik dan perpindahan lajur (U_TURN), atau berhenti darurat

Tabel 3.13 Hubungan input-output FIS 1, FIS 2, FSM

Pembentukan perintah motor menggabungkan kecepatan dasar, faktor perlambatan FIS 1, dan koreksi FIS 2 sesuai mode yang dipilih FSM. Saat pelacakan normal, FIS 2 mengoreksi arah gerak. Saat perlambatan, kecepatan dasar dikurangi tanpa menghilangkan kebutuhan koreksi lintasan. Saat penghindaran halangan, FSM menjalankan perintah manuver atau menggunakan lintasan acuan yang diperbarui oleh navigasi. Ketika menerima perintah berhenti, keluaran gerak dihentikan. Perintah akhir dibatasi pada rentang aktuasi yang ditetapkan, kemudian dikirim sebagai sinyal PWM dan arah putaran kepada dua driver BTS7960 yang masing-masing mengendalikan satu motor wiper Canter MK461340.

Pada sistem terintegrasi, komunikasi Orange Pi–ESP32 perlu membawa informasi kedekatan halangan, status operasi, dan data acuan lintasan yang cukup untuk menghitung kedua galat. Lintasan acuan dan estimasi posisi robot harus menggunakan kerangka koordinat lokal yang sama. GPS menyediakan referensi posisi global bagi navigasi dan belum digunakan sebagai masukan fusi lokal encoder–IMU. Model kedua FIS dirancang serta disimulasikan di MATLAB, kemudian diterapkan dalam C++ pada ESP32. Kesamaan keluaran MATLAB dan

ESP32 diperiksa menggunakan kombinasi masukan yang identik sebelum pengujian kendali gerak pada robot fisik.



Gambar 3.16 Arsitektur kendali gerak menggunakan FIS 1, FIS 2, dan Fusi encoder-IMU

3.4.4.2 Penentuan Variabel Masukan dan Keluaran

Variabel kendali ditetapkan berdasarkan pembagian fungsi FIS 1 sebagai pengatur perlambatan dan FIS 2 sebagai pengendali kinematika gerak. FIS 1 mengolah informasi kedekatan halangan dari subsistem persepsi untuk menghasilkan faktor pengurangan kecepatan. FIS 2 mengolah galat arah dan galat lateral lintasan untuk menghasilkan koreksi perbedaan kecepatan penggerak kiri dan kanan. Kedua pengendali dirancang menggunakan inferensi fuzzy Sugeno orde

nol, disimulasikan pada MATLAB, kemudian diimplementasikan menggunakan bahasa C++ pada ESP32.

Pengendali	Jenis variabel	Variabel dan simbol	Satuan/rentang	Sumber dan fungsi
FIS 1	Masukan	Jarak_Piksel, (Y_{obs})	Piksel; [450640]	Koordinat vertikal titik bawah *bounding box* objek hidup hasil pengolahan citra pada Orange Pi. Menjadi indikator kedekatan objek terhadap robot.
FIS 1	Keluaran	Pengali_Kecepatan, (k_v)	Tanpa satuan; [01]	Dihasilkan oleh rule base FIS 1 untuk mengalikan PWM kedua motor. Nilai 1 mempertahankan perintah, sedangkan nilai 0 menetapkan perintah PWM nol.
FIS 2	Masukan	Heading_Error, (e_θ)	Derajat; [−3030]	Selisih arah lintasan acuan terhadap orientasi hasil fusi encoder–IMU. Galat yang dihitung dalam radian dikonversi menjadi derajat sebelum diproses oleh FIS 2.
FIS 2	Masukan	CrossTrack_Error, (e_{ct})	Meter; [−0.50.5]	Jarak bertanda posisi estimasi robot terhadap garis lintasan acuan.
FIS 2	Keluaran	PWM_Kiri (u_L), PWM_Kanan (u_R)	Nilai PWM 8-bit; [0 255]	Perintah motor kanan hasil inferensi Sugeno yang bekerja bersama keluaran motor kiri untuk menghasilkan gerak pelacakan lintasan.

Tabel 3.14 Variabel masukan dan keluaran pengendali fuzzy

a. Masukan dan keluaran FIS 1

Masukan FIS 1 berupa Jarak_Piksel, yaitu koordinat vertikal titik bawah *bounding box* objek hidup, seperti manusia atau hewan, yang dikirimkan oleh Orange Pi kepada ESP32. Nilai piksel yang semakin besar menunjukkan posisi objek semakin dekat dengan robot pada konfigurasi kamera yang digunakan. Masukan ini diproses melalui lima himpunan keanggotaan, yaitu Aman, Waspada, Dekat, Kritis, dan Stop. Rule base FIS 1 menghasilkan Pengali_Kecepatan \(k_v\) pada rentang 0–1 untuk mengalikan PWM kedua motor, sehingga perintah gerak berkurang secara bertahap ketika objek mendekat dan menjadi nol pada kondisi Stop.

Klasifikasi objek, pemilihan arah penghindaran, dan keputusan penghentian darurat dikelola oleh FSM. Ketika FSM menetapkan STATE_EMERGENCY_STOP, perintah PWM kedua motor ditetapkan nol dengan mengesampingkan keluaran kedua FIS. Dengan demikian, FIS 1 menyediakan perlambatan bertahap, sedangkan FSM memberikan prioritas penghentian darurat ketika kondisi bahaya terpenuhi.

b. Masukan galat arah FIS 2

Galat arah dihitung dari selisih orientasi lintasan acuan (θ_{ref}) terhadap orientasi robot ($\hat{\theta}$). Orientasi robot diperoleh dari estimasi fusi dua rotary encoder 1000 PPR dan MPU-9250. Selisih sudut dibatasi pada interval $[-\pi, \pi]$ agar perubahan melewati batas sudut tidak menghasilkan galat yang keliru.

Untuk segmen lintasan dari titik $A(x_A, y_A)$ menuju $B(x_B, y_B)$, arah acuan dihitung sebagai:

$$\theta_{ref} = atan2(y_B - y_A,\ x_B - x_A)$$

Persamaan 3.1

Galat arah dihitung menggunakan :

$$e_\theta = atan2\left(\sin\left(\theta_{ref} - \hat{\theta}\right), \cos\left(\theta_{ref} - \hat{\theta}\right)\right)$$

Persamaan 3.2

Nilai (e_θ) dikonversi dari radian menjadi derajat sebelum digunakan sebagai masukan Heading_Error pada FIS 2. Domain fuzzy ditetapkan sebesar $[-30,30]°$, sedangkan hasil perhitungan galat arah dapat berada di luar domain tersebut.

Pada sistem koordinat dengan arah putaran positif berlawanan jarum jam, ($e_\theta > 0$) menunjukkan kebutuhan koreksi arah ke kiri, sedangkan($e_\theta < 0$) menunjukkan kebutuhan koreksi ke kanan. Nilai mendekati nol menunjukkan kesesuaian arah robot dengan arah lintasan.

c. Masukan galat lateral FIS 2

Galat lateral menunjukkan penyimpangan posisi robot terhadap garis lintasan acuan. Perhitungan menggunakan posisi estimasi robot ($\hat{x}, \hat{y}$), titik awal segmen $A(x_A, y_A)$, dan arah lintasan (θ_{ref}) dalam kerangka koordinat lokal yang sama.

$$e_{\mathrm{ct}} = -(\hat{x} - x_A)\sin\theta_{\mathrm{ref}} + (\hat{y} - y_A)\cos\theta_{\mathrm{ref}}$$

Persamaan 3.3

Nilai $(e_{\mathrm{ct}} > 0)$ menunjukkan posisi robot berada di sebelah kiri lintasan menurut arah perjalanan dari A menuju B, sehingga diperlukan koreksi menuju kanan. Sebaliknya, $(e_{\mathrm{ct}} < 0)$ menunjukkan posisi robot berada di sebelah kanan lintasan, sehingga diperlukan koreksi menuju kiri. Nilai nol menunjukkan posisi estimasi robot berada pada garis acuan. Galat lateral dinyatakan dalam meter dan digunakan sebagai masukan CrossTrack_Error pada FIS 2 dengan domain fuzzy $[-0{,}5,\ +0{,}5]$ m. Informasi GPS digunakan sebagai referensi navigasi global, sedangkan estimasi gerak lokal diperoleh dari fusi encoder–IMU.d. Keluaran FIS 2 dan konvensi kendali

Keluaran FIS 2 berupa pasangan PWM_Kiri (u_R) dan PWM_Kanan (u_L) pada rentang 0–255. Kedua keluaran dihasilkan melalui rule base Sugeno untuk mengoreksi arah dan posisi robot terhadap lintasan acuan. ESP32 menerapkan perintah tersebut melalui dua driver BTS7960 yang mengendalikan motor wiper MK461340 kiri dan kanan. Koreksi diferensial (Δu) menyatakan setengah selisih keluaran kanan terhadap kiri dan digunakan untuk menjelaskan kecenderungan arah belok.

Kondisi keluaran	Makna perintah gerak
$(u_R > u_L), sehingga (\Delta u > 0)$	Koreksi belok kiri melalui perintah motor kanan yang lebih besar daripada motor kiri.
$(u_R < u_L), sehingga (\Delta u < 0)$	Koreksi belok kanan melalui perintah motor kiri yang lebih besar daripada motor kanan.
$(u_R = u_L), sehingga (\Delta u = 0)$	Perintah kedua motor sama; tidak terdapat koreksi diferensial dari FIS 2.

Tabel 3.15 Hubungan kondisi keluaran dan makna perintah gerak

Konvensi tersebut menjadi dasar penyusunan aturan fuzzy. Galat arah positif memerlukan koreksi positif, sedangkan galat lateral positif memerlukan koreksi negatif. Perbedaan ini harus dipertahankan pada simulasi MATLAB dan program C++ ESP32. Kesamaan perintah PWM belum menjamin kesamaan kecepatan kedua track, sehingga hubungan PWM terhadap kecepatan masing-masing motor perlu dikarakterisasi melalui pengujian encoder.

e. Hubungan Variabel fuzzy dengan *Finite State Machine* (FSM)

FSM mengatur penggunaan keluaran FIS 1 dan FIS 2 berdasarkan status navigasi yang diterima ESP32 dari Orange Pi. Status tersebut merupakan variabel diskrit pengatur mode operasi, sehingga tidak melalui proses fuzzifikasi. FIS 1 menghasilkan faktor kecepatan (k_v), sedangkan FIS 2 menghasilkan koreksi diferensial (Δu). FSM menentukan kapan keluaran tersebut diterapkan, kapan lintasan acuan diganti untuk penghindaran halangan, dan kapan perintah motor harus dihentikan.

Status FSM	Kondisi pemicu	Peran FIS 1	Peran FIS 2	Tindakan penggerak
STATE_PATROL	Jalur kerja dinyatakan bebas dari halangan yang memerlukan respons	Faktor kecepatan ditetapkan pada kondisi normal, $(k_v = 1)$	Mengoreksi galat arah dan lateral terhadap lintasan CPP	Mengikuti lintasan pemotongan dengan kecepatan dasar yang ditetapkan
STATE_SLOWDOWN	Halangan memasuki Zona Kuning; estimasi jarak awal 1,5–2,5 m	Mengurangi faktor kecepatan berdasarkan kedekatan halangan	Mempertahankan arah dan posisi terhadap lintasan aktif	Bergerak lebih lambat sambil mempertahankan kendali lintasan
STATE_SMART_DODGE	Halangan statis memerlukan manuver penghindaran dan lintasan deviasi tersedia	Membatasi kecepatan selama manuver	Mengikuti lintasan deviasi dan lintasan kembali yang diberikan navigasi	Membelok melalui perbedaan kecepatan track kiri dan kanan
STATE_EMERGENCY_STOP	Objek berisiko tinggi, seperti manusia atau hewan, memasuki Zona Merah; estimasi jarak awal <1,5 m	Keluaran diabaikan	Keluaran diabaikan	ESP32 menetapkan perintah PWM kedua penggerak menjadi nol
STATE_U_TURN	Robot mencapai ujung segmen aktif dan lajur berikutnya tersedia	Pengurangan kecepatan tetap diterapkan apabila kondisi halangan mengharuskannya	Mengikuti referensi manuver perpindahan lajur dan mengoreksi penyelarasan terhadap lajur berikutnya	ESP32 mengendalikan kedua motor melalui BTS7960 untuk berbelok, bergeser menuju lajur berikutnya, dan menyelaraskan arah sebelum kembali ke STATE_PATROL

Tabel 3.16 Hubungan status FSM dengan pengendali fuzzy dan penggerak robot

3.4.4.3 Fungsi Keanggotaan Fuzzy dan Logika FSM

Fungsi keanggotaan dirancang untuk mengubah masukan numerik menjadi derajat keanggotaan pada interval [0,1]. FIS 1 mengatur perlambatan berdasarkan estimasi jarak halangan, sedangkan FIS 2 menghasilkan koreksi diferensial berdasarkan galat arah dan galat lateral. FSM mengatur penggunaan kedua pengendali sesuai kondisi operasi, termasuk penghindaran halangan, perpindahan lajur melalui STATE_U_TURN, dan penghentian darurat. Parameter berikut merupakan rancangan awal untuk simulasi MATLAB, yang akan diperbaiki berdasarkan kalibrasi kamera dan pengujian robot.

a. Bentuk Fungsi Keanggotaan

Fungsi segitiga digunakan untuk himpunan di bagian tengah domain, sedangkan fungsi trapesium digunakan pada kedua ujung domain. Bentuk tersebut dipilih agar perhitungan derajat keanggotaan dapat diimplementasikan pada ESP32 menggunakan bahasa C++.

Fungsi keanggotaan segitiga dinyatakan sebagai :

$$\boldsymbol{\mu(x,a,b,c) = \begin{cases} 0, \\ \dfrac{x-a}{b-a}, \\ \dfrac{c-x}{c-b}, \\ 0, \end{cases} , x \le a, a < x \le b, b < x < c, x \ge c}$$

Persamaan 3.4

Parameter (a) dan (c) merupakan batas himpunan, sedangkan (b) merupakan titik dengan derajat keanggotaan maksimum

Fungsi keanggotaan trapesium dinyatakan sebagai :

$$\boldsymbol{\mu(x,a,b,c) = \begin{cases} 0, \\ \dfrac{x-a}{b-a}, \\ 1, \\ \dfrac{d-x}{d-c}, \\ 0 \end{cases} , x-a, a \le x < b, b \le x \le c, c < x \le d, x > d}$$

Persamaan 3.5

Pada himpunan bahu dengan $(a = b)$ atau $(c = d)$, bagian transisi yang memiliki lebar nol tidak dihitung. Program C++ menggunakan kondisi percabangan

untuk mempertahankan nilai keanggotaan pada bahu dan menghindari pembagian dengan nol.

b. Fungsi Keanggotaan FIS 1 : Perlambatan

Masukan Jarak_Piksel menggunakan koordinat vertikal titik bawah *bounding box* manusia atau hewan yang diterima dari Orange Pi. Domain masukan ditetapkan pada 450–640 piksel. Nilai piksel yang meningkat menunjukkan objek semakin dekat pada konfigurasi citra dan kamera yang digunakan.

Setup input & Output

INPUT 1	Jarak_Piksel
OUTPUT 1	Pengali_Kecepatan

Tabel 3.17 Masukan dan keluaran FIS 1

Membership Function Input 1 : Jarak_Piksel

Setup Range : [450 640]

Himpunan	Bentuk fungsi	Parameter	Makna
Aman	TraMF	[400 400 460 500]	Posisi objek relatif jauh; perlambatan belum diperlukan
Waspada	TriMF	[480 520 560]	Objek mulai mendekati ruang gerak robot
Dekat	TriMF	[540 580 620]	Objek semakin dekat; pengurangan perintah motor ditingkatkan
Kritis	TriMF	[600 620 640]	Objek berada pada posisi citra yang memerlukan perlambatan besar
Stop	TrapMF	[630 640 700 700]	Posisi objek memerlukan perintah motor nol

Tabel 3.18 Parameter fungsi keanggotaan masukan FIS 1

Keluaran FIS 1 berupa pengali kecepatan (k_v) dengan domain 0–1. Nama konsekuen diseragamkan agar setiap aturan memiliki pasangan masukan dan keluaran yang jelas.

Membership Function Output 1 : Pengali_Kecepatan

Setup Range : [0 1]

No	Kondisi masukan	Konsekuen keluaran	Nilai (k_v)
1	Jarak_Piksel Aman	Penuh	1.00
2	Jarak_Piksel Waspada	Agak_Penuh	0.75
3	Jarak_Piksel Dekat	Setengah	0.50
4	Jarak_Piksel Kritis	Agak_Pelan	0.25
5	Jarak_Piksel Stop	Nol	0.00

Tabel 3.19 Konsekuen dan aturan FIS 1

Keluaran hasil inferensi dapat berada di antara konstanta tersebut karena menggunakan rata-rata berbobot. Faktor (k_v) diterapkan pada perintah kedua motor untuk mengurangi laju tanpa memindahkan fungsi perlambatan ke FIS 2. Ketika tidak terdapat objek hidup yang memerlukan perlambatan dan data persepsi valid, faktor pengali ditetapkan sebesar 1. Penghentian darurat oleh FSM tetap memiliki prioritas terhadap hasil inferensi.

c. Fungsi Keanggotaan FIS 2 : Pelacakan Lintasan

Membership Function Input 1 : Heading_Error

Setup Range : [-30 30]

Masukan Heading_Error menggunakan galat arah dalam derajat dengan domain −30° hingga +30°. Galat yang diperoleh dari perhitungan orientasi dalam radian dikonversi menjadi derajat sebelum diproses oleh FIS 2.

Himpunan	Bentuk fungsi	Parameter, derajat	Makna
NB (Negatif Besar)	TrapMF	[-180 -180 -30 -15]	Memerlukan kecenderungan koreksi kanan besar
NK (Negatif Kecil)	TriMF	[-25 -10 0]	Memerlukan kecenderungan koreksi kanan kecil
ZERO	TriMF	[-10 0 10]	Arah robot mendekati arah lintasan

PK (Positif Kecil)	TriMF	[0 10 25]	Memerlukan kecenderungan koreksi kiri kecil
PB (Positif Besar)	TrapMF	[15 30 180 180]	Memerlukan kecenderungan koreksi kiri besar

Tabel 3.20 Parameter fungsi keanggotaan galat arah

Membership Function Input 2 : CrossTrack_Error

Setup Range : [-0.5 0.5]

Masukan CrossTrack_Error menggunakan galat lateral dalam meter dengan domain −0.5 hingga +0.5 m. Sesuai konvensi perhitungan, galat negatif menunjukkan posisi robot di kanan lintasan, sedangkan galat positif menunjukkan posisi robot di kiri lintasan.

Himpunan	Bentuk fungsi	Parameter, meter	Makna posisi
KaJ (Kanan Jauh)	TrapMF	[-2 -2 -0.5 -0.2]	Robot berada jauh di kanan lintasan
KaD (Kanan Dekat)	TriMF	[-0.4 -0.15 0]	Robot berada dekat di kanan lintasan
ZERO	TriMF	[-0.15 0 0.15]	Posisi robot mendekati garis lintasan
KiD (Kiri Dekat)	TriMF	[0 0.15 0.4]	Robot berada dekat di kiri lintasan
KiJ (Kiri Jauh)	TrapMF	[0.2 0.5 2 2]	Robot berada jauh di kiri lintasan

Tabel 3.21 Parameter fungsi keanggotaan galat lateral

Parameter pembentuk fungsi keanggotaan dapat melampaui domain masukan. Domain dan parameter kurva merupakan pengaturan yang berbeda; nilai aktual di luar domain harus ditangani secara eksplisit oleh program. Penamaan himpunan kanan–kiri mengikuti tanda galat pada persamaan agar aturan motor menghasilkan koreksi menuju lintasan.

Keluaran FIS 2 berupa dua variabel, yaitu PWM_Kiri dan PWM_Kanan, masing-masing dengan domain 0–255 pada konfigurasi PWM 8-bit. Kedua variabel menggunakan konstanta konsekuen yang sama.

Membership Function Output 1 & 2 : PWM_Kiri & PWM_Kanan

Setup Range : [0 255]

Konsekuen	Nilai PWM kiri atau kanan
Sangat_Lambat	50
Lambat	120
Normal	180
Cepat	220
Sangat_Cepat	255

Tabel 3.22 Konsekuen keluaran PWM FIS 2

Kombinasi lima himpunan galat arah dan lima himpunan galat lateral menghasilkan 25 kombinasi aturan, masing-masing dengan dua konsekuen. Galat arah positif memiliki kecenderungan koreksi kiri, sedangkan galat lateral positif memiliki kecenderungan koreksi kanan. Ketika kedua masukan muncul bersamaan, keluaran ditentukan oleh aktivasi aturan dan pembobotan konsekuennya. Simbol (Δu) digunakan untuk menjelaskan setengah selisih PWM kanan terhadap kiri.

d. Logika FSM dan Pemnilihan Mode Operasi

FSM pada ESP32 mengatur pemilihan perintah gerak berdasarkan informasi objek, zona bahaya, lintasan aktif, serta estimasi posisi dan orientasi robot. Lima status operasi digunakan untuk mengoordinasikan pelacakan, perlambatan, penghindaran, perpindahan lajur, dan penghentian.

Status	Kondisi pemicu	Pelaksanaan kendali	Kondisi penyelesaian
STATE_PATROL	Lintasan aktif tersedia dan jalur dapat dilalui	FIS 2 menghasilkan PWM kiri–kanan untuk pelacakan; $(k_v = 1)$ ketika tidak diperlukan perlambatan	Halangan memerlukan respons atau ujung lajur tercapai
STATE_SLOWDOWN	Objek hidup terdeteksi pada zona perlambatan	FIS 1 menghasilkan pengali PWM; FIS 2 tetap mengoreksi lintasan aktif	Kondisi perlambatan selesai atau penghentian darurat diperlukan
STATE_SMART_DODGE	Halangan statis menghalangi lintasan	FSM menjalankan urutan deviasi, melewati halangan,	Halangan telah dilewati dan posisi serta arah robot

	dan manuver penghindaran tersedia	dan kembali ke lajur dengan umpan balik encoder–IMU	selaras dengan lintasan
STATE_U_TURN	Ujung lajur aktif tercapai dan lajur berikutnya tersedia	FSM menjalankan urutan putar dan perpindahan lajur; FIS 2 digunakan pada tahap penyelarasan yang sesuai	Robot memasuki lajur berikutnya dalam toleransi arah dan lateral
STATE_EMERGENCY_STOP	Objek hidup memasuki zona kritis atau terdapat permintaan penghentian keselamatan	Keluaran kedua FIS dikesampingkan; PWM kedua penggerak ditetapkan nol	Kondisi bahaya selesai dan izin melanjutkan operasi terpenuhi

Tabel 3.23 Logika FSM subsistem kendali gerak

1. Pelacakan dan perlambatan

   Pada STATE_PATROL, ESP32 menggunakan pasangan PWM dari FIS 2 untuk mengikuti lintasan. Ketika objek hidup memasuki zona perlambatan, FSM mengaktifkan STATE_SLOWDOWN. FIS 1 menghasilkan pengali kecepatan yang diterapkan pada kedua perintah motor. Kelas objek dan zona menentukan status operasi, sedangkan koordinat piksel menjadi masukan inferensi perlambatan.
2. Penghindaran halangan statis

   Pada STATE_SMART_DODGE, robot menjalankan tahapan perubahan arah, gerak melewati halangan, dan penyelarasan kembali. ESP32 memantau perubahan orientasi melalui MPU-9250 dan perpindahan melalui encoder. Waktu digunakan sebagai batas pemantauan pelaksanaan tahap; pencapaian target gerak dinilai menggunakan umpan balik sensor. Robot kembali ke pelacakan setelah jalur tersedia dan manuver selesai.
3. Putar balik dan perpindahan lajur

   Pada STATE_U_TURN, robot berpindah dari ujung lajur aktif menuju lajur berikutnya yang diberikan oleh perencanaan lintasan. Manuver mencakup perubahan arah dan perpindahan lateral sesuai jarak antarlajur. Putaran di tempat sebesar 180° hanya mengubah arah hadap, sehingga harus disertai gerak perpindahan agar robot tidak kembali memotong lajur yang sama.

   Perintah maju dan mundur pada tahap putar diatur oleh FSM melalui arah aktuasi BTS7960. Dua keluaran FIS 2 yang bernilai nonnegatif digunakan untuk pelacakan dan penyelarasan; keluaran tersebut tidak langsung dianggap sebagai perintah pivot dengan satu motor mundur. Setelah posisi dan orientasi memenuhi toleransi lajur berikutnya, FSM kembali ke STATE_PATROL. Apabila tidak tersedia lajur berikutnya, robot menghentikan gerak dan melaporkan penyelesaian misi.
4. Penghentian darurat

STATE_EMERGENCY_STOP memiliki prioritas tertinggi dan dapat menginterupsi pelacakan, perlambatan, penghindaran, maupun putar balik. ESP32 menetapkan perintah PWM kedua penggerak nol dan menahan gerak sampai persyaratan melanjutkan operasi terpenuhi. Waktu serta jarak berhenti fisik tetap dinilai melalui pengujian robot.

Program FSM menggunakan perulangan *non-blocking* agar pembacaan encoder, MPU-9250, komunikasi Orange Pi, dan pemeriksaan kondisi penghentian tetap berlangsung selama setiap manuver.

3.4.4.4 Basis Aturan Fuzzy *(Rule Base)*

Basis aturan fuzzy menghubungkan kondisi masukan dengan konsekuen keluaran menggunakan pernyataan IF–THEN. Jumlah kombinasi aturan lengkap ditentukan menggunakan aturan (k^n), dengan (k) sebagai jumlah himpunan keanggotaan pada setiap masukan dan (n) sebagai jumlah variabel masukan, apabila setiap masukan memiliki jumlah himpunan yang sama. FIS 1 memiliki satu masukan dengan lima himpunan keanggotaan sehingga menghasilkan $5^1 = 5$ aturan. FIS 2 memiliki dua masukan, masing-masing dengan lima himpunan keanggotaan, sehingga menghasilkan $5^2 = 25$ aturan. Setiap aturan FIS 1 menghasilkan konsekuen Pengali_Kecepatan, sedangkan setiap aturan FIS 2 menghasilkan pasangan konsekuen PWM_Kiri dan PWM_Kanan. Konsekuen berupa konstanta Sugeno orde nol, sementara keluaran akhir diperoleh melalui pembobotan aturan yang aktif.

a. Basis Aturan FIS 1

FIS 1 menerima Jarak_Piksel dan menghasilkan Pengali_Kecepatan (k_v). Semakin dekat posisi objek hidup pada citra, semakin kecil faktor pengali yang diterapkan pada perintah kedua motor.

1	If Jarak_Piksel is Aman, then Pengali_Kecepatan is Penuh.
2	If Jarak_Piksel is Waspada, then Pengali_Kecepatan is Agak_Pelan.
3	If Jarak_Piksel is Dekat, then Pengali_Kecepatan is Setengah.
4	If Jarak_Piksel is Kritis, then Pengali_Kecepatan is Sangat_Pelan.
5	If Jarak_Piksel is Stop, then Pengali_Kecepatan is Nol.

Tabel 3.24 Basis aturan FIS 1 untuk perlambatan

Untuk FIS 1, bobot aturan berasal dari derajat keanggotaan masukan Jarak_Piksel. Keluaran faktor pengali kecepatan dihitung sebagai :

$$k_v = \frac{\sum_{i=1}^{5} w_i c_i}{\sum_{i=1}^{5} w_i}$$

Persamaan 3.6

(w_i) merupakan derajat aktivasi aturan ke$-(i)$, sedangkan (c_i) merupakan konstanta konsekuen pengali kecepatan, yaitu $(1), (0{,}75), (0{,}50), (0{,}25)$, atau (0).

Ketika masukan berada pada daerah tumpang tindih fungsi keanggotaan, beberapa aturan dapat aktif bersamaan sehingga keluaran (k_v) berubah secara bertahap. Konsekuen Nol menghasilkan keluaran nol apabila hanya aturan Stop yang aktif. Pada daerah tumpang tindih Kritis dan Stop, hasil inferensi masih dapat bernilai lebih besar dari nol; penghentian darurat tetap dilaksanakan melalui prioritas FSM.

b. Basis Aturan FIS 2

Setiap aturan FIS 2 menggabungkan Heading_Error dan CrossTrack_Error menggunakan operator AND, kemudian menghasilkan dua konsekuen, yaitu PWM_Kiri dan PWM_Kanan. Penyusunan aturan menggunakan konvensi berikut:

- Galat arah negatif memerlukan kecenderungan koreksi kanan; galat arah positif memerlukan kecenderungan koreksi kiri.
- Galat lateral negatif menunjukkan posisi robot di kanan lintasan sehingga memerlukan kecenderungan koreksi kiri.
- Galat lateral positif menunjukkan posisi robot di kiri lintasan sehingga memerlukan kecenderungan koreksi kanan.

Ketika kebutuhan koreksi kedua galat searah, selisih PWM diperbesar. Ketika kebutuhan koreksinya berlawanan, aturan memberikan koreksi lebih kecil atau pasangan PWM seimbang sebagai respons awal yang akan diperiksa melalui simulasi.

1	If Heading_Error is NB and CrossTrack_Error is KiJ then PWM_Kiri is Sangat_Cepat, PWM_Kanan is Sangat_Lambat
2	If Heading_Error is NB and CrossTrack_Error is KiD then PWM_Kiri is Sangat_Cepat, PWM_Kanan is Sangat_Lambat
3	If Heading_Error is NB and CrossTrack_Error is Z then PWM_Kiri is Cepat, PWM_Kanan is Lambat

4	If Heading_Error is NB and CrossTrack_Error is KaD then PWM_Kiri is Cepat, PWM_Kanan is Lambat
5	If Heading_Error is NB and CrossTrack_Error is KaJ then PWM_Kiri is Normal, PWM_Kanan is Normal
6	If Heading_Error is NK and CrossTrack_Error is KiJ then PWM_Kiri is Sangat_Cepat, PWM_Kanan is Sangat_Lambat
7	If Heading_Error is NK and CrossTrack_Error is KiD then PWM_Kiri is Cepat, PWM_Kanan is Lambat
8	If Heading_Error is NK and CrossTrack_Error is Z then PWM_Kiri is Cepat, PWM_Kanan is Lambat
9	If Heading_Error is NK and CrossTrack_Error is KaD then PWM_Kiri is Normal, PWM_Kanan is Normal
10	If Heading_Error is NK and CrossTrack_Error is KaJ then PWM_Kiri is Lambat, PWM_Kanan is Cepat
11	If Heading_Error is Z and CrossTrack_Error is KiJ then PWM_Kiri is Sangat_Cepat, PWM_Kanan is Sangat_Lambat
12	If Heading_Error is Z and CrossTrack_Error is KiD then PWM_Kiri is Cepat, PWM_Kanan is Lambat
13	If Heading_Error is Z and CrossTrack_Error is Z then PWM_Kiri is Normal, PWM_Kanan is Normal
14	If Heading_Error is Z and CrossTrack_Error is KaD then PWM_Kiri is Lambat, PWM_Kanan is Cepat
15	If Heading_Error is Z and CrossTrack_Error is KaJ then PWM_Kiri is Sangat_Lambat, PWM_Kanan is Sangat_Cepat
16	If Heading_Error is PK and CrossTrack_Error is KiJ then PWM_Kiri is Cepat, PWM_Kanan is Lambat
17	If Heading_Error is PK and CrossTrack_Error is KiD then PWM_Kiri is Normal, PWM_Kanan is Normal
18	If Heading_Error is PK and CrossTrack_Error is Z then PWM_Kiri is Lambat, PWM_Kanan is Cepat
19	If Heading_Error is PK and CrossTrack_Error is KaD then PWM_Kiri is Lambat, PWM_Kanan is Cepat
20	If Heading_Error is PK and CrossTrack_Error is KaJ then PWM_Kiri is Sangat_Lambat, PWM_Kanan is Sangat_Cepat
21	If Heading_Error is PB and CrossTrack_Error is KiJ then PWM_Kiri is Normal, PWM_Kanan is Normal
22	If Heading_Error is PB and CrossTrack_Error is KiD then PWM_Kiri is Lambat, PWM_Kanan is Cepat

23	If Heading_Error is PB and CrossTrack_Error is Z then PWM_Kiri is Lambat, PWM_Kanan is Cepat
24	If Heading_Error is PB and CrossTrack_Error is KaD then PWM_Kiri is Sangat_Lambat, PWM_Kanan is Sangat_Cepat
25	If Heading_Error is PB and CrossTrack_Error is KaJ then PWM_Kiri is Sangat_Lambat, PWM_Kanan is Sangat_Cepat

Tabel 3.25 Basis aturan FIS 2 untuk pelacakan lintasan

Untuk FIS 2, setiap aturan menggabungkan kondisi Heading_Error dan CrossTrack_Error menggunakan operator AND minimum. Bobot aturan dihitung sebagai:

$$\boldsymbol{w_i = \min\left[\mu_{A_i}(e_\theta), \mu_{B_i}(e_{ct})\right]}$$

Persamaan 3.7

(μ_{A_i}) dan (μ_{B_i}) merupakan derajat keanggotaan kedua masukan pada aturan ke-(i). Dengan bobot aturan yang sama, keluaran PWM kiri dan kanan dihitung secara terpisah:

$$\boldsymbol{u_L = \frac{\sum_{i=1}^{25} w_i c_{L,i}}{\sum_{i=1}^{25} w_i}, \quad u_R = \frac{\sum_{i=1}^{25} w_i c_{R,i}}{\sum_{i=1}^{25} w_i}}$$

Persamaan 3.8

$c_{L,i}$ dan $c_{R,i}$ merupakan konstanta konsekuen PWM kiri dan kanan pada aturan ke$-(i)$, yaitu $(50), (120), (180), (220), atau\ (255)$.

Sebagai contoh, apabila dua aturan aktif memiliki bobot 0,6 dan 0,4, dengan pasangan konsekuen (180,180) dan (220,120), keluaran FIS 2 adalah:

$$\boldsymbol{u_L = \frac{0.6(180) + 0.4(220)}{0.6 + 0.4} = 196, \quad u_R = \frac{0.6(180) + 0.4(120)}{0.6 + 0.4} = 156}$$

c. Penerapan Aturan pada Sistemn Kendali

Pada MATLAB, aturan FIS 2 menggunakan bobot awal 1 dan operator AND berupa minimum. Hasil inferensi dihitung secara terpisah untuk keluaran kiri dan kanan menggunakan rata-rata berbobot. Parameter serta aturan yang sama diterapkan pada program C++ ESP32. Pada mode perlambatan, faktor keluaran FIS 1 diterapkan pada pasangan perintah motor. Pada mode penghindaran dan U_TURN, FSM menentukan penggunaan perintah manuver atau keluaran

pelacakan. Ketika STATE_EMERGENCY_STOP aktif, perintah PWM kedua motor ditetapkan nol dengan mengesampingkan keluaran fuzzy. Evaluasi aturan mencakup kesesuaian arah koreksi, kesinambungan perubahan PWM, dan kemampuan robot mengurangi galat lintasan. Data simulasi serta pengujian encoder–IMU digunakan untuk menilai kebutuhan penyetelan konsekuen sebelum pengujian terintegrasi.

3.4.4.5 Simulasi dan Implementasi

Simulasi dilakukan menggunakan MATLAB dengan Fuzzy Logic Toolbox untuk memeriksa fungsi keanggotaan, aktivasi aturan, dan keluaran kedua pengendali Sugeno sebelum diterapkan pada robot. FIS 1 menghasilkan Pengali_Kecepatan, sedangkan FIS 2 menghasilkan PWM_Kiri dan PWM_Kanan. Setelah model diperiksa, perhitungan fuzzy diimplementasikan menggunakan bahasa C++ pada ESP32 dengan parameter yang sama.

Model FIS 1 disusun dengan satu masukan Jarak_Piksel, lima fungsi keanggotaan, dan lima aturan. Model FIS 2 disusun dengan dua masukan, yaitu Heading_Error dan CrossTrack_Error, masing-masing memiliki lima fungsi keanggotaan sehingga membentuk 25 aturan. Kedua model menggunakan konsekuen konstan dan metode rata-rata berbobot. Pemeriksaan menggunakan *Rule Viewer* untuk mengamati aturan yang aktif pada masukan tertentu. Grafik keluaran FIS 1 digunakan untuk memeriksa perubahan pengali kecepatan terhadap posisi objek pada citra. Pada FIS 2, *Surface Viewer* digunakan untuk menampilkan permukaan keluaran PWM kiri dan kanan terhadap kombinasi galat arah dan galat lateral.

Model/kondisi	Masukan yang diperiksa	Respons yang diperiksa
FIS 1—seluruh domain	Jarak_Piksel pada rentang 450–640 piksel	Pengali berada pada rentang 0–1 dan menurun ketika objek semakin mendekati batas bawah citra
FIS 1—batas dan tumpang tindih	Titik batas serta daerah perpotongan fungsi keanggotaan	Aturan aktif menghasilkan transisi keluaran yang sesuai tanpa pembagian dengan jumlah bobot nol
FIS 2—galat nol	$(e_\theta = 0°);\ (e_{ct} = 0)m$	Keluaran kiri dan kanan sama, yaitu PWM 180
FIS 2—galat arah	$(e_\theta)\ divariasikan$; $(e_{ct} = 0)$ m	Galat arah positif menghasilkan koreksi ke kiri; galat negatif menghasilkan koreksi ke kanan
FIS 2—galat lateral	$(e_\theta = 0°);\ (e_{ct})\ divariasikan$	Galat lateral positif menghasilkan koreksi ke kanan; galat negatif menghasilkan koreksi ke kiri
FIS 2—kombinasi galat	(e_θ) pada −30° hingga 30° dan (e_{ct}) pada −0,5 hingga 0,5 m	Respons kedua PWM sesuai kombinasi aturan, termasuk ketika kebutuhan koreksi kedua masukan berlawanan
Integrasi FIS dan FSM	Pengali kecepatan, keluaran PWM, dan perubahan status operasi	Perlambatan mengalikan kedua PWM; perintah berhenti darurat mengesampingkan keluaran fuzzy

Tabel 3.26 Rencana pemeriksaan model fuzzy pada MATLAB

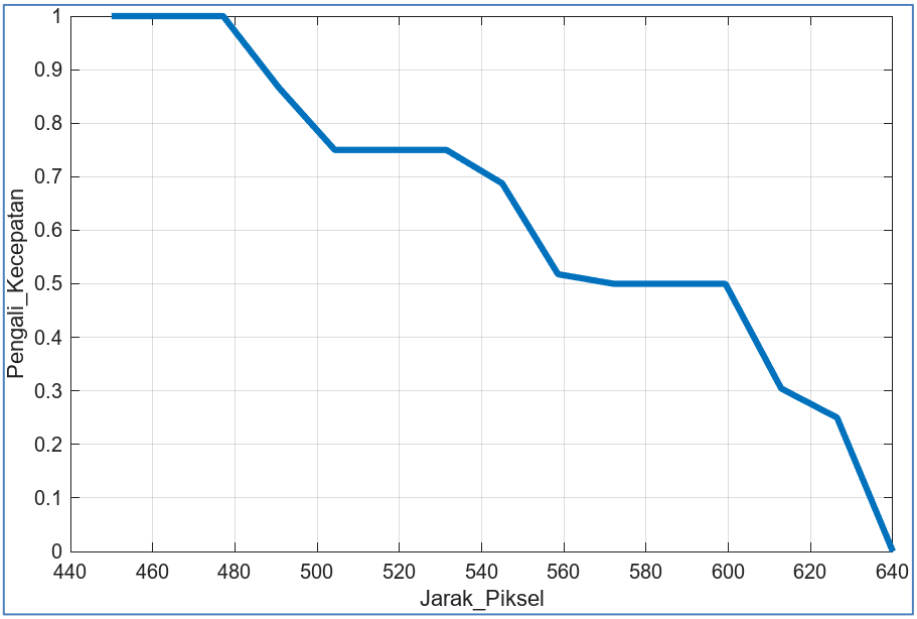
a. Simulasi FIS 1 : Seluruh Domain

Input Jarak_Piksel (piksel)	Output Pengali_Kecepatan	Kondisi
450	1	Kecepatan penuh, perintah PWM dipertahankan 100%
470	1	Kecepatan penuh, perintah PWM dipertahankan 100%
490	0,875	Perlambatan awal, perintah PWM menjadi 87,5%
510	0,75	Perintah PWM menjadi 75%
530	0,75	Perintah PWM menjadi 75%

550	0,625	Perintah PWM menjadi 62,5%
570	0,5	Perintah PWM menjadi 50%
590	0,5	Perintah PWM menjadi 50%
610	0,333	Perintah PWM menjadi sekitar 33,3%
630	0,25	Perintah PWM menjadi 25%
640	0	Perintah PWM kedua motor menjadi nol

Tabel 3.27 Hasil simulasi FIS 1 pada seluruh domain masukan

Hasil simulasi menunjukkan bahwa pengali kecepatan menurun dari 1 menjadi 0 seiring peningkatan koordinat Jarak_Piksel dari 450 hingga 640 piksel. Pada titik yang diuji, tidak terdapat kenaikan keluaran ketika nilai masukan meningkat. Nilai antara seperti 0,875; 0,625; dan 0,333 menunjukkan pembobotan konsekuen dari aturan yang aktif pada daerah tumpang tindih fungsi keanggotaan. Persentase tersebut menyatakan proporsi perintah PWM, sedangkan perubahan kecepatan fisik robot perlu dibuktikan melalui pengukuran encoder.



Gambar 3.17 Control surface simulasi FIS 1 : Perlambatan

Grafik *control surface* FIS 1 menunjukkan hubungan antara masukan Jarak_Piksel dan keluaran Pengali_Kecepatan. Karena model memiliki satu masukan dan satu keluaran, respons ditampilkan sebagai kurva dua dimensi. Pengali menurun dari 1 pada 450 piksel menjadi 0 pada 640 piksel, tanpa peningkatan keluaran sepanjang kurva yang ditampilkan. Pola tersebut sesuai dengan tujuan perlambatan ketika posisi bawah *bounding box* objek semakin mendekati batas bawah citra.

Bagian mendatar pada pengali 1; 0,75; dan 0,5 menunjukkan daerah dengan konsekuen keluaran yang tetap. Bagian miring menunjukkan transisi akibat perubahan bobot aturan pada daerah tumpang tindih fungsi keanggotaan. Mendekati 640 piksel, pengali turun menuju nol sehingga perintah PWM kedua motor dikurangi hingga nol. Grafik ini menunjukkan karakteristik perlambatan

pada model fuzzy; perubahan kecepatan dan jarak berhenti robot tetap perlu diperiksa melalui pengujian fisik.

b. Simulasi FIS 2 : Galat nol

Input Heading_Error (e_θ)	Input CrossTrack_Error (e_{ct})	Output PWM [Kiri Kanan]	Kondisi
0	0	[180 180]	Robot berada pada garis lintasan acuan dengan arah yang sesuai; perintah bergerak lurus tanpa koreksi diferensial

Tabel 3.28 Hasil simulasi FIS 2 dengan kondisi galat nol

Pada masukan galat arah 0° dan galat lateral 0 m, kedua himpunan ZERO memiliki derajat keanggotaan penuh sehingga aturan ZERO–ZERO menghasilkan PWM kiri dan kanan sebesar 180. Kesamaan keluaran menunjukkan bahwa FIS 2 tidak memberikan koreksi belok karena posisi dan orientasi estimasi robot telah sesuai dengan lintasan acuan. Nilai tersebut setara dengan duty cycle sekitar 70,6% pada PWM 8-bit, sebelum dikalikan keluaran FIS 1. Gerak lurus secara fisik tetap perlu diverifikasi menggunakan encoder karena PWM yang sama belum menjamin kecepatan kedua track sama.

c. Simulasi FIS 2 : Galat Arah

Input Heading_Error (e_θ)	Input CrossTrack_Error (e_{ct})	Output PWM [Kiri Kanan]	Kondisi
-30	0	[220 120]	Koreksi ke kanan
-24	0	[220 120]	Koreksi ke kanan
-18	0	[220 120]	Koreksi ke kanan
-12	0	[220 120]	Koreksi ke kanan
-6	0	[204 144]	Koreksi ke kanan dengan selisih PWM lebih kecil
6	0	[144 204]	Koreksi ke kiri dengan selisih PWM lebih kecil
12	0	[120 220]	Koreksi ke kiri
18	0	[120 220]	Koreksi ke kiri
24	0	[120 220]	Koreksi ke kiri
30	0	[120 220]	Koreksi ke kiri

Tabel 3.29 Hasil simulasi FIS 2 dengan variasi galat arah dan galat lateral nol

Hasil simulasi menunjukkan bahwa galat arah negatif menghasilkan PWM kiri lebih besar daripada kanan, sehingga robot diperintahkan berbelok ke kanan. Sebaliknya, galat arah positif menghasilkan PWM kanan lebih besar daripada kiri, sehingga robot diperintahkan berbelok ke kiri. Respons ini sesuai dengan konvensi ($e_\theta = \theta_{ref} - \hat{\theta}$), dengan arah putaran positif berlawanan jarum jam.

Pada galat ±6°, selisih PWM sebesar 60, sedangkan pada titik uji ±12° hingga ±30°, selisihnya sebesar 100. Hal ini menunjukkan koreksi yang lebih kecil mendekati galat nol serta keluaran tetap pada galat yang lebih besar karena

konsekuen aturan yang aktif sama. Pasangan keluaran untuk galat positif dan negatif saling bertukar, menunjukkan respons simetris pada titik yang diuji. Simulasi ini membuktikan kesesuaian arah perintah kendali; kemampuan mengurangi galat dan kestabilan gerak tetap perlu dinilai melalui simulasi lintasan serta pengujian robot.

d. Simulasi FIS 2 : Galat Lateral

Input Heading_Error (e_θ)	Input CrossTrack_Error (e_{ct})	Output PWM [Kiri Kanan]	Kondisi
0	-0.5	[50 255]	Robot di kanan lintasan; koreksi ke kiri
0	-0.4	[50 255]	Robot di kanan lintasan; koreksi ke kiri
0	-0.3	[88.2 236]	Robot di kanan lintasan; koreksi ke kiri
0	-0.2	[120 220]	Robot di kanan lintasan; koreksi ke kiri
0	-0.1	[140 207]	Robot di kanan lintasan; koreksi ke kiri dengan selisih PWM lebih kecil
0	0.1	[207 140]	Robot di kiri lintasan; koreksi ke kanan dengan selisih PWM lebih kecil
0	0.2	[220 120]	Robot di kiri lintasan; koreksi ke kanan
0	0.3	[236 88.2]	Robot di kiri lintasan; koreksi ke kanan
0	0.4	[255 50]	Robot di kiri lintasan; koreksi ke kanan
0	0.5	[255 50]	Robot di kiri lintasan; koreksi ke kanan

Tabel 3.30 Hasil simulasi FIS 2 dengan galat arah nol dan variasi galat lateral

Hasil simulasi menunjukkan arah koreksi yang sesuai dengan konvensi galat lateral. Galat negatif menghasilkan PWM kanan lebih besar sehingga robot diperintahkan berbelok ke kiri, sedangkan galat positif menghasilkan PWM kiri lebih besar untuk berbelok ke kanan. Kedua respons mengarahkan robot menuju lintasan acuan ketika galat arah nol.

Besar selisih PWM meningkat dari 67 pada penyimpangan 0,1 m, menjadi 100 pada 0,2 m, 147,8 pada 0,3 m, dan 205 pada 0,4–0,5 m. Respons juga simetris: perubahan tanda galat menukar pasangan keluaran kiri–kanan. Nilai pecahan merupakan hasil rata-rata berbobot Sugeno dan dibulatkan saat diterapkan pada ESP32. Hasil ini memverifikasi arah serta besar perintah koreksi pada titik yang diuji; keberhasilan mengurangi galat selama robot bergerak masih perlu diperiksa melalui simulasi lintasan dan pengujian fisik.

e. Simulasi FIS 2 : Kombinasi Galat

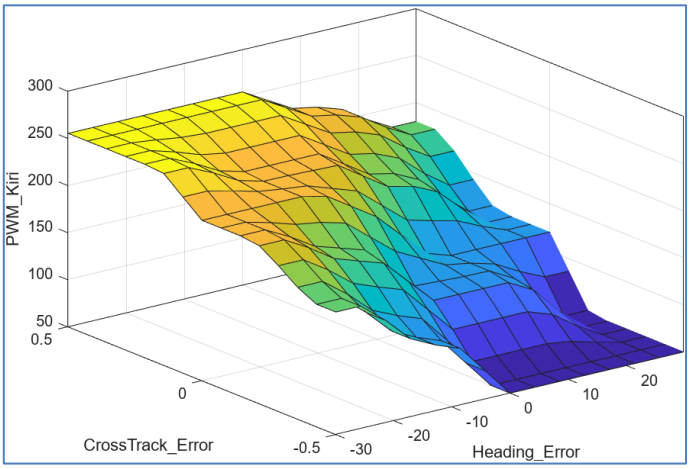
No.	$(e_\theta)(°)$	$(e_{ct})(m)$	PWM kiri	PWM kanan	Kondisi keluaran
1	0	0	180	180	Tanpa koreksi diferensial
2	−30	0,5	255	50	Koreksi kanan; kedua kebutuhan koreksi searah
3	30	−0,5	50	255	Koreksi kiri; kedua kebutuhan koreksi searah
4	−18	0,3	242	76,7	Koreksi kanan; kedua kebutuhan koreksi searah
5	18	−0,3	76,7	242	Koreksi kiri; kedua kebutuhan koreksi searah

6	−6	0,1	215	128	Koreksi kanan; kedua kebutuhan koreksi searah
7	6	−0,1	128	215	Koreksi kiri; kedua kebutuhan koreksi searah
8	−30	−0,5	180	180	Kebutuhan koreksi berlawanan; keluaran seimbang
9	30	0,5	180	180	Kebutuhan koreksi berlawanan; keluaran seimbang
10	−18	−0,3	167	183	Koreksi kiri kecil; kebutuhan koreksi berlawanan
11	18	0,3	183	167	Koreksi kanan kecil; kebutuhan koreksi berlawanan
12	−6	−0,1	172	179	Koreksi kiri kecil; kebutuhan koreksi berlawanan
13	6	0,1	179	172	Koreksi kanan kecil; kebutuhan koreksi berlawanan
14	6	0,4	234	92	Koreksi kanan; respons lateral lebih dominan
15	−6	−0,4	92	234	Koreksi kiri; respons lateral lebih dominan

Tabel 3.31 Hasil simulasi FIS 2 pada kombinasi galat arah dan galat lateral

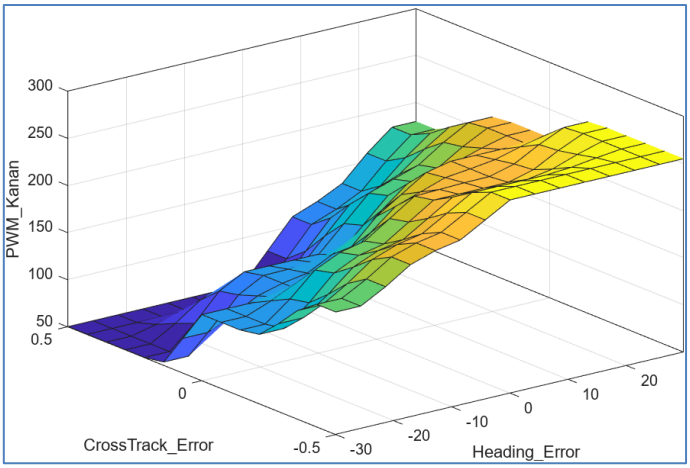
Hasil simulasi menunjukkan bahwa ketika galat arah dan lateral membutuhkan koreksi searah, FIS 2 menghasilkan selisih PWM yang memperkuat arah koreksi tersebut. Pada kombinasi $(-30°;\ 0{,}5\ \text{m})$, keluaran [255, 50] memberikan perintah belok kanan, sedangkan kombinasi dengan tanda berlawanan menghasilkan [50, 255] untuk belok kiri. Selisih PWM pada kelompok ini meningkat dari 87 hingga 205 seiring perubahan kombinasi masukan yang diuji. Ketika kebutuhan koreksi berlawanan, keluaran ditentukan oleh pembobotan aturan aktif. Kombinasi $(-18°;\ -0{,}3\ \text{m})$ menghasilkan [167, 183], sedangkan $(-6°;\ -0{,}1\ \text{m})$ menghasilkan [172, 179], yang keduanya memberikan koreksi kiri kecil. Pada galat arah kecil dan penyimpangan lateral besar, yaitu $(6°;\ 0{,}4\ \text{m})$, keluaran [234, 92] menunjukkan koreksi kanan sehingga respons lateral lebih dominan. Seluruh pasangan masukan yang tandanya dibalik menghasilkan pertukaran PWM kiri–kanan, menunjukkan simetri respons pada titik pengujian.

Keluaran [180, 180] pada kombinasi $(-30°;\ -0{,}5\ \text{m})$ dan $(30°;\ 0{,}5\ \text{m})$ menunjukkan keseimbangan konsekuen untuk kebutuhan koreksi yang berlawanan, bukan berarti kedua galat telah nol. Robot masih diperintahkan maju tanpa koreksi diferensial pada saat tersebut. Kemampuan respons ini mengurangi galat perlu diperiksa melalui simulasi gerak terhadap waktu; hasil inferensi statis belum membuktikan kestabilan atau keberhasilan robot kembali ke lintasan.



Gambar 3.18 Control surface PWM_Kiri

Permukaan keluaran PWM kiri menunjukkan respons terhadap kombinasi galat arah pada rentang −30° hingga 30° dan galat lateral pada rentang −0,5 hingga 0,5 m. PWM kiri meningkat pada kombinasi yang membutuhkan koreksi ke kanan, terutama ketika galat arah negatif dan galat lateral positif. Hasil simulasi pada masukan −30° ; 0,5 m menghasilkan PWM kiri 255. Sebaliknya, pada kombinasi 30° ; −0,5 ,PWM kiri turun menjadi 50 untuk mendukung koreksi ke kiri. Bagian mendatar menunjukkan daerah keluaran yang didominasi konsekuen konstan, sedangkan bagian miring menunjukkan transisi akibat pembobotan beberapa aturan aktif.



Gambar 3.19 Control surface PWM_Kanan

Permukaan PWM kanan memperlihatkan pola yang berlawanan dengan PWM kiri. Nilainya meningkat ketika kombinasi masukan membutuhkan koreksi ke kiri, yaitu galat arah positif dan galat lateral negatif. Pada masukan (30°; , −0,5 m), PWM kanan mencapai 255, sedangkan pada (−30°; ,0,5 m) turun menjadi 50. Perbedaan kedua keluaran membentuk perintah belok melalui pengaturan penggerak kanan relatif terhadap penggerak kiri. Transisi pada permukaan menunjukkan bahwa keluaran Sugeno dapat mengambil nilai antara konstanta konsekuen, sesuai derajat aktivasi aturan.

Kedua permukaan menunjukkan pola koreksi yang saling berlawanan dan sesuai dengan hasil pengujian kombinasi masukan. Pada galat nol, pasangan keluaran [180, 180] menunjukkan perintah maju tanpa koreksi diferensial. Namun, keluaran seimbang juga muncul pada kombinasi galat yang kebutuhan koreksinya berlawanan, seperti (−30°; , −0,5 m). Kondisi tersebut merupakan hasil keseimbangan aturan, bukan bukti bahwa robot telah berada pada lintasan. Secara keseluruhan, grafik menunjukkan karakteristik pemetaan masukan terhadap PWM; kestabilan, kemungkinan osilasi, dan kemampuan kembali ke lintasan harus dinilai melalui simulasi gerak terhadap waktu serta pengujian fisik.

3.5 Teknik Uji Kelayakan Hasil

Uji kelayakan hasil dilakukan secara bertahap untuk menilai kinerja subsistem kendali gerak, meliputi pembacaan dua rotary encoder 1000 PPR dan IMU MPU-9250, karakterisasi hubungan PWM terhadap kecepatan dua motor wiper MK461340 melalui driver BTS7960, ketelitian estimasi odometri–IMU, serta respons pengendali fuzzy Sugeno. Evaluasi dilakukan melalui simulasi MATLAB, pemeriksaan kesesuaian keluaran program C++ pada ESP32, dan pengujian gerak robot pada petak seluas 100 m² di Lapangan Woodball Universitas Negeri Semarang. Pengujian dilaksanakan berulang dengan mencatat kondisi operasi dan membandingkan hitungan encoder, kecepatan penggerak, orientasi, posisi, keluaran PWM kiri–kanan, serta galat arah dan lateral terhadap pengukuran acuan dan indikator kuantitatif yang ditetapkan. FIS 1 dan FIS 2 diuji sesuai fungsi masing-masing, sedangkan pengujian integrasi dengan FSM menunggu penetapan logika transisi dan parameter operasinya.

3.5.1 Pengujian Sensor dan Karateristik Sistem Penggerak

Pengujian dilakukan untuk memeriksa ketelitian pembacaan dua rotary encoder 1000 PPR dan MPU-9250, serta memperoleh karakteristik hubungan PWM terhadap kecepatan motor wiper MK461340 kiri dan kanan. ESP32 menjalankan program pengujian C++, membaca sensor, dan mengirimkan data melalui komunikasi serial ke laptop untuk diolah menggunakan MATLAB. Pengujian awal dilakukan pada dudukan pengujian, kemudian dilanjutkan pada robot dengan beban operasional. Tegangan suplai, konfigurasi pembacaan encoder, interval pencatatan, dan kondisi beban dicatat agar hasil antarpercobaan dapat dibandingkan.

3.5.1.1 Pengujian Rotary Encoder

Pengujian dilakukan dengan membandingkan hitungan ESP32 terhadap putaran poros yang diberi penanda, sebanyak 1, 5, dan 10 putaran pada kedua arah, masing-masing diulang lima kali untuk setiap encoder. Nilai nominal 1000 PPR dibedakan dari hitungan efektif per putaran sesuai pembacaan kuadratur ×1, ×2, atau ×4; rasio transmisi diperhitungkan apabila poros encoder berbeda dari poros sprocket. Selanjutnya, kecepatan hasil encoder dibandingkan dengan takometer optik atau alat ukur putaran independen untuk mengevaluasi galat pembacaan saat motor beroperasi.

Parameter	Variasi pengujian	Acuan pembanding	Data yang dicatat
Hitungan efektif per putaran	1, 5, dan 10 putaran; lima pengulangan	Putaran poros berdasarkan penanda	Hitungan acuan, hitungan ESP32, dan selisih hitungan
Identifikasi arah	Putaran maju dan mundur	Arah putaran poros yang diamati	Tanda hitungan dan kesesuaian arah
Pembacaan saat motor berputar	Beberapa tingkat PWM dalam rentang operasi	Alat ukur putaran independen	Kecepatan acuan, kecepatan encoder, dan galat
Konsistensi pembacaan	Pengulangan pada kondisi yang sama	Hasil antarpercobaan	Rata-rata dan simpangan baku

Tabel 3.32 Rencana pengujian rotary encoder

3.5.1.2 Pengujian MPU-9250

Pengujian MPU-9250 diawali dengan pencatatan saat diam untuk mengestimasi bias giroskop, kemudian dilanjutkan dengan pengukuran drift setelah koreksi bias dan perubahan orientasi terhadap sudut acuan. Sumbu sensor diselaraskan dengan sumbu pengujian, sedangkan galat sudut dinormalisasi untuk menghindari selisih semu di sekitar batas ±180°. Pembacaan akselerometer diperiksa pada kemiringan terukur, sementara pengaruh motor terhadap data IMU dievaluasi pada kondisi motor mati dan menyala; pemeriksaan magnetometer digunakan untuk menilai gangguan lingkungan pemasangan.

Jenis pengujian	Kondisi yang direncanakan	Acuan pembanding	Parameter evaluasi
Bias giroskop	Sensor diam selama 60 detik	Kecepatan sudut nol	Rata-rata pembacaan dan simpangan baku
Drift setelah koreksi bias	Sensor diam selama 60 detik setelah kalibrasi	Orientasi tetap	Perubahan estimasi sudut terhadap waktu
Perubahan arah	Putaran ±30°, ±90°, dan ±180°; lima pengulangan	Meja putar atau penanda sudut terukur	Galat sudut akhir dan kesesuaian arah
Kemiringan badan	Posisi datar dan kemiringan ±5° serta ±10°	Inklinometer atau alat ukur sudut independen	Galat perkiraan *roll* dan *pitch*
Pengaruh motor	Motor mati dan menyala pada beberapa tingkat PWM	Posisi dan orientasi sensor dipertahankan	Perubahan derau dan pembacaan medan magnet

Tabel 3.33 Rencana pengujian MPU-9250

3.5.1.3 Karateristik Motor Wiper dan Driver BTS7960

Karakterisasi dilakukan dengan memberikan PWM sebesar 0, 50, 120, 180, 220, dan 255 pada arah maju dan mundur, masing-masing diulang lima kali. Motor kiri dan kanan diuji secara terpisah, kemudian bersamaan, pada kondisi penggerak tanpa kontak tanah dan robot dengan beban operasional. Kecepatan diukur setelah respons relatif stabil, disertai pencatatan tegangan dan arus motor. Data diolah menjadi grafik PWM–kecepatan untuk menentukan PWM minimum yang menghasilkan gerak, konsistensi respons, serta perbedaan karakteristik kiri–kanan sebagai dasar pemetaan penggerak pada simulasi dan implementasi kendali.

Sisi motor	Arah	PWM	Kondisi beban	Tegangan motor (V)	Arus motor (A)	Kecepatan acuan (rpm)	Kecepatan encoder (rpm)
Kiri/kanan	Maju/mundur	0–255	Tanpa kontak tanah/beban operasional	Hasil ukur	Hasil ukur	Hasil ukur	Hasil pembacaan

Tabel 3.34 Format pencatatan karakterisasi sistem penggerak

3.5.2 Pengujian Ketelitian Estimasi Odometri-IMU

Pengujian dilakukan untuk membandingkan ketelitian estimasi posisi dan orientasi dari odometri encoder saja dengan fusi encoder–IMU. ESP32 mencatat hitungan dua encoder 1000 PPR, kecepatan sudut terkoreksi MPU-9250, waktu pengambilan data, serta estimasi gerak untuk dianalisis menggunakan MATLAB. Kedua metode dihitung dari rekaman percobaan yang sama, dengan posisi awal, orientasi awal, dan parameter kinematika yang seragam. Pengukuran acuan

diperoleh secara independen menggunakan penanda posisi, meteran, dan penanda sudut terukur; GPS tetap digunakan sebagai referensi navigasi global.

3.5.2.1 Skenario dan Pelaksanaan Pengujian

Pengujian direncanakan pada permukaan keras yang relatif datar dan permukaan berumput di Lapangan Woodball UNNES, masing-masing sebanyak lima pengulangan per skenario. Posisi dan orientasi aktual diukur pada titik pemeriksaan setelah robot berhenti, sedangkan data estimasi dicatat secara kontinu. Bobot fusi ditentukan melalui data kalibrasi, kemudian dipertahankan selama pengujian evaluasi agar hasilnya tidak disesuaikan terhadap data yang sedang dinilai.

Skenario	Variasi yang direncanakan	Titik pemeriksaan	Tujuan evaluasi
Gerak lurus	Jarak 2 m dan 5 m	Posisi dan orientasi akhir	Galat perpindahan dan penyimpangan arah
Putaran di tempat	±90° dan ±180°	Pusat robot dan orientasi akhir	Galat perubahan arah dan pergeseran pusat
Lintasan berbentuk L	Dua segmen sepanjang 2 m dengan belokan ±90°	Ujung setiap segmen	Akumulasi galat setelah gerak dan belokan
Lintasan persegi	Sisi 2 m; arah searah dan berlawanan jarum jam	Setiap sudut dan titik akhir	Galat posisi serta orientasi setelah lintasan tertutup

Tabel 3.35 Skenario pengujian odometri-IMU

Parameter	Ketentuan
Metode pembanding	Odometri encoder saja dan fusi encoder–IMU
Posisi awal	Pusat robot ditetapkan sebagai ((0,0)) dalam koordinat lokal
Orientasi awal	Diselaraskan terhadap garis acuan
Permukaan	Permukaan keras relatif datar dan permukaan berumput
Pengulangan	Lima kali untuk setiap skenario dan permukaan
Parameter fusi	Bobot giroskop (\alpha) dicatat dan dipertahankan selama evaluasi
Data kontinu	Waktu, hitungan encoder kiri–kanan, kecepatan sudut, posisi, dan orientasi estimasi
Kondisi operasi	Tegangan baterai, beban robot, PWM, dan kondisi permukaan
Pengukuran acuan	Posisi pusat robot dan arah badan pada setiap titik pemeriksaan

Tabel 3.36 Kondisi dan parameter pencatatan pengujian

3.5.2.2 Parameter Evaluasi Ketelitian

Ketelitian dinilai dari jarak antara posisi estimasi dan posisi acuan, selisih orientasi yang dinormalisasi pada interval ±180°, serta konsistensi hasil antar pengulangan. Galat posisi dilaporkan dalam meter dan galat orientasi dalam derajat; keduanya dianalisis secara terpisah. Pengurangan galat fusi dihitung terhadap odometri pada percobaan yang sama, sedangkan ketelitian alat ukur acuan dicatat untuk menentukan apakah perbedaan hasil cukup berarti.

Indikator	Cara perhitungan	Satuan	Makna
Galat posisi	Jarak Euclidean antara posisi estimasi dan posisi acuan	m	Ketelitian estimasi posisi pada titik pemeriksaan
Galat orientasi	Nilai absolut selisih sudut yang dinormalisasi	°	Ketelitian estimasi arah badan robot
Rata-rata galat	Jumlah galat dibagi jumlah pengukuran	m atau °	Besar galat rata-rata per skenario
RMSE	Akar dari rata-rata kuadrat galat	m atau °	Ukuran galat dengan penekanan pada penyimpangan besar
Simpangan baku	Sebaran galat antar pengulangan	m atau °	Konsistensi hasil pengujian
Pengurangan RMSE	Selisih RMSE odometri dan fusi, dibagi RMSE odometri, dikalikan 100%	%	Perubahan ketelitian akibat fusi; dihitung jika RMSE odometri bukan nol

Tabel 3.37 Indikator evaluasi estimasi gerak local

Skenario/permukaan	Metode	Jumlah pengulangan	RMSE posisi (m)	RMSE orientasi (°)	Galat posisi maksimum (m)	Galat orientasi maksimum (°)
Diisi sesuai pengujian	Odometri encoder	5	Hasil uji	Hasil uji	Hasil uji	Hasil uji
Kondisi yang sama	Fusi encoder–IMU	5	Hasil uji	Hasil uji	Hasil uji	Hasil uji

Tabel 3.38 Format rekapitulasi hasil pengujian odometri-IMU

3.5.2.3 Penyajian dan Penilaian Hasil

Hasil disajikan melalui grafik posisi pada titik pemeriksaan dan perbandingan galat kedua metode untuk setiap skenario serta permukaan. Fusi dinilai memberikan perbaikan apabila galatnya lebih kecil dan hasilnya konsisten pada data evaluasi, dengan mempertimbangkan ketelitian pengukuran acuan. Perbaikan orientasi tidak langsung dianggap menghilangkan galat perpindahan akibat slip maju atau menyamping; batas penerimaan kuantitatif ditetapkan dalam matriks indikator capaian.

3.5.3 Pengujian Pengendali Fuzzy Sugeno dan Pelacakan Lintasan

3.5.3.1 Pengujian Respon FIS 1

FIS 1 diuji dengan memberikan masukan koordinat vertikal bawah *bounding box* objek yang relevan pada domain 450–640 piksel. Keluaran berupa pengali kecepatan (k_v) diperiksa terhadap hasil simulasi yang sudah diperoleh, kemudian diterapkan pada kedua keluaran PWM FIS 2. Pengujian ini menilai ketepatan perhitungan perlambatan; akurasi deteksi objek tetap menjadi pengujian subsistem persepsi. Pada pengujian kamera, resolusi citra, posisi pemasangan, dan pemilihan objek harus mengikuti konfigurasi yang digunakan untuk menetapkan domain piksel.

Masukan piksel	Pengali kecepatan (k_v)	Proporsi perintah PWM (%)	Kondisi keluaran
450	1	100	Perintah PWM dipertahankan
470	1	100	Perintah PWM dipertahankan
490	0,875	87,5	Perlambatan awal
510	0,750	75	Perlambatan
530	0,750	75	Perlambatan
550	0,625	62,5	Perlambatan meningkat
570	0,500	50	Setengah perintah PWM
590	0,500	50	Setengah perintah PWM
610	0,333	33,3	Perlambatan kuat
630	0,250	25	Seperempat perintah PWM
640	0	0	Perintah PWM menjadi nol

Tabel 3.39 Acuan pengujian keluaran FIS 1

3.5.5.2 Pengujian Respon FIS 2

FIS 2 diuji pada domain galat arah −30° hingga 30° dan galat lateral −0,5 hingga 0,5 m, dengan keluaran langsung PWM kiri dan kanan. Galat arah positif

memerlukan koreksi ke kiri, sedangkan galat lateral positif menunjukkan robot berada di kiri lintasan sehingga memerlukan koreksi ke kanan. Pengujian mencakup kondisi netral, variasi satu masukan, dan kombinasi kedua masukan; keluaran yang seimbang pada galat berlawanan tidak dianggap sebagai bukti bahwa robot telah berada pada lintasan.

Galat arah (°)	Galat lateral (m)	PWM kiri	PWM kanan	Respons yang diperiksa
0	0	180	180	Perintah maju tanpa koreksi diferensial
−30	0,5	255	50	Koreksi kanan kuat
30	−0,5	50	255	Koreksi kiri kuat
−18	0,3	242	76,7	Koreksi kanan
18	−0,3	76,7	242	Koreksi kiri
−6	0,1	215	128	Koreksi kanan
6	−0,1	128	215	Koreksi kiri
−30	−0,5	180	180	Pengaruh kedua galat saling menyeimbangkan
30	0,5	180	180	Pengaruh kedua galat saling menyeimbangkan
−18	−0,3	167	183	Koreksi kiri kecil
18	0,3	183	167	Koreksi kanan kecil
−6	−0,1	172	179	Koreksi kiri kecil
6	0,1	179	172	Koreksi kanan kecil
6	0,4	234	92	Koreksi kanan; pengaruh lateral lebih besar
−6	−0,4	92	234	Koreksi kiri; pengaruh lateral lebih besar

Tabel 3.40 Acuan pengujian kombinasi masukan FIS 2

3.5.5.3 Pengujian Kesesuaian MATLAB dan ESP32

Masukan yang identik diberikan kepada model MATLAB dan program C++ ESP32 untuk membandingkan pengali FIS 1 serta kedua keluaran FIS 2. Perbandingan dilakukan menggunakan nilai berpresisi penuh sebelum pembulatan, karena angka pada tabel simulasi merupakan nilai tampilan yang dapat dibulatkan. Pengujian kemudian dilanjutkan pada perkalian pengali kecepatan dengan PWM kiri–kanan, pembatasan keluaran pada rentang 0–255, dan pemeriksaan sinyal PWM ESP32 menggunakan osiloskop atau *logic analyzer*.

Bagian yang diuji	Data pengujian	Parameter pembanding	Bukti pengujian
FIS 1	Seluruh domain dengan langkah 1 piksel	Selisih (k_v) MATLAB–ESP32	Rekaman keluaran kedua platform
FIS 2	Kisi galat arah langkah 3° dan lateral langkah 0,05 m	Selisih PWM kiri dan kanan sebelum pembulatan	Rekaman 441 pasangan masukan
Batas fungsi keanggotaan	Titik batas dan titik di sekitarnya	Kesesuaian keluaran dan keberadaan nilai tidak valid	Tabel pemeriksaan batas
Penggabungan FIS	Variasi (k_v) dan pasangan PWM FIS 2	Kesesuaian hasil perkalian dan pembulatan	Tabel PWM akhir
Sinyal PWM ESP32	Nilai perintah 0, 50, 120, 180, 220, dan 255	Duty cycle aktual terhadap perintah 8-bit	Hasil osiloskop atau *logic analyzer*

Tabel 3.41 Rencana pemeriksaan keseuaian implementasi pengendali

3.5.3.4 Pengujian Pelacakan Lintasan Robot

Pengujian pelacakan dilakukan pada lintasan lurus sepanjang 5 m di dalam petak pengujian 100 m², dengan variasi penyimpangan awal dan lima pengulangan untuk setiap kondisi. FIS 2 menerima galat yang dihitung dari estimasi encoder–IMU, sedangkan penilaian kinerja menggunakan posisi serta orientasi robot yang diukur secara independen. Pengujian awal menggunakan $(k_v = 1)$, kemudian dilanjutkan dengan beberapa nilai pengali FIS 1 untuk memeriksa pelacakan saat

perintah PWM dikurangi; kondisi beban, permukaan, tegangan baterai, dan parameter pengendali dicatat.

Skenario	Galat arah awal	Galat lateral awal	Pengali kecepatan	Tujuan
Kondisi sejajar	0°	0 m	1	Memeriksa kemampuan mempertahankan lintasan
Penyimpangan arah	−10° dan 10°	0 m	1	Memeriksa koreksi orientasi
Penyimpangan lateral	0°	−0,2 dan 0,2 m	1	Memeriksa koreksi posisi lateral
Galat saling menguatkan	−10°; 10°	0,2 m; −0,2 m	1	Memeriksa koreksi gabungan searah
Galat saling berlawanan	−10°; 10°	−0,2 m; 0,2 m	1	Memeriksa respons terhadap kebutuhan koreksi yang berlawanan
Pelacakan saat perlambatan	0°	−0,2 dan 0,2 m	0,75; 0,50; 0,25	Memeriksa pengaruh pengurangan PWM terhadap koreksi lintasan

Tabel 3.42 Skenario pengujian pelacakan lintasan

Indikator	Data penilaian	Satuan
RMSE galat lateral	Penyimpangan posisi aktual terhadap garis acuan pada titik pengukuran	m
Galat lateral maksimum	Penyimpangan lateral absolut terbesar yang terukur	m
Rata-rata galat arah absolut	Selisih orientasi aktual terhadap arah lintasan	°
Jarak pemulihan	Jarak tempuh hingga robot masuk dan bertahan dalam toleransi lintasan	m
Waktu penyelesaian	Waktu dari awal gerak hingga mencapai ujung lintasan	s
Tingkat keberhasilan	Proporsi percobaan yang memenuhi seluruh kriteria pelacakan	%

Tabel 3.43 Indikator evaluasi pelacakan lintasan

3.5.4 Matriks Indikator Capaian Kuantitatif

Matriks indikator digunakan untuk menilai kelayakan sensor, estimasi gerak, implementasi fuzzy Sugeno, dan pelacakan lintasan berdasarkan pengukuran yang dapat diverifikasi. Batas numerik pada Tabel 3.44 merupakan target awal yang diusulkan, bukan hasil pengujian, dan ditetapkan sebelum evaluasi setelah mempertimbangkan ketelitian alat ukur serta karakteristik robot. Penilaian dilakukan per indikator agar keberhasilan satu bagian tidak menutupi kegagalan bagian lain; integrasi FSM belum dimasukkan karena logika operasinya masih dalam tahap penentuan.

No.	Aspek pengujian	Indikator kuantitatif	Target awal yang diusulkan	Metode verifikasi
1	Hitungan encoder	Galat hitungan terhadap putaran acuan	≤1% pada setiap encoder	Pengujian 1, 5, dan 10 putaran, masing-masing lima pengulangan
2	Identifikasi arah encoder	Kesesuaian tanda hitungan terhadap arah putaran	100% percobaan sesuai	Pengamatan putaran maju dan mundur
3	Kecepatan encoder	Galat relatif terhadap takometer	≤5% pada kondisi putaran stabil dan kecepatan acuan bukan nol	Perbandingan pada beberapa tingkat PWM
4	Perubahan orientasi IMU	Rata-rata galat sudut absolut	≤5°	Pengujian putaran ±30°, ±90°, dan ±180°
5	Drift giroskop	Perubahan estimasi sudut saat diam setelah koreksi bias	≤3° selama 60 detik	Pencatatan orientasi dengan sensor tetap diam
6	Karakterisasi penggerak	Kelengkapan data PWM–kecepatan	Seluruh nilai PWM 0, 50, 120, 180, 220, dan 255 diuji pada kedua motor dan arah	Tabel serta grafik karakterisasi, masing-masing lima pengulangan
7	Estimasi posisi lokal	RMSE posisi pada titik pemeriksaan	≤0,20 m per skenario dan permukaan	Perbandingan estimasi dengan posisi acuan independen
8	Estimasi orientasi lokal	RMSE orientasi pada titik pemeriksaan	≤5° per skenario dan permukaan	Perbandingan estimasi dengan sudut acuan

9	Manfaat fusi encoder–IMU	Perubahan RMSE terhadap odometri encoder saja	RMSE orientasi lebih rendah tanpa peningkatan RMSE posisi yang melampaui ketelitian pengukuran acuan	Evaluasi kedua metode dari rekaman percobaan yang sama
10	Implementasi FIS 1	Selisih absolut keluaran MATLAB–ESP32	≤0,001 untuk (k_v) pada seluruh masukan uji	Pemeriksaan domain 450–640 piksel dengan langkah 1 piksel
11	Implementasi FIS 2	Selisih absolut keluaran MATLAB–ESP32	≤1 satuan PWM untuk setiap keluaran sebelum pembulatan	Pemeriksaan 441 pasangan masukan
12	Penggabungan keluaran FIS	Kesesuaian perkalian, pembulatan, dan pembatasan PWM	100% kasus sesuai; PWM akhir berada pada rentang 0–255	Perbandingan hasil perhitungan dan keluaran program
13	Pelacakan lintasan	RMSE galat lateral aktual	≤0,15 m pada lintasan lurus 5 m	Pengukuran posisi aktual terhadap garis acuan
14	Pelacakan lintasan	Galat lateral maksimum setelah pemulihan	≤0,25 m	Pengukuran pada bagian lintasan setelah robot masuk toleransi
15	Pelacakan arah	Rata-rata galat arah absolut setelah pemulihan	≤5°	Pengukuran orientasi aktual terhadap arah lintasan
16	Pemulihan lintasan	Jarak untuk masuk toleransi lateral ±0,15 m dan arah ±5°	≤2 m serta bertahan hingga akhir lintasan	Pengujian variasi galat awal yang direncanakan
17	Keberhasilan pelacakan	Proporsi percobaan yang memenuhi seluruh kriteria pelacakan	≥80%; sasaran pengembangan ≥90%	Rekapitulasi percobaan berhasil dan total percobaan

Tabel 3.44 Matriks indicator capaian kuantitatif subsistem kendali gerak

No	Kondisi/skenario	Target	Hasil pengukuran	Status	Keterangan
Sesuai Tabel 3.44	Kondisi pengujian	Batas yang ditetapkan	Diisi setelah pengujian	Memenuhi/belum memenuhi/belum diuji	Penyimpangan dan tindak lanjut

Tabel 3.45 Format rekapitulasi penilaian kalayakan hasil

DAFTAR PUSTAKA

Aji, B., & Sutikno. (2023). Fuzzy logic of Sugeno method for controlling line follower mobile robot. *ILKOM Jurnal Ilmiah, 15*(2), 283–289. https://doi.org/10.33096/ilkom.v15i2.1558.283-289

Akmal, M. D., Al Tahtawi, A. R., Wijayanto, K., & Wahab, F. (2024). Trajectory tracking control design for mobile robot using interval type-2 fuzzy logic. *Journal of Fuzzy Systems and Control, 2*(2), 67–73. https://doi.org/10.59247/jfsc.v2i2.200

Assefinew, W., Jeong, J.-S., Kim, S.-K., Debele, Y., & Kang, B.-S. (2021). Improved attitude and heading accuracy with double quaternion parameters estimation and magnetic disturbance rejection. *Sensors, 21*(16), Article 5475. https://doi.org/10.3390/s21165475

Cantelli, L., Ligama, S., Muscato, G., & Spina, D. (2016). Auto-calibration methods of kinematic parameters and magnetometer offset for the localization of a tracked mobile robot. *Robotics, 5*(4), Article 23. https://doi.org/10.3390/robotics5040023

Costanzi, R., Fanelli, F., Monni, N., Ridolfi, A., & Allotta, B. (2016). An attitude estimation algorithm for mobile robots under unknown magnetic disturbances. *IEEE/ASME Transactions on Mechatronics, 21*(4), 1900–1911. https://doi.org/10.1109/TMECH.2016.2559941

Fazekas, M., Gáspár, P., & Németh, B. (2021). Calibration and improvement of an odometry model with dynamic wheel and lateral dynamics integration. *Sensors, 21*(2), Article 337. https://doi.org/10.3390/s21020337

Inoue, K., Kaizu, Y., Igarashi, S., Furuhashi, K., & Imou, K. (2022). Autonomous navigation and obstacle avoidance in an orchard using machine vision techniques for a robotic mower. *Engineering in Agriculture, Environment and Food, 15*(4), 87–99. https://doi.org/10.37221/eaef.15.4_87

InvenSense. (2016). *MPU-9250 product specification* (Revision 1.1, Document PS-MPU-9250A-01).

Kok, M., Hol, J. D., & Schön, T. B. (2017). Using inertial sensors for position and orientation estimation. *Foundations and Trends in Signal Processing, 11*(1–2), 1–153. https://doi.org/10.1561/2000000094

Lucet, E., Lucazeau, A., & Chemin, J. (2024). Autonomous forklift navigation inside a cluttered logistics factory. In *Proceedings of the 21st International Conference on Informatics in Control, Automation and Robotics* (Vol. 2, pp. 327–335). https://doi.org/10.5220/0013067600003822

Nguyen, A.-T., & Vu, C.-T. (2022). Mobile robot motion control using a combination of fuzzy logic method and kinematic model. In *Intelligent systems and networks* (Lecture Notes in Networks and Systems, Vol. 471, pp. 495–503). Springer. https://doi.org/10.1007/978-981-19-3394-3_56

Okawara, T., Koide, K., Oishi, S., Yokozuka, M., Banno, A., Uno, K., & Yoshida, K. (2024). *Tightly-coupled LiDAR-IMU-wheel odometry with online calibration of a kinematic model for skid-steering robots* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2404.02515

Omrane, H., Masmoudi, M. S., & Masmoudi, M. (2016). Fuzzy logic based control for autonomous mobile robot navigation. *Journal of Robotics, 2016*, Article 9548482. https://doi.org/10.1155/2016/9548482

Radočaj, D., Plaščak, I., Heffer, G., & Jurišić, M. (2022). A low-cost global navigation satellite system positioning accuracy assessment method for agricultural machinery. *Applied Sciences, 12*(2), Article 693. https://doi.org/10.3390/app12020693

Rahimi Azghadi, M., Olsen, A., Wood, J., Saleh, A., Calvert, B., Granshaw, T., Fillols, E., & Philippa, B. (2024). *Precise robotic weed spot-spraying for reduced herbicide usage and improved environmental outcomes—A real-world case study* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2401.13931

Reitbauer, E., & Schmied, C. (2021). Bridging GNSS outages with IMU and odometry: A case study for agricultural vehicles. *Sensors, 21*(13), Article 4467. https://doi.org/10.3390/s21134467

Sathyamoorthy, A. J., Weerakoon, K., Guan, T., Liang, J., & Manocha, D. (2022). *TerraPN: Unstructured terrain navigation using online self-supervised learning* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2202.12873

Tomaštík, J., & Everett, T. (2023). Static positioning under tree canopy using low-cost GNSS receivers and adapted RTKLIB software. *Sensors, 23*(6), Article 3136. https://doi.org/10.3390/s23063136

u-blox. (2025). *u-blox 20 HPG 2.00: High precision GNSS receiver—Interface description* (Document UBXDOC-304424225-19888, R01).

Valenti, R. G., Dryanovski, I., & Xiao, J. (2015). Keeping a good attitude: A quaternion-based orientation filter for IMUs and MARGs. *Sensors, 15*(8), 19302–19330. https://doi.org/10.3390/s150819302

Yamauchi, G., Nagatani, K., Hashimoto, T., & Fujino, K. (2017). Slip-compensated odometry for tracked vehicle on loose and weak slope. *ROBOMECH Journal, 4*, Article 27. https://doi.org/10.1186/s40648-017-0095-1

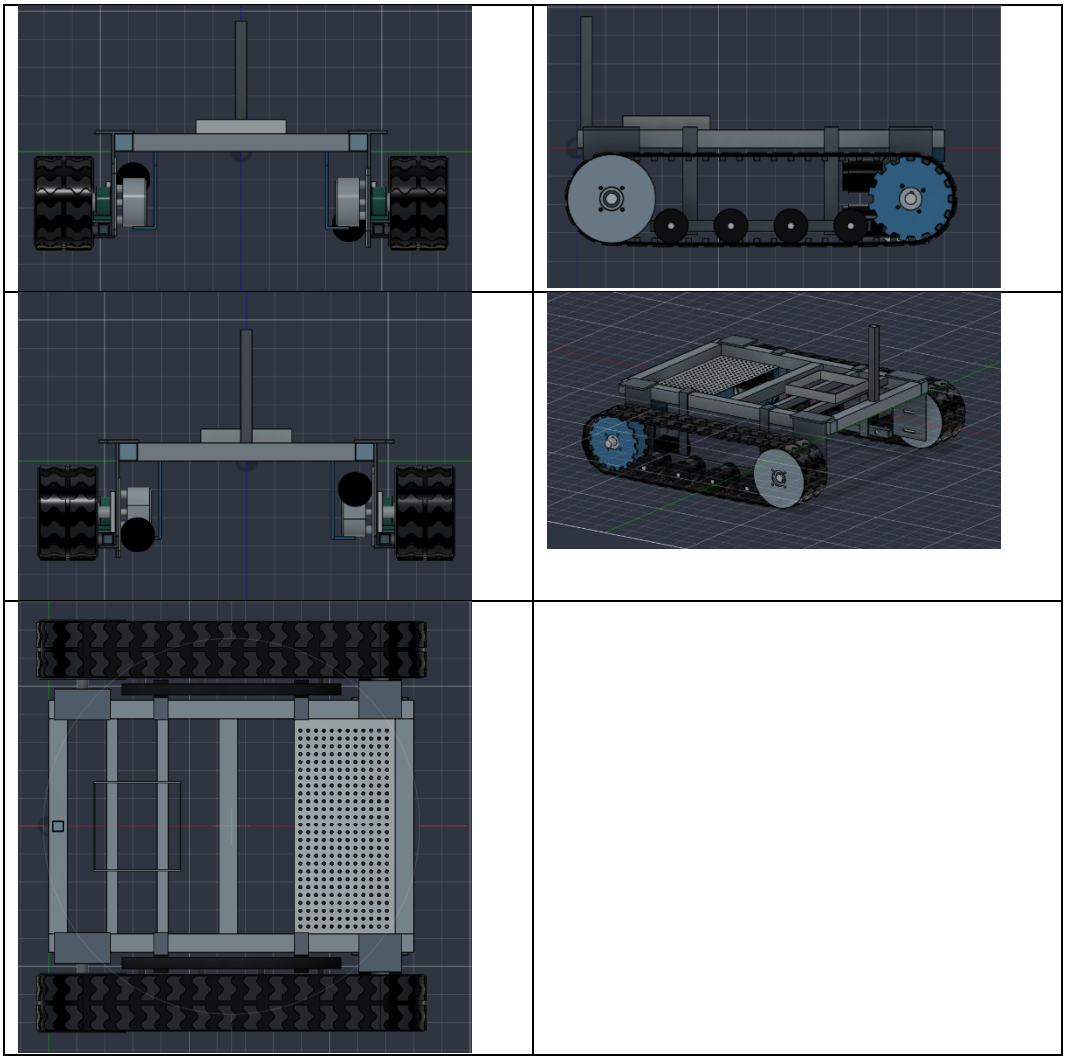
Zuo, X., Zhang, M., Wang, M., Chen, Y., Huang, G., Liu, Y., & Li, M. (2020). *Visual-based kinematics and pose estimation for skid-steering robots* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2006.04335

LAMPIRAN

Lampiran 1 : Biodata Penulis

Nama Lengkap	:	Nicodemus Oloan Simanjuntak
Nomor Induk Mahasiswa	:	2305110016
Tempat, Tanggal Lahir	:	Jakarta, 06 April 2005
Program Studi	:	Teknik Komputer (S1)
Fakultas	:	Fakultas Teknik
Perguruan Tinggi	:	Universitas Negeri Semarang (UNNES)
Alamat (Email)	:	juntaknicodemus@students.unnes.ac.id
Bidang Minat Penelitian	:	Embedded System, AI (Artificial Intelligence), Robotics, Control System, Electronics

Lampiran 2 : Desain Robot



Lampiran 3 : Surat Keputusan / Lembar Penugasan Dosen Pembimbing

KEMENTERIAN PENDIDIKAN TINGGI, SAINS, DAN TEKNOLOGI
UNIVERSITAS NEGERI SEMARANG
FAKULTAS TEKNIK

Gedung Dekanat, Kampus Sekaran,
Gunungpati, Semarang 50229.
Telepon: 0248508101
Laman: http://ft.unnes.ac.id, surel:
ft@mail.unnes.ac.id

KEPUTUSAN
DEKAN FAKULTAS TEKNIK
UNIVERSITAS NEGERI SEMARANG
Nomor: B/950/UN37.1.5/TD.06/2026
Tentang
PENETAPAN DOSEN PEMBIMBING SKRIPSI/TUGAS AKHIR SEMESTER GASAL/GENAP
TAHUN AKADEMIK 2026/2027

Menimbang : Bahwa untuk memperlancar mahasiswa Prodi Teknik Komputer, S1 Fakultas Teknik membuat Skripsi/Tugas Akhir, maka perlu menetapkan Dosen-dosen Prodi Teknik Komputer, S1 Fakultas Teknik UNNES untuk menjadi pembimbing.

Mengingat :
1. Undang-Undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional;
2. Peraturan Pemerintah Nomor 4 Tahun 2014 tentang Penyelenggaraan Pendidikan Tinggi dan Pengelolaan Perguruan Tinggi;
3. Peraturan Pemerintah Nomor 36 Tahun 2022 tentang Perguruan Tinggi Negeri Badan Hukum Universitas Negeri Semarang;
4. Peraturan Rektor Universitas Negeri Semarang Nomor 88 Tahun 2024 tentang Panduan Tugas Akhir Sarjana dan Diploma Universitas Negeri Semarang;

Menimbang : Usulan Koordinator Prodi Teknik Komputer, S1 Tanggal 1 September 2026

MEMUTUSKAN

Menetapkan :

PERTAMA : Menunjuk dan menugaskan kepada:

Nama	: Dr. Feddy Setio Pribadi, S.Pd., M.T.
NIP	: 197808222003121002
Pangkat/Golongan	: Penata Tk. I - III/d
Jabatan Akademik	: Lektor

Sebagai Pembimbing

Untuk membimbing mahasiswa penyusun skripsi/Tugas Akhir :

Nama	: NICODEMUS OLOAN SIMANJUNTAK
NIM	: 2305110016
Prodi	: Teknik Komputer, S1
Topik	: Perancangan dan Implementasi Robot Pemotong Rumput Otonom dengan Sistem Navigasi Cerdas untuk Pemeliharaan Halaman Rumah

KEDUA : Keputusan ini mulai berlaku sejak tanggal ditetapkan.

DITETAPKAN DI : SEMARANG
PADA TANGGAL : 1 September 2026
DEKAN

Prof. Dr. Wirawan Sumbodo, M.T.
NIP 196601051990021002

Tembusan
1. WD Bid. Akademik dan Mawa
2. Ketua Prodi
3. Pertinggal

2305110016

...:: FM-03-AKD-24/Rev. 00 ::...

Dokumen ini telah ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan oleh Balai Besar Sertifikasi Elektronik (BSrE), BSSN.