



© 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 *MODEL*
*PREDICTIVE CONTROL (MPC)***

SKRIPSI

**POLITEKNIK
NEGERI
JAKARTA**

MUHAMMAD RAIHAN RIFQO ARDANDI

2203431029

**PROGRAM STUDI INSTRUMENTASI DAN KONTROL
INDUSTRI**

**JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI JAKARTA**

2026



© 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 *MODEL
PREDICTIVE CONTROL (MPC)***

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar
Sarjana Terapan**

MUHAMMAD RAIHAN RIFQO ARDANDI

2203431029

**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

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Muhammad Raihan Rifqo Ardandi

NIM : 2203431029

Tanda Tangan : 

Tanggal : 15 Juni 2026

**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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

**LEMBAR PENGESAHAN
TUGAS AKHIR**

Tugas Akhir diajukan oleh

Nama : Muhammad Raihan Rifqo Ardandi
NIM : 2203431029
Program Studi : Instrumentasi dan Kontrol Industri
Judul Tugas Akhir : Kendali Robot *Mobile* IKIBOT Berbasis Model
Predictive Control (MPC)

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

Pembimbing : Hariyanto, S.Pd., M.T.

NIP. 199101282020121008

(.....)

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

Disahkan Oleh

Ketua Jurusan Elektro



Dr. Murie Dwiyaniti, S.T., M.T.

NIP. 19780331200312220022



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

Puji syukur saya panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat-Nya sehingga Skripsi berjudul “ Rancang Bangun Kendali *Mobile* IKIBOT dalam Menghindari Rintangan Secara *Real-Time* ” dapat diselesaikan dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Terapan. Penyusunan laporan ini tidak terlepas dari dukungan berbagai pihak, baik selama masa studi maupun dalam proses penyelesaian laporan. Untuk itu, saya mengucapkan terima kasih kepada semua pihak yang telah memberikan bantuan dan dukungan.

1. Dr. Murie Dwiyanti, S.T., M.T. selaku Ketua Jurusan Teknik Elektro;
2. Sulis Setiowati, S.Pd., M.Eng. selaku Ketua Program Studi Instrumentasi dan Kontrol Industri;
3. Hariyanto, S.Pd., M.T. selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan penulis dalam penyusunan skripsi ini;
4. Orang tua dan keluarga yang telah memberikan bantuan dukungan moral dan materi;
5. Soraya Putri Aulia dan Yosef Dwi Krisnata, selaku rekan satu tim dalam pelaksanaan Tugas Akhir ini yang telah mendukung, membantu, dan memotivasi dalam menyelesaikan skripsi ini;
6. Vivian Oktavia Ningrum selaku kekasih penulis yang selalu memberikan dukungan, bantuan, dan pendengar keluh kesah selama penulis menjalankan masa skripsi dan menyelesaikan laporan skripsi;
7. Teman-teman IKI 2022 A yang saling mendukung dan berjuang dalam menyelesaikan perkuliahan dan skripsi selama masa kuliah.

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga laporan Skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 15 Juni 2026

Penulis



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

ABSTRAK

Perkembangan sistem navigasi otonom pada robot differential drive menuntut algoritma kendali yang mampu melacak lintasan secara presisi sekaligus menghindari rintangan secara dinamis. Model Predictive Control (MPC) menawarkan keunggulan dalam memprediksi pergerakan robot di masa depan dengan memperhitungkan batasan kinematika dan rintangan secara simultan. Namun, implementasi MPC pada perangkat keras dengan kapasitas komputasi terbatas, seperti Raspberry Pi 4, sering kali memicu delay yang dapat mengakibatkan keterlambatan eksekusi pada aktuator dan kegagalan manuver. Penulisan ini berfokus pada implementasi dan optimasi algoritma MPC menggunakan kerangka kerja CasADi dan IPOPT pada robot differential drive berbasis Robot Operating System (ROS) Noetic. Sistem ini dirancang untuk menerima rute dari global planner dan merespons rintangan berdasarkan data laser dari LiDAR. Untuk mengatasi delay tersebut, penelitian ini melakukan penyetelan prediction horizon ke titik ekuilibrium operasional guna menyeimbangkan responsivitas aktuator dengan keandalan prediksi matematis. Algoritma ini juga diintegrasikan dengan normalisasi sudut (angle wrap-around) untuk menjamin efisiensi rotasi lintasan terpendek. Pengujian perangkat keras secara langsung membuktikan bahwa algoritma MPC yang telah disesuaikan mampu menjalankan navigasi otonom dan penghindaran rintangan secara stabil tanpa mengalami pembengkakan waktu komputasi.

Kata Kunci: Model Predictive Control (MPC), Navigasi Otonom, Robot Differential Drive, ROS Noetic, CasADi IPOPT, Penghindaran Rintangan

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

Design and Development of the IKIBOT Mobile Control System for Real-Time Obstacle Avoidance

ABSTRACT

Advances in autonomous navigation systems for differential-drive robots require control algorithms capable of precisely tracking a path while dynamically avoiding obstacles. Model Predictive Control (MPC) offers the advantage of predicting a robot's future movements by simultaneously accounting for kinematic constraints and obstacles. However, implementing MPC on hardware with limited computational capacity, such as the Raspberry Pi 4, often causes delays that can lead to execution delays in the actuators and maneuvering failures. This paper focuses on the implementation and optimization of the MPC algorithm using the CasADi and IPOPT frameworks on a differential-drive robot based on the Robot Operating System (ROS) Noetic. This system is designed to receive routes from a global planner and respond to obstacles based on LiDAR data. To address these delays, the prediction horizon is adjusted to the operational equilibrium point to balance actuator responsiveness with the reliability of mathematical predictions. This algorithm is also integrated with angle wrap-around normalization to ensure the efficiency of the shortest path rotation. Direct hardware testing demonstrates that the adapted MPC algorithm is capable of performing autonomous navigation and obstacle avoidance stably without incurring excessive computational time.

Keyword: *Model Predictive Control, Autonomous Navigation, Differential Drive Robot, ROS Noetic, CasADi IPOPT, Obstacle Avoidance.*

**POLITEKNIK
NEGERI
JAKARTA**



DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PERNYATAAN ORISINALITAS	iii
LEMBAR PENGESAHAN TUGAS AKHIR.....	iv
KATA PENGANTAR.....	v
ABSTRAK	vi
ABSTRACT.....	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	2
1.3 Tujuan.....	2
1.4 Batasan Masalah.....	3
1.5 Luaran	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 State of The Art	5
2.2 Robot Mobile dan Kinematika.....	8
2.2.1 Pengertian Robot <i>Mobile</i>	8
2.2.2 Model Kinematika <i>Differential Drive</i>	9
2.2.3 Sistem Kendali Robot Mobile.....	11
2.2.4 Sistem Navigasi Otonom	12
2.3 Algoritma Kendali Model Predictive Control (MPC).....	13
2.3.1 Pengertian <i>Model Predictive Control (MPC)</i>	14
2.3.2 Kerangka Kerja CasADi dan Solver IPOPT	15
2.3.3 Obstacle Avoidance pada Algoritma MPC.....	17
2.4 Komponen.....	18
2.4.1 Raspberry PI 4.....	18
2.4.2 Teensy 4.0.....	20



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

2.4.3 YDLIDAR X3	21
2.4.4 IMU MPU 6050	22
2.4.5 Baterai Lipo 3S	23
2.4.6 L298N IC	24
2.4.7 Motor DC <i>Encoder</i>	25
2.4.8 ACS712	27
2.4.9 MicroSD	28
2.4.10 Kabel Data USB to Micro B	29
2.4.11 <i>Universal Battery Eliminator Circuit</i> (UBEC)	30
2.5 Perangkat Lunak.....	31
2.5.1 Ubuntu 20.04 LTS	31
2.5.2 Robot Operating System (ROS).....	33
2.5.3 Python3	35
2.5.4 Visual Studio Code (VSCode)	36
2.5.5 Robot Visualization (Rviz).....	36
BAB III PERANCANGAN DAN REALISASI	38
3.1 Perancangan Sistem	38
3.1.1 Deskripsi Sistem	38
3.1.2 Cara Kerja Alat.....	41
3.1.3 Spesifikasi Alat	45
3.1.4 Peta Lintasan	47
3.1.5 Perancangan Skematik	48
3.1.6 Diagram Blok Sistem	51
3.1.7 Perancangan Mobile IKIBOT	56
3.1.8 Cara Kerja Alat Sub Sistem.....	58
3.2 Realisasi Alat.....	60
3.2.1 Realisasi Penerapan Penggunaan PCB	61
3.2.2 Realisasi Skematik Alat.....	63
3.2.3 Realisasi Robot Mobile IKIBOT	66
3.2.4 Realisasi Lingkungan Kerja Robot	71
3.2.5 Realisasi Implementasi MPC pada Robot Mobile IKIBOT.....	80
BAB IV PEMBAHASAN	86



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 Pengujian.....	86
4.1.1 Deskripsi Pengujian	87
4.1.2 Daftar Alat dan Bahan Pengujian.....	87
4.1.3 Prosedur Pengujian	88
4.1.4 Data Pengujian 3 Titik.....	89
4.1.5 Data Pengujian 3 Titik dengan Obstacle	95
4.1.6 Data Pengujian Full Map	101
4.2 Analisis Data Hasil Pengujian.....	108
4.2.1 Analisis Data Pengujian Navigasi 3 Titik	108
4.2.2 Analisis Data Pengujian Navigasi 3 Titik dengan Obstacle.....	110
4.2.3 Analisis Data Pengujian Navigasi Full Map	111
BAB V PENUTUP.....	114
5.1 Kesimpulan	114
5.2 Saran.....	115
DAFTAR PUSTAKA	117
LAMPIRAN.....	120

**POLITEKNIK
NEGERI
JAKARTA**



DAFTAR GAMBAR

Gambar 2. 1 Raspberry Pi 4B	18
Gambar 2. 2 Teensy 4.0.....	20
Gambar 2. 3 YDLiDAR X3	21
Gambar 2. 4 IMU MPU 6050	22
Gambar 2. 5 Baterai LiPO 3S	24
Gambar 2. 6 L298N Motor Driver	25
Gambar 2. 7 Motor DC Encoder	26
Gambar 2. 8 Sensor ACS712	28
Gambar 2. 9 Micro SD	29
Gambar 2. 10 USB to Micro B.....	29
Gambar 2. 11 UBEC	31
Gambar 2. 12 Ubuntu Operating System	32
Gambar 2. 13 Robot Operating System (ROS).....	33
Gambar 2. 14 Python3	35
Gambar 2. 15 VSCode	36
Gambar 2. 16 Robot Visualization	36
Gambar 3. 1 RQT Graph MPC	42
Gambar 3. 2 TF Tree IKIBOT.....	43
Gambar 3. 3 Peta Lintasan Robot	48
Gambar 3. 4 Diagram Skematik IKIBOT	49
Gambar 3. 5 Diagram Blok Sistem	51
Gambar 3. 6 Diagram Blok Sub Sistem	54
Gambar 3. 7 Model Robot Mobile IKIBOT 3D.....	56
Gambar 3. 8 Flowchart Cara Kerja Sub Sistem	58
Gambar 3. 9 Penampilan fisik PCB Embedded System IKIBOT	61
Gambar 3. 10 Hasil Akhir PCB Embedded IKIBOT	62
Gambar 3. 11 Realisasi Rangkaian Power Distribution	64
Gambar 3. 12 Realisasi Rangkaian Kontroler.....	65
Gambar 3. 13 Realisasi Layer 1 IKIBOT.....	67
Gambar 3. 14 Realisasi Layer 2 IKIBOT.....	68
Gambar 3. 15 Realisasi Layer 3 IKIBOT Tampak Bawah.....	69
Gambar 3. 16 Realisasi Layer 3 IKIBOT Tampak Atas.....	69
Gambar 3. 17 Robot Mobile IKIBOT	70
Gambar 3. 18 Tampilan Roscore.....	72
Gambar 3. 19 Parameter Bringup.....	73
Gambar 3. 20 Inisialisasi Rosserial.....	74
Gambar 3. 21 Penerimaan Data Odom	75
Gambar 3. 22 Node Slam Gmapping	76
Gambar 3. 23 Node Rviz.....	77

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



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 3. 24 Tampilan Slam Gmapping pada Rviz.....	78
Gambar 3. 25 Tampilan Navigasi setelah Save Map	79
Gambar 3. 26 Node MPC Navigator.....	80
Gambar 3. 27 Rviz Navigasi MPC.....	85
Gambar 4. 1 Grafik Kecepatan Linear Horizon $N = 10$	90
Gambar 4. 2 Grafik Kecepatan Sudut Horizon $N = 10$	90
Gambar 4. 3 Grafik Lintasan Gerak Horizon $N = 10$	91
Gambar 4. 4 Grafik Kecepatan Linear Horizon $N = 15$	92
Gambar 4. 5 Grafik Kecepatan Sudut Horizon $N = 15$	92
Gambar 4. 6 Grafik Lintasan Gerak Horizon $N = 15$	93
Gambar 4. 7 Grafik Kecepatan Linear Horizon $N = 20$	94
Gambar 4. 8 Grafik Kecepatan Sudut Horizon $N = 20$	94
Gambar 4. 9 Grafik Lintasan Gerak Horizon $N = 20$	95
Gambar 4. 10 Grafik Kecepatan Linear Horizon $N = 10$ Obstacle	96
Gambar 4. 11 Grafik Kecepatan Sudut Horizon $N = 10$ Obstacle.....	96
Gambar 4. 12 Grafik Lintasan Gerak Horizon $N = 10$ Obstacle	97
Gambar 4. 13 Grafik Kecepatan Linear Horizon $N = 15$ Obstacle	98
Gambar 4. 14 Grafik Kecepatan Sudut Horizon $N = 15$ Obstacle	98
Gambar 4. 15 Grafik Lintasan Gerak Horizon $N = 15$ Obstacle	99
Gambar 4. 16 Grafik Kecepatan Linear Horizon $N = 20$ Obstacle	100
Gambar 4. 17 Grafik Kecepatan Sudut Horizon $N = 20$ Obstacle	100
Gambar 4. 18 Grafik Lintasan Gerak Horizon $N = 20$ Obstacle	101
Gambar 4. 19 Grafik Kecepatan Linear Horizon $N = 10$ Full Map.....	102
Gambar 4. 20 Grafik Kecepatan Sudut Horizon $N = 10$ Full Map.....	103
Gambar 4. 21 Grafik Lintasan Gerak Horizon $N = 10$ Full Map	103
Gambar 4. 22 Grafik Kecepatan Linear Horizon $N = 15$ Full Map.....	104
Gambar 4. 23 Grafik Kecepatan Sudut Horizon $N = 15$ Full Map.....	105
Gambar 4. 24 Grafik Lintasan Gerak Horizon $N = 15$ Full Map	105
Gambar 4. 25 Grafik Kecepatan Linear Horizon $N = 20$ Full Map.....	106
Gambar 4. 26 Grafik Kecepatan Sudut Horizon $N = 20$ Full Map.....	107
Gambar 4. 27 Grafik Lintasan Gerak Horizon $N = 20$ Full Map	107



DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu.....	5
Tabel 3. 1 Spesifikasi Mekanis.....	45
Tabel 3. 2 Spesifikasi Perangkat Keras	46
Tabel 3. 3 Keterangan Komponen Layer 1 IKIBOT.....	67
Tabel 3. 4 Keterangan Komponen Layer 2 IKIBOT.....	68
Tabel 3. 5 Keterangan Komponen Layer 3 IKIBOT.....	70
Tabel 3. 6 Spesifikasi IKIBOT.....	71
Tabel 4. 1 Daftar Alat dan Bahan Pengujian.....	88
Tabel 4. 2 Data Pengujian Horizon $N=10$	90
Tabel 4. 3 Data Hasil Pengujian Horizon $N = 15$	92
Tabel 4. 4 Data Pengujian Horizon $N = 20$	94
Tabel 4. 5 Data Hasil Pengujian Horizon $N = 10$ Obstacle.....	96
Tabel 4. 6 Data Hasil Pengujian Horizon $N = 15$ Obstacle.....	98
Tabel 4. 7 Data Hasil Pengujian Horizon $N = 20$ Obstacle.....	100
Tabel 4. 8 Data Hasil Pengujian Horizon $N = 10$ Full Map.....	102
Tabel 4. 9 Data Hasil Pengujian Horizon $N = 15$ Full Map.....	104
Tabel 4. 10 Data Hasil Pengujian Horizon $N = 20$ Full Map.....	106
Tabel 4. 11 Analisis Perbandingan Parameter Horizon Pengujian 3 Titik	108
Tabel 4. 12 Analisis Perbandingan Parameter Horizon Pengujian 3 Titik Obstacle.....	110
Tabel 4. 13 Analisis Perbandingan Parameter Horizon Full Map.....	112

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 mengumunkan 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

DAFTAR LAMPIRAN

Lampiran 1 Daftar Riwayat Hidup Penulis.....	120
Lampiran 2 Dokumentasi Alat	121
Lampiran 3 Program Firmware Teensy.....	122
Lampiran 4 Program Python MPC.....	130





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

Perkembangan teknologi robotika pada robot *differential drive* menuntut sistem navigasi otonom yang tangguh dalam lingkungan dinamis. Menurut Nirmala & Hidayati (2024) integrasi Robot Operating System (ROS) telah menjadi standar arsitektur perangkat lunak untuk menjembatani sensor, aktuator, dan algoritma pemrosesan tinggi. Dalam navigasi otonom, kapabilitas melacak lintasan sekaligus menghindari rintangan secara aman merupakan parameter fundamental penentu keandalan pergerakan robot.

Untuk menjawab tantangan tersebut, *Model Predictive Control* (MPC) menjadi metode kendali lanjutan yang cukup menjanjikan. Studi yang dilakukan oleh Ali et al. (2024) menjelaskan bahwa keunggulan utama MPC terletak pada kemampuannya memprediksi trayektori masa depan, seraya memperhitungkan batasan kinematika secara eksplisit. Hal ini diperkuat oleh temuan Gao et al. (2023), yang membuktikan bahwa formulasi MPC menghasilkan presisi pelacakan lintasan tingkat tinggi. Lebih lanjut, Sudianto et al. (2022) menegaskan efektivitas MPC dalam penghindaran tabrakan (*collision avoidance*), mengingat algoritma ini secara matematis mengkalkulasi rute paling optimal di sekitar rintangan pada setiap siklus kontrol.

Meskipun memiliki keunggulan teoretis komprehensif, transisi implementasi MPC dari simulasi virtual ke perangkat keras fisik kerap menghadapi kendala komputasi. Mayoritas literatur, seperti penelitian yang dilakukan oleh Hariyanto et al. (2025) masih bertumpu pada evaluasi simulasi Gazebo dan RViz. Ketika algoritma sarat beban komputasi ini dieksekusi langsung pada *minicomputer* berspesifikasi terbatas seperti Raspberry Pi 4, kalkulasi matriks yang masif memicu tantangan teknis signifikan. Penelitian oleh Gerdpratoom & Yamamoto (2025) menyoroti bahwa komputasi optimasi real-time pada arsitektur skala kecil (*edge computing*) merupakan hambatan besar robotika otonom. Keterbatasan prosesor ini memicu masalah yang diidentifikasi oleh Dirckx et al. (2024) sebagai latensi



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

komputasi (computational delay). Durasi penyelesaian persoalan optimasi menyebabkan eksekusi aktuator tertinggal dari kondisi spasial aktual robot, yang berpotensi memicu kegagalan manuver.

Berlandaskan fenomena tersebut, penelitian skripsi ini difokuskan pada implementasi dan optimasi algoritma MPC secara fisik pada robot *differential drive* menggunakan kerangka kerja CasADi dan *solver* IPOPT dalam ekosistem ROS Noetic. Menurut Ortolani et al. (2026) pemanfaatan kerangka optimasi matematika yang efisien esensial untuk mereduksi beban komputasi tingkat lanjut. Dalam penelitian ini, parameter *prediction horizon* disetel presisi guna mencapai ekuilibrium antara jangkauan prediksi dan kapasitas eksekusi Raspberry Pi 4, dipadukan dengan normalisasi sudut (*angle wrap-around*) untuk efisiensi rotasi terpendek. Melalui pendekatan operasional tersebut, robot diharapkan mampu melaksanakan navigasi dan manuver penghindaran rintangan secara *real-time* tanpa terdegradasi oleh lonjakan latensi komputasi.

1.2 Perumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang arsitektur dan mengimplementasikan sistem kendali Model Predictive Control (MPC) pada robot *mobile* IKIBOT?
2. Bagaimana mengimplementasikan kerangka kerja CasADi dan *solver* IPOPT pada perangkat keras fisik robot *mobile* IKIBOT?
3. Bagaimana pengaruh penyesuaian parameter prediksi *horizon* terhadap pergerakan robot *mobile* IKIBOT?
4. Bagaimana hasil kinerja robot setelah dioptimasi dengan CasADi dan *solver* IPOPT dalam melakukan pelacakan lintasan dan penghindaran rintangan?

1.3 Tujuan

Berdasarkan rumusan masalah diatas maka diperoleh tujuan dari penelitian sebagai berikut:



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

1. Merancang bangun arsitektur dan mengimplementasikan algoritma kendali Model Predictive Control (MPC) pada perangkat keras robot mobile IKIBOT menggunakan ROS Noetic dan Teensy 4.0.
2. Mengintegrasikan kerangka kerja optimasi CasADi dan *solver* IPOPT pada robot *mobile* IKIBOT
3. Menganalisis pengaruh penyesuaian parameter prediksi *horizon* terhadap dinamika pergerakan robot *mobile* IKIBOT.
4. Mengevaluasi kinerja sistem kendali MPC dalam pelacakan lintasan dan penghindaran rintangan secara real-time pada robot *mobile* IKIBOT.

1.4 Batasan Masalah

Agar penelitian ini tetap terarah dan fokus pada tujuan yang ingin dicapai, maka ditetapkan batasan-batasan masalah sebagai berikut:

1. Implementasi dan pengujian sistem kendali dilakukan pada robot *mobile* IKIBOT.
2. *Model Predictive Control* (MPC) pada sistem kendali ini menggunakan formulasi matematis dan penyelesaian optimasi matriks dengan kerangka kerja CasADi dan *solver* numerik IPOPT.
3. Metode pengujian parameter difokuskan pada penyetelan batas horizon prediksi (*prediction horizon*), pembatasan waktu komputasi (*max CPU time*) pada IPOPT.
4. Pengujian manuver robot dilakukan pada lingkungan nyata dengan *map* yang sudah ditentukan
5. Penelitian ini merupakan bagian dari Tugas Akhir berkelompok dengan pembagian sub-sistem. Sub-sistem yang dibahas dalam laporan ini adalah sistem kendali berbasis Model Predictive Control (MPC). Sub-sistem lain seperti kendali PID dan perbandingan PID vs MPC dibahas dalam laporan anggota tim yang berbeda.

1.5 Luaran

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

1. Laporan Skripsi.
2. Publikasi Jurnal.



© 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

3. Rancang Bangun Kendali Mobile IKIBOT dalam menghindari rintangan secara *Real-Time*
4. Sistem kendali berbasis MPC yang terimplementasi pada robot mobile IKIBOT, mencakup hasil evaluasi kinerja pelacakan lintasan dan penghindaran rintangan dengan variasi parameter prediction horizon



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 penelitian, perancangan, dan serangkaian pengujian implementasi *Model Predictive Control* (MPC) pada navigasi otonom *robot mobile* IKIBOT yang telah dilaksanakan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Algoritma navigasi berbasis *Nonlinear Model Predictive Control* (NMPC) dengan *solver* CasADi/IPOPT telah berhasil diimplementasikan pada Robot *Mobile* IKIBOT. Sistem kendali ini terbukti mampu menjalankan fungsi pelacakan lintasan (*trajectory tracking*) dan penghindaran rintangan dinamis (*obstacle avoidance*) yang terintegrasi dengan pemindaian sensor YDLidar X3 secara *real-time*.
2. Integrasi kerangka kerja optimasi CasADi dan solver IPOPT pada perangkat keras robot mobile IKIBOT berhasil dilaksanakan dalam ekosistem ROS Noetic di atas Raspberry Pi 4. Penerapan pembatasan waktu komputasi pada IPOPT terbukti penting untuk menjaga eksekusi kendali tetap *real-time*, mencegah latensi komputasi yang dapat menyebabkan kegagalan manuver pada platform *edge computing* berspesifikasi terbatas.
3. Panjang *horizon* prediksi (*prediction horizon*, N) memiliki korelasi yang kuat terhadap profil dinamika aktuator dan akurasi pelacakan lintasan. Penggunaan *horizon* yang pendek ($N = 10$) menghasilkan respons kendali yang reaktif dan agresif. Keterbatasan jangkauan prediksi ini membuat algoritma terlambat merespons tikungan maupun rintangan dadakan, sehingga memicu efek osilasi (*chattering*) dan menghasilkan nilai *Tracking Error* rata-rata yang tinggi yaitu 0,5756 m pada rute statis dan 0,8230 m pada rute kontinu.
4. Penggunaan *horizon* prediksi yang terlalu panjang ($N = 20$) tidak menjamin akurasi yang lebih baik pada implementasi perangkat keras riil. Skala matriks optimasi yang masif membebani kapasitas *onboard computer* (Raspberry Pi 4), memicu terjadinya latensi komputasi. Jeda respons



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

komputasi ini menyebabkan robot terlambat mengeksekusi manuver belokan, mengambil lintasan penghindaran yang terlalu melebar (*overshoot*), dan mencatatkan *Tracking Error* yang lebih besar dibandingkan parameter menengah.

5. Konfigurasi *horizon* prediksi $N = 15$ terbukti sebagai parameter yang paling optimal untuk arsitektur IKIBOT. Parameter ini memberikan keseimbangan yang optimal antara kemampuan antisipasi masa depan (*predictive capability*) dengan batas toleransi kecepatan komputasi perangkat keras. Keseimbangan ini terbukti secara empiris dengan pencapaian rata-rata *Tracking Error* terendah di seluruh skenario pengujian: 0,4854 m pada rute statis 3 titik, 0,5914 m pasca-penghindaran rintangan, dan 0,7344 m pada navigasi *Full Map* berkelanjutan.

5.2 Saran

Meskipun sistem navigasi MPC pada *robot mobile* IKIBOT telah beroperasi dengan baik, terdapat beberapa ruang pengembangan yang dapat direkomendasikan untuk penelitian selanjutnya, yaitu:

1. Pada penelitian ini, nilai *horizon* prediksi ditetapkan secara tetap. Penelitian selanjutnya dapat mengimplementasikan *Adaptive MPC*, di mana nilai N dapat berubah secara dinamis merespons kondisi lingkungan, misalnya nilai N mengecil saat di keramaian untuk menghemat komputasi, dan membesar di lintasan lurus bebas hambatan.
2. Mengingat penyelesaian optimasi *Nonlinear MPC* khususnya untuk $N \geq 20$ cukup menguras CPU, disarankan untuk menggunakan komputer pendamping dengan arsitektur GPU terintegrasi, seperti NVIDIA Jetson Nano atau seri sejenisnya. Hal ini akan memungkinkan eksekusi perhitungan matriks CasADi berjalan secara paralel tanpa latensi.
3. Untuk meningkatkan ketahanan navigasi jarak jauh (*Full Map*) berkelanjutan, sistem odometri dari mikrokontroler dapat digabungkan dengan metode *Extended Kalman Filter* (EKF).
4. Pengujian *obstacle avoidance* saat ini dievaluasi terhadap satu rintangan dinamis dominan. Formulasi fungsi penalti rintangan (J_{obs}) ke depannya

dapat dikembangkan dengan metode pendeteksian untuk menghindari berbagai rintangan majemuk yang bergerak dari arah yang saling bersilangan.



© 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





DAFTAR PUSTAKA

- Ali, A. M., Shen, C., & Hashim, H. A. (2024). A linear MPC with control barrier functions for differential drive robots. *IET Control Theory and Applications*, 18(18). <https://doi.org/10.1049/cth2.12709>
- Celik, M. O., Koseoglu, M., & Deniz, F. N. (2026). Experimental performance analysis of an AMR controlled by a hybrid MPC-PID method. *Bulletin of the Polish Academy of Sciences Technical Sciences*, 156764–156764. <https://doi.org/10.24425/bpasts.2025.156764>
- Dirckx, D., Bos, M., Vandewal, B., Vanroye, L., Decré, W., Swevers, J., Make, F., & Leuven, K. (2024). *ASAP-MPC: An Asynchronous Update Scheme for Online Motion Planning with Nonlinear Model Predictive Control*.
- Gao, X., Su, X., An, A., & Zhang, H. (2023). Robust model predictive tracking control for the wheeled mobile robot with boundary uncertain based on linear matrix inequalities. *IET Cyber-Systems and Robotics*, 5(1). <https://doi.org/10.1049/csy2.12086>
- Gerdpratoom, N., & Yamamoto K. (2025). Decentralized nonlinear model predictive control-based flock navigation with real-time obstacle avoidance in unknown obstructed environments. *Frontiers in Robotics and AI*, 12. <https://doi.org/10.3389/frobt.2025.1540808>
- Guffanti, D., Mora Murillo, M. F., Bustamante Sanchez, S., Oswaldo Obregón Gutiérrez, J., Hinojosa, M. A., Brunete, A., Hernando, M., & Álvarez, D. (2026). Robust Model Predictive Control with a Dynamic Look-Ahead Re-Entry Strategy for Trajectory Tracking of Differential-Drive Robots. *Sensors*, 26(2). <https://doi.org/10.3390/s26020520>
- Hariyanto, Raihan Rifqo Ardandi, M., Putri Aulia, S., Dwi Krisnata, Y., & Sakhinatul Putri, Y. (2026). *Evaluasi Eksperimental Horizon Prediksi MPC pada Navigasi Otonom Robot TurtleBot3 Burger Experimental Evaluation of Prediction Horizon in MPC for Autonomous Navigation of TurtleBot3 Burger*.

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 mengumunkan 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

ELECTRICES: Jurnal Otomasi Kelistrikan dan Energi Terbarukan.
<https://jurnal.pnj.ac.id/index.php/electrices/article/view/8300>

Hariyanto, Sakhinatul Putri, Y., Raihan Rifqo Ardandi, M., Dwi Krisnata, Y., & Soraya Putri Aulia, dan. (2025). *Perbandingan Kinerja Kontrol MPC dan PID pada Navigasi Robot TurtleBot3 Menggunakan Simulasi Gazebo dan Rviz. Prosiding Seminar Nasional Inovasi Vokasional (SNIV).*
<https://prosiding.pnj.ac.id/index.php/sniv/article/view/3918>

Ismael, O. Y., Almaged, M., & Abdulla, A. I. (2024). Nonlinear Model Predictive Control-based Collision Avoidance for Mobile Robot. *Journal of Robotics and Control (JRC)*, 5(1). <https://doi.org/10.18196/jrc.v5i1.20615>

Jian, Z., Yan, Z., Lei, X., Lu, Z., Lan, B., Wang, X., & Liang, B. (2023). Dynamic control barrier function-based model predictive control to safety-critical obstacle-avoidance of mobile robot. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 3679–3685.
<https://doi.org/10.1109/ICRA48891.2023.10161376>

Nirmala, I., & Hidayati, R. (2024). SISTEM NAVIGASI OTONOM ROBOT MOBIL BERBASIS ROS PADA ROBOT PENGGERAK DIFERENSIAL. *INFOTECH Journal*, 10(2), 288–296.
<https://doi.org/10.31949/infotech.v10i2.11219>

Ortolani, L., Voss, G., Beltrami, G., Dorati, F., & Banfi, T. F. (2026). *Bayesian Optimization for Learning Nonlinear MPC in Autonomous Agent Navigation.*

Rawlings, J. B., Mayne, D. Q., Diehl, M. M., & Barbara, S. (2024). *Model Predictive Control: Theory, Computation, and Design 2nd Edition.*
<http://www.nobhillpublishing.com>

Sudianto, A. I., Muslim, M. A., & Rusli, M. (2022). Collision Avoidance System with Model Predictive Control for Mobile Robot Navigation. *2022 International Conference of Science and Information Technology in Smart*



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

Administration,

ICSINTESA

2022.

<https://doi.org/10.1109/ICSINTESA56431.2022.10041582>

Zhao, J., Liu, S., & Li, J. (2022). Research and implementation of autonomous navigation for mobile robots based on SLAM algorithm under ROS. *Sensors*, 22(11), 4172. <https://doi.org/10.3390/s22114172>



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



Penulis bernama lengkap Muhammad Raihan Rifqo Ardandi, yang merupakan anak pertama dari dua bersaudara. Lahir di Depok, 27 Juni 2004. Latar belakang pendidikan penulis adalah Sekolah Dasar di SDN Sukamaju 4 dan lulus pada tahun 2016. Melanjutkan ke Sekolah Menengah Pertama di SMP Negeri 11 Depok dan lulus pada tahun 2019. Kemudian melanjutkan Sekolah Menengah Atas di SMA Negeri 7 Depok dan lulus pada tahun 2022. Lalu 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.

Email: raihanrifqo118@gmail.com

**POLITEKNIK
NEGERI
JAKARTA**

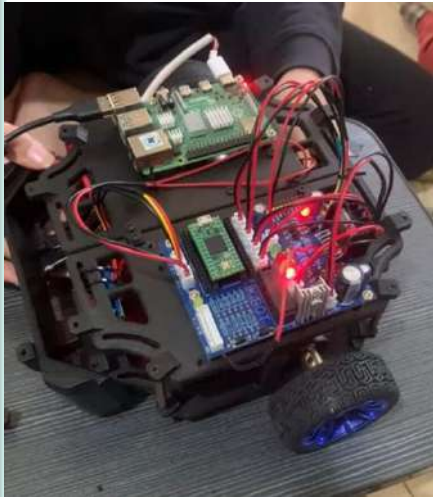


© Hak Cipta milik Politeknik Negeri Jakarta

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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Proses Perancangan Komunikasi IKIBOT



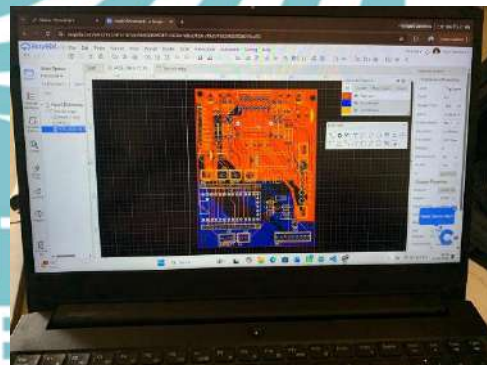
Robot *Mobile* IKIBOT



Lampiran 2 Dokumentasi Alat



Perancangan Desain PCB



Pengujian Robot IKIBOT pada *Track Map*



Program Teensy pada Arduino IDE

```
// =====  
// IKIBOT TEENSY 4.0 - FIRMWARE NAVIGASI & POWER MONITORING  
// =====  
  
#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
```

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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
#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;

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;
```



© 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

```
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) {
    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) {
```



© 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 mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

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));
  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);

```



© 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

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

    // -----
    // AUTO-REBOOT WATCHDOG (PENGGANTI 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) {
        // Jika koneksi terputus (karena Ctrl+C), mulai menghitung waktu
        if (disconnect_timer == 0) disconnect_timer = millis();

        // Jika terputus selama 3 detik penuh, paksa Teensy untuk Software Reset
        if (millis() - disconnect_timer > 3000) {
            // Perintah spesifik ARM Cortex-M untuk melakukan system reboot
            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;

```



© 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 mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

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

// -----
// KALKULASI 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);

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;

```



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

```
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;
}

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;
```



© Hak Cipta milik Politeknik Negeri Jakarta

```
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

Program mpc_navigation.py

```
#!/usr/bin/env python3
import rospy
import numpy as np
import math
from casadi import *
from geometry_msgs.msg import Twist, PoseStamped
from nav_msgs.msg import Odometry, Path
from sensor_msgs.msg import LaserScan

class MPCController:
    def __init__(self, N=20, dt=0.1):
        self.N = N
        self.dt = dt
        self.nx = 3 # [x, y, theta]
        self.nu = 2 # [v, w]

        # Sesuai data fisik & parameter DWA Ikibot
        self.v_min = -0.20
        self.v_max = 0.65
        self.w_min = -3.0
        self.w_max = 3.0

        # Cost weights: Fokus pada posisi (X,Y) terlebih dahulu, baru Heading (Theta)
        self.Q = np.diag([20.0, 20.0, 5.0])
        self.R = np.diag([1.0, 0.5]) # Penalty untuk gerakan agresif
        self.Qf = np.diag([40.0, 40.0, 10.0])
        self.obstacle_cost_weight = 500.0
        self.MAX_OBS = 5 # 5 obstacle terdekat

    def create_solver(self):
        x = SX.sym('x', self.nx)
        u = SX.sym('u', self.nu)

        # Kinematik Unicycle
        xdot = vertcat(
            u[0] * cos(x[2]),
            u[0] * sin(x[2]),
            u[1]
        )
        self.f = Function('f', [x, u], [xdot])

        opti = Opti()
        X = opti.variable(self.nx, self.N + 1)
        U = opti.variable(self.nu, self.N)

        x0 = opti.parameter(self.nx)
        x_ref = opti.parameter(self.nx)
        obstacles = opti.parameter(self.MAX_OBS, 3) # [x, y, radius]

        cost = 0
        for k in range(self.N):
            state_error = X[:, k] - x_ref
```

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



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

```

cost += state_error.T @ self.Q @ state_error
cost += U[:, k].T @ self.R @ U[:, k]

# BARRIER FUNCTION UNTUK OBSTACLE
for i in range(self.MAX_OBS):
    obs_x = obstacles[i, 0]
    obs_y = obstacles[i, 1]
    obs_r = obstacles[i, 2] # Robot Radius (0.12) + Safety Margin
    dist = sqrt((X[0, k] - obs_x)**2 + (X[1, k] - obs_y)**2 + 1e-5)
    cost += self.obstacle_cost_weight * (fmax(0, obs_r - dist)**3)

x_next = X[:, k] + self.f(X[:, k], U[:, k]) * self.dt
opti.subject_to(X[:, k + 1] == x_next)

# Terminal cost
state_error = X[:, self.N] - x_ref
cost += state_error.T @ self.Qf @ state_error

opti.minimize(cost)
opti.subject_to(X[:, 0] == x0)

for k in range(self.N):
    opti.subject_to(U[0, k] >= self.v_min)
    opti.subject_to(U[0, k] <= self.v_max)
    opti.subject_to(U[1, k] >= self.w_min)
    opti.subject_to(U[1, k] <= self.w_max)

opts = {'ipopt.print_level': 0, 'print_time': 0, 'ipopt.sb': 'yes'}
opti.solver('ipopt', opts)

self.solver = opti.to_function('mpc_solver', [x0, x_ref, obstacles], [U[:, 0], X])

def solve(self, x0, x_ref, obstacles_np):
    return self.solver(x0, x_ref, obstacles_np)

class IkibotMPCNavigator:
    def __init__(self):
        rospy.init_node('ikibot_mpc_navigator', anonymous=True)
        self.mpc = MPCController(N=20, dt=0.1)
        self.mpc.create_solver()

        self.current_pose = None
        self.global_path = []
        self.obstacles = []

        self.lookahead_distance = 0.4 # Jarak target bayangan di garis path (40 cm)

        self.cmd_vel_pub = rospy.Publisher('/cmd_vel', Twist, queue_size=10)
        self.cmd_vel_murni_pub = rospy.Publisher('/cmd_vel_mpc_murni', Twist, queue_size=10)
        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)

        self.rate = rospy.Rate(10)
        rospy.loginfo("[MPC] Siap! Menunggu Global Plan dari move_base...")

```



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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

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

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

def scan_callback(self, msg):
    if self.current_pose is None: return
    self.obstacles = []
    q = self.current_pose.orientation
    theta = math.atan2(2*(q.w*q.z + q.x*q.y), 1 - 2*(q.y*q.y + q.z*q.z))

    # Filter rintangan terdekat
    pts = []
    for i, r in enumerate(msg.ranges):
        if 0.1 < r < 0.6: # Jarak deteksi
            ang = msg.angle_min + i * msg.angle_increment
            pts.append((r, ang))

    pts.sort(key=lambda x: x[0])
    for r, ang in pts[:self.mpc.MAX_OBS]:
        ox = self.current_pose.position.x + r * math.cos(theta + ang)
        oy = self.current_pose.position.y + r * math.sin(theta + ang)
        self.obstacles.append([ox, oy, 0.22])

def get_current_state(self):
    if self.current_pose is None: return None
    x = self.current_pose.position.x
    y = self.current_pose.position.y
    q = self.current_pose.orientation
    theta = math.atan2(2*(q.w*q.z + q.x*q.y), 1 - 2*(q.y*q.y + q.z*q.z))
    return np.array([x, y, theta])

def get_lookahead_goal(self, x0):
    if not self.global_path: return None
    target_pt = self.global_path[-1]

    for pt in self.global_path:
        dist = math.sqrt((pt[0] - x0[0])**2 + (pt[1] - x0[1])**2)
        if dist > self.lookahead_distance:
            target_pt = pt
            break

    dx = target_pt[0] - x0[0]
    dy = target_pt[1] - x0[1]
    target_theta_raw = math.atan2(dy, dx)

    # --- NORMALISASI SUDUT TERPENDEK ---
    current_theta = x0[2]
    error_theta = target_theta_raw - current_theta

    while error_theta > math.pi:
        error_theta -= 2.0 * math.pi
    while error_theta < -math.pi:
        error_theta += 2.0 * math.pi
```



© 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_theta_continuous = current_theta + error_theta
# -----
return np.array([target_pt[0], target_pt[1], target_theta_continuous])

def navigate(self):
    while not rospy.is_shutdown():
        x0 = self.get_current_state()
        if x0 is None or not self.global_path:
            self.rate.sleep()
            continue

        x_ref = self.get_lookahead_goal(x0)

        final_goal = self.global_path[-1]
        if math.sqrt((x0[0]-final_goal[0])**2 + (x0[1]-final_goal[1])**2) < 0.15:
            self.cmd_vel_pub.publish(Twist())
            self.global_path = [] # Clear path
            rospy.loginfo_throttle(2.0, "[MPC] Goal Tercapai!")
            self.rate.sleep()
            continue

        obs_matrix = np.zeros((self.mpc.MAX_OBS, 3))
        for i in range(min(len(self.obstacles), self.mpc.MAX_OBS)):
            obs_matrix[i, :] = self.obstacles[i]

        try:
            u_opt, _ = self.mpc.solve(x0, x_ref, obs_matrix)
            cmd = Twist()

            # --- PENGIRIMAN PERINTAH KE TEENSY ---
            cmd.linear.x = float(u_opt[0])
            cmd.angular.z = float(u_opt[1])

            self.cmd_vel_pub.publish(cmd)
            self.cmd_vel_murni_pub.publish(cmd)

        except Exception as e:
            rospy.logwarn_throttle(1.0, "[MPC] Optimasi Gagal! Menunggu ruang aman...")
            self.cmd_vel_pub.publish(Twist())

            self.rate.sleep()

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

```