



© 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



**Navigasi Robot *Turtlebot3 Burger* berbasis MPC
Menggunakan Raspberry Pi 5 Dan Rviz Secara *Real-Time***

TESIS

Nama : Panjiyana

Nim : 2409511001

**POLITEKNIK
NEGERI
JAKARTA**

**PROGRAM STUDI MAGISTER TERAPAN
TEKNIK ELEKTRO
REKAYASA KONTROL INDUSTRI
PASCASARJANA POLITEKNIK NEGERI JAKARTA
DEPOK
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



**Navigasi Robot *Turtlebot3 Burger* berbasis MPC
Menggunakan *Raspberry Pi 5* Dan Rvis Secara *Real-Time***

TESIS

Diajukan untuk Persyaratan Melaksanakan Penelitian dan Menulis Tesis
dalam Bidang Rekayasa Kontrol Industri

Nama : Panjiyana

Nim : 2409511001

**PROGRAM STUDI MAGISTER TERAPAN TEKNIK
ELEKTRO
REKAYASA KONTROL INDUSTRI
PASCASARJANA POLITEKNIK NEGERI JAKARTA
DEPOK
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

HALAMAN PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini menyatakan dengan sebenarnya bahwa tesis ini saya susun tanpa tindakan plagiarisme sesuai dengan peraturan yang berlaku di Politeknik Negeri Jakarta.

Jika di kemudian hari ternyata saya melakukan tindakan plagiarisme, saya akan bertanggung jawab sepenuhnya dan menerima sanksi yang diajukan oleh Politeknik Negeri Jakarta kepada saya.

Depok, 08 juli 2026

Panjiyana
2409511001

POLITEKNIK
NEGERI
JAKARTA



HALAMAN PERNYATAAN ORISINALITAS

Dengan ini menyatakan bahwa tesis yang saya susun ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun yang dirujuk telah saya nyatakan dengan benar.

Nama : Panjiyana
NIM : 2409511001
Tanda Tangan :

Tanggal : 08 Juli 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

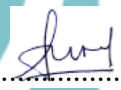
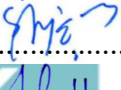
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan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

HALAMAN PENGESAHAN

Tesis ini yang telah diajukan oleh :

Nama : Panjiyana
NIM : 2409511001
Program Studi : Magister Terapan Teknik Elektro
Judul : Navigasi Robot *Turtlebot3 Burger* Berbasis MPC
Menggunakan *Raspberry Pi 5* Dan Rvis Secara *Real-Time*

Telah diuji oleh Tim Penguji dalam sidang Tesis pada hari kamis tanggal 16 juli tahun 2026 dan dinyatakan LULUS untuk memperoleh derajat gelar Magister Terapan pada Program Studi Magister Terapan Teknik Elektro Politeknik Negeri Jakarta

Pembimbing I	: Nana Sutarna, S.T., M.T, Ph.D.	(..... )
Pembimbing II	: Dr. Prihatin Oktivasari, S.Si, M.Si.	(..... )
Penguji I	: Dr. Isdawimah, S.T., M.T.	(..... )
Penguji II	: Dr. Murie Dwiyaniti, S.T., M.T.	(..... )
Penguji III	: Dr. Dra. Yogi Widiawati, M.Hum	(..... )

Depok, 16 Juli 2026

Diketahui Oleh

Ketua Pascasarjana Politeknik Negeri Jakarta



Prof. Dr. Tatun Hayatun Nufus, M.Si.
NIP. 19660416199512200



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 PERSETUJUAN PUBLIKASI TESIS UNTUK KEPENTINGAN AKADEMIK

Sebagai sivitas akademik Politeknik Negeri Jakarta, saya yang bertanda tangan dibawah ini

Nama : Panjiyana
NIM : 2409511001
Program Studi : Magister Terapan Teknik Elektro
Jenis Karya : Tesis

Demi pengembangan ilmu pengetahuan, saya menyetujui untuk memberikan kepada Politeknik Negeri Jakarta Hak Bebas Royalti Noneksklusif (Non-exclusive Royalti – Free Right) atas karya ilmiah saya berjudul : *NAVIGASI ROBOT TURTLEBOT3 BURGER BERBASIS MPC MENGGUNAKAN RASBERRY PI 5 DAN RVIS SECARA REAL-TIME*

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Politeknik Negeri Jakarta berhak menyimpan, mengalihmediakan/mengalihinformasikan, mengelola dalam bentuk pangkalan data (data base), merawat dan memublikasikan tesis saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta. Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 08 Juli 2026

Yang menyatakan

Panjiyana

2409511001



ABSTRAK

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

Nama : Panjiyana

Program Studi : Magister Terapan Teknik Elektro

Judul Tesis : NAVIGASI ROBOT *TURTLEBOT3 BURGER* BERBASIS MPC
MENGUNAKAN *RASBERRY PI 5* DAN RVIS SECARA
REAL-TIME

Robot *mobile* otonom membutuhkan sistem kendali yang mampu menghasilkan navigasi yang akurat, stabil, dan efisien. Pengendali Proportional-Integral-Derivative (PID) masih banyak digunakan, namun memiliki keterbatasan pada lintasan yang kompleks. Penelitian ini bertujuan merancang dan mengimplementasikan MPC serta membandingkan kinerjanya dengan PID pada robot *TurtleBot3 Burger* berbasis *Raspberry Pi 5* menggunakan ROS Noetic. Penelitian dilakukan melalui 25 kali pengujian pada lintasan indoor dengan lima waypoint. Parameter yang dievaluasi meliputi akurasi lintasan, kualitas manuver, waktu navigasi, dan konsumsi daya. Hasil penelitian menunjukkan bahwa MPC menghasilkan konsumsi daya rata-rata 9,935 W, lebih hemat 5,6% dibandingkan PID sebesar 10,524 W, serta memiliki akurasi lintasan yang lebih baik dengan nilai RMSE 0,6788 m dibandingkan 0,7288 m pada PID. Selain itu, MPC menghasilkan manuver yang lebih halus dan waktu navigasi yang lebih singkat. Implementasi MPC pada *Raspberry Pi 5* berjalan stabil secara real-time sehingga layak diterapkan sebagai metode kendali navigasi robot *TurtleBot3 Burger* di lingkungan indoor.

Kata kunci: MPC, PID, *TurtleBot3 Burger*, ROS Noetic, Navigasi.



KATA PENGANTAR

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

Alhamdulillah syukur ke hadirat Allah SWT atas segala rahmat, hidayah, taufik, serta karunia-Nya sehingga penulis dapat menyelesaikan tesis yang berjudul "ANALISIS DAN PERANCANGAN SISTEM NAVIGASI ROBOT *TURTLEBOT3 BURGER* BERBASIS MPC MENGGUNAKAN *RASBERRY PI 5* DAN RVIS SECARA *REAL-TIME*" sebagai salah satu syarat untuk memperoleh gelar Magister Terapan Teknik Elektro pada Program Studi Magister Terapan Teknik Elektro, Pascasarjana Politeknik Negeri Jakarta.

Shalawat serta salam semoga senantiasa tercurah kepada junjungan kita Nabi Muhammad SAW, beserta keluarga, sahabat, dan seluruh umatnya yang istiqamah mengikuti ajaran beliau hingga akhir zaman.

Penulis menyadari bahwa penyusunan tesis ini tidak terlepas dari bantuan, bimbingan, arahan, dukungan, serta doa dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Istri saya tercinta (Nurul Hasanah) dan kedua anak saya (Mas Giandra dan Kayyisa) yang berikan doa dan motivasi saya sehingga dapat menyusun tesis ini dengan baik.
2. Direktur Politeknik Negeri Jakarta beserta seluruh jajaran yang telah memberikan kesempatan kepada penulis untuk menempuh pendidikan pada Program Magister Terapan Teknik Elektro.
3. Ibu Prof. Dr. Tatun Hayatun Nufus, M.Si., selaku Ketua Jurusan Teknik Elektro Politeknik Negeri Jakarta.
4. Bapak Nana Sutarna, S.T., M.T., Ph.D., selaku Ketua Program Studi Magister Terapan Teknik Elektro Politeknik Negeri Jakarta Dosen Pembimbing I, yang telah meluangkan waktu, memberikan bimbingan, arahan, motivasi, serta masukan yang sangat berharga selama proses penyusunan tesis ini. Semoga pak nana dan keluarga diberikan kesehatan , kelancaran dan riski yang berlimpah.



© 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

5. Ibu Dr. Prihatin Oktivasari, S.Si., M.Si., selaku Dosen Pembimbing II, yang telah memberikan perhatian, bimbingan, saran, dan masukan yang sangat bermanfaat dalam penyempurnaan tesis ini. Semoga pak nana dan keluarga diberikan kesehatan, kelancaran dan riski yang berlimpah
6. Seluruh dosen dan tenaga kependidikan Program Studi Magister Terapan Teknik Elektro Politeknik Negeri Jakarta yang telah memberikan ilmu, pengalaman, serta pelayanan akademik selama penulis menempuh pendidikan.
7. Hariyanto, S.Pd., M.T., Dendy G.A. S.pd., Alan Devri Susanto S.pd. dan rekan-rekan ELKA2010 yang sudah memberikan saran dan motivasi dalam penyelesaian tesis ini.
8. Rekan-rekan mahasiswa (Yosef, Raihan, dan Soraya) Politeknik Negeri Jakarta yang telah memberikan semangat, bantuan, dan kebersamaan selama penyusunan tesis.
9. Kepada ibu Lise Tiasanty selaku pimpinan CS division dan rekan-rekan PT.Sharp Electronics Indonesia yang sudah memberikan doa dan motivasi dalam pekerjaan maupun perkuliahan sampai penyusunan tesis ini.
10. Semua rekan-rekan mahasiswa MTTE24 dan para dosen pasca sarjana politeknik negeri jakarta, kemudian pihak yang tidak dapat penulis sebutkan satu per satu yang telah memberikan doa, dukungan moral maupun material, sehingga penulis dapat menyelesaikan tesis ini.

Penulis menyadari bahwa tesis ini masih memiliki keterbatasan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat diharapkan demi penyempurnaan penelitian pada masa mendatang. Penulis berharap semoga tesis ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan, khususnya di bidang sistem kendali, robotika, dan navigasi robot mobile, serta menjadi referensi bagi penelitian selanjutnya.

Depok, 08 Juli 2026

Penulis



DAFTAR ISI

HALAMAN PERNYATAAN BEBAS PLAGIARISME	i
HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN	Error! Bookmark not defined.
ABSTRAK.....	v
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR.....	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Batasan Penelitian	4
1.4.1 Platform Robot yang Digunakan	4
1.4.2 Ruang Lingkup Lingkungan Uji	4
1.4.3 Metode Kendali yang Dibandingkan	4
1.4.4 Adaptivitas	4
1.5 Manfaat penelitian.....	4
BAB II.....	6
2.1 Model Predictive Control (MPC).....	6
2.2 PID	7
2.3 Evaluasi Kinerja Navigasi	9
2.3.1 Trajectory Tracking	9
2.3.2 Kecepatan Linear dan Kecepatan Angular	9
2.3.3 Tracking Error dan Root Mean Square Error (RMSE).....	10
2.3.4 Konsumsi Daya Robot.....	11
2.4 Robot <i>turtlebot3 burger</i>	12

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 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

2.5	Rviz (ROS <i>Visualization</i>)	14
2.6	ROS (<i>Robot Operating System</i>)	15
2.7	<i>Differential Drive</i>	16
2.8	Penelitian Terkait	17
BAB III		12
3.1	Desain Penelitian	12
3.2	Tahapan Penelitian	13
3.2.1	Studi Literatur dan Analisis Kebutuhan	13
3.3	Pembuatan Alat/ Robot	13
3.3.1	Desain Elektrikal	13
3.3.2	Desain Mekanik	14
3.3.3	Perancangan Perangkat Lunak (<i>Software</i>)	15
3.4	Pemodelan Robot <i>Differential Drive</i>	17
3.5	Perancangan Model Predictive Control (MPC)	21
3.5.1	Formulasi <i>Nonlinear</i> MPC (NMPC)	21
3.5.2	Pemilihan Matriks Bobot	22
3.5.3	Formulasi Linear MPC (LMPC)	24
3.5.4	Sketsa Implementasi	25
3.5.5	Ringkasan Alur Perancangan	25
3.6	Pengujian Alat/ Robot	26
3.7	Pengambilan data penelitian	27
3.8	Metode Dan Teknis Analisis Data Penelitian	28
3.8.1	Teknik Analisis Data	29
3.8.2	Output yang Dihasilkan	29
3.8.3	Parameter Evaluasi	30
BAB IV		31
4.1	Pengujian Tuning Parameter PID	31
4.2	Pengujian perbandingan antara MPC dan PID	35
4.3	Pengukuran Kecepatan Jalur Lurus	36
4.3.1	Kecepatan Linear Titik 1	37
4.3.2	Kecepatan Linear Titik 2	37



© 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

4.3.3	Kecepatan Linear Titik 3	38
4.3.4	Kecepatan Linear Titik 4	39
4.3.5	Kecepatan Linear Titik 5	40
4.4	Pengukuran Kecepatan Sudut	41
4.4.1	Kecepatan Sudut Titik 1	41
4.4.2	Kecepatan Sudut Titik 2	42
4.4.3	Kecepatan Sudut Titik 3	43
4.4.4	Kecepatan Sudut Titik 4	44
4.4.5	Kecepatan Sudut Titik 5	44
4.5	Pengujian <i>Trajectory Tracking Error</i>	45
4.6	Pengukuran Konsumsi Daya	46
4.7	Rekapitulasi Hasil Pengujian	47
4.8	Pembahasan	52
BAB V		54
5.1	Kesimpulan	54
5.2	Saran	55
DAFTAR PUSTAKA		58
LAMPIRAN		61

**POLITEKNIK
NEGERI
JAKARTA**



DAFTAR TABEL

Tabel 2.1 Parameter Fisik TurtleBot3 Burger.....	13
Tabel 2.2 Perbandingan Penelitian Terdahulu	21
Tabel 3.1 Pemilihan Matriks Bobot Q, R, dan P	22
Tabel 4. 1 Perbandingan Parameter Tuning PID	34
Tabel 4. 2 Rekapitulasi Perbandingan Rata-Rata Hasil Pengujian PID dan MPC.....	48

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

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

Gambar 2. 1 Robot Turtlebot3 Burger	14
Gambar 2. 2 Logo Resmi Rviz.....	15
Gambar 3. 1 Diagram Alir Tahapan Penelitian.....	12
Gambar 3. 2 Blok Diagram Robot Turtlebot3 Burger	15
Gambar 3. 3 Rancangan Robot Turtlebot3 Burger	16
Gambar 3. 4 Flowchart Implementasi Sistem Navigasi Robot Berbasis ROS.....	17
Gambar 3. 5 Pemodel yang digunakan	20
Gambar 3. 6 Track Pengujian Robot Turtlebot3	27
Gambar 3. 7 Visualisasi Lidar pada ROS	28
Gambar 4. 1 Perbandingan Kecepatan Linear PID	31
Gambar 4. 2 Perbandingan Kecepatan Sudut PID	32
Gambar 4. 3 Perbandingan Error Terhadap Waktu PID	33
Gambar 4. 4 Lintasan PID dan MPC	35
Gambar 4. 5 Perbandingan Kecepatan Linear Titik 1.....	37
Gambar 4. 6 Perbandingan Kecepatan Linear Titik 2.....	38
Gambar 4. 7 Perbandingan Kecepatan Linear Titik 3.....	39
Gambar 4. 8 Perbandingan Kecepatan Linear Titik 4.....	40
Gambar 4. 9 Perbandingan Kecepatan Linear Titik 5.....	41
Gambar 4. 10 Perbandingan Kecepatan Sudut Titik 1.....	42
Gambar 4. 11 Perbandingan Kecepatan Sudut Titik 2.....	43
Gambar 4. 12 Perbandingan Kecepatan Sudut Titik 3.....	43
Gambar 4. 13 Perbandingan Kecepatan Sudut Titik 4.....	44
Gambar 4. 14 Perbandingan Kecepatan Sudut Titik 5.....	45
Gambar 4. 15 Perbandingan Tracking Error PID dan MPC	46
Gambar 4. 16 Perbandingan Konsumsi Daya PID dan MPC.....	47
Gambar 4. 17 Komparasi Rata Rata Kecepatan Linear antara PID Dan MPC	50
Gambar 4. 18 Komparasi Rata Rata Kecepatan Sudut Parameter antara PID Dan MPC ..	51
Gambar 4. 19 Komparasi Rata Rata Trajectory Error Parameter antara PID Dan MPC ..	51
Gambar 4. 20 Komparasi Rata Rata Konsumsi Daya Parameter antara PID Dan MPC...	51



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

BAB 1 PENDAHULUAN

1. Latar Belakang

Robot merupakan alat bantu yang dirancang untuk meningkatkan kemampuan manusia dalam menyelesaikan pekerjaan secara lebih aman, efisien, dan presisi. Salah satu bidang robotika yang mengalami perkembangan pesat adalah robot *mobile* otonom, yaitu robot yang mampu bergerak, bernavigasi, dan berinteraksi dengan lingkungan secara real-time tanpa intervensi langsung manusia [1]. Robot *mobile* otonom telah banyak diterapkan pada sektor industri manufaktur, logistik otomatis, pelayanan kesehatan, pencarian dan penyelamatan, serta eksplorasi trek atau area berbahaya yang sulit dijangkau manusia [2]. Dalam konteks tersebut, sistem navigasi menjadi komponen krusial karena kegagalan navigasi dapat menyebabkan tabrakan, penurunan efisiensi operasional, serta peningkatan risiko keselamatan bagi lingkungan dan pengguna.

Agar robot *mobile* dapat beroperasi secara aman dan efisien, diperlukan sistem kendali yang mampu menjaga stabilitas gerak, ketepatan lintasan, serta memberikan respons cepat terhadap perubahan lingkungan. Salah satu metode kendali yang paling umum digunakan pada robot *mobile* dengan konfigurasi *differential drive* adalah PID [3]. Metode ini banyak dipilih karena strukturnya sederhana, mudah diimplementasikan, dan memiliki kebutuhan komputasi yang rendah. PID bekerja dengan memperbaiki kesalahan berdasarkan umpan balik kondisi aktual sistem, sehingga efektif pada lingkungan yang relatif stabil [4]. Namun, pada lintasan kompleks, belokan tajam, serta lingkungan yang dinamis dan bersifat nonlinier, kendali PID sering mengalami keterbatasan berupa osilasi, respons lambat, dan ketidakmampuan mengantisipasi perubahan kondisi secara proaktif.

Sebagai upaya mengatasi keterbatasan tersebut, metode kendali seperti MPC banyak dikembangkan pada sistem navigasi robot *mobile*. MPC memanfaatkan model matematika untuk memprediksi perilaku robot di masa



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

cepatan dengan mempertimbangkan batasan fisik aktuator dan lingkungan [5]. Namun, sebagian besar penelitian sebelumnya masih terbatas pada simulasi dengan konfigurasi perangkat keras tertentu yang tidak selalu ekonomis [6]. Kelebihan MPC dibandingkan PID mampu menangani constraint secara eksplisit. Pada belokan tajam, kontrol PID memaksa aktuator mencapai saturasi, sedangkan kontrol MPC mampu memperlambat aktuator sebelum tikungan agar tidak keluar jalur.

Robot *mobile turtlebot3 burger* dikembangkan sebagai platform penelitian dan pembelajaran yang mengadopsi kinematika *differential drive*, namun menggunakan perangkat keras yang lebih ekonomis, yaitu Raspberry Pi 5 sebagai unit komputasi berbasis *Robot Operating System* (ROS) dan *openCR* sebagai pengendali aktuator tingkat rendah. Tantangan utama yang dihadapi robot *turtlebot3 burger* tetap sama, yaitu menjaga stabilitas gerak, ketepatan lintasan, dan kemampuan penghindaran rintangan secara *real-time* pada lingkungan nyata [7]. Berdasarkan kondisi tersebut, terdapat urgensi untuk melakukan penelitian yang membandingkan kinerja kendali PID dan MPC pada platform robot mobile [8][9]. Keterbaruan (*novelty*) penelitian ini berupa penerapan MPC sebagai navigator arah jalannya robot *turtlebot3 burger* yang menggunakan Raspberry Pi 5 dan program MPC minimalis diterapkan langsung pada robot fisik kemudian dibandingkan dengan PID menggunakan platform, lintasan, lima *waypoint*, perangkat keras, serta prosedur pengujian yang sama. Setiap metode diuji sebanyak 25 kali untuk memperoleh hasil yang lebih representatif. Parameter evaluasi meliputi akurasi *trajectory tracking*, kualitas manuver berdasarkan kecepatan linear dan kecepatan sudut, waktu navigasi, kestabilan operasi *real-time*, serta konsumsi daya. Hasil penelitian diharapkan memberikan kontribusi berupa program MPC yang ringkas dan mudah direplikasi serta menghasilkan bukti empiris mengenai kelayakan MPC dibandingkan PID pada navigasi robot mobile di lingkungan indoor.



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 Perumusan Masalah

Berdasarkan latar belakang tersebut, penelitian ini akan berfokus pada beberapa permasalahan utama, yaitu:

1. Bagaimana mengimplementasikan MPC pada robot *TurtleBot3 Burger* berbasis *Raspberry Pi 5* menggunakan *ROS Noetic* untuk sistem navigasi robot mobile?
2. Bagaimana perbandingan kinerja MPC dan PID dalam navigasi robot *TurtleBot3 Burger* berdasarkan akurasi *trajectory tracking*, kualitas manuver, waktu navigasi, dan konsumsi daya?
3. Apakah implementasi MPC pada *Raspberry Pi 5* mampu memberikan kinerja navigasi yang lebih baik serta berjalan secara stabil dan *real-time* dibandingkan PID?

1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk mengembangkan dan menerapkan MPC sebagai metode kendali navigasi pada *turtlebot3 burger* agar mampu menghasilkan navigasi yang akurat, stabil, dan efisien. Untuk mencapai tujuan utama tersebut, penelitian ini memiliki beberapa tujuan spesifik sebagai berikut:

1. Mengimplementasikan MPC pada robot *TurtleBot3 Burger* berbasis *Raspberry Pi 5* menggunakan *ROS Noetic* sebagai pengendali navigasi robot *mobile*.
2. Menganalisis dan membandingkan kinerja MPC dan PID berdasarkan parameter akurasi *trajectory tracking*, kualitas manuver, waktu navigasi, dan konsumsi daya.
3. Mengevaluasi efektivitas implementasi MPC pada *Raspberry Pi 5* dari aspek performa navigasi, efisiensi energi, serta kemampuan komputasi *real-time* dibandingkan PID.



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

1.4 Batasan Penelitian.

1.4.1 Platform Robot yang Digunakan

Penelitian ini dibatasi pada penggunaan robot *Turtlebot3 Burger* sebagai platform utama. Semua implementasi kontrol dilakukan langsung pada robot fisik dengan komponen standar bawaan (*Raspberry Pi 5*, *OpenCR*, sensor LiDAR. Modifikasi hardware tambahan tidak dilakukan. Setiap arah navigasi dibatasi satu kali tujuan dan robot memiliki jarak aman 10.5cm untuk setiap arah (depan, belakang dan samping kanan-kiri).

1.4.2 Ruang Lingkup Lingkungan Uji

Pengujian hanya dilakukan pada lingkungan indoor dengan kondisi lantai rata (datar), pencahayaan normal, dan tanpa gangguan elektromagnetik yang signifikan. Lingkungan dibatasi pada area laboratorium atau ruangan uji dengan obstacle sederhana (misalnya dinding tiruan, box, dan objek lain), bahwa robot ini menggunakan dua buah roda sebagai penggeraknya.

1.4.3 Metode Kendali yang Dibandingkan

Tipe kendali yang digunakan hanya pada Kendali PID menggunakan model konvensional dan MPC *base line*.

1.4.4 Adaptivitas

Adaptivitas dalam penelitian ini dibatasi pada kemampuan robot menyesuaikan lintasan dan kecepatan secara *real-time*.

1.5 Manfaat penelitian

- 1) Meningkatkan Efisiensi Navigasi Robot dengan membandingkan dua metode pengendalian PID dan MPC penelitian ini menunjukkan

bagaimana MPC dapat meningkatkan efisiensi navigasi robot dalam menghindari rintangan.

- 2) Penelitian ini memperkenalkan desain MPC untuk robot mobile dalam menghindari navigasi
- 3) Penelitian ini mengintegrasikan MPC dalam ekosistem ROS dan Rviz untuk pengukuran secara *real-time* pada robot *turtlebot3 burger*.



© 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





BAB V PENUTUP

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

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan terhadap implementasi MPC dan PID pada navigasi robot *TurtleBot3 Burger* menggunakan *Raspberry Pi 5* dalam lingkungan *indoor* dengan 5 titik *waypoint* serta 25 percobaan untuk masing-masing metode, dapat ditarik kesimpulan sebagai berikut.

- 1) MPC terbukti menghasilkan konsumsi daya yang lebih efisien dibandingkan PID. Rata-rata konsumsi daya MPC dari 25 percobaan adalah 9,935 W, sedangkan PID sebesar 10,524 W, sehingga MPC lebih hemat energi sebesar 0,589 W atau sekitar 5,6%. Perbedaan ini signifikan pada aplikasi robot otonom berbaterai karena secara langsung memperpanjang durasi operasional robot.
- 2) MPC menghasilkan kualitas manuver yang lebih baik dari PID, khususnya pada segmen navigasi yang membutuhkan belokan tajam dan manuver kompleks. Hal ini terlihat dari profil kecepatan sudut MPC yang lebih terstruktur dan terencana, dengan osilasi yang lebih sedikit dibandingkan PID yang cenderung melakukan koreksi arah berulang. Keunggulan ini paling menonjol pada titik 4 di mana MPC menyelesaikan segmen dalam sekitar 13 detik sementara PID membutuhkan lebih dari 40 detik.
- 3) MPC menghasilkan akurasi *trajectory tracking* yang lebih baik dengan nilai RMSE sebesar 0,6788 m dibandingkan PID sebesar 0,7288 m (selisih 6,9%). Meskipun rata-rata *trajectory error* MPC dari 25 percobaan lebih tinggi akibat adanya fase inisialisasi pada percobaan ke-1 hingga ke-6 dengan error di atas 1,0 m, setelah kondisi stabil pada percobaan ke-7 dan seterusnya nilai error MPC konsisten berada di bawah 0,7 m.



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

MPC menghasilkan total durasi navigasi yang lebih singkat dibandingkan PID. Dari percobaan representatif, total durasi MPC adalah 85,3 detik sedangkan PID adalah 88,7 detik. Meskipun tidak selalu unggul di setiap segmen — PID lebih cepat pada segmen lurus sederhana seperti titik 5, secara keseluruhan MPC menunjukkan efisiensi waktu yang lebih baik, terutama pada segmen yang membutuhkan manuver kompleks.

Implementasi MPC pada Raspberry Pi 5 dalam ekosistem ROS Noetic menggunakan Docker terbukti feasible secara teknis dan praktis. Overhead komputasi *real-time* MPC tidak menyebabkan degradasi performa yang signifikan, dan sistem berjalan stabil selama seluruh sesi pengujian. Hal ini menunjukkan bahwa Raspberry Pi 5 merupakan platform komputasi yang memadai untuk menjalankan algoritma kontrol berbasis optimasi seperti MPC pada robot *mobile* skala kecil.

- 6) Secara keseluruhan, MPC unggul atas PID pada sebagian besar aspek yang diukur, yaitu efisiensi energi, kualitas manuver, akurasi lintasan, dan total waktu navigasi. Temuan ini konsisten dengan hipotesis penelitian bahwa MPC mampu memberikan performa navigasi yang lebih baik pada robot TurtleBot3 Burger dalam lingkungan indoor, khususnya pada lintasan dengan geometri yang kompleks.

5.2 Saran

Berdasarkan hasil penelitian dan keterbatasan yang ditemukan selama pelaksanaan, terdapat beberapa saran untuk pengembangan penelitian selanjutnya, yaitu sebagai berikut.

- 1) Perlu dilakukan optimasi parameter MPC yang lebih sistematis, terutama untuk mengurangi variabilitas *trajectory error* pada percobaan awal (percobaan 1–6). Penggunaan metode *tuning* otomatis seperti *Bayesian Optimization* atau *grid search* dapat membantu menemukan parameter horizon prediksi dan bobot fungsi biaya yang lebih optimal untuk lingkungan



© 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

pengujian yang spesifik.

Penelitian selanjutnya disarankan untuk mengintegrasikan data pose dari AMCL (Adaptive Monte Carlo Localization) sebagai basis perhitungan *trajectory error*, menggantikan data odometri murni. AMCL menghasilkan estimasi posisi yang lebih akurat karena terkoreksi oleh data LiDAR, sehingga hasil evaluasi akurasi lintasan akan lebih representatif terhadap kondisi nyata.

Pengujian sebaiknya diperluas ke lingkungan yang lebih dinamis, termasuk skenario dengan rintangan bergerak (*dynamic obstacles*) dan peta yang berubah secara real-time. Hal ini penting untuk mengevaluasi keunggulan prediktif MPC pada kondisi yang lebih menantang dan relevan dengan aplikasi dunia nyata, seperti navigasi di koridor gedung atau gudang yang aktif.

- 4) Perlu dikaji implementasi MPC berbasis model kinematik yang lebih akurat dengan mempertimbangkan slip roda dan ketidakidealan aktuator pada TurtleBot3 Burger. Model diferensial drive yang lebih presisi akan meningkatkan akurasi prediksi MPC dan berpotensi menghasilkan performa yang lebih baik, khususnya pada manuver belok dengan kecepatan tinggi.
- 5) Pengukuran konsumsi daya pada penelitian selanjutnya dapat diperbuat lebih granular dengan memasang sensor INA219 secara terpisah pada jalur daya motor drive dan jalur daya komputasi (Raspberry Pi 5 dan LiDAR). Pemisahan ini akan memberikan gambaran yang lebih jelas tentang seberapa besar *overhead* komputasi MPC berkontribusi terhadap total konsumsi daya sistem.
- 6) Implementasi sistem dapat dikembangkan ke arah ROS 2 (Humble atau Jazzy) yang lebih modern, mengingat ROS 1 Noetic telah mencapai *End of Life* pada Mei 2025. Migrasi ke ROS 2 juga membuka peluang pemanfaatan fitur-fitur terbaru seperti real-time scheduling dan komunikasi DDS yang lebih handal, yang berpotensi meningkatkan konsistensi performa MPC pada siklus kontrol yang lebih presisi.

Penelitian lanjutan dapat mempertimbangkan penggabungan MPC dengan metode pembelajaran mesin (machine learning), misalnya menggunakan neural network untuk memprediksi hambatan atau mempelajari model kinematik robot secara adaptif. Pendekatan hybrid ini berpotensi menghasilkan kontroler yang lebih robust terhadap variasi lingkungan tanpa memerlukan tuning manual yang intensif.



© 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





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

- [1] I. Nirmala and R. Hidayati, "Sistem Navigasi Otonom Robot Mobil Berbasis Ros Pada Robot Penggerak Diferensial," *INFOTECH J.*, vol. 10, no. 2, pp. 288–296, 2024, doi: 10.31949/infotech.v10i2.11219.
- [2] M. Sobirin and A. S. Hidayat, "Rancang Bangun Robot Wall Follower Dengan Kendali PID Berbasis Sensor Ultrasonik," *J. Kaji. Tek. Elektro*, vol. 10, no. 1, pp. 79– 87, 2025, doi: 10.52447/jkte.v10i1.7839.
- [3] B. D. Perkuliahan, "Ok Um Ok Um," vol. 23221040, 2024.
- [4] T. Bowo Setiyo, A. Triwiyatno, and Sumardi, "Perancangan Model Predictive Control (Mpc) Pada Proses Quadruple Tank," *Transient*, vol. 2, pp. 565–572, 2013.
- [5] B. R. Putra, A. Habibi, and T. Alawiy, "Rancang Bangun Penstabilan Robot Berkaki Wall - Following Menggunakan Sensor Gp2Y0a21 Untuk Menjaga Posisi Robot Diantara Dinding," vol. 19, no. 1, pp. 146–151, 2025.
- [6] Y. S. Putri, M. Raihan, R. Ardandi, Y. Dwi, and S. P. Aulia, "Perbandingan Kinerja Kontrol MPC dan PID pada Navigasi Robot *Turtlebot3* Menggunakan Simulasi Gazebo dan Rviz Abstrak," vol. 4, no. 1, pp. 599–606.
- [7] T. Informasi, "Penghindaran Rintangan Otomatis Pada AgenOtonomBerbasis End-to- End Deep Imitation Learning," vol. 5, pp. 264–269, 2023.
- [8] D. Erlangga, D. Endang, H. S. Rosalia, Sunarto, and T. S. K. Rahardjo, "Sistem Navigasi Mobile Robot dalam Ruangan Berbasis Autonomous Navigation," *Jurnal MEM*, 2019.
- [9] D. Santoso, A. Nurhadi, and P. Wijaya, "Penerapan Model Predictive Control pada Robot Mobile untuk Peningkatan Efisiensi Navigasi," *Jurnal Robotika Cerdas Indonesia*, 2023.



© 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

- [10] T. Rahman and B. Kurniawan, "Perbandingan Algoritma Navigasi Mobile Robot Menggunakan ROS," *Jurnal RESTI*, 2022.
- [11] I. Papadimitriou et al., "Comparative Analysis of PID and MPC on Differential-Drive Mobile Robots," *IEEE Access*, 2021.
- [12] D. Lee and W. Kim, "Autonomous Navigation of *Turtlebot3* Using ROS and LiDAR," *Sensors*, 2022.
- [13] M. Körner et al., "Model Predictive Control for Collision-Free Navigation of Mobile Robots," *Robotics and Autonomous Systems*, 2023. doi: 10.1016/j.robot.2023.104332
- [14] F. Muhammad, "Literature Review Sistem Navigasi Autonomous Mobile Robot," *Jurnal Ilmiah Sistem Informasi*, vol. 17, no. 2, 2021.
- [15] I. Papadimitriou et al., "Comparative Analysis of PID and MPC on Differential-Drive Mobile Robots," *IEEE Access*, 2021.
- [16] R. Pratama and H. Firmansyah, "Tren Perkembangan Robot Otonom dan Aplikasinya di Indonesia," *Jurnal Sistem dan Informatika*, vol. 14, no. 1, 2022.
- [17] T. Rahman and B. Kurniawan, "Perbandingan Algoritma Navigasi Mobile Robot Menggunakan ROS," *Jurnal RESTI*, 2022.
- [18] U. C. Patkar, V. Mandhalkar, A. Chavan, S. Songire, dan H. Kothawade, "Robot Operating System: A Comprehensive Analysis and Evaluation," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 7s, pp. 516–520, 2024.
- [19] A. Salam, "Literature Review Sistem Navigasi Autonomous Mobile Robot Berbasis ROS," *Jurnal Ilmiah Setrum*, vol. 10, no. 1, 2024.
- [20] D. Santoso, A. Nurhadi, and P. Wijaya, "Penerapan Model Predictive Control pada Robot Mobile untuk Efisiensi Navigasi," *Jurnal Robotika Cerdas Indonesia*, vol. 2, no. 1, 2023.



© 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

- [21] M. Ubaidillah et al., “Implementasi ROS pada Mobile Robot untuk Navigasi Otonom di Lingkungan Indoor,” *Jurnal Teknologi dan Sistem Komputer*, 2023.
- [22] P. Rezeck, H. Azpúrua, M. F. S. Corrêa, dan L. Chaimowicz, “HeRo 2.0: A Low-Cost Robot for Swarm Robotics Research,” arXiv preprint, arXiv:2202.12391, 2022.
- [23] ROBOTIS, “TurtleBot3 e-Manual: Specifications,” 2021. [Online]. Tersedia: <https://emanual.robotis.com/docs/en/platform/turtlebot3/specifications/>
- [24] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, Introduction to Autonomous Mobile Robots, 2nd ed. Cambridge, MA: MIT Press, 2011.
- [25] J. B. Rawlings, D. Q. Mayne, and M. Diehl, Model Predictive Control: Theory, Computation, and Design, 2nd ed. Madison, WI: Nob Hill Publishing, 2017.
- [26] G. Klančar and I. Škrjanc, “Tracking-error model-based predictive control for mobile robots in real time,” *Robotics and Autonomous Systems*, vol. 55, no. 6, pp. 460–469, 2007.
- [27] F. Kühne, W. F. Lages, and J. M. G. da Silva Jr., “Model predictive control of a mobile robot using linearization,” in *Proc. Mechatronics and Robotics*, 2004.
- [28] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, “A stable tracking control method for an autonomous mobile robot,” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 1990, pp. 384–389.
- [29] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi: a software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [30] N. Vinukonda, R. Madduri, R. Chadha, and V. Desai, “TurtleBot Path Tracking using PID Controller,” Project Report, 2020.



LAMPIRAN

Lampiran Kode Program

Kode Program MPC

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

class MPCController:
    def __init__(self, N=15, dt=0.1):
        self.N = N # Prediction horizon
        self.dt = dt # Time step

        # State: [x, y, theta]
        self.nx = 3
        # Control: [v, w]
        self.nu = 2

        # Model parameters
        self.L = 0.16 # Wheelbase (approximate for TurtleBot3 Burger)

        # Constraints
        self.v_min = -0.22
        self.v_max = 0.22
        self.w_min = -2.84
        self.w_max = 2.84

        # Cost weights
        self.Q = np.diag([10.0, 10.0, 1]) # State error
        self.R = np.diag([0.1, 0.1]) # Control effort
        self.Qf = np.diag([20.0, 20.0, 10.0]) # Final state error

        # Obstacle parameters
        self.obstacle_cost_weight = 100.0
        self.obstacle_threshold = 0.5

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

        # Define dynamics
        xdot = vertcat(
```

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 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

```
u[0] * cos(x[2]),
u[0] * sin(x[2]),
u[1]
)

# Continuous time dynamics
self.f = Function('f', [x, u], [xdot])

# Create optimization problem
opti = Opti()

# Decision variables
X = opti.variable(self.nx, self.N+1)
U = opti.variable(self.nu, self.N)

# Parameters
x0 = opti.parameter(self.nx)
x_ref = opti.parameter(self.nx)
obstacles = opti.parameter(2, 3) # Max 3 obstacles (x,y,radius)

# Cost function
cost = 0

for k in range(self.N):
    # Running cost
    state_error = X[:,k] - x_ref
    cost += state_error.T @ self.Q @ state_error
    cost += U[:,k].T @ self.R @ U[:,k]

    # Obstacle avoidance cost
    for i in range(obstacles.size1()):
        obs_x = obstacles[i,0]
        obs_y = obstacles[i,1]
        obs_r = obstacles[i,2]

        dist = sqrt((X[0,k]-obs_x)**2 + (X[1,k]-obs_y)**2)
        cost += self.obstacle_cost_weight * exp(-(dist - obs_r)**2)

    # Dynamics constraint
    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)

# Constraints
opti.subject_to(X[:,0] == x0)
```



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

```
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)

# Solver options
opts = {'ipopt.print_level': 0, 'print_time': 0}
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):
    return self.solver(x0, x_ref, obstacles)

class TurtleBot3MPCNavigator:
    def __init__(self):
        rospy.init_node('turtlebot3_mpc_navigator', anonymous=True)

        # MPC Controller
        self.mpc = MPCController(N=10, dt=0.1)
        self.mpc.create_solver()

        # Robot state
        self.current_pose = None
        self.goal_pose = None
        self.scan_data = None
        self.obstacles = []

        # Publishers and Subscribers
        self.cmd_vel_pub = rospy.Publisher('/cmd_vel', Twist, queue_size=10)
        rospy.Subscriber('/odom', Odometry, self.odom_callback)
        rospy.Subscriber('/scan', LaserScan, self.scan_callback)
        rospy.Subscriber('/move_base_simple/goal', PoseStamped, self.goal_callback)

        self.rate = rospy.Rate(10)

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

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

    def goal_callback(self, msg):
        self.goal_pose = msg.pose

    def update_obstacles(self):
```




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

```
if self.scan_data is None or self.current_pose is None:
    return

self.obstacles = []

# Convert quaternion to Euler angles
q = self.current_pose.orientation
siny_cosp = 2 * (q.w * q.z + q.x * q.y)
cosy_cosp = 1 - 2 * (q.y * q.y + q.z * q.z)
theta = math.atan2(siny_cosp, cosy_cosp)

# Detect obstacles from laser scan
for i in range(len(self.scan_data.ranges)):
    if self.scan_data.ranges[i] < self.mpc.obstacle_threshold:
        angle = self.scan_data.angle_min + i * self.scan_data.angle_increment

        # Convert to global frame
        obs_x = self.current_pose.position.x + self.scan_data.ranges[i] * math.cos(theta
+ angle)
        obs_y = self.current_pose.position.y + self.scan_data.ranges[i] * math.sin(theta
+ angle)

        self.obstacles.append([obs_x, obs_y, 0.2]) # [x, y, radius]

# Limit number of obstacles to consider
if len(self.obstacles) >= 3:
    break

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

    # Convert quaternion to Euler angles
    q = self.current_pose.orientation
    siny_cosp = 2 * (q.w * q.z + q.x * q.y)
    cosy_cosp = 1 - 2 * (q.y * q.y + q.z * q.z)
    theta = math.atan2(siny_cosp, cosy_cosp)

    return np.array([x, y, theta])

def get_goal_state(self):
    if self.goal_pose is None:
        return None

    x = self.goal_pose.position.x
    y = self.goal_pose.position.y
```



© 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

```
# For goal, we don't care about orientation (point towards goal)
dx = x - self.current_pose.position.x
dy = y - self.current_pose.position.y
theta = math.atan2(dy, dx)

return np.array([x, y, theta])

def navigate(self):
    while not rospy.is_shutdown():
        if self.current_pose is None or self.goal_pose is None:
            self.rate.sleep()
            continue

        # Get current and goal states
        x0 = self.get_current_state()
        x_ref = self.get_goal_state()

        # Prepare obstacles (pad with zeros if less than 3)
        obstacles = np.zeros((3, 3))
        for i in range(min(len(self.obstacles), 3)):
            obstacles[i,:] = self.obstacles[i]

        # Solve MPC problem
        u_opt, _ = self.mpc.solve(x0, x_ref, obstacles)

        # Apply control
        cmd_vel = Twist()
        cmd_vel.linear.x = float(u_opt[0])
        cmd_vel.angular.z = float(u_opt[1])

        self.cmd_vel_pub.publish(cmd_vel)
        self.rate.sleep()

if __name__ == '__main__':
    try:
        navigator = TurtleBot3MPCNavigator()
        navigator.navigate()
    except rospy.ROSInterruptException:
        pass
```



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

Kode Program PID

```
#!/usr/bin/env python3

=====
TURTLEBOT3 PURE PID NAVIGATOR (PATH FOLLOWER MODE)
Versi: Disesuaikan untuk TurtleBot3 (OpenCR menangani Low-Level Motor)
=====

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 PID_Controller:
    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 TB3Navigator:
    def __init__(self):
        rospy.init_node('tb3_pure_pid_navigator', anonymous=True)

        # 1. HIGH-LEVEL PID (Navigasi)
        # Nilai ini bisa Anda tuning ulang via topic /tune_nav_linear & angular
        self.linear_pid = PID_Controller(kp=0.91, ki=0.07, kd=0.10, integral_max=1.0)
        self.angular_pid = PID_Controller(kp=1.91, ki=0.07, kd=0.09, integral_max=1.5)

        # 2. KINEMATIKA TURTLEBOT3 (Default diset ke spesifikasi aman TB3 Burger)
        # Jika menggunakan Waffle Pi, ubah v_max ke 0.26 dan w_max ke 1.82
        self.v_max = 0.22
```




© 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

```
self.v_min = -0.22
self.w_max = 2.84
self.w_min = -2.84
self.min_w_abs = 0.1 # Sedikit diturunkan agar TB3 lebih mulus

# Akselerasi diturunkan agar pergerakan Dynamixel tidak menyentak (jerk)
self.acc_x = 1.0
self.acc_theta = 2.0

self.lookahead_distance = 0.28

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.cmd_vel_murni_pub = rospy.Publisher('/cmd_vel_pid_murni', 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/DWAPlannerROS/global_plan', Path, queue_size=2)
self.clear_local_pub = rospy.Publisher('/move_base/DWAPlannerROS/local_plan',
Path, queue_size=2)

rospy.Subscriber('/odom', Odometry, self.odom_callback)
rospy.Subscriber('/scan', LaserScan, self.scan_callback)
rospy.Subscriber('/move_base/DWAPlannerROS/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)
self.sleep(1.0)

rospy.loginfo("\n=====")
rospy.loginfo(" [ NAVIGATOR PID ] AKTIF UNTUK TURTLEBOT3 ")
rospy.loginfo("=====\\n")

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()
    rospy.loginfo(f"Linear PID Update: P={msg.x}, I={msg.y}, D={msg.z}")

def tune_angular_cb(self, 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
self.angular_pid.kp, self.angular_pid.ki, self.angular_pid.kd = msg.x, msg.y, msg.z
self.angular_pid.reset()
rospy.loginfo(f'Angular PID Update: P={msg.x}, I={msg.y}, D={msg.z}')

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):
    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):
        # LIDAR LDS-01/02 TurtleBot3 biasanya berjarak 0.12 - 3.5 meter
        if not (0.12 < r < 3.5): 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)
```




© 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

```

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

        # Toleransi jarak sampai ke tujuan (dalam meter)
        if dist_to_final <= 0.15:
            self.cmd_vel_pub.publish(Twist())
            self.cancel_pub.publish(GoalID())

            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

```




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

```
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()

# NOTE: Pada script awal, fitur obstacle di matikan secara paksa (False).
# Hapus atau ubah baris di bawah jika ingin mengaktifkan deteksi LIDAR
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):
    target_v = 0.0
    self.linear_pid.reset()
else:
    target_v = target_v * max(0.1, 1.0 - (abs(head_err) / (math.pi/4)))

# Pembatasan Kecepatan Kinematik
target_v = max(min(target_v, self.v_max), self.v_min)
target_w = max(min(target_w, self.w_max), self.w_min)

# Pembatasan Akselerasi Kinematik
max_dv = self.acc_x * dt
max_dw = self.acc_theta * dt
```



© 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

```
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_v
cmd.angular.z = cmd_w

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

if __name__ == '__main__':
    try:
        TB3Navigator().navigate()
    except rospy.ROSInterruptException:
        pass
```

**POLITEKNIK
NEGERI
JAKARTA**