



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



RANCANG BANGUN KENDALI *MOBILE* IKIBOT DALAM MENGHINDARI RINTANGAN SECARA *REAL-TIME*

Sub Judul:

**KENDALI ROBOT *MOBILE* IKIBOT BERBASIS
PROPORTIONAL INTEGRAL DERIVATIVE (PID)**

SKRIPSI

Yosef Dwi Krisnata
2203431025
**POLITEKNIK
NEGERI
JAKARTA**

**PROGRAM STUDI INSTRUMENTASI DAN KONTROL
INDUSTRI**

**JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI JAKARTA**

2026



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**RANCANG BANGUN KENDALI *MOBILE* IKIBOT DALAM
MENGHINDARI RINTANGAN SECARA *REAL-TIME***

Sub Judul:

**KENDALI ROBOT *MOBILE* IKIBOT BERBASIS
PROPORTIONAL INTEGRAL DERIVATIVE (PID)**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar

Sarjana Terapan

**POLITEKNIK
NEGERI
JAKARTA**

**Yosef Dwi Krisnata
2203431025**

**PROGRAM STUDI INSTRUMENTASI DAN KONTROL
INDUSTRI**

**JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI JAKARTA**

2026



Hak Cipta :

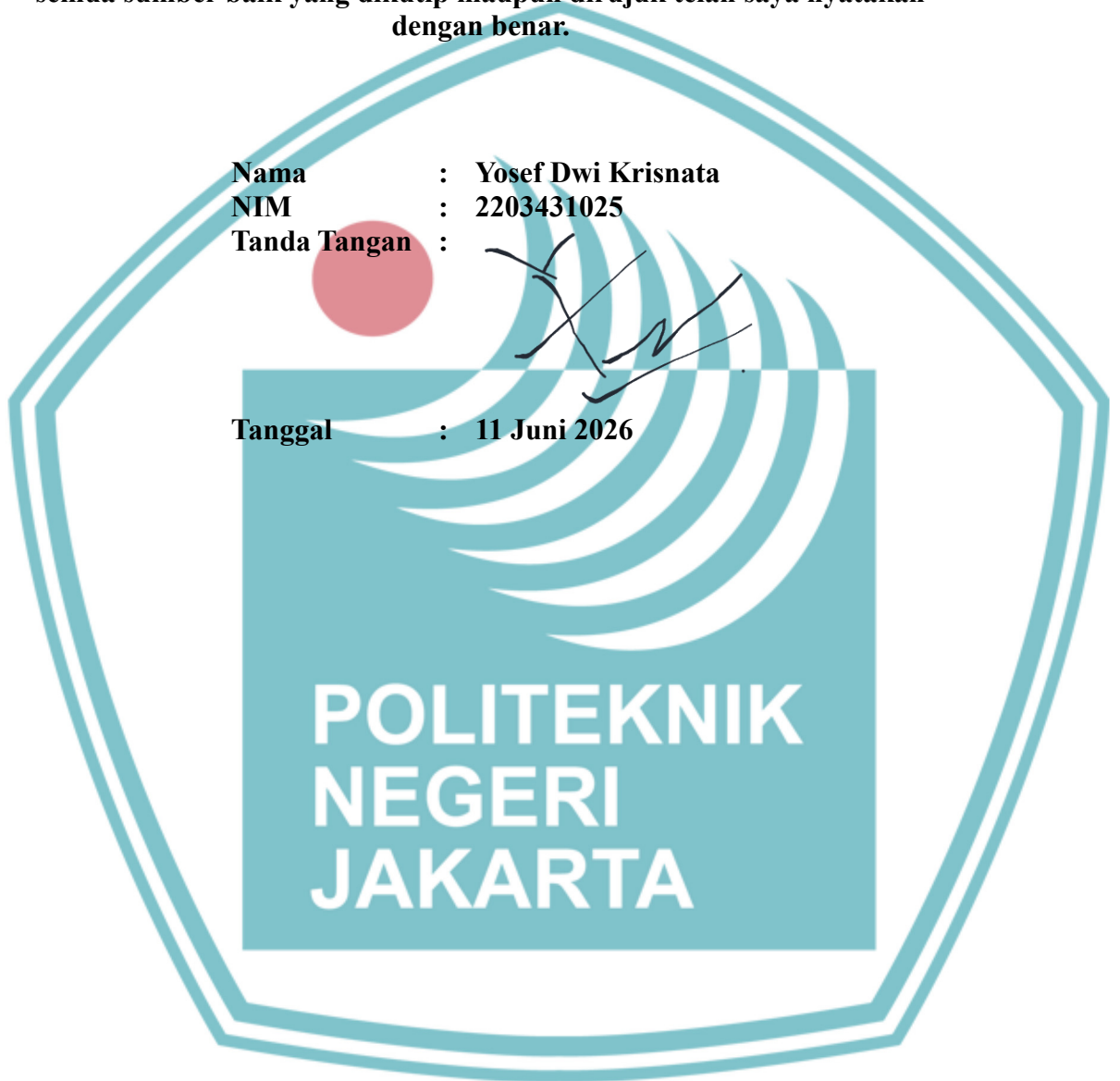
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Yosef Dwi Krisnata
NIM : 2203431025
Tanda Tangan :

Tanggal : 11 Juni 2026





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LEMBAR PENGESAHAN
SKRIPSI

Skripsi diajukan oleh :
Nama : Yosef Dwi Krisnata
NIM : 2203431025
Program Studi : Instrumentasi dan Kontrol Industri
Judul Tugas Akhir : Kendali Robot *mobile* IKIBOT Berbasis Proportional
Integral Derivative (PID)

Telah diuji oleh tim penguji dalam Sidang Tugas Akhir pada 19 Juni 2026 dan dinyatakan **LULUS**

Pembimbing : Harivanto, S.Pd., M.T. ()
NIP. 199101282020121008

**POLITEKNIK
NEGERI
JAKARTA**
Depok, 19 Juni 2026
Disahkan oleh

Ketua Jurusan Teknik Elektro



Dr. Murie Dwivaniti, S.T., M.T.

NIP. 197803312003122002



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas berkat rahmat dan karunia-Nya sehingga penulis mampu menyelesaikan penulisan skripsi yang berjudul “Kendali Robot *mobile* IKIBOT Berbasis *Proportional Integral Derivative* (PID)”. Skripsi ini ditulis dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Terapan pada Program Studi Instrumentasi dan Kontrol Industri Jurusan Teknik Elektro Politeknik Negeri Jakarta.

Penulis menyadari bahwa dalam menyelesaikan penulisan skripsi ini banyak memperoleh bantuan, bimbingan, arahan dan nasihat dari berbagai pihak yang akan selalu diingat penulis. Untuk itu penulis ingin mengucapkan terima kasih yang setulus-tulusnya kepada :

1. Dr. Murie Dwiyaniti, S.T.,M.T. selaku Ketua Jurusan Teknik Elektro;
2. Sulis Setiowati, S.Pd.,M.Eng. selaku Kepala Program Studi Instrumentasi dan Kontrol Industri;
3. Hariyanto, S.Pd., M.T. selaku Dosen Pembimbing yang telah menyediakan waktu, tenaga, serta pikiran untuk membimbing penulis dalam menyelesaikan skripsi;
4. Orang tua yang senantiasa memberikan bantuan moral dan material;
5. Stephanie Dwi Lestari selaku teman hidup yang selalu mendukung dan membantu dalam pengerjaan skripsi;
6. Soraya Putri Aulia dan Muhammad Raihan Rifqo Ardandi serta rekan-rekan IKI-22;

Akhir kata, semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 11 Juni 2026

Yosef Dwi Krisnata



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Kendali Robot *mobile* IKIBOT Berbasis Proportional Integral Derivative (PID)

Abstrak

Penelitian ini bertujuan merancang dan mengimplementasikan sistem kendali Proportional-Integral-Derivative (PID) pada robot *mobile* IKIBOT untuk meningkatkan stabilitas navigasi dan akurasi pergerakan menuju target secara otonom. Sistem dikembangkan menggunakan Robot Operating System (ROS) Noetic pada Raspberry Pi 4B sebagai pengendali tingkat tinggi dan Teensy 4.0 sebagai pengendali tingkat rendah. Robot dilengkapi sensor YDLIDAR X3, IMU MPU6050, dan enkoder roda untuk mendukung lokalisasi, navigasi, dan pengendalian gerak. Pengendali PID diterapkan pada dua tingkat kendali, yaitu PID Low-Level untuk pengaturan kecepatan roda dan PID High-Level untuk navigasi robot. Parameter PID ditentukan menggunakan Particle Swarm Optimization (PSO) dengan fungsi objektif Integral Time Absolute Error (ITAE). Hasil optimasi menghasilkan parameter PID Low-Level sebesar $K_p = 376,47$, $K_i = 23,40$, dan $K_d = 0,78$, sedangkan PID High-Level menghasilkan parameter linear $K_p = 2,19$, $K_i = 0,12$, $K_d = 0,00$ dan parameter angular $K_p = 1,89$, $K_i = 0,14$, $K_d = 0,00$. Kinerja sistem dievaluasi menggunakan Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Integral Time Absolute Error (ITAE), Final Position Error (FPE), dan waktu tempuh. Hasil pengujian menunjukkan bahwa sistem kendali PID yang dioptimasi menggunakan PSO mampu menghasilkan navigasi yang stabil dan akurat pada skenario satu titik maupun tiga titik, baik pada lingkungan tanpa hambatan maupun dengan hambatan. Hasil penelitian menunjukkan bahwa integrasi PID dan PSO efektif dalam meningkatkan performa navigasi robot *mobile* IKIBOT berbasis ROS.

Kata Kunci: Robot Mobile IKIBOT, PID, Particle Swarm Optimization, ROS, Navigasi Otonom, RMSE, MAE, ITAE, Final Position Error

**POLITEKNIK
NEGERI
JAKARTA**



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Proportional-Integral-Derivative (PID) Control of the IKIBOT Mobile Robot

Abstract

This research aims to design and implement a Proportional-Integral-Derivative (PID) control system for the IKIBOT mobile robot to improve navigation stability and motion accuracy toward autonomous target locations. The system was developed using Robot Operating System (ROS) Noetic on a Raspberry Pi 4B as the high-level controller and a Teensy 4.0 as the low-level controller. The robot was equipped with a YDLIDAR X3, MPU6050 IMU, and wheel encoders to support localization, navigation, and motion control. PID controllers were implemented at two control levels: a Low-Level PID for wheel speed regulation and a High-Level PID for navigation control. Particle Swarm Optimization (PSO) was employed to determine the PID parameters using Integral Time Absolute Error (ITAE) as the objective function. The optimized Low-Level PID parameters were $K_p = 376.47$, $K_i = 23.40$, and $K_d = 0.78$, while the High-Level PID parameters were $K_p = 2.19$, $K_i = 0.12$, and $K_d = 0.00$ for linear control, and $K_p = 1.89$, $K_i = 0.14$, and $K_d = 0.00$ for angular control. System performance was evaluated using Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Integral Time Absolute Error (ITAE), Final Position Error (FPE), and Travel Time. Experimental results showed that the PSO-tuned PID controller provided stable and accurate navigation performance in both single-waypoint and three-waypoint scenarios under obstacle-free and obstacle-present environments. The results demonstrate that the integration of PID control and PSO optimization effectively improves the navigation performance of the ROS-based IKIBOT mobile robot.

Keywords: *IKIBOT Mobile Robot, PID Control, Particle Swarm Optimization, ROS, Autonomous Navigation, RMSE, MAE, ITAE, Final Position Error.*

**POLITEKNIK
NEGERI
JAKARTA**



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR ISI

HALAMAN SAMPUL.....	i
HALAMAN JUDUL.....	ii
HALAMAN PERNYATAAN ORISINALITAS	iii
LEMBAR PENGESAHAN	iv
KATA PENGANTAR.....	v
<i>Abstrak</i>	vi
<i>Abstract</i>	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Tujuan Penelitian.....	4
1.4 Batasan Masalah.....	4
1.5 Luaran	5
BAB II TINJAUAN PUSTAKA	6
2.1 <i>State Of The Art</i>	6
2.2 <i>Robot Mobile</i>	9
2.2.1 Pengertian <i>Robot Mobile</i>	9
2.2.2 <i>Differential Drive Robot Mobile</i>	9
2.2.3 Sistem Kendali <i>Robot Mobile</i>	11
2.2.4 Sistem Navigasi <i>Robot Mobile</i>	12
2.3 Kendali <i>Proportional Integral Derivative (PID)</i>	13
2.3.1 Pengertian Kendali <i>Proportional Integral Derivative (PID)</i>	13
2.3.2 Model Matematis <i>Proportional Integral Derivative (PID)</i>	14
2.3.3 Parameter <i>Proportional Integral Derivative (PID)</i>	14
2.3.4 Kontrol Gerak Berbasis PID	15
2.3.5 <i>Particle Swarm Optimization (PSO)</i> untuk <i>Tuning PID</i>	17
2.4 Komponen	20
2.4.1 Raspberry Pi 4B	20
2.4.2 Teensy 4.0	21
2.4.3 YDLIDAR X3	22
2.4.4 IMU MPU6050	23
2.4.5 L298N	24



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

2.4.6	Motor DC Encoder.....	25
2.4.7	Modul ACS712	26
2.4.8	<i>MicroSD Card</i>	27
2.4.9	Baterai LiPo	27
2.4.10	Kabel Data.....	28
2.4.11	UBEC (Buck Converter)	28
2.5	Perangkat Lunak.....	29
2.5.1	Ubuntu 20.04 LTS	29
2.5.2	Robot Operating System (ROS).....	30
2.5.3	Python3	32
2.5.4	Visual Studio Code.....	33
2.5.5	Robot Visualization (RViz)	34
2.6	Parameter Evaluasi Kinerja Sistem.....	34
2.6.1	<i>Root Mean Square Error (RMSE)</i>	35
2.6.2	<i>Mean Absolute Error (MAE)</i>	35
2.6.3	<i>Integral Time Absolute Error (ITAE)</i>	36
2.6.4	<i>Final Position Error (FPE)</i>	36
2.6.5	<i>Travel Time</i>	37
BAB III PERANCANGAN DAN REALISASI		38
3.1	Perancangan Sistem	38
3.1.1	Deskripsi Sistem	40
3.1.2	Cara Kerja Sistem	43
3.1.3	Spesifikasi Alat	48
3.1.4	Perancangan Sistem Embedded	51
3.1.5	Diagram Blok Sistem	54
3.1.6	Perancangan Mekanis Alat	61
3.2	Realisasi Sistem	62
3.2.1	Realisasi Pembuatan Sistem Embedded	62
3.2.2	Skematik Rangkaian Alat	62
3.2.3	Realisasi Pembuatan Alat	63
3.2.4	Realisasi Pembuatan <i>Map</i>	67
3.2.5	Realisasi Sistem ROS pada Robot <i>Mobile</i> IKIBOT	68
3.2.6	<i>Flowchart</i> Subsistem.....	72
3.2.7	Realisasi Kendali PID	74
BAB IV PEMBAHASAN.....		80
4.1	Pengujian Kendali PID.....	80
4.1.1	Deskripsi Pengujian	82



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

4.1.2	Daftar Alat Pengujian	82
4.1.3	Prosedur Pengujian	83
4.1.4	Data Pengujian Kendali <i>PID low-level</i>	83
4.1.5	<i>PSO Convergence Curve for Low-Level PID</i>	86
4.1.6	Data Pengujian Kendali <i>PID low-level</i> dengan <i>PSO-Tuned</i>	87
4.1.7	Data Pengujian Kendali <i>PID Navigasi</i>	90
4.1.8	<i>PSO Convergence Curve for Navigation PID</i>	93
4.1.9	Data Pengujian Kendali <i>PID Navigasi</i> dengan <i>PSO-Tuned</i>	94
4.1.10	Hasil Pengujian Navigasi Robot	98
4.2	Hasil Pengujian Kendali <i>PID</i> pada Robot <i>Mobile IKIBOT</i>	107
4.2.1	Kinerja <i>PID Low-Level</i> Pra dan Pasca <i>Tuning PSO</i>	107
4.2.2	Kinerja <i>PID High-Level</i> Pra dan Pasca <i>Tuning PSO</i>	109
4.2.3	Performa Navigasi Skenario Tiga Titik <i>Full Map</i>	111
4.2.4	Performa Navigasi Skenario Satu Titik <i>Full Map</i>	114
4.2.5	Pencapaian Tujuan Penelitian	115
BAB V PENUTUP		116
5.1	Kesimpulan	116
5.2	Saran.....	117
DAFTAR PUSTAKA		118
LAMPIRAN		123

**POLITEKNIK
NEGERI
JAKARTA**

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR GAMBAR

Gambar 2.1 Raspberry Pi 4B	21
Gambar 2.2 Teensy 4.0.....	22
Gambar 2.3 YDLIDAR X3	23
Gambar 2.4 MPU6050	24
Gambar 2.5 IC L298N.....	25
Gambar 2.6 Motor DC Enkoder.....	26
Gambar 2.7 Modul ACS712.....	26
Gambar 2.8 MicroSD Card	27
Gambar 2.9 Baterai LiPo.....	28
Gambar 2.10 Kabel Data.....	28
Gambar 2.11 UBEC DC to DC Converter	29
Gambar 2.12 Logo Resmi Ubuntu	30
Gambar 2.13 Logo Resmi ROS	32
Gambar 2.14 Logo Resmi Python3	33
Gambar 2.15 Logo Resmi Visual Studio Code	33
Gambar 2.16 Logo Resmi Rviz.....	34
Gambar 3.1 Flowchart Perancangan Sistem	39
Gambar 3.2 tf Tree Robot Mobile.....	43
Gambar 3.3 ROS Computation Graph System	44
Gambar 3.4 Skematik Sistem Kontrol Tingkat Rendah.....	52
Gambar 3.5 PCB Sistem Kontrol Tingkat Rendah.....	54
Gambar 3.6 Diagram Blok Alat	55
Gambar 3.7 Diagram Blok Subsistem.....	57
Gambar 3.8 Diagram Blok Cascade Control Closed Loop.....	59
Gambar 3.9 Desain Mekanis Alat	61
Gambar 3.10 Modul Sistem Kontrol Tingkat Rendah	62
Gambar 3.11 Skematik Rangkaian Power.....	63
Gambar 3.12 Skematik Rangkaian Kontrol	63
Gambar 3.13 Tampak Depan Alat.....	64
Gambar 3.14 Layer Bawah	64
Gambar 3.15 Layer Tengah.....	65
Gambar 3.16 Layer Atas Tampak Bawah.....	65
Gambar 3.17 Layer Atas Tampak Atas.....	66
Gambar 3.18 Map Pengujian.....	67
Gambar 3.19 Inisialisasi ROS Master.....	69
Gambar 3.20 Inisialisasi Komunikasi Serial Robot	69
Gambar 3.21 SLAM Nodes.....	70
Gambar 3.22 Rviz Nodes	70
Gambar 3.23 Lokalisasi Robot.....	71
Gambar 3.24 Visualisasi Sistem Navigasi di Rviz.....	72
Gambar 3.25 Flowchart Subsistem	72
Gambar 4.1 Linear Velocity Tracking.....	84
Gambar 4.2 Angular Velocity Tracking	85
Gambar 4.3 Tracking Error	85
Gambar 4.4 Grafik Konvergensi PSO Low-Level PID	87
Gambar 4.5 Proses Tuning PSO Low-Level PID	87



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Gambar 4.6 Linear Velocity After PSO tuning	88
Gambar 4.7 Angular Velocity After PSO Tuning	88
Gambar 4.8 Tracking Error After PSO Tuning	89
Gambar 4.9 Linear Velocity Navigation PID	90
Gambar 4.10 Angular Velocity Navigation PID	91
Gambar 4.11 Tracking Error Navigation PID	92
Gambar 4.12 Trajectory Tracking Navigation PID	92
Gambar 4.13 Grafik Konvergensi PSO Navigation PID	94
Gambar 4.14 Proses Tuning High-Level PID	94
Gambar 4.15 Linear Velocity Nav PID After PSO Tuning	95
Gambar 4.16 Angular Velocity Nav PID After PSO Tuning	95
Gambar 4.17 Tracking Error Nav PID After PSO Tuning	96
Gambar 4.18 Trajectory Tracking Nav PID After PSO Tuning	96
Gambar 4.19 Linear Velocity Pengujian Tiga Titik Full Map Environment	98
Gambar 4.20 Angular Velocity Pengujian Tiga Titik Full Map Environment	99
Gambar 4.21 Tracking Error Pengujian Tiga Titik Full Map Environment	100
Gambar 4.22 Trajectory Tracking Pengujian Tiga Titik Full Map Environment	100
Gambar 4.23 Linear Velocity Pengujian Tiga Titik FME with Obstacle	102
Gambar 4.24 Angular Velocity Pengujian Tiga Titik Full Map With Obstacle ..	102
Gambar 4.25 Tracking Error Pengujian Tiga Titik Full Map With Obstacle	103
Gambar 4.26 Trajectory Tracking Pengujian Tiga Titik FME With Obstacle	104
Gambar 4.27 Linear Velocity Pengujian Satu Titik Full Map Environment	105
Gambar 4.28 Angular Velocity Pengujian Satu Titik Full Map Environment	106
Gambar 4.29 Tracking Error Pengujian Satu Titik Full Map Environment	106
Gambar 4.30 Trajectory Tracking Pengujian Satu Titik Full Map Environment	107

**POLITEKNIK
NEGERI
JAKARTA**



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR TABEL

Tabel 2.1 Tabel State of the Art.....	6
Tabel 3.1 Spesifikasi Mekanis.....	48
Tabel 3.2 Spesifikasi Hardware.....	49
Tabel 3.3 Spesifikasi PCB.....	54
Tabel 3.4 Tabel Keterangan Modul Sistem Kontrol Tingkat Rendah	62
Tabel 3.5 Keterangan Gambar Tampak Depan Alat.....	64
Tabel 3.6 Tabel Keterangan Gambar Layer Bawah	65
Tabel 3.7 Tabel Keterangan Gambar Layer Tengah	65
Tabel 3.8 Tabel Keterangan Gambar Layer Atas Tampak Bawah.....	65
Tabel 3.9 Tabel Keterangan Gambar Layer Atas Tampak Atas.....	66
Tabel 3.10 Tabel Spesifikasi Robot Mobile IKIBOT.....	66
Tabel 3.11 Parameter Optimasi PSO.....	77
Tabel 3.12 Batas Pencarian Parameter PID.....	77
Tabel 3.13 Skenario Pengujian Sistem.....	78
Tabel 3.14 Parameter Evaluasi Kinerja Sistem	79
Tabel 4.1 Daftar Alat Pengujian	82
Tabel 4.2 PID Low-Level Pra-PSO.....	86
Tabel 4.3 PID Low-Level Pasca-PSO	89
Tabel 4.4 Navigation PID Pra-PSO.....	92
Tabel 4.5 Tabel Navigation PID After PSO Tuning	97
Tabel 4.6 Tabel Pengujian Tiga Titik Full Map Environment.....	101
Tabel 4.7 Tabel Pengujian Tiga Titik Full Map Environment With Obstacle	104
Tabel 4.8 Tabel Pengujian Satu Titik Full Map Environment.....	107
Tabel 4.9 Nilai PID Low-Level.....	108
Tabel 4.10 Perbandingan Kinerja PID Low-Level.....	108
Tabel 4.11 Nilai PID High-Level	110
Tabel 4.12 Perbandingan Kinerja PID High-Level	111
Tabel 4.13 Nilai PID yang Digunakan	113
Tabel 4.14 Perbandingan Kinerja PID IKIBOT Navigasi Tiga Titik	113



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR LAMPIRAN

Lampiran 1 Daftar Riwayat Hidup.....	121
Lampiran 2 Program Teensy 4.0	122
Lampiran 3 Program Kontrol PID.....	129
Lampiran 4 Dokumentasi Alat	134



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB I PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi robotika terutama pada model robot *mobile* saat ini memberikan pengaruh yang signifikan terhadap aktivitas manusia di berbagai bidang, terutama di sektor industri dan sosial. Robot *mobile* ini dituntut untuk bergerak menyesuaikan dengan kondisi lingkungan tempat robot *mobile* digunakan. Robot *mobile* dipengaruhi oleh kemampuan pengendalian respons sistemnya. Respons dari sistem kendali tersebut berperan penting dalam menjaga kestabilan, arah gerak dan kecepatan robot *mobile*.

Menurut Nahavandi et al. (2025), perkembangan robot *mobile* terus mengalami peningkatan terutama pada bidang navigasi, *path planning*, dan sistem pengendalian robot. Perkembangan tersebut dipengaruhi oleh kemampuan sistem dalam mengatur pergerakan robot secara stabil dan akurat. Sistem kontrol umpan balik berpengaruh terhadap pergerakan robot *mobile* supaya dapat bergerak dengan baik. Sistem tersebut diletakkan di dalam *microcontroller* yang mengatur respons sistem dengan suatu metode kendali.

Sistem metode kendali tersebut yang banyak digunakan pada robot *mobile* IKIBOT adalah metode *Proportional – Integral – Derivative* (PID). Metode tersebut dipilih dikarenakan mampu menghasilkan respons yang cukup baik dengan struktur kendali yang sederhana. Menurut Park (2025), sistem kendali pada robot *mobile* merupakan komponen penting yang menentukan performa robot secara keseluruhan, karena secara langsung memengaruhi mobilitas, akurasi pergerakan, serta kestabilan robot dalam mengikuti lintasan yang telah direncanakan. Selain itu, sistem kendali juga berperan dalam meminimalkan kesalahan pergerakan (*path-tracking-error*) sehingga robot *mobile* dapat bergerak menuju target secara stabil dan efisien. Metode PID telah mencakup kemampuan untuk memberikan respons yang stabil, meningkatkan akurasi pergerakan, serta mengurangi kesalahan pada proses navigasi robot *mobile*.



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Meskipun metode PID dikatakan telah dapat memberikan peningkatan performa secara signifikan yang membuat robot *mobile* IKIBOT dapat bergerak secara stabil dan efisien, namun proses penentuan parameter PID secara manual memiliki banyak kelemahan dalam menentukan parameter *proportional* (K_p), *integral* (K_i), dan *derivative* (K_d). Salah satu metode manual untuk menentukan parameter PID adalah dengan melakukan *trial and error*. Percobaan tersebut membutuhkan waktu yang cukup lama dan bergantung pada pengalaman pengguna. Selain itu, perubahan lingkungan berpengaruh pada performa kendali robot *mobile* IKIBOT. Hal tersebut dapat mengakibatkan respons kendali menjadi kurang optimal dan tidak stabil.

Menurut Reda et al. (2025), performa PID bergantung pada proses *tuning* parameter yang tepat, sedangkan metode *tuning* konvensional seperti Ziegler-Nichols masih menggunakan parameter statis dan sering memerlukan penyesuaian ulang ketika terjadi perubahan dinamika sistem atau gangguan eksternal. Selain itu, PID konvensional juga memiliki keterbatasan dalam melakukan penyesuaian parameter secara adaptif terhadap perubahan kondisi sistem secara *real-time*.

Oleh sebab itu, diperlukan metode *tuning* yang dapat menutupi kekurangan metode manual dan konvensional seperti Ziegler-Nichols. Metode tersebut harus dapat menghasilkan parameter PID yang lebih hemat waktu, tidak bergantung pada pengalaman pengguna dan lebih optimal ketika terjadi perubahan dinamika sistem. Salah satu metode tersebut yang paling banyak digunakan adalah *Particle Swarm Optimization* (PSO). Menurut Rahayu et al. (2022), *Particle Swarm Optimization* (PSO) merupakan salah satu metode optimasi yang digunakan untuk melakukan *tuning* parameter PID secara optimal dengan memanfaatkan pencarian solusi terbaik berdasarkan pergerakan partikel secara iteratif dan dinilai mampu menghasilkan parameter PID yang lebih stabil dibandingkan metode *trial and error* dan metode konvensional.

Metode PSO dilakukan secara otomatis hingga diperoleh performa sistem yang optimal dan mampu menghasilkan respons sistem yang lebih baik dengan nilai RMSE, MAE, ITAE, *Final Position Error* dan waktu tempuh yang lebih optimal dibandingkan metode *tuning* manual. Selain itu, metode PSO juga mampu



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

membantu proses pencarian parameter PID secara lebih efisien sehingga proses *tuning* dapat dilakukan dengan waktu yang lebih singkat dan hasil yang lebih akurat. Robot *mobile* IKIBOT dipilih sebagai objek penelitian dengan penerapan PID dengan proses *tuning* menggunakan metode PSO karena memiliki sistem pergerakan yang sesuai untuk pengujian pada proses sistem kendalinya. Selain itu, robot *mobile* IKIBOT memiliki struktur perangkat keras yang mendukung implementasi sistem kontrol berbasis Raspberry Pi 4B dan Teensy yang memungkinkan implementasi algoritma kendali, pengambilan data, serta pengujian respons sistem dilakukan secara langsung dan *real-time*. Dengan penerapan metode tersebut, diharapkan sistem kendali yang dihasilkan mampu meningkatkan kestabilan, akurasi pergerakan, serta respons navigasi robot *mobile* IKIBOT secara lebih optimal.

Meskipun PID dan PSO telah banyak diteliti dalam aplikasi robot *mobile*, penerapan pengendali PID yang disesuaikan dengan PSO pada robot *mobile* IKIBOT untuk evaluasi kinerja navigasi berbasis titik jalan menggunakan RMSE, MAE, ITAE, *Final Position Error*, dan Waktu Tempuh belum diteliti secara mendalam. Oleh karena itu, penelitian ini bertujuan untuk mengisi celah tersebut dengan menerapkan dan mengevaluasi pengendali navigasi PID yang dioptimalkan dengan PSO pada platform IKIBOT.

1.2 Perumusan Masalah

Berdasarkan pemaparan dalam latar belakang dapat diambil perumusan masalah pada penelitian ini sebagai berikut:

- 1) Bagaimana merancang dan mengimplementasikan sistem kendali robot *mobile* IKIBOT berbasis *Proportional Integral Derivative* (PID) yang mengintegrasikan kendali *high-level* untuk pengaturan posisi dan orientasi (θ) serta kendali *low-level* untuk pengaturan kecepatan motor?
- 2) Bagaimana mengoptimalkan parameter PID (K_p , K_i , dan K_d) menggunakan metode *Particle Swarm Optimization* (PSO) agar robot *mobile* IKIBOT dapat bergerak stabil dan mencapai target navigasi dengan akurat?



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- 3) Bagaimana performa sistem kendali PID pada Robot *mobile* IKIBOT berdasarkan parameter pengujian seperti RMSE, MAE, ITAE, *Final Position Error* dan waktu tempuh?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas maka diperoleh tujuan dari penelitian sebagai berikut:

- 1) Merancang dan mengimplementasikan sistem kendali robot *mobile* IKIBOT berbasis PID yang mengintegrasikan kendali *high-level* untuk pengaturan posisi dan orientasi (θ) serta kendali *low-level* untuk pengaturan kecepatan motor.
- 2) Mengoptimalkan parameter PID (K_p , K_i , dan K_d) menggunakan *Particle Swarm Optimization* (PSO) untuk meningkatkan performa navigasi robot *mobile* IKIBOT
- 3) Menguji performa sistem kendali PID pada Robot *mobile* IKIBOT berdasarkan parameter pengujian seperti RMSE, MAE, ITAE, *Final Position Error* dan waktu tempuh.

1.4 Batasan Masalah

Adapun batasan masalah dari penelitian ini sebagai berikut:

- 1) Robot yang digunakan adalah robot *mobile* IKIBOT.
- 2) Penelitian ini berfokus pada penerapan sistem kendali PID pada robot *mobile* IKIBOT.
- 3) Sistem kendali PID yang diteliti merupakan kendali *low-level* dan *high-level* dengan optimasi parameter menggunakan PSO.
- 4) Pengujian dilakukan pada lintasan yang telah ditentukan dan parameter yang diuji meliputi RMSE, MAE, ITAE, *Final Position Error* dan waktu tempuh.
- 5) Penelitian ini merupakan bagian dari proyek tugas akhir kelompok IKIBOT dan hanya berfokus pada perancangan, implementasi, optimasi, dan evaluasi sistem kendali navigasi berbasis PID menggunakan metode PSO.



1.5 Luaran

Luaran yang diharapkan dari penelitian ini untuk tugas akhir sebagai berikut:

- 1) Laporan skripsi.
- 2) Publikasi jurnal.
- 3) Purwarupa sistem kendali Robot *mobile* IKIBOT berbasis *Proportional Integral Derivative* (PID).
- 4) Sistem navigasi robot *mobile* IKIBOT berbasis PID yang dioptimasi menggunakan *Particle Swarm Optimization* (PSO), terdiri atas perangkat keras robot *mobile*, perangkat lunak berbasis ROS Noetic, serta algoritma kendali PID *Low-Level* dan PID *High-Level* yang telah diimplementasikan dan diuji pada berbagai skenario navigasi.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem kendali pada robot *mobile* IKIBOT, diperoleh beberapa kesimpulan sebagai berikut:

- 1) Sistem kendali robot *mobile* IKIBOT berbasis *Proportional Integral Derivative* (PID) berhasil dirancang dan diimplementasikan dengan mengintegrasikan kendali PID *High-Level* dan PID *Low-Level*. Kendali PID *High-Level* digunakan untuk menghasilkan kecepatan linear dan kecepatan sudut berdasarkan kesalahan posisi dan orientasi robot terhadap target navigasi, sedangkan kendali PID *Low-Level* digunakan untuk mengatur kecepatan motor berdasarkan umpan balik enkoder agar mampu mengikuti nilai referensi yang diberikan. Integrasi kedua lapisan kendali tersebut mampu menghasilkan sistem navigasi yang bekerja secara terkoordinasi sehingga robot dapat bergerak menuju target dengan stabil dan mengikuti lintasan yang telah direncanakan.
- 2) Parameter PID pada kendali *High-Level* maupun *Low-Level* berhasil dioptimalkan menggunakan metode *Particle Swarm Optimization* (PSO). Proses optimasi dilakukan dengan menggunakan fungsi objektif *Integral Time Absolute Error* (ITAE) sehingga diperoleh kombinasi parameter PID yang mampu meningkatkan performa sistem. Hasil optimasi menunjukkan bahwa penggunaan PSO dapat membantu memperoleh parameter kendali yang lebih optimal dibandingkan parameter awal, sehingga respons sistem menjadi lebih stabil, kesalahan pelacakan menjadi lebih kecil, dan kemampuan navigasi robot meningkat pada berbagai kondisi pengujian.
- 3) Berdasarkan hasil evaluasi kinerja menggunakan parameter *Root Mean Square Error* (RMSE), *Mean Absolute Error* (MAE), *Integral Time Absolute Error* (ITAE), *Final Position Error* (FPE), dan waktu tempuh, sistem kendali PID yang dioptimasi menggunakan PSO menunjukkan performa yang baik pada robot *mobile* IKIBOT. Robot mampu



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

mencapai target navigasi pada skenario satu titik maupun tiga titik, baik pada lingkungan tanpa hambatan maupun dengan hambatan. Hasil pengujian menunjukkan bahwa integrasi kendali PID dan metode optimasi PSO mampu meningkatkan kestabilan pergerakan, akurasi pencapaian target, serta efisiensi navigasi robot mobile IKIBOT secara keseluruhan.

5.2 Saran

Berdasarkan temuan penelitian dan hasil pengujian yang diuraikan dalam studi ini, diajukan beberapa rekomendasi untuk penelitian lanjutan guna mengembangkan lebih lanjut dan meningkatkan kinerja sistem kendali navigasi berbasis PID pada robot *mobile* IKIBOT:

- 1) Melakukan pengujian berulang pada setiap skenario navigasi sehingga dapat dihitung nilai rata-rata, simpangan baku, dan tingkat konsistensi performa sistem untuk meningkatkan validitas hasil penelitian.
- 2) Mengembangkan metode optimasi parameter PID dengan membandingkan *Particle Swarm Optimization* (PSO) terhadap metode optimasi berbasis metaheuristik lainnya, seperti *Genetic Algorithm* (GA), *Grey Wolf Optimizer* (GWO), atau *Ant Colony Optimization* (ACO), untuk memperoleh parameter pengendali yang lebih optimal pada berbagai kondisi lingkungan.
- 3) Menambahkan variasi skenario pengujian yang lebih kompleks, seperti penggunaan rintangan dinamis, perubahan tata letak lingkungan secara *real-time*, serta lintasan yang lebih panjang dan kompleks untuk mengevaluasi ketahanan sistem navigasi terhadap perubahan kondisi operasional yang lebih beragam.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR PUSTAKA

- Abdrabbo, M. S., Ibrahim, A. A., & El-samanty, M. (2026). High-Level Control and Navigation of Autonomous Mobile Robots Using A* and RRT* Algorithms. *International Journal of Engineering and Applied Sciences-October 6 University*, 3(1), 45–59. <https://doi.org/10.21608/ijeasou.2025.437120.1071>
- Adiuku, N., Avdelidis, N. P., Tang, G., Plastropoulos, A., & Diallo, Y. (2024). Mobile Robot Obstacle Detection and Avoidance with NAV-YOLO. *International Journal of Mechanical Engineering and Robotics Research*, 13(2), 219–226. <https://doi.org/10.18178/ijmerr.13.2.219-226>
- Aremu, M. B., Ahmed, G., Elferik, S., & Saif, A.-W. A. (2026). Autonomous Mobile Robot Path Planning Techniques—A Review: Metaheuristic and Cognitive Techniques. *Robotics*, 15(1), 23. <https://doi.org/10.3390/robotics15010023>
- Gameiro, T., Pereira, T., Moghadaspoura, H., Di Giorgio, F., Viegas, C., Ferreira, N., Ferreira, J., Soares, S., & Valente, A. (2025). Evaluation of PID-Based Algorithms for UGVs. *Algorithms*, 18(2). <https://doi.org/10.3390/a18020063>
- He, Q., Wang, Z., Li, K., Zhang, Y., & Li, M. (2025). Research on autonomous navigation of mobile robots based on IA-DWA algorithm. *Scientific Reports*, 15(1). <https://doi.org/10.1038/s41598-024-84858-3>
- Jaramillo-Martínez, R., Chavero-Navarrete, E., & Ibarra-Pérez, T. (2026). Hierarchical Autonomous Navigation for Differential-Drive Mobile Robots Using Deep Learning, Reinforcement Learning, and Lyapunov-Based Trajectory Control. *Technologies*, 14(2). <https://doi.org/10.3390/technologies14020125>
- Kumar, A., & Kim, S. J. (2025). Real-Time Map Generation by VLP-16 LiDAR sensor in GPS-denied environment. *Peninsulares International Journal of Innovations and Sustainability*, 3(1). <https://www.bpsu.edu.ph/PIJIS>
- Mahdi, S. M., Abdulkareem, A. I., & Humaidi, A. J. (2025). Improved Tracking Accuracy of Par-4 Delta Parallel Robot Using Optimized FOPID Control with PSO Technique. *Journal of Robotics and Control (JRC)*, 6(4), 1721–1728. <https://doi.org/10.18196/jrc.v6i4.26607>
- Matei, A., Precup, S. A., Circa, D., Gellert, A., & Zamfirescu, C. B. (2023). Estimating Travel Time for Autonomous Mobile Robots through Long Short-Term Memory. *Mathematics*, 11(7). <https://doi.org/10.3390/math11071723>
- Nahavandi, S., Alizadehsani, R., Nahavandi, D., Mohamed, S., Mohajer, N., Rokonzaman, M., & Hossain, I. (2025). A Comprehensive Review on Autonomous Navigation. *ACM Computing Surveys*, 57(9). <https://doi.org/10.1145/3727642>



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- Nwankwo, L., Fritze, C., Bartsch, K., & Rueckert, E. (2023). ROMR: A ROS-based open-source mobile robot. *HardwareX*, 14, e00426. <https://doi.org/10.17605/OSF.IO/K83X7>
- Park, J. H. (2025). Optimal Controller Design for a Mobile Robot Using Genetic Algorithm and Adaptive PID Controller. *IEEE Access*, 13, 86167–86184. <https://doi.org/10.1109/ACCESS.2025.3570472>
- Rahayu, E. S., Ma'arif, A., & Cakan, A. (2022). Particle Swarm Optimization (PSO) Tuning of PID Control on DC Motor. *International Journal of Robotics and Control Systems*, 2(2), 435–447. <https://doi.org/10.31763/ijres.v2i2.476>
- Reda, M., Onsy, A., Haikal, A. Y., & Ghanbari, A. (2025). ROS-Based Autonomous Driving System with Enhanced Path Planning Node Validated in Chicane Scenarios. *Actuators*, 14(8). <https://doi.org/10.3390/act14080375>
- Salwa, M., & Krzysztofik, I. (2025). Influence of Control System Architecture on Mobile Robot Stability and Performance. *Sensors*, 25(23). <https://doi.org/10.3390/s25237353>
- Setiawan, F. A., Marzani, G., Arkana, D., Budianto, S., & Hadinata, J. (2026). PID-Based Motion Control of a Differential Drive Robot: A Simulation Study in CoppeliaSim. In *Indonesian Journal of Engineering and Technology (INAJET)* (Vol. 8, Number 2). <https://journal.unesa.ac.id/index.php/inajet>
- Soebhakti, H., & Pangantar, R. H. (2024). Simulation of Mobile Robot Navigation System using Hector SLAM on ROS. *Jurnal Integrasi*, 16(1), 11–20. <https://doi.org/10.30871/ji.v16i1.5755>
- Temsamani, P., & Gadner, T. (2024). Power Distribution System for Robotics Applications. In *Online Conference Proceedings Mechatronics: Graduates of 2024*. Management Center Innsbruck (MCI). <https://sakai.mci4me.at/>
- Tinoco, V., Silva, M. F., Santos, F. N., Morais, R., Magalhães, S. A., & Oliveira, P. M. (2025). A review of advanced controller methodologies for robotic manipulators. *International Journal of Dynamics and Control*, 13(1). <https://doi.org/10.1007/s40435-024-01533-1>
- Varela-Aldás, J., & Palacios-Navarro, G. (2024). A ROS-Based Open Tool for Controlling an Educational Mobile Robot. *International Journal of Online and Biomedical Engineering*, 20(1), 23–39. <https://doi.org/10.3991/ijoe.v20i01.46573>
- Waga, A., Benhlila, S., Bekri, A., Abdouni, J., & Saber, F. Z. (2025). A survey on autonomous navigation for mobile robots: From traditional techniques to deep learning and large language models. In *Journal of King Saud University - Computer and Information Sciences* (Vol. 37, Number 7). Springer International Publishing. <https://doi.org/10.1007/s44443-025-00216-x>



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Wang, Z., Wang, S., Xie, Y., Xiong, T., & Wang, C. (2025). Autonomous Navigation of Mobile Robots: A Hierarchical Planning–Control Framework with Integrated DWA and MPC. *Sensors*, 25(7). <https://doi.org/10.3390/s25072014>

Yankova, A., Gamberov, T., & Tsoy, T. (2024). Implementation of Alg1 and Alg2 Path Planning Algorithms for Mobile Robots Using ROS Noetic. In *The 2024 International Conference on Artificial Life and Robotics (ICAROB2024)* (pp. 276–280). <https://doi.org/10.5954/icarob.2024.os10-3>.

Zhang, M., Chen, N., Wang, H., Qiu, J., Han, Z., Ren, Q., Xu, C., Gao, F., & Cao, Y. (2025). Universal Trajectory Optimization Framework for Differential Drive Robot Class. *IEEE Transactions on Automation Science and Engineering*. <https://doi.org/10.1109/TASE.2025.3550676>

Zhou, Y., He, X., Shao, F., & Zhang, X. (2024). Research on the Optimization of the PID Control Method for an EOD Robotic Manipulator Using the PSO Algorithm for BP Neural Networks. *Actuators*, 13(10). <https://doi.org/10.3390/act13100386>



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LAMPIRAN

Lampiran 1 Daftar Riwayat Hidup



Penulis bernama Yosef Dwi Krisnata, anak pertama dari dua bersaudara dan lahir di Bogor pada 15 Maret 2002. Latar belakang pendidikan formal penulis yaitu Sekolah Dasar di SDS Eka Wijaya, lulus pada tahun 2014. Melanjutkan ke Sekolah Menengah Pertama di SMPN 1 Cibinong, lulus pada tahun 2017. Kemudian melanjutkan ke Sekolah Menengah Atas di SMKN 1 Cibinong, lulus pada tahun 2021. Setelah itu penulis bekerja sebagai *Electrical Engineering* sampai pada tahun 2022. Kemudian penulis melanjutkan studi ke jenjang perkuliahan Sarjana Terapan (S.Tr) di Politeknik Negeri Jakarta Jurusan Teknik Elektro Program Studi Instrumentasi dan Kontrol Industri sejak tahun 2022. Penulis dapat dihubungi melalui email yosef.dkrisnata@gmail.com

**POLITEKNIK
NEGERI
JAKARTA**



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Lampiran 2 Program Teensy 4.0

```
#include <ros.h>
#include <geometry_msgs/Twist.h>
#include <geometry_msgs/Vector3.h>
#include <nav_msgs/Odometry.h>
#include <sensor_msgs/Imu.h>
#include <std_msgs/Empty.h>
#include <tf/transform_broadcaster.h>
#include <Wire.h>
#include <Encoder.h>
#include <MPU6050.h>

// TRIPLE PORT SERIAL - Teensy 4.0
class NewHardware0 : public ArduinoHardware {
public:
  NewHardware0() : ArduinoHardware(&Serial, 115200) {}
};
ros::NodeHandle_<NewHardware0> nh_teleop;

class NewHardware1 : public ArduinoHardware {
public:
  NewHardware1() : ArduinoHardware() {}
  void init() { SerialUSB1.begin(115200); }
  int read() { return SerialUSB1.read(); }
  void write(uint8_t* d, int l) { for (int i = 0; i < l; i++) SerialUSB1.write(d[i]); }
};
ros::NodeHandle_<NewHardware1> nh_odom;

class NewHardware2 : public ArduinoHardware {
public:
  NewHardware2() : ArduinoHardware() {}
  void init() { SerialUSB2.begin(115200); }
  int read() { return SerialUSB2.read(); }
  void write(uint8_t* d, int l) { for (int i = 0; i < l; i++) SerialUSB2.write(d[i]); }
};
ros::NodeHandle_<NewHardware2> nh_imu;

// PIN DEFINITIONS & KONSTANTA KINEMATIKA
#define ENC_L_A 2
#define ENC_L_B 3
#define ENC_R_A 4
#define ENC_R_B 5
#define IN1_L 8
#define IN2_L 11
#define IN3_R 6
#define IN4_R 7
#define ENA 10
#define ENB 9
#define MPU_INT_PIN 12

const float WHEEL_DIAMETER = 0.07f;
const float WHEEL_BASE = 0.225f;
const int CPR = 2059;
const float METERS_PER_TICK = (PI * WHEEL_DIAMETER) / CPR;
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

const int ARAH_KIRI = 1;
const int ARAH_KANAN = -1;
const int ARAH_MAJU = -1;

const int PWM_DEADBAND = 40;
const float MIN_ROT_VEL = 0.35f;
const float PELAMBAT_VAR_L = 0.90f;

// KONSTANTA POWER MONITORING
const int PIN_ACS = A0;
const float V_REF = 3.3f;
const float ADC_RES = 1023.0f;
const float V_BATTERY = 12.0f;
const float ACS_OFFSET = 1.258f;
const float SENSITIVITY = 0.033f;

float total_joules = 0.0f;
geometry_msgs::Vector3 power_msg;
ros::Publisher power_pub("power_consumption", &power_msg);

// VARIABEL STATE & PID
const unsigned long CMD_TIMEOUT_MS = 500;
const unsigned long CONTROL_PERIOD_MS = 50;
unsigned long last_control_time = 0;

Encoder encLeft(ENC_L_A, ENC_L_B);
Encoder encRight(ENC_R_A, ENC_R_B);

long oldLeft = 0, oldRight = 0;
float linear_x = 0.0f, angular_z = 0.0f;
double x_pos = 0.0, y_pos = 0.0, yaw = 0.0;
unsigned long last_cmd_time = 0;

float Kp = 260.0f, Ki = 15.0f, Kd = 0.5f;
float integral_L = 0.0f, integral_R = 0.0f;
float prev_error_L = 0.0f, prev_error_R = 0.0f;
const float INTEGRAL_MAX = 80.0f;

MPU6050 mpu;
int16_t ax, ay, az, gx, gy, gz;
int16_t ax_offset = 0, ay_offset = 0, az_offset = 0;
int16_t gx_offset = 0, gy_offset = 0, gz_offset = 0;

const float ACCEL_SCALE = 9.80665f / 16384.0f;
const float GYRO_SCALE = (PI / 180.0f) / 131.0f;

nav_msgs::Odometry odom_msg;
ros::Publisher odom_pub("odom", &odom_msg);
sensor_msgs::Imu imu_msg;
ros::Publisher imu_pub("imu/data", &imu_msg);
tf::TransformBroadcaster broadcaster;

// WATCHDOG VARIABLES
bool was_connected = false;
unsigned long disconnect_timer = 0;

// HELPER FUNCTIONS
geometry_msgs::Quaternion quaternionFromYaw(double yaw_rad) {

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

geometry_msgs::Quaternion q;
q.x = 0.0; q.y = 0.0;
q.z = sin(yaw_rad / 2.0); q.w = cos(yaw_rad / 2.0);
return q;
}

float normalizeAngle(float angle) {
while (angle > PI) angle -= 2.0f * PI;
while (angle < -PI) angle += 2.0f * PI;
return angle;
}

// ROS CALLBACKS
void cmdVelCallback(const geometry_msgs::Twist& msg) {
linear_x = msg.linear.x;
angular_z = msg.angular.z;
last_cmd_time = millis();
}
ros::Subscriber<geometry_msgs::Twist> sub_cmd("cmd_vel", cmdVelCallback);

void pidCallback(const geometry_msgs::Vector3& msg) {
Kp = msg.x; Ki = msg.y; Kd = msg.z;
integral_L = integral_R = 0.0f;
prev_error_L = prev_error_R = 0.0f;
}
ros::Subscriber<geometry_msgs::Vector3> sub_pid("tune_pid", pidCallback);

void pwmCallback(const std_msgs::Int16& pwm_msg) {
int target_pwm = pwm_msg.data;
// Pastikan logika kode Anda me-override PID saat topik ini menerima data
analogWrite(PIN_PWM_KIRI, target_pwm);
analogWrite(PIN_PWM_KANAN, target_pwm);
}
ros::Subscriber<std_msgs::Int16> sub_pwm("cmd_pwm", &pwmCallback);

void resetOdomCallback(const std_msgs::Empty& msg) {
x_pos = 0.0; y_pos = 0.0; yaw = 0.0;
encLeft.write(0); encRight.write(0);
oldLeft = 0; oldRight = 0;
}
ros::Subscriber<std_msgs::Empty> sub_reset("reset_odom", resetOdomCallback);

// MOTOR DRIVER
void setMotor(int pL, int pR) {
if (pL != 0 && abs(pL) < PWM_DEADBAND) pL = (pL > 0) ? PWM_DEADBAND : -
PWM_DEADBAND;
if (pR != 0 && abs(pR) < PWM_DEADBAND) pR = (pR > 0) ? PWM_DEADBAND : -
PWM_DEADBAND;

if (pL > 0) { digitalWrite(IN1_L, HIGH); digitalWrite(IN2_L, LOW); }
else if (pL < 0) { digitalWrite(IN1_L, LOW); digitalWrite(IN2_L, HIGH); }
else { digitalWrite(IN1_L, LOW); digitalWrite(IN2_L, LOW); }

if (pR > 0) { digitalWrite(IN3_R, HIGH); digitalWrite(IN4_R, LOW); }
else if (pR < 0) { digitalWrite(IN3_R, LOW); digitalWrite(IN4_R, HIGH); }
else { digitalWrite(IN3_R, LOW); digitalWrite(IN4_R, LOW); }

analogWrite(ENA, constrain(abs(pL), 0, 255));

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

analogWrite(ENB, constrain(abs(pR), 0, 255));
}

// SETUP
void setup() {
  pinMode(IN1_L, OUTPUT); pinMode(IN2_L, OUTPUT);
  pinMode(IN3_R, OUTPUT); pinMode(IN4_R, OUTPUT);
  pinMode(ENA, OUTPUT); pinMode(ENB, OUTPUT);
  setMotor(0, 0);

  pinMode(PIN_ACS, INPUT);

  Wire.setSDA(18); Wire.setSCL(19);
  Wire.begin(); Wire.setClock(400000);

  pinMode(MPU_INT_PIN, INPUT);
  mpu.initialize();
  mpu.setFullScaleAccelRange(MPU6050_ACCEL_FS_2);
  mpu.setFullScaleGyroRange(MPU6050_GYRO_FS_250);

  nh_teleop.initNode();
  nh_teleop.subscribe(sub_cmd);
  nh_teleop.subscribe(sub_pid);
  nh_teleop.subscribe(sub_reset);

  nh_odom.initNode();
  broadcaster.init(reinterpret_cast<ros::NodeHandle*>(nh_odom));
  nh_odom.advertise(odom_pub);
  nh_odom.advertise(power_pub);

  nh_imu.initNode();
  nh_imu.advertise(imu_pub);

  for (int i = 0; i < 36; i++) {
    odom_msg.pose.covariance[i] = 0.0; odom_msg.twist.covariance[i] = 0.0;
  }
  odom_msg.pose.covariance[0] = 0.001; odom_msg.pose.covariance[7] = 0.001;
  odom_msg.pose.covariance[35] = 0.010;
  odom_msg.twist.covariance[0] = 0.001; odom_msg.twist.covariance[35] = 0.010;

  for (int i = 0; i < 9; i++) {
    imu_msg.orientation_covariance[i] = 0.0; imu_msg.angular_velocity_covariance[i] = 0.0;
    imu_msg.linear_acceleration_covariance[i] = 0.0;
  }
  imu_msg.orientation_covariance[0] = -1;
  imu_msg.angular_velocity_covariance[0] = 0.02; imu_msg.angular_velocity_covariance[4] =
0.02; imu_msg.angular_velocity_covariance[8] = 0.02;
  imu_msg.linear_acceleration_covariance[0] = 0.04;
  imu_msg.linear_acceleration_covariance[4] = 0.04; imu_msg.linear_acceleration_covariance[8]
= 0.04;

  last_cmd_time = millis();
  last_control_time = millis();
}

// LOOP UTAMA
void loop() {
  nh_teleop.spinOnce();

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

nh_odom.spinOnce();
nh_imu.spinOnce();

// AUTO-REBOOT WATCHDOG (PENGANTI CABUT-COLOK POWER)
bool is_connected = nh_teleop.connected() || nh_odom.connected() || nh_imu.connected();

if (is_connected) {
    was_connected = true;
    disconnect_timer = 0;
} else if (was_connected) {
    if (disconnect_timer == 0) disconnect_timer = millis();
    if (millis() - disconnect_timer > 3000) {
        SCB_AIRCR = 0x05FA0004;
    }
}

unsigned long now = millis();
if (now - last_control_time < CONTROL_PERIOD_MS) return;

float dt = (now - last_control_time) / 1000.0f;
last_control_time = now;
if (dt <= 0.005f || dt > 0.5f) return;

if (now - last_cmd_time > CMD_TIMEOUT_MS) {
    linear_x = 0.0f;
    angular_z = 0.0f;
}

// PENGGUNAAN ENERGI
int raw_adc = analogRead(PIN_ACS);
float v_out = (raw_adc * V_REF) / ADC_RES;

float current_amps = ((v_out - ACS_OFFSET) / SENSITIVITY) * -1.0f;
if (abs(current_amps) < 0.2f) current_amps = 0.0f;

float power_watts = V_BATTERY * current_amps;
total_joules += power_watts * dt;

power_msg.x = current_amps;
power_msg.y = power_watts;
power_msg.z = total_joules;

if (nh_odom.connected()) power_pub.publish(&power_msg);

// ODOMETRI & KINEMATIKA
long newLeft = encLeft.read();
long newRight = -encRight.read();

float dLeft = (newLeft - oldLeft) * METERS_PER_TICK;
float dRight = (newRight - oldRight) * METERS_PER_TICK;
oldLeft = newLeft; oldRight = newRight;

float dCenter = (dLeft + dRight) / 2.0f;
float dTheta = (dRight - dLeft) / WHEEL_BASE;

x_pos += dCenter * cos(yaw + dTheta / 2.0f);
y_pos += dCenter * sin(yaw + dTheta / 2.0f);
yaw += dTheta; yaw = normalizeAngle(yaw);

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

float actual_vL = dLeft / dt;
float actual_vR = dRight / dt;

geometry_msgs::TransformStamped tf_msg;
tf_msg.header.stamp = nh_odom.now();
tf_msg.header.frame_id = "odom"; tf_msg.child_frame_id = "base_link";
tf_msg.transform.translation.x = x_pos;
tf_msg.transform.translation.y = y_pos; tf_msg.transform.translation.z = 0.0;
tf_msg.transform.rotation = quaternionFromYaw(yaw);

odom_msg.header.stamp = nh_odom.now();
odom_msg.header.frame_id = "odom"; odom_msg.child_frame_id = "base link";
odom_msg.pose.pose.position.x = x_pos; odom_msg.pose.pose.position.y = y_pos;
odom_msg.pose.pose.position.z = 0.0;
odom_msg.pose.pose.orientation = quaternionFromYaw(yaw);
odom_msg.twist.twist.linear.x = dCenter / dt; odom_msg.twist.twist.linear.y = 0.0;
odom_msg.twist.twist.angular.z = dTheta / dt;

if (nh_odom.connected()) {
    broadcaster.sendTransform(tf_msg);
    odom_pub.publish(&odom_msg);
}

// IMU PROCESSING
mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
ax -= ax_offset; ay -= ay_offset; az -= az_offset;
gx -= gx_offset; gy -= gy_offset; gz -= gz_offset;

imu_msg.header.stamp = nh_imu.now(); imu_msg.header.frame_id = "imu_link";
imu_msg.linear_acceleration.x = ax * ACCEL_SCALE; imu_msg.linear_acceleration.y = ay *
ACCEL_SCALE; imu_msg.linear_acceleration.z = az * ACCEL_SCALE;
imu_msg.angular_velocity.x = gx * GYRO_SCALE; imu_msg.angular_velocity.y = gy *
GYRO_SCALE; imu_msg.angular_velocity.z = gz * GYRO_SCALE;
imu_msg.orientation.w = 1.0; imu_msg.orientation.x = 0.0; imu_msg.orientation.y = 0.0;
imu_msg.orientation.z = 0.0;

if (nh_imu.connected()) imu_pub.publish(&imu_msg);

// PID & KONTROL MOTOR
float v = linear_x;
float w = angular_z;

float target_vL = ((v * ARAH_MAJU) - (w * WHEEL_BASE / 2.0f));
float target_vR = ((v * ARAH_MAJU) + (w * WHEEL_BASE / 2.0f));

if (v == 0.0f && w != 0.0f) {
    if (abs(target_vL) < MIN_ROT_VEL) target_vL = (target_vL >= 0.0f) ? MIN_ROT_VEL : -
MIN_ROT_VEL;
    if (abs(target_vR) < MIN_ROT_VEL) target_vR = (target_vR >= 0.0f) ? MIN_ROT_VEL : -
MIN_ROT_VEL;
}

if (v == 0.0f && w == 0.0f) {
    integral_L = integral_R = 0.0f;
    prev_error_L = prev_error_R = 0.0f;
    setMotor(0, 0);
    return;
}

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

}

float error_L = target_vL - actual_vL;
float error_R = target_vR - actual_vR;

integral_L = constrain(integral_L + error_L * dt, -INTEGRAL_MAX, INTEGRAL_MAX);
integral_R = constrain(integral_R + error_R * dt, -INTEGRAL_MAX, INTEGRAL_MAX);

float derivative_L = (error_L - prev_error_L) / dt;
float derivative_R = (error_R - prev_error_R) / dt;

float out_L = (Kp * error_L) + (Ki * integral_L) + (Kd * derivative_L);
float out_R = (Kp * error_R) + (Ki * integral_R) + (Kd * derivative_R);

int pwm_L = constrain((int)((out_L * PELAMBAT_VAR_L) * ARAH_KIRI), -255, 255);
int pwm_R = constrain((int)(out_R * ARAH_KANAN), -255, 255);

prev_error_L = error_L; prev_error_R = error_R;

setMotor(pwm_L, pwm_R);
}

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Lampiran 3 Program Kontrol PID

```
import rospy
from geometry_msgs.msg import Twist, PoseStamped, Vector3
from nav_msgs.msg import Odometry, Path
from sensor_msgs.msg import LaserScan
from actionlib_msgs.msg import GoalID
import math

class PIDController:
    def __init__(self, kp, ki, kd, integral_max=1.0):
        self.kp = kp
        self.ki = ki
        self.kd = kd
        self.integral_max = integral_max
        self.prev_error = 0.0
        self.integral = 0.0

    def compute(self, error, dt):
        if dt <= 0.0: return 0.0
        self.integral += error * dt
        self.integral = max(min(self.integral, self.integral_max), -self.integral_max)
        derivative = (error - self.prev_error) / dt
        output = (self.kp * error) + (self.ki * self.integral) + (self.kd * derivative)
        self.prev_error = error
        return output

    def reset(self):
        self.prev_error = 0.0
        self.integral = 0.0

class IkibotNavigator:
    def __init__(self):
        rospy.init_node('ikibot_pure_pid_navigator', anonymous=True)

        # 1. LOW-LEVEL PID
        self.ll_kp = 274.0
        self.ll_ki = 20.0
        self.ll_kd = 0.5
        self.tune_pid_pub = rospy.Publisher('/tune_pid', Vector3, queue_size=10)

        # 2. HIGH-LEVEL PID
        self.linear_pid = PIDController(kp=1.70, ki=0.12, kd=0.33, integral_max=1.0)
        self.angular_pid = PIDController(kp=2.50, ki=0.15, kd=0.10, integral_max=1.5)

        # KINEMATIKA
        self.v_max = 0.45
        self.v_min = -0.45
        self.w_max = 1.3
        self.w_min = -1.3
        self.min_w_abs = 0.2
        self.acc_x = 1.5
        self.acc_theta = 4.0

        self.lookahead_distance = 0.28
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

self.current_pose = None
self.global_path = []
self.scan_data = None
self.last_v = 0.0
self.last_w = 0.0
self.last_time = rospy.Time.now().to_sec()

# Komunikasi ROS
self.cmd_vel_pub = rospy.Publisher('/cmd_vel', Twist, queue_size=10)
self.cancel_pub = rospy.Publisher('/move_base/cancel', GoalID, queue_size=10)

# PUBLISHER UNTUK MENGHAPUS GARIS RVIZ
self.clear_global_pub = rospy.Publisher('/move_base/DWAPlanerROS/global_plan', Path,
queue_size=2)
self.clear_local_pub = rospy.Publisher('/move_base/DWAPlanerROS/local_plan', Path,
queue_size=2)

rospy.Subscriber('/odom', Odometry, self.odom_callback)
rospy.Subscriber('/scan', LaserScan, self.scan_callback)
rospy.Subscriber('/move_base/DWAPlanerROS/global_plan', Path, self.path_callback)
rospy.Subscriber('/tune_nav_linear', Vector3, self.tune_linear_cb)
rospy.Subscriber('/tune_nav_angular', Vector3, self.tune_angular_cb)

self.rate = rospy.Rate(10)
rospy.sleep(1.0)
self.send_ll_pid_to_tensy()

rospy.loginfo("\n=====")
rospy.loginfo(" [ NAVIGATOR PID ] AKTIF! ")
rospy.loginfo("=====\\n")

def send_ll_pid_to_tensy(self):
    pid_msg = Vector3(x=self.ll_kp, y=self.ll_ki, z=self.ll_kd)
    for _ in range(3):
        self.tune_pid_pub.publish(pid_msg)
        rospy.sleep(0.2)

def tune_linear_cb(self, msg):
    self.linear_pid.kp, self.linear_pid.ki, self.linear_pid.kd = msg.x, msg.y, msg.z
    self.linear_pid.reset()

def tune_angular_cb(self, msg):
    self.angular_pid.kp, self.angular_pid.ki, self.angular_pid.kd = msg.x, msg.y, msg.z
    self.angular_pid.reset()

def odom_callback(self, msg):
    self.current_pose = msg.pose.pose

def scan_callback(self, msg):
    self.scan_data = msg

def path_callback(self, msg):
    self.global_path = [(p.pose.position.x, p.pose.position.y) for p in msg.poses]
    self.linear_pid.reset()
    self.angular_pid.reset()

def get_yaw_from_quaternion(self, q):

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

return math.atan2(2 * (q.w * q.z + q.x * q.y), 1 - 2 * (q.y * q.y + q.z * q.z))

def get_current_yaw(self):
    if self.current_pose is None: return 0.0
    return self.get_yaw_from_quaternion(self.current_pose.orientation)

def get_lookahead_target(self):
    if not self.global_path or self.current_pose is None: return None
    cx = self.current_pose.position.x
    cy = self.current_pose.position.y
    target_pt = self.global_path[-1]

    for pt in self.global_path:
        dist = math.sqrt((pt[0] - cx)**2 + (pt[1] - cy)**2)
        if dist > self.lookahead_distance:
            target_pt = pt
            break
    return target_pt

def check_obstacles(self):
    if self.scan_data is None: return False, None
    threshold = 0.35
    front_min, left_min, right_min = float('inf'), float('inf'), float('inf')

    for i, r in enumerate(self.scan_data.ranges):
        if not (0.18 < r < 8.0): continue
        angle_deg = math.degrees(self.scan_data.angle_min + i * self.scan_data.angle_increment)
        while angle_deg > 180: angle_deg -= 360
        while angle_deg < -180: angle_deg += 360

        if -20 <= angle_deg <= 20: front_min = min(front_min, r)
        elif 20 < angle_deg <= 60: left_min = min(left_min, r)
        elif -60 <= angle_deg < -20: right_min = min(right_min, r)

    if front_min < threshold or left_min < threshold or right_min < threshold:
        if front_min < 0.20 or (front_min < threshold and left_min < threshold and right_min <
threshold):
            return True, 'backward'
        elif left_min > right_min:
            return True, 'left'
        else:
            return True, 'right'
    return False, None

def navigate(self):
    while not rospy.is_shutdown():
        if self.current_pose is None or not self.global_path:
            self.last_time = rospy.Time.now().to_sec()
            self.rate.sleep()
            continue

        current_time = rospy.Time.now().to_sec()
        dt = current_time - self.last_time
        self.last_time = current_time
        if dt <= 0: continue

        final_goal = self.global_path[-1]

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

dist to final = math.sqrt((self.current_pose.position.x - final_goal[0])**2 +
(self.current_pose.position.y - final_goal[1])**2)

if dist_to_final <= 0.15:
    self.cmd_vel_pub.publish(Twist())
    self.cancel_pub.publish(GoalID())

# EKSEKUSI HAPUS GARIS DI RVIZ
empty_path = Path()
empty_path.header.frame_id = "map"
empty_path.header.stamp = rospy.Time.now()
self.clear_global_pub.publish(empty_path)
self.clear_local_pub.publish(empty_path)

self.global_path = []
self.last_v, self.last_w = 0.0, 0.0
self.linear_pid.reset(); self.angular_pid.reset()
rospy.loginfo_throttle(2.0, "[NAVIGATOR PID] Sampai di Tujuan! (Rute
Dibersihkan)")
self.rate.sleep()
continue

target_pt = self.get_lookahead_target()
if target_pt is None: continue

dx = target_pt[0] - self.current_pose.position.x
dy = target_pt[1] - self.current_pose.position.y
dist_err = math.sqrt(dx**2 + dy**2)

target_theta = math.atan2(dy, dx)
current_yaw = self.get_current_yaw()
head_err = math.atan2(math.sin(target_theta - current_yaw), math.cos(target_theta -
current_yaw))

target_v, target_w = 0.0, 0.0

has_obstacle, free_side = self.check_obstacles()
has_obstacle = False

if has_obstacle:
    if free_side == 'backward':
        target_v = self.v_min
        target_w = 0.0
    else:
        target_v = 0.0
        target_w = self.w_max if free_side == 'left' else -self.w_max

self.linear_pid.reset(); self.angular_pid.reset()
rospy.logwarn_throttle(1.0, f"[NAVIGATOR PID] Terhalang! Mengeksekusi manuver:
{free_side}")
else:
    target_v = self.linear_pid.compute(dist_err, dt)
    target_w = self.angular_pid.compute(head_err, dt)

if abs(target_w) > 0 and abs(target_w) < self.min_w_abs:
    target_w = self.min_w_abs if target_w > 0 else -self.min_w_abs

if abs(head_err) > (math.pi / 3.5):

```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

target_v = 0.0
self.linear_pid.reset()
else:
    target_v = target_v * max(0.1, 1.0 - (abs(head_err) / (math.pi/4)))

target_v = max(min(target_v, self.v_max), self.v_min)
target_w = max(min(target_w, self.w_max), self.w_min)
max_dv, max_dw = self.acc_x * dt, self.acc_theta * dt

cmd_v, cmd_w = target_v, target_w
if target_v > self.last_v + max_dv: cmd_v = self.last_v + max_dv
elif target_v < self.last_v - max_dv: cmd_v = self.last_v - max_dv

if target_w > self.last_w + max_dw: cmd_w = self.last_w + max_dw
elif target_w < self.last_w - max_dw: cmd_w = self.last_w - max_dw

self.last_v, self.last_w = cmd_v, cmd_w
cmd = Twist()
cmd.linear.x, cmd.angular.z = cmd_v, cmd_w
self.cmd_vel_pub.publish(cmd)
self.rate.sleep()

if __name__ == '__main__':
    try:
        IkibotNavigator().navigate()
    except rospy.ROSInterruptException:
        pass

```

POLITEKNIK
NEGERI
JAKARTA



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Lampiran 4 Dokumentasi Alat

