

Implementasi Potential Field pada Robot Omnidirectional Berbasis LiDAR Menggunakan ROS

Ari Prayudi* , Elang Derdian Marindani, F. Trias Pontia W.
Universitas Tanjungpura, Indonesia
Email: ariprayudi19@gmail.com*

Abstrak

Sistem navigasi otonom dikembangkan untuk robot mobile omnidirectional dengan memanfaatkan algoritma Potential Field yang terintegrasi dengan data sensor LiDAR. Penelitian ini bertujuan untuk mengimplementasikan algoritma Potential Field pada robot omnidirectional berbasis ROS guna mencapai navigasi yang akurat dan responsif dalam lingkungan statis maupun dinamis, serta mengoptimalkan penghindaran rintangan secara real-time. Implementasi menggunakan Robot Operating System (ROS) memungkinkan koordinasi modular antara algoritma kendali, masukan sensor, dan aktuator. Metode penelitian meliputi pemodelan kinematik robot, integrasi data LiDAR untuk deteksi rintangan, dan penerapan medan gaya atraktif-repulsif dalam algoritma Potential Field. Hasil pengujian menunjukkan bahwa robot dapat mencapai titik tujuan baik dalam kondisi tanpa halangan maupun dengan halangan, dengan rata-rata galat posisi akhir di bawah 40 mm dan waktu tempuh rata-rata sekitar 12 detik. Sensor LiDAR menunjukkan akurasi tinggi, dengan galat pengukuran jarak pendek sebesar 2,23% dan jarak jauh sebesar 5,38%. Delay pada topik ROS untuk data LiDAR dan odometri masing-masing rata-rata sebesar 0,50 detik dan 0,08 detik, yang cukup untuk mendukung kendali waktu nyata. Hasil penelitian ini memiliki implikasi signifikan dalam pengembangan sistem robotik otonom, terutama untuk aplikasi logistik, manufaktur, dan lingkungan dinamis lainnya, dengan menyediakan solusi navigasi yang adaptif dan efisien secara komputasi.

kata kunci: Robot omnidirectional, Potential Field, LiDAR, ROS

Abstract

An autonomous navigation system was developed for an omnidirectional mobile robot using the Potential Field algorithm integrated with LiDAR sensor data. This study aims to implement the Potential Field algorithm on an ROS-based omnidirectional robot to achieve accurate and responsive navigation in both static and dynamic environments, while optimizing real-time obstacle avoidance. The implementation leverages the Robot Operating System (ROS) to enable modular coordination between the control algorithm, sensor input, and actuator commands. The methodology includes kinematic modeling of the robot, LiDAR data integration for obstacle detection, and the application of attractive-repulsive force fields in the Potential Field algorithm. Experimental trials showed that in both obstacle-free and obstacle-present environments, the robot reached target points with average position errors below 40 mm and an average completion time of approximately 12 seconds. The LiDAR sensor demonstrated a short-range measurement error of 2.23% and a long-range error of 5.38%, indicating reliable distance estimation. ROS topic delays for LiDAR and odometry data averaged 0.50 and 0.08 seconds, respectively, enabling real-time feedback and control. The findings highlight the practical implications of this research for autonomous robotic systems, particularly in logistics, manufacturing, and dynamic environments, by offering a computationally efficient and adaptive navigation solution.

keywords: Omnidirectional robo, Potential Field, LiDAR, ROS

PENDAHULUAN

Robot mobile omnidirectional merupakan salah satu jenis robot yang memiliki keunggulan dalam hal mobilitas, karena mampu bergerak ke segala arah tanpa perlu melakukan rotasi pada sumbu utama tubuhnya. Kemampuan ini memberikan fleksibilitas tinggi, terutama untuk aplikasi navigasi dalam ruang sempit atau lingkungan kompleks, seperti gudang, rumah sakit, dan pabrik dengan tata letak dinamis. Dalam sistem navigasi otonom, dua tantangan utama yang harus dihadapi adalah perencanaan jalur (path planning) dan penghindaran rintangan (obstacle avoidance) secara real-time. Untuk itu, diperlukan metode kendali yang tidak hanya tanggap terhadap perubahan lingkungan, tetapi juga efisien dan ringan secara komputasi (Kacprzyk, 2017).

Salah satu pendekatan yang banyak digunakan untuk navigasi reaktif adalah algoritma Potential Field. Algoritma ini bekerja dengan memodelkan target sebagai gaya tarik dan rintangan sebagai gaya tolak, sehingga menghasilkan lintasan yang seolah-olah terbentuk oleh medan gaya (Khatib, 1986; Ge & Cui, 2002). Metode ini populer karena kesederhanaannya dan efisiensi komputasinya dalam lingkungan dinamis (Zhu et al., 2015). Namun, salah satu kelemahan utama dari algoritma ini adalah kemungkinan terjebak dalam titik lokal minimum (Sharma & Kapoor, 2020). Untuk mengatasi hal ini, berbagai pendekatan hibrida telah dikembangkan dengan menggabungkan algoritma potential field dengan metode lain seperti algoritma genetika atau fuzzy logic (Liu et al., 2019). Robot dapat merespons secara langsung terhadap keberadaan rintangan tanpa perlu membangun peta global secara eksplisit (Wang et al., 2017). Sensor LiDAR (Light Detection and Ranging) digunakan sebagai komponen utama untuk mendeteksi lingkungan sekitar robot dengan tingkat ketelitian tinggi (Yin et al., 2021). Sensor ini memungkinkan akuisisi data spasial secara real-time, sehingga sangat cocok untuk navigasi robot di lingkungan kompleks (Sun et al., 2020).

LiDAR memberikan informasi jarak secara real-time terhadap objek-objek di sekeliling robot, yang selanjutnya dimanfaatkan dalam algoritma penghindaran rintangan. Untuk mengintegrasikan berbagai komponen sistem seperti sensor, aktuator, dan kendali, digunakan Robot Operating System (ROS) sebagai kerangka kerja utama. ROS memungkinkan pengembangan sistem secara modular dan terdistribusi, mendukung komunikasi antar komponen melalui sistem publish-subscribe, dan menyediakan pustaka yang relevan untuk pengembangan aplikasi robotik (Herath & St-Onge Eds, 2022).

Beberapa penelitian sebelumnya telah mengembangkan berbagai metode untuk navigasi robot omnidirectional. Misalnya, Rijalusalam & Iswanto (2021) memodelkan kinematik robot omnidirectional dengan empat roda dan mengimplementasikan odometri untuk estimasi posisi. Sementara itu, Wahab (2022) menggunakan metode Potential Field untuk navigasi robot roda omni dengan fokus pada penghindaran rintangan. Namun, penelitian tersebut masih memiliki keterbatasan dalam hal akurasi dan responsivitas terhadap lingkungan yang dinamis. Alatise & Hancke (2020) juga menyoroti tantangan dalam integrasi sensor LiDAR untuk deteksi rintangan yang akurat, terutama pada jarak jauh.

Berdasarkan penelitian terdahulu, terdapat beberapa celah penelitian (*research gap*) yang perlu diatasi. Pertama, akurasi dan *real-time processing*: sebagian besar penelitian belum sepenuhnya mengoptimalkan akurasi navigasi dan kecepatan pemrosesan data sensor dalam lingkungan dinamis. Kedua, integrasi sensor dan algoritma: integrasi antara sensor LiDAR dan algoritma *Potential Field* pada platform ROS masih memerlukan pengujian lebih mendalam untuk memastikan kinerja yang stabil. Ketiga, adaptabilitas lingkungan: kemampuan robot untuk beradaptasi dengan lingkungan yang berubah-ubah, seperti rintangan bergerak atau tata letak yang kompleks, masih menjadi tantangan.

Penelitian ini menawarkan beberapa kebaruan (*novelty*). Pertama, implementasi ROS modular: penggunaan Robot Operating System (ROS) untuk mengintegrasikan sensor LiDAR,

algoritma *Potential Field*, dan kinematik robot secara modular, sehingga memungkinkan skalabilitas dan fleksibilitas dalam pengembangan. Kedua, optimasi *Potential Field*: modifikasi algoritma *Potential Field* untuk meningkatkan responsivitas terhadap rintangan statis dan dinamis, dengan memanfaatkan data LiDAR yang akurat. Ketiga, evaluasi komprehensif: pengujian kinerja sistem secara menyeluruh, termasuk akurasi posisi, waktu tempuh, dan *delay* pemrosesan data, untuk memastikan keandalan dalam berbagai skenario navigasi.

Penelitian ini bertujuan untuk mengimplementasikan algoritma Potential Field pada robot omnidirectional berbasis ROS dengan menggunakan sensor LiDAR sebagai sumber utama informasi lingkungan. Fokus utama diarahkan pada pengujian kemampuan robot dalam melakukan navigasi dan penghindaran rintangan secara langsung, serta evaluasi kinerja sistem melalui parameter akurasi posisi dan waktu tempuh pada berbagai kondisi lingkungan.

Hasil penelitian ini diharapkan dapat memberikan manfaat di berbagai bidang. Di bidang akademik, penelitian ini dapat memberikan kontribusi dalam pengembangan algoritma navigasi robot yang lebih efisien dan adaptif. Di bidang industri, penelitian ini dapat menyediakan solusi praktis untuk aplikasi robot mobile di lingkungan industri, seperti logistik otomatis dan manufaktur. Sementara itu, di bidang teknologi robotik, penelitian ini dapat memperluas penerapan integrasi sensor LiDAR dan ROS dalam pengembangan sistem robotik otonom.

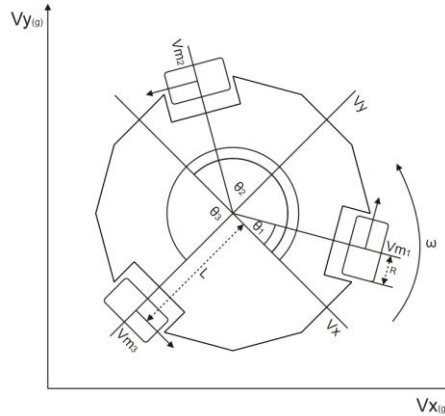
METODE PENELITIAN

ROS

Robot Operating System (ROS) merupakan *framework* perangkat lunak open-source yang dirancang untuk memudahkan pengembangan sistem robotik secara modular. ROS menyediakan berbagai pustaka dan alat bantu yang memungkinkan komponen robot berkomunikasi satu sama lain melalui arsitektur berbasis node. Setiap node menjalankan fungsi tertentu, seperti membaca sensor atau mengendalikan motor, dan dapat saling berkomunikasi melalui saluran bernama *topic*. Komunikasi antar node dilakukan dengan format data standar yang disebut *message*. Dalam sistem ini, node yang mengirim data disebut *publisher*, sedangkan node yang menerima data disebut *subscriber*. Seluruh proses komunikasi dikelola oleh *master*, yang berfungsi mengatur koneksi antar node dan memastikan aliran data berjalan dengan baik. Dengan struktur ini, ROS mendukung pengembangan sistem robotik yang fleksibel, terdistribusi, dan real-time (Alsadik & Karam, 2021).

Kinematik

Pemodelan kinematik merupakan studi tentang pergerakan robot berdasarkan hubungan geometris antar komponen sistem, tanpa mempertimbangkan gaya, torsi, atau momen yang memengaruhinya (Rijalusalam & Iswanto, 2021). Pada robot omnidirectional, terdapat dua jenis pemodelan kinematik utama, yaitu forward kinematics dan inverse kinematics. *Inverse kinematics* digunakan untuk menghitung kecepatan masing-masing roda (V_{m1} , V_{m2} , V_{m3}) berdasarkan kecepatan gerak yang diinginkan, sedangkan *forward kinematics* digunakan untuk menentukan kecepatan linear dan rotasi robot (V_x , V_y , ω) dari kecepatan roda (Rijalusalam & Iswanto, 2021).



Gambar 1 Kinematik Robot Omni

Sumber: Diolah dari Rijalusalam & Iswanto (2021); Wahab (2022)

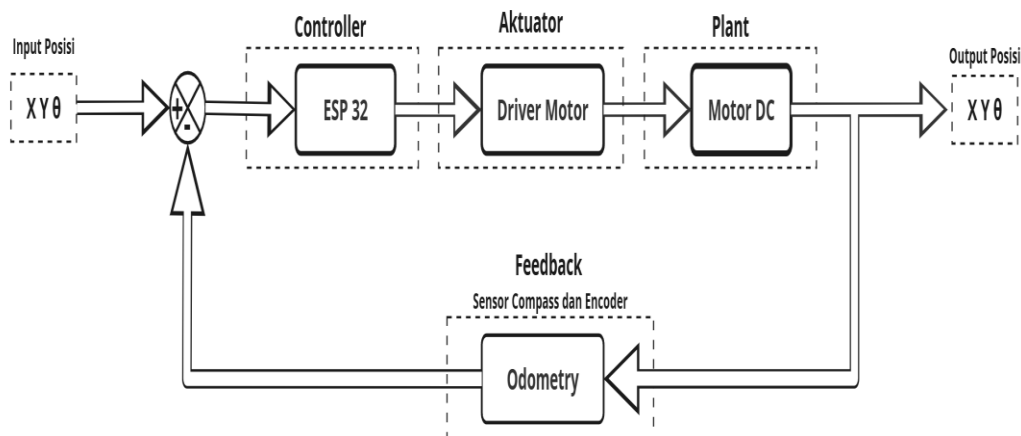
Berdasarkan Gambar 1, konfigurasi kinematik robot omnidirectional dengan tiga roda yang tersusun membentuk pola “Y” memiliki selisih sudut antar roda sebesar 120°. Dengan konfigurasi ini, masing-masing roda dipasang pada sudut $\theta_1 = 30^\circ$, $\theta_2 = 150^\circ$, dan $\theta_3 = 270^\circ$ terhadap sumbu acuan robot. Mengacu pada Persamaan 1 dan 2 yang menjelaskan model kinematik robot (Wahab, 2022), maka dapat diturunkan persamaan forward kinematics yang menggambarkan hubungan antara kecepatan setiap roda dengan kecepatan linear dan angular robot sebagai berikut:

$$\begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} = r \begin{bmatrix} \cos(\theta_1 + \frac{\pi}{2}) & \cos(\theta_2 + \frac{\pi}{2}) & \cos(\theta_3 + \frac{\pi}{2}) \\ \sin(\theta_1 + \frac{\pi}{2}) & \sin(\theta_2 + \frac{\pi}{2}) & \sin(\theta_3 + \frac{\pi}{2}) \\ \frac{1}{3L} & \frac{1}{3L} & \frac{1}{3L} \end{bmatrix} \begin{bmatrix} Vm_1 \\ Vm_2 \\ Vm_3 \end{bmatrix} \quad (1)$$

Untuk menentukan kecepatan masing-masing roda pada robot omnidirectional, dilakukan proses inversi terhadap persamaan forward kinematics. Melalui pendekatan ini, diperoleh persamaan inverse kinematics sebagai berikut:

$$\begin{bmatrix} Vm_1 \\ Vm_2 \\ Vm_3 \end{bmatrix} = r \begin{bmatrix} \cos(\theta_1 + \frac{\pi}{2}) & \sin(\theta_1 + \frac{\pi}{2}) & 1/3L \\ \cos(\theta_2 + \frac{\pi}{2}) & \sin(\theta_2 + \frac{\pi}{2}) & 1/3L \\ \cos(\theta_3 + \frac{\pi}{2}) & \sin(\theta_3 + \frac{\pi}{2}) & 1/3L \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} \quad (2)$$

Diagram Blok Sistem



Gambar 2 Diagram Blok Sistem

Sumber: Data diolah (2024)

Gambar 2 menggambarkan diagram blok sistem kendali robot omnidirectional berbasis *close-loop control*. Sistem ini menerima masukan berupa koordinat target (X, Y, θ), di mana sudut orientasi θ diset tetap pada 0 guna mempertahankan arah hadap robot selama menuju titik tujuan. Masukan ini dijadikan referensi oleh kontroler untuk menghitung kecepatan masing-masing motor melalui algoritma inverse kinematics.

Hasil perhitungan kecepatan motor (V_{m1}, V_{m2}, V_{m3}) kemudian diteruskan ke motor driver untuk mengatur putaran motor, sehingga robot bergerak menuju posisi yang diinginkan. Nilai kecepatan kemudian dikirimkan ke modul motor driver untuk menggerakkan masing-masing motor pada robot. Sistem menghasilkan keluaran berupa posisi robot terkini dalam bentuk koordinat (X, Y, θ).

Selama pergerakan berlangsung, sensor encoder pada tiap motor serta sensor kompas mendeteksi perubahan posisi dan orientasi robot. Data dari sensor ini digunakan dalam sistem odometri, yang mengestimasi posisi robot dengan menggabungkan informasi rotasi roda dan sudut arah. Estimasi ini menjadi umpan balik (*feedback*) bagi sistem kendali, sehingga gerakan robot dapat disesuaikan secara dinamis untuk mencapai presisi dalam navigasi.

Odometri Robot

Penggunaan odometri dalam sistem kendali robot omnidirectional bertujuan untuk memperkirakan perubahan posisi dan orientasi robot selama bergerak. Untuk meningkatkan akurasi estimasi heading dari odometri, sistem ini didukung oleh data heading yang diperoleh dari sensor kompas. Namun demikian, pembacaan arah dari kompas memiliki keterbatasan, yakni menghasilkan nilai absolut dalam rentang 0° hingga 360° , yang bersifat tetap terhadap medan magnet bumi (Setiawan et al., 2024). Nilai ini tidak langsung sesuai dengan format sudut yang digunakan dalam sistem odometri, yang biasanya memerlukan rentang kontinu seperti -180° hingga 180° untuk representasi arah. Oleh karena itu, diperlukan modifikasi terhadap data kompas agar dapat disesuaikan dengan kebutuhan sistem, sehingga menghasilkan estimasi heading yang konsisten dengan kerangka odometri. Adapun transformasi data heading dari kompas dilakukan dengan menggunakan persamaan sebagai berikut:

$$\theta_{(0^\circ \leq n \leq 360^\circ)} = (\theta - \theta_0 + 360^\circ) \% 360^\circ \quad (3)$$

Pada persamaan tersebut, θ merepresentasikan data sudut yang diperoleh dari sensor kompas, sedangkan θ_0 merupakan nilai sudut awal yang dibaca saat sistem pertama kali diaktifkan. Dengan demikian, sistem mengkalibrasi data heading kompas sehingga sudut awal robot selalu dianggap sebagai 0° , memungkinkan perhitungan orientasi relatif terhadap posisi awal.

Namun, agar sudut yang dihasilkan oleh sensor kompas dapat digunakan secara konsisten dalam perhitungan odometri, nilai heading perlu dikonversi dari rentang 0° hingga 360° menjadi rentang -180° hingga 180° . Transformasi ini penting untuk menjaga kontinuitas sudut dan menghindari diskontinuitas saat robot melintasi arah utara ($0^\circ/360^\circ$). Oleh karena itu, dilakukan modifikasi terhadap data kompas sehingga diperoleh persamaan transformasi berikut:

$$\theta_{(0^\circ \leq n \leq 360^\circ)} = \begin{cases} -\theta & \theta < 180^\circ \\ 360^\circ - \theta & \theta \geq 180^\circ \end{cases} \quad (4)$$

Dengan mempertimbangkan hasil transformasi sudut, maka yang semula digunakan untuk menghitung odometri dapat dimodifikasi agar sesuai dengan data heading dari kompas yang telah disesuaikan. Hasil modifikasi tersebut menghasilkan bentuk persamaan baru sebagai berikut:

$$X = \cos(\theta_1 + \frac{\pi}{2} + \theta)R_1 + \cos(\theta_2 + \frac{\pi}{2} + \theta)R_2 + \cos(\theta_3 + \frac{\pi}{2} + \theta)R_3 \quad (5)$$

$$Y = \sin(\theta_1 + \frac{\pi}{2} + \theta)R_1 + \sin(\theta_2 + \frac{\pi}{2} + \theta)R_2 + \sin(\theta_3 + \frac{\pi}{2} + \theta)R_3 \quad (6)$$

$$\theta = \text{Data Kompas } (\theta) \quad (7)$$

Data heading yang diperoleh dari kompas merepresentasikan orientasi dalam kerangka lokal robot. Untuk mengintegrasikan data ini ke dalam sistem koordinat global, diperlukan transformasi melalui matriks rotasi. Dengan mengacu pada rotasi dari kerangka lokal ke kerangka global dapat dilakukan, sehingga diperoleh persamaan transformasi sebagai berikut:

$$\begin{bmatrix} X_* \\ Y_* \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} \quad (8)$$

Metode Potensial Field

Metode *Potential Field* pertama kali diperkenalkan oleh Khatib pada tahun 1986, yang membagi medan gaya menjadi dua komponen utama, yaitu medan atraktif (*attractive field*) dan medan repulsif (*repulsive field*). Medan atraktif berperan dalam menarik robot menuju posisi tujuan, sedangkan medan repulsif menghasilkan gaya tolak untuk menjauhkan robot dari rintangan di sekitarnya. Kombinasi dari kedua medan ini memungkinkan robot untuk bergerak secara otonom menuju target sambil secara aktif menghindari Mengacu pada formulasi metode *Potential Field* (Alatise & Hancke, 2020), maka untuk komponen medan atraktif, diperoleh persamaan sebagai berikut:

$$[X_\alpha \ Y_\alpha]^T = \begin{cases} [0 \ 0]^T & d < r \\ \begin{bmatrix} \alpha(d-r) \cos \theta \\ \alpha(d-r) \sin \theta \end{bmatrix} & r \leq d \leq s+r \\ \begin{bmatrix} \alpha \cos \theta \\ \alpha \sin \theta \end{bmatrix} & d > s+r \end{cases} \quad (9)$$

Untuk menghitung jarak antara robot dan posisi tujuan, digunakan persamaan berikut.

$$d = \sqrt{(X_{tujuan} - X_{aktual})^2 + (Y_{tujuan} - Y_{aktual})^2} \quad (10)$$

Untuk menentukan sudut robot dengan posisi tujuan, diperoleh persamaan berikut.

$$\theta = \tan^{-1} \frac{Y_{tujuan} - Y_{aktual}}{X_{tujuan} - X_{aktual}} \quad (11)$$

Untuk persamaan medan *repulsive* menghindari rintangan diperoleh persamaan berikut.

$$[X_r \ Y_r]^T = \begin{cases} -\delta \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} & d < r \\ -\beta(s+r-d) \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} & r \leq d \leq s+r \\ [0 \ 0]^T & d > s+r \end{cases} \quad (12)$$

Untuk menentukan sudut robot dengan posisi rintangan, diperoleh persamaan berikut.

$$d = \sqrt{(X_{halangan} - X_{aktual})^2 + (Y_{halangan} - Y_{aktual})^2} \quad (13)$$

Untuk menentukan sudut robot dengan posisi rintangan, diperoleh persamaan berikut.

$$\theta = \tan^{-1} \frac{Y_{halangan} - Y_{aktual}}{X_{halangan} - X_{aktual}} \quad (14)$$

Dengan menjumlahk gaya, didapatkan persamaan gaya dari *potential field*, di mana F_x adalah gaya pada koordinat X dan F_y adalah gaya pada koordinat Y. Sehingga diperoleh persamaan sebagai berikut.

$$F_x = X_r + X_\alpha \quad (15)$$

$$F_y = Y_r + Y_\alpha \quad (16)$$

Penerapan LiDAR Terhadap Potential field

Dalam penelitian ini RPLiDAR digunakan untuk mendapatkan data halangan dari robot dimana jarak ditentukan dengan mengalikan kecepatan cahaya dalam suatu medium dengan waktu yang dibutuhkan pulsa cahaya untuk menempuh jarak ke target. Karena kecepatan cahaya konstan, maka jarak ke objek berbanding lurus dengan waktu yang ditempuh (Nikmah et al., 2024).

$$d = \frac{c \times t}{2} \quad (17)$$

Pada persamaan 3.8, dapat diperoleh jarak dari LiDAR ke halangan robot dengan konfigurasi di mana letak LiDAR berada pada pusat robot. Berdasarkan konsep *potential field*, data yang diperlukan untuk menghindari halangan adalah data dari titik halangan.

Sedangkan data yang diperoleh dari LiDAR adalah data jarak dan sudut ke objek halangan. Maka didapatkan persamaan berikut.

$$\begin{bmatrix} X_o \\ Y_o \end{bmatrix} = \begin{bmatrix} X \\ Y \end{bmatrix} + d_{(n)} \begin{bmatrix} \cos(\theta_{(n)}) \\ \sin(\theta_{(n)}) \end{bmatrix} \quad (18)$$

Berdasarkan (X,Y) merupakan koordinat dari robot aktual, (X_o, Y_o) merupakan koordinat dari halangan d adalah jarak dari pusat robot ke halangan, θ adalah sudut yang diukur oleh LiDAR, dan n adalah jumlah data jarak yang diperoleh dari LiDAR.

Untuk meningkatkan pelacakan halangan terhadap robot, digunakan sensor kompas agar ketika robot berotasi, sudut yang diukur oleh LiDAR dapat mengikuti perubahan rotasi robot. Dengan demikian, perubahan sudut dari LiDAR sebanding dengan rotasi robot. Maka didapatkan persamaan berikut.

$$\begin{bmatrix} X_o \\ Y_o \end{bmatrix} = \begin{bmatrix} X \\ Y \end{bmatrix} + d_{(n)} \begin{bmatrix} \cos(\theta_{(n)} + \theta) \\ \sin(\theta_{(n)} + \theta) \end{bmatrix} \quad (19)$$

HASIL DAN PEMBAHASAN

Pengujian Sensor LiDAR

Pengujian ini dilakukan untuk mengukur jarak minimal dan maksimal yang dapat dideteksi oleh sensor RPLiDAR A1M8-R6 terhadap objek. Data yang diuji mengacu pada spesifikasi yang tercantum dalam datasheet sensor.

Tabel 1 Pengukuran Jarak Minimal

Pengujian	Jarak (m)	Jarak Terbaca (m)	Error (%)
1	0,05	0,00	-
2	0,15	0,16	6,67
3	0,25	0,26	4,00
4	0,35	0,35	0,00
5	0,45	0,47	4,44

Pengujian	Jarak (m)	Jarak Terbaca (m)	Error (%)
6	0,55	0,55	0,00
7	0,65	0,64	1,54
8	0,75	0,75	0,00
9	0,85	0,86	1,18
10	0,95	0,97	2,11
Rata - rata error (%)			2,23

Sumber: Hasil pengujian eksperimen (2024)

Pada Tabel 1 data hasil pembacaan jarak LiDAR dengan rentang jarak pengujian antara 0,05 – 0,95 m, dengan rata-rata persentase error pembacaan adalah 2,23%. Ketika LiDAR mengukur jarak yang < dari 0,15 m sensor tidak dapat membaca jarak dari objek. Sesuai dengan datasheet, sensor memiliki pengukuran jarak minimal $\pm 0,15$ m.

Tabel 2 Pengukuran Jarak Maksimal

Pengujian	Jarak (m)	Jarak Terbaca (m)	Error (%)
1	1,00	0,98	2,00
2	2,00	2,02	1,00
3	3,00	3,04	1,33
4	4,00	4,09	2,25
5	5,00	5,16	3,20
6	6,00	6,28	4,67
7	7,00	7,34	4,86
8	8,00	8,47	5,88
9	9,00	9,64	7,11
10	10,00	10,69	6,90
11	11,00	12,47	13,36
12	12,00	13,44	12,00
Rata - rata error (%)			5,38

Sumber: Hasil pengujian eksperimen (2024)

Pada Tabel 2, data hasil pembacaan jarak LiDAR menunjukkan bahwa dalam rentang pengujian antara 1 hingga 12 meter, rata-rata *error* pembacaan adalah 5,38 %. Ketika LiDAR mengukur jarak ≤ 3 meter, hasil pembacaannya mendekati jarak sebenarnya. Namun, pada jarak 3 - 5 meter, akurasi mulai menurun, dan pada jarak 5 - 25 meter, akurasi semakin berkurang.

Sesuai dengan spesifikasi dalam datasheet, akurasi LiDAR mencapai $\pm 1\%$ dari jarak yang diukur untuk jarak ≤ 3 meter. Pada jarak 3 - 5 meter, akurasi menurun menjadi $\pm 2\%$, sedangkan pada jarak 5 - 25 meter, akurasi semakin berkurang menjadi $\pm 2,5\%$ dari jarak yang diukur. Hal ini menunjukkan bahwa semakin jauh jarak yang diukur, semakin besar persentase *error* dalam pembacaan.

Pengujian Sistem ROS

Pengujian sistem dilakukan dengan mengukur delay pada topik */scan* (data LiDAR) dan */odom* (data *odometri*) untuk mengevaluasi kinerja sistem ROS. Pengukuran delay ini bertujuan untuk menganalisis keterlambatan

Tabel 3. Data Pengujian Delay Data LiDAR

Pengujian	Delay (s)	Jumlah Data
1	0,26	55
2	0,28	107

3	0,26	150
4	0,28	200
5	0,64	256
6	0,73	304
7	0,74	354
8	0,67	404
9	0,62	454
10	0,58	505
Rata - rata	0,50	278,90

Sumber: Hasil simulasi dan pengujian sistem (2024)

Berdasarkan Tabel 3, hasil pengujian *delay* pada data LiDAR menunjukkan bahwa *delay* terlama terjadi pada pengujian ketujuh, yaitu sebesar 0,74 detik dengan jumlah data 354. Sementara itu, *delay* tercepat terjadi pada pengujian pertama dan ketiga, 0,26 detik, dengan jumlah data 55 dan 150. Rata-rata *delay* dari sepuluh pengujian 0,50 detik dengan rata-rata jumlah data 278,90.

Tabel 4 Data Pengujian Delay Data Odometry

Pengujian	Delay (s)	Jumlah Data
1	0,06	50
2	0,06	100
3	0,07	150
4	0,07	200
5	0,08	250
6	0,08	300
7	0,08	350
8	0,09	401
9	0,09	451
10	0,10	501
Rata - rata	0,08	275,30

Sumber: Hasil simulasi dan pengujian sistem (2024)

Berdasarkan Tabel 4, hasil pengujian *delay* pada data *odometry* menunjukkan bahwa *delay* terlama terjadi pada pengujian kesepuluh, yaitu sebesar 0,10 detik dengan jumlah data 501. Sementara itu, *delay* tercepat terjadi pada pengujian pertama dan kedua, 0,06 detik, dengan jumlah data 50 dan 100. Rata-rata *delay* dari sepuluh pengujian 0,08 detik dengan rata-rata jumlah data 275,30.

Delay dari data LiDAR dan *odometry* dipengaruhi oleh kondisi jaringan saat pembacaan data berlangsung. Selain itu, Keterlambatan dapat disebabkan oleh beban komputasi sistem ROS dalam mengolah data, serta kecepatan transfer data antar node dalam sistem ROS.

Pengujian Gerak Robot Pada Sumbu +Y Tanpa Halangan

Pengujian dilakukan untuk menguji robot bergerak sepanjang sumbu +Y di lingkungan yang tanpa ada halangan, robot diuji untuk mencapai titik tujuan, yaitu $X = 0$, $Y = 3000$. Tujuan pengujian adalah untuk melihat sejauh mana robot dapat bergerak menuju posisi yang diinginkan dengan akurat. Untuk itu, data posisi robot aktual dibandingkan dengan data *odometri* pada robot.

Tabel 5 Data Pengujian Robot ($X = 0$, $Y = 3000$) Tanpa Halangan

No	Target (mm)			Terukur Nyata (mm)			Terukur Robot (mm)			Waktu (detik)
	X	Y	Jarak Euclidean	X	Y	Jarak Euclidean	X	Y	Jarak Euclidean	
1	0	3000	3000	30	3020	3020,15	40,75	3010,60	3010,88	11,36

Implementasi Potential Field pada Robot Omnidirectional Berbasis LiDAR Menggunakan ROS

2	0	3000	3000	40	3020	3020,26	-41,68	3026,89	3027,18	11,69	
3	0	3000	3000	50	3015	3015,41	-2,78	3026,87	3026,87	11,90	
4	0	3000	3000	-30	2975	2975,15	-4,66	2987,28	2987,28	11,45	
5	0	3000	3000	50	3030	3030,41	13,35	3014,85	3014,88	11,71	
<i>Mean Absolute Error</i>				40	22	22,22	41,44	11,12	11,06	Waktu Rata-Rata	11,62

Sumber: Dokumentasi pengujian robot (2024)

Pada Tabel 5 hasil pengujian pergerakan robot yang dilakukan sebanyak lima kali percobaan. Pada setiap percobaan, robot bergerak dari posisi awal ($X = 0, Y = 0$) menuju target pada posisi ($X = 0, Y = 3000$). Hasil pengukuran menunjukkan nilai Mean Absolute Error (MAE) sebesar 40,00 mm pada sumbu X dan 22,00 mm pada sumbu Y, dengan jarak Euclidean sebesar 22,22 mm. Sementara itu, odometri robot mencatat posisi akhir di $X = 41,44$ mm dan $Y = 11,96$ mm, menghasilkan jarak Euclidean sebesar 11,06 mm. Selisih absolut antara hasil pengukuran dan pembacaan odometri adalah 1,44 mm pada sumbu X, 10,04 mm pada sumbu Y, dan 11,16 mm untuk jarak Euclidean. Robot membutuhkan waktu rata-rata 11,59 detik untuk mencapai tujuan. Waktu tercepat yang dicapai 11,36 detik, sedangkan waktu terlama 11,90 detik. Selisih antara waktu tercepat dan terlama 0,54 detik.

Pengujian Gerak Robot Pada Sumbu +Y Halangan

Pengujian dilakukan untuk menguji robot bergerak sepanjang sumbu +Y di lingkungan dengan halangan, robot diuji untuk mencapai titik tujuan, yaitu $X = 0, Y = 3000$. Tujuan pengujian adalah untuk melihat sejauh mana robot dapat bergerak menuju posisi yang diinginkan dengan menghindari halangan. Untuk itu, data posisi robot aktual dibandingkan dengan data odometri pada robot.

Tabel 6 Data Pengujian Robot ($X = 0, Y = 3000$) Halangan

Sumber: Dokumentasi pengujian robot (2024)

No	Target (mm)			Terukur Nyata (mm)			Terukur Robot (mm)			Waktu (detik)	
	X	Y	Jarak Euclidean	X	Y	Jarak Euclidean	X	Y	Jarak Euclidean		
1	0	3000	3000	20	2990	2990,07	-8,28	3002,01	3002,02	14,24	
2	0	3000	3000	30	2980	2980,15	4,93	2991,00	2991,00	12,42	
3	0	3000	3000	20	2970	2970,07	10,87	2995,48	2995,50	12,75	
4	0	3000	3000	-65	2985	2985,71	-20,20	3019,85	3019,92	12,52	
5	0	3000	3000	50	2980	2980,42	6,16	2980,35	2980,36	12,77	
<i>Mean Absolute Error</i>				37	19	18,72	30,22	16,74	16,50	Waktu Rata-Rata	12,94

Pada Tabel 6 hasil pengujian pergerakan robot yang dilakukan sebanyak lima kali percobaan. Pada setiap percobaan, robot bergerak dari posisi awal ($X = 0, Y = 0$) menuju target pada posisi ($X = 0, Y = 3000$). Hasil pengukuran menunjukkan nilai Mean Absolute Error (MAE) sebesar 37 mm pada sumbu X dan 19 mm pada sumbu Y, dengan jarak Euclidean sebesar 18,72 mm. Sementara itu, odometri robot mencatat posisi akhir di $X = 30,22$ mm dan $Y = 16,74$ mm, menghasilkan jarak Euclidean sebesar 16,50 mm. Selisih absolut antara hasil pengukuran dan pembacaan odometri adalah 6,78 mm pada sumbu X, 2,26 mm pada sumbu Y, dan 2,22 mm untuk jarak Euclidean. Pada Gambar 6 hasil dari 5 kali percobaan menunjukkan bahwa robot membutuhkan waktu rata-rata 12,77 detik untuk mencapai tujuan. Waktu tercepat yang dicapai 12,42 detik, sedangkan waktu terlama 14,24 detik. Selisih antara waktu tercepat dan terlama 1,82 detik.

Pengujian Gerak Robot Pada Sumbu -Y Tanpa halangan

Pengujian dilakukan untuk menguji robot bergerak sepanjang sumbu -Y di lingkungan yang tidak ada halangan. Robot diuji untuk mencapai titik tujuan, yaitu $X = 0$, $Y = -3000$. Tujuan pengujian adalah untuk melihat sejauh mana robot dapat bergerak menuju posisi yang diinginkan dengan akurat. Untuk itu, data posisi robot aktual dibandingkan dengan data odometri pada robot.

Tabel 6 Data Pengujian Robot ($X=0$, $Y = -3000$) Tanpa Halangan

No	Target (mm)			Terukur Nyata (mm)			Terukur Robot (mm)			Waktu (detik)
	X	Y	Jarak Euclidean	X	Y	Jarak Euclidean	X	Y	Jarak Euclidean	
1	0	-3000	3000	20	-2955	2955,07	29,10	-2989,89	2990,03	11,20
2	0	-3000	3000	30	-2960	2960,15	-21,76	-2972,10	2972,18	11,03
3	0	-3000	3000	65	-2965	2965,71	-58,24	-2988,51	2989,08	11,44
4	0	-3000	3000	40	-3025	3025,26	4,07	-3008,20	3008,20	12,05
5	0	-3000	3000	30	-2980	2980,15	5,61	-2981,61	2981,62	11,94
<i>Mean Absolute Error</i>				37	33	32,84	48,88	17,78	17,78	Waktu Rata-Rata 11,53

Sumber: Dokumentasi pengujian robot (2024)

Pada Tabel 7 hasil pengujian pergerakan robot yang dilakukan sebanyak lima kali percobaan. Pada setiap percobaan, robot bergerak dari posisi awal ($X = 0$, $Y = 0$) menuju target pada posisi ($X = 0$, $Y = -3000$). Hasil pengukuran menunjukkan nilai Mean Absolute Error (MAE) sebesar 37,00 mm pada sumbu X dan 33,00 mm pada sumbu Y, dengan jarak Euclidean sebesar 32,84 mm. Sementara itu, odometri robot mencatat posisi akhir di $X = 48,88$ mm dan $Y = 17,78$ mm, menghasilkan jarak Euclidean sebesar 17,78 mm. Selisih absolut antara hasil pengukuran dan pembacaan odometri adalah 11,88 mm pada sumbu X, 15,06 mm pada sumbu Y, dan 15,06 mm untuk jarak Euclidean. Robot membutuhkan waktu rata-rata 11,53 detik untuk mencapai tujuan. Waktu tercepat yang dicapai 11,03 detik, sedangkan waktu terlama 12,05 detik. Selisih antara waktu tercepat dan terlama 1,02 detik.

Pengujian Gerak Robot Pada Sumbu -Y Halangan

Pengujian dilakukan untuk menguji robot bergerak sepanjang sumbu Y di lingkungan dengan halangan. Robot diuji untuk mencapai titik tujuan, yaitu $X = 0$, $Y = -3000$. Tujuan pengujian adalah untuk melihat sejauh mana robot dapat bergerak menuju posisi yang diinginkan dengan akurat. Untuk itu, data posisi robot aktual dibandingkan dengan data odometri pada robot.

Tabel 7 Data Pengujian Robot ($X=0$, $Y = -3000$) Halangan

No	Target (mm)			Terukur Nyata (mm)			Terukur Robot (mm)			Waktu (detik)
	X	Y	Jarak Euclidean	X	Y	Jarak Euclidean	X	Y	Jarak Euclidean	
1	0	-3000	3000	60	-3020	3020,60	-15,09	-3020,47	3020,51	12,48
2	0	-3000	3000	65	-2950	2950,72	33,23	-2967,76	2967,95	12,51
3	0	-3000	3000	40	-3015	3015,27	0,17	-3006,73	3006,73	12,29
4	0	-3000	3000	80	-2980	2981,07	-6,26	-3009,54	3009,55	12,30
5	0	-3000	3000	-50	-2980	2980,42	27,70	-2978,71	2978,84	14,18
<i>Mean Absolute Error</i>				59	25	24,73	62,13	11,47	11,18	Waktu Rata-Rata 12,75

Sumber: Dokumentasi pengujian robot (2024)

Pada Tabel 8 hasil pengujian pergerakan robot yang dilakukan sebanyak lima kali percobaan. Pada setiap percobaan, robot bergerak dari posisi awal ($X = 0, Y = 0$) menuju target pada posisi ($X = 0, Y = -3000$). Hasil pengukuran menunjukkan nilai Mean Absolute Error (MAE) sebesar 37,00 mm pada sumbu X dan 33,00 mm pada sumbu Y, dengan jarak Euclidean sebesar 32,84 mm. Sementara itu, odometri robot mencatat posisi akhir di $X = 48,88$ mm dan $Y = 17,78$ mm, menghasilkan jarak Euclidean sebesar 17,78 mm. Selisih absolut antara hasil pengukuran dan pembacaan odometri adalah 11,88 mm pada sumbu X, 15,06 mm pada sumbu Y, dan 15,06 mm untuk jarak Euclidean. Robot membutuhkan waktu rata-rata 11,53 detik untuk mencapai tujuan. Waktu tercepat yang dicapai 11,03 detik, sedangkan waktu terlama 12,05 detik. Selisih antara waktu tercepat dan terlama 1,02 detik.

Pengujian Gerak Robot Pada Sumbu +X Tanpa Halangan

Pengujian dilakukan untuk menguji robot bergerak sepanjang sumbu +X di lingkungan yang tidak ada halangan. Robot diuji untuk mencapai titik tujuan, yaitu $X = 3000, Y = 0$. Tujuan pengujian adalah untuk melihat sejauh mana robot dapat bergerak menuju posisi yang diinginkan dengan akurat. Untuk itu, data posisi robot aktual dibandingkan dengan data odometri pada robot.

Tabel 8 Data Pengujian Robot ($X = 3000, Y = 0$) Tanpa Halangan

No	Target (mm)			Terukur Nyata (mm)			Terukur Robot (mm)			Waktu (detik)	
	X	Y	Jarak Euclidean	X	Y	Jarak Euclidean	X	Y	Jarak Euclidean		
1	3000	0	3000	3020	50	3020,41	3024,44	48,01	3024,82	11,17	
2	3000	0	3000	3065	50	3065,41	3044,00	8,41	3044,01	11,11	
3	3000	0	3000	3015	-10	3015,02	3005,89	0,37	3005,89	11,14	
4	3000	0	3000	3025	40	3025,26	3014,26	12,05	3014,28	11,38	
5	3000	0	3000	3030	25	3030,10	3019,22	19,71	3019,28	11,42	
<i>Mean Absolute Error</i>				31	35	31,24	11,21	17,44	11,35	Waktu Rata-Rata	11,24

Sumber: Dokumentasi pengujian robot (2024)

Pada Tabel 9 ditampilkan hasil pengujian pergerakan robot sebanyak lima kali percobaan. Pada setiap percobaan, robot bergerak dari posisi awal ($X = 0, Y = 0$) menuju target pada posisi ($X = 3000, Y = 0$) tanpa halangan. Hasil pengukuran menunjukkan nilai Mean Absolute Error (MAE) sebesar 31,00 mm pada sumbu X dan 35,00 mm pada sumbu Y, dengan jarak Euclidean sebesar 31,24 mm. Sementara itu, odometri robot mencatat posisi akhir di $X = 11,21$ mm dan $Y = 17,44$ mm, menghasilkan jarak Euclidean sebesar 11,35 mm. Selisih absolut antara hasil pengukuran dan pembacaan odometri adalah 19,79 mm pada sumbu X, 17,56 mm pada sumbu Y, dan 19,89 mm untuk jarak Euclidean. Robot membutuhkan waktu rata-rata 11,24 detik untuk mencapai tujuan. Waktu tercepat yang dicapai adalah 11,11 detik, sedangkan waktu terlama adalah 11,42 detik, dengan selisih waktu sebesar 0,31 detik.

Pengujian Gerak Robot Pada Sumbu +X Tanpa Halangan

Pengujian dilakukan untuk menguji robot bergerak sepanjang sumbu +Y di lingkungan dengan halangan. Robot diuji untuk mencapai titik tujuan, yaitu $X = 3000, Y = 0$. Tujuan pengujian adalah untuk melihat sejauh mana robot dapat bergerak menuju posisi yang diinginkan dengan menghindari halangan. Untuk itu, data posisi robot aktual dibandingkan dengan data odometri pada robot.

Tabel 9 Data Pengujian Robot ($X = 3000, Y = 0$) Halangan

No	Target (mm)			Terukur Nyata (mm)			Terukur Robot (mm)			Waktu (detik)	
	X	Y	Jarak Euclidean	X	Y	Jarak Euclidean	X	Y	Jarak Euclidean		
1	3000	0	3000	3040	20	3040,07	2986,13	7,98	2986,14	12,19	
2	3000	0	3000	3030	40	3030,26	2996,84	-4,90	2996,84	12,49	
3	3000	0	3000	3010	-45	3010,34	2993,69	-14,29	2993,72	12,25	
4	3000	0	3000	2975	70	2975,82	2949,77	49,21	2950,18	12,64	
5	3000	0	3000	3035	-40	3035,26	3012,88	6,65	3012,89	12,62	
<i>Mean Absolute Error</i>				28	43	28,02	30,14	31,01	30,40	Waktu Rata-Rata	12,44

Sumber: Dokumentasi pengujian robot (2024)

Pada Tabel 10 ditampilkan hasil pengujian pergerakan robot sebanyak lima kali percobaan dengan adanya halangan. Pada setiap percobaan, robot bergerak dari posisi awal ($X = 0$, $Y = 0$) menuju target pada posisi ($X = 3000$, $Y = 0$). Hasil pengukuran menunjukkan nilai Mean Absolute Error (MAE) sebesar 28 mm pada sumbu X dan 43 mm pada sumbu Y, dengan jarak Euclidean sebesar 28,02 mm. Sementara itu, odometri robot mencatat posisi akhir di $X = 30,14$ mm dan $Y = 31,01$ mm, menghasilkan jarak Euclidean sebesar 30,40 mm. Selisih absolut antara hasil pengukuran dan pembacaan odometri adalah 2,14 mm pada sumbu X, 11,99 mm pada sumbu Y, dan 2,38 mm untuk jarak Euclidean. Robot membutuhkan waktu rata-rata 12,44 detik untuk mencapai tujuan.

Pengujian Gerak Robot Pada Sumbu -X Tanpa halangan

Pengujian dilakukan untuk menguji robot bergerak sepanjang sumbu X di lingkungan yang tidak ada halangan. Robot diuji untuk mencapai titik tujuan, yaitu $X = -3000$, $Y = 0$. Tujuan pengujian adalah untuk melihat sejauh mana robot dapat bergerak menuju posisi yang diinginkan dengan akurat. Untuk itu, data posisi robot aktual dibandingkan dengan data odometri pada robot.

Tabel 10 Data Pengujian Robot ($X = -3000$, $Y = 0$) Tanpa Halangan

No	Target (mm)			Terukur Nyata (mm)			Terukur Robot (mm)			Waktu (detik)	
	X	Y	Jarak Euclidean	X	Y	Jarak Euclidean	X	Y	Jarak Euclidean		
1	-3000	0	3000	-3030	-55	3030,50	-3028,10	46,10	3028,45	11,36	
2	-3000	0	3000	-3010	35	3010,20	-3005,55	28,44	3005,68	11,69	
3	-3000	0	3000	-3030	-45	3030,33	-3010,87	-18,11	3010,92	11,90	
4	-3000	0	3000	-3030	10	3030,02	-2995,42	18,95	2995,48	11,45	
5	-3000	0	3000	-3050	-40	3050,26	-3019,97	-2,19	3019,97	11,71	
<i>Mean Absolute Error</i>				30	37	30,26	18,02	36,26	18,16	Waktu Rata-Rata	11,62

Sumber: Dokumentasi pengujian robot (2024)

Pada Tabel 11 ditampilkan hasil pengujian pergerakan robot yang dilakukan sebanyak lima kali percobaan tanpa adanya halangan. Pada setiap percobaan, robot bergerak dari posisi awal ($X = 0$, $Y = 0$) menuju target pada posisi ($X = -3000$, $Y = 0$). Hasil pengukuran menunjukkan nilai Mean Absolute Error (MAE) sebesar 30 mm pada sumbu X dan 37 mm pada sumbu Y, dengan jarak Euclidean sebesar 30,26 mm. Sementara itu, odometri robot mencatat posisi akhir di $X = 18,02$ mm dan $Y = 36,26$ mm, menghasilkan jarak Euclidean sebesar 18,16 mm. Selisih absolut antara hasil pengukuran dan pembacaan odometri adalah 11,98 mm pada sumbu X, 0,74 mm pada sumbu Y, dan 12,10 mm untuk jarak Euclidean. Robot membutuhkan waktu rata-rata 11,62 detik untuk mencapai tujuan.

Pengujian Gerak Robot Pada Sumbu -X halangan

Pengujian dilakukan untuk menguji robot bergerak sepanjang sumbu X di lingkungan yang ada halangan. Robot diuji untuk mencapai titik tujuan, yaitu $X = -3000$, $Y = 0$. Tujuan pengujian adalah untuk melihat sejauh mana robot dapat bergerak menuju posisi yang diinginkan dengan akurat. Untuk itu, data posisi robot aktual dibandingkan dengan data odometri pada robot.

Tabel 11 Data Pengujian Robot ($X = -3000$, $Y = 0$) Halangan

No	Target (mm)			Terukur Nyata (mm)			Terukur Robot (mm)			Waktu (detik)	
	X	Y	Jarak Euclidean	X	Y	Jarak Euclidean	X	Y	Jarak Euclidean		
1	-3000	0	3000	-3030	-10	3030,02	-2995,37	-6,70	2995,38	12,82	
2	-3000	0	3000	-2985	-60	2985,60	-2980,11	49,88	2980,53	12,98	
3	-3000	0	3000	-3035	-35	3035,20	-3009,35	-10,02	3009,37	13,02	
4	-3000	0	3000	-2990	20	2990,07	-2985,90	14,73	2985,94	13,04	
5	-3000	0	3000	-3010	15	3010,04	-3008,96	-18,43	3009,02	12,63	
<i>Mean Absolute Error</i>				20	28	19,92	14,06	35,37	14,14	Waktu Rata-Rata	12,90

Sumber: Dokumentasi pengujian robot (2024)

Pada Tabel 12 ditampilkan hasil pengujian pergerakan robot sebanyak lima kali percobaan dengan adanya halangan. Pada setiap percobaan, robot bergerak dari posisi awal ($X = 0$, $Y = 0$) menuju target pada posisi ($X = -3000$, $Y = 0$). Hasil pengukuran menunjukkan nilai Mean Absolute Error (MAE) sebesar 20 mm pada sumbu X dan 28 mm pada sumbu Y, dengan jarak Euclidean sebesar 19,92 mm. Sementara itu, odometri robot mencatat posisi akhir di $X = 14,06$ mm dan $Y = 35,37$ mm, menghasilkan jarak Euclidean sebesar 14,14 mm. Selisih absolut antara hasil pengukuran dan pembacaan odometri adalah 5,94 mm pada sumbu X, 7,37 mm pada sumbu Y, dan 5,78 mm untuk jarak Euclidean. Robot membutuhkan waktu rata-rata 12,90 detik untuk mencapai tujuan.

KESIMPULAN

Penelitian berhasil mengimplementasikan sistem navigasi otonom pada robot omnidirectional menggunakan algoritma Potential Field yang dikombinasikan dengan sensor LiDAR dan ROS. Dari hasil pengujian, robot mampu bergerak ke berbagai arah dengan akurasi posisi yang baik, baik pada kondisi tanpa halangan maupun dengan halangan, sesuai dengan harapan. Pengujian sensor LiDAR menunjukkan rata-rata error sebesar 2,23% pada jarak minimal (0,05–0,95 m) dan 5,38% pada jarak maksimal (1–12 m), sesuai dengan spesifikasi sensor. Delay pengolahan data LiDAR dan odometri di sistem ROS masing-masing rata-rata 0,50 detik dan 0,08 detik, mendukung proses kendali real-time. Pada pengujian gerak robot, nilai Mean Absolute Error (MAE) posisi berkisar antara 20 mm hingga 40 mm pada sumbu X dan Y, dengan jarak Euclidean rata-rata di bawah 32 mm, baik pada pengujian tanpa halangan maupun dengan halangan. Waktu rata-rata tempuh robot mencapai tujuan berkisar antara 11,24 hingga 12,94 detik, dengan selisih waktu terpendek dan terpanjang kurang dari 2 detik. Hasil ini membuktikan bahwa integrasi algoritma Potential Field dengan sensor LiDAR dan ROS mampu mewujudkan navigasi otonom yang efisien dan akurat pada robot omnidirectional dalam berbagai kondisi lingkungan.

REFERENSI

- Alatise, M. B., & Hancke, G. P. (2020). A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods. *IEEE Access*, 8, 39830–39846. <https://doi.org/10.1109/ACCESS.2020.2975643>
- Alsadik, B., & Karam, S. (2021). The Simultaneous Localization and Mapping (SLAM)-An Overview. *Journal of Applied Science and Technology Trends*, 2(02), 147–158. <https://doi.org/10.38094/jastt204117>
- Herath, D., & St-Onge Eds, D. (2022). *Foundations of Robotics A Multidisciplinary Approach with Python and ROS A Multidisciplinary Approach with Python and ROS*. <https://doi.org/10.1007/978-981-19-1983-1>
- Kacprzyk, J. (2017). *Robot Operating System (ROS) The Complete Reference (Volume 1)* (Anis Koubaa, Vol. 1). <https://doi.org/10.1007/978-3-319-26054-9>
- Nikmah, A., Ramadani, R., Khoiro, M., Aminullah Firdaus, R., & Dzulkiifli. (2024). Analisis Kinerja Sistem LiDAR (Light Detection and Ranging) dalam Pengukuran Jarak dengan Pendekatan Simulasi: Evaluasi Ketepatan dan Keandalan Pengukuran. *Jurnal Kolaboratif Sains*, 7(5), 1569–1576. <https://doi.org/10.56338/jks.v7i5.5250>
- Rijalusalam, D. U., & Iswanto, I. (2021). Implementation Kinematics Modeling and Odometry of Four Omni Wheel Mobile Robot on The Trajectory Planning and Motion Control Based Microcontroller. *Journal of Robotics and Control (JRC)*, 2(5), 448–455. <https://doi.org/10.18196/jrc.25121>
- Setiawan, S., Derdian Marindani, E., & Wibowo Sanjaya, B. (2024). Navigasi Berbasis Koordinat dengan Penghindaran Rintangan untuk Robot Omnidireksional. *Jurnal Teknik Elektro Dan Komputasi (ELKOM)*, 6(2), 191–200. <https://doi.org/10.32528/elkom.v6i2.22493>
- Wahab, F. (2022). Metode Potential Field sebagai Kendali Robot Roda Omni untuk Menuju Target dan Menghindari Rintangan. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 10(1), 177. <https://doi.org/10.26760/elkomika.v10i1.177>
- Ge, S. S., & Cui, Y. J. (2002). Dynamic motion planning for mobile robots using potential field method. *Autonomous Robots*, 13(3), 207–222. <https://doi.org/10.1023/A:1019620809787>
- Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1), 90–98. <https://doi.org/10.1177/027836498600500106>
- Liu, Y., Ma, H., & Wang, D. (2019). A hybrid approach of improved potential field and genetic algorithm for path planning of mobile robots. *Journal of Intelligent & Robotic Systems*, 95(1), 175–186. <https://doi.org/10.1007/s10846-018-0939-3>
- Sharma, A., & Kapoor, S. (2020). Local minima problem in artificial potential field-based path planning and a novel solution approach. *Procedia Computer Science*, 167, 2401–2410. <https://doi.org/10.1016/j.procs.2020.03.296>
- Sun, L., Du, C., & Li, H. (2020). Real-time mapping and obstacle detection using LiDAR for autonomous driving. *Sensors*, 20(23), 6876. <https://doi.org/10.3390/s20236876>
- Wang, D., Liu, M., & Chen, H. (2017). Potential field-based real-time path planning for autonomous mobile robots. *Robotics and Autonomous Systems*, 89, 161–171. <https://doi.org/10.1016/j.robot.2016.12.003>
- Yin, T., Jiang, L., & Xie, L. (2021). High-resolution LiDAR-based obstacle detection and navigation for autonomous robots. *IEEE Sensors Journal*, 21(4), 4711–4721. <https://doi.org/10.1109/JSEN.2020.3035862>