

Robot Omnidireksional Tiga Roda dengan Pemetaan 2D Berbasis LiDAR Menggunakan Algoritma GMapping

Ari Prayudi*, Elang Dardian Marindani, F. Trias Pontia W

Teknik Elektro, Fakultas Teknik, Universitas Tanjungpura
Jl. Prof. Dr. Hadari Nawawi, Pontianak, Kalimantan Barat, Indonesia
E-mail: ariprayudi19@gmail.com

Naskah Masuk: 13 Juli 2025; Diterima: 28 September 2026; Terbit: 30 September 2026

ABSTRAK

Abstrak - Robot *mobile omnidirectional* membutuhkan kemampuan navigasi dan pemetaan yang baik agar dapat beroperasi secara mandiri pada lingkungan yang belum diketahui. Penelitian ini bertujuan merancang dan mengimplementasikan robot tiga roda *omnidirectional* berbasis *Robot Operating System* (ROS) dengan sensor *LiDAR* untuk melakukan pemetaan lingkungan menggunakan algoritma *GMapping*. Untuk menentukan area eksplorasi berikutnya, metode *Frontier Cells Detection* digunakan untuk mendeteksi batas antara area yang telah diketahui dan area yang belum diketahui. Kandidat *frontier* dipilih berdasarkan jarak terjauh dari posisi robot, kemudian algoritma *A** digunakan untuk merencanakan jalur menuju target dengan mempertimbangkan kondisi peta dan *obstacle*. Sistem juga menggunakan *odometri* dan *kinematik* untuk memperkirakan posisi serta mengatur pergerakan robot secara *real-time*. Pengujian dilakukan pada beberapa bentuk ruangan, baik tanpa maupun dengan *obstacle*, dengan masing-masing skenario dilakukan sebanyak 10 kali percobaan. Hasil pengujian menunjukkan rata-rata error pemetaan sebesar 0,22%–1,72% pada ruangan tanpa *obstacle* dan mencapai 4,86% pada ruangan dengan *obstacle*, dengan error tertinggi sebesar 8,29%. Hasil tersebut menunjukkan bahwa kombinasi *GMapping*, *Frontier Cells Detection*, dan *A** dapat mendukung proses pemetaan, eksplorasi, dan navigasi robot pada lingkungan dalam ruangan.

Kata kunci: Robot *Omnidirectional*, ROS, LiDAR, Gmapping, *Frontier Detection*, Algoritma *A**

ABSTRACT

Abstract - A *mobile omnidirectional robot* requires reliable navigation and mapping capabilities to operate autonomously in an unknown environment. This study aims to design and implement a three-wheeled *omnidirectional robot* based on the *Robot Operating System* (ROS) and equipped with a *LiDAR* sensor for environmental mapping using the *GMapping* algorithm. To determine the next exploration area, the *Frontier Cells Detection* method is used to identify the boundary between known and unknown areas. The detected *frontier* candidates are selected based on the farthest distance from the robot's current position. Subsequently, the *A** algorithm is applied to plan a path toward the selected target while considering the map conditions and obstacles. The navigation system also utilizes *odometry* and *kinematics* to estimate the robot's position and control its movement in real time. Experiments were conducted in several room configurations, both with and without obstacles, with each scenario tested in 10 trials. The results show an average mapping error of 0.22%–1.72% in rooms without obstacles and up to 4.86% in rooms with obstacles, with the highest recorded error being 8.29%. These results demonstrate that the integration of *GMapping*, *Frontier Cells Detection*, and *A** can support the robot's mapping, exploration, and navigation processes in indoor environments.

Keywords: *Omnidirectional Robot*, ROS, LiDAR, GMapping, *Frontier Detection*, *A* Algorithm*

Copyright © 2026 Jurnal Teknik Elektro dan Komputasi (ELKOM)

1. PENDAHULUAN

Robot *mobile omnidirectional* merupakan robot yang mampu bergerak ke berbagai arah tanpa perlu mengubah orientasinya [1]-[3]. Kemampuan tersebut memungkinkan robot melakukan navigasi secara fleksibel pada lingkungan yang belum diketahui. Agar dapat beroperasi secara mandiri, robot harus mampu melakukan pemetaan, menentukan posisi, merencanakan jalur, dan menghindari *obstacle* [4]-[6]. Salah satu metode yang dapat digunakan adalah *Simultaneous Localization and Mapping* (SLAM), yaitu proses pemetaan lingkungan sekaligus menentukan posisi robot [7]-[9].

Penelitian ini menggunakan robot tiga roda omnidirectional yang dikendalikan menggunakan *Robot Operating System* (ROS) dan dilengkapi sensor *LiDAR* untuk memperoleh informasi lingkungan dalam bentuk peta 2D. Metode *GMapping* digunakan untuk membangun peta berdasarkan data *LiDAR* dan *odometri*. Namun, hasil pemetaan belum menentukan area berikutnya yang harus dieksplorasi, terutama pada lingkungan yang masih memiliki area *unknown*. Oleh karena itu, digunakan metode *Frontier Cells Detection* untuk mendeteksi batas antara area yang telah diketahui dan area yang belum diketahui [10].

Frontier yang terdeteksi digunakan sebagai kandidat tujuan eksplorasi. Dalam penelitian ini, pemilihan target dilakukan berdasarkan jarak terjauh dari posisi robot untuk memperluas area pemetaan. Setelah target ditentukan, algoritma A* digunakan untuk mencari jalur menuju target dengan mempertimbangkan kondisi peta dan keberadaan *obstacle* [11]. Kinematik dan *odometri* digunakan untuk mengatur pergerakan serta memperkirakan posisi robot selama proses navigasi.

Tujuan penelitian ini adalah merancang sistem eksplorasi otomatis pada robot tiga roda omnidirectional yang mampu melakukan pemetaan, menentukan target eksplorasi, dan merencanakan jalur secara otomatis. Penelitian ini diharapkan dapat memberikan manfaat dalam pengembangan sistem navigasi robot yang mampu melakukan pemetaan dan eksplorasi lingkungan secara mandiri.

2. KAJIAN PUSTAKA

2.1 Kinematik

Kinematik adalah kajian mengenai gerakan robot yang didasarkan pada analisis struktur geometris dari sistem kerangka acuan, baik yang tetap maupun bergerak, tanpa mempertimbangkan gaya, torsi, atau momen yang menyebabkan gerakan tersebut [12].

Dalam pemodelan kinematik robot, terdapat dua jenis model utama, yaitu *invers kinematik* (IK) dan *forward kinematik* (FK). *Invers kinematik* digunakan untuk menghitung kecepatan setiap roda pada robot omni (V_{m1}, V_{m2}, V_{m3}) dan *forward kinematik* menentukan kecepatan linear robot terhadap koordinat lokal (V_x, V_y, ω) [12].

Robot omnidirectional dengan tiga roda yang dipasang dengan model “Y” memiliki perbedaan sudut pemasangan setiap roda sebesar 120° . Dengan demikian, sudut setiap roda adalah $\theta_1 = 30^\circ, \theta_2 = 150^\circ, \text{ dan } \theta_3 = 270^\circ$ [13].

Dari kinematika robot omnidirectional tersebut maka didapatkan persamaan *forward kinematik* sebagai berikut:

$$\begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} = r \begin{bmatrix} \cos(\theta_1 + \frac{\pi}{2}) & \cos(\theta_2 + \frac{\pi}{2}) & \cos(\theta_3 + \frac{\pi}{2}) \\ \sin(\theta_1 + \frac{\pi}{2}) & \sin(\theta_2 + \frac{\pi}{2}) & \sin(\theta_3 + \frac{\pi}{2}) \\ \frac{1}{3L} & \frac{1}{3L} & \frac{1}{3L} \end{bmatrix} \begin{bmatrix} V_{m1} \\ V_{m2} \\ V_{m3} \end{bmatrix} \quad (1)$$

Untuk menentukan kecepatan dari setiap roda robot omni maka dengan melakukan *invers* pada persamaan *forward kinematik*, sehingga diperoleh persamaan *invers kinematik* berikut:

$$\begin{bmatrix} V_{m1} \\ V_{m2} \\ V_{m3} \end{bmatrix} = r \begin{bmatrix} \cos(\theta_1 + \frac{\pi}{2}) & \sin(\theta_1 + \frac{\pi}{2}) & 1/3L \\ \cos(\theta_2 + \frac{\pi}{2}) & \sin(\theta_2 + \frac{\pi}{2}) & 1/3L \\ \cos(\theta_3 + \frac{\pi}{2}) & \sin(\theta_3 + \frac{\pi}{2}) & 1/3L \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} \quad (2)$$

2.2 Odometri

Odometri adalah cara untuk menghitung perubahan posisi sebuah robot berdasarkan data dari sensor gerak. Metode ini sering digunakan pada robot berkaki maupun beroda untuk memperkirakan posisi robot relatif terhadap posisi awalnya. *Odometri* bekerja dengan menghitung putaran roda menggunakan *encoder*. Setiap kali roda berputar, *encoder* menghasilkan pulsa yang dihitung secara terus-menerus sampai robot mencapai tujuan [14]. Jarak yang ditempuh roda dapat dihitung dengan rumus:

$$R_{123} = \frac{\text{jumlah pulse}}{PPR} \times 2\pi r \quad (3)$$

Untuk mendapatkan koordinat X dan koordinat Y lokal pada robot, digunakan persamaan kinematik maju dengan mengganti jari-jari masing-masing roda menjadi jarak yang ditempuh masing-masing roda. Dengan cara tersebut, maka didapatkan persamaan berikut :

$$V_x = \cos(\theta_1 + \frac{\pi}{2})R_1 + \cos(\theta_2 + \frac{\pi}{2})R_2 + \cos(\theta_3 + \frac{\pi}{2})R_3 \quad (1)$$

$$V_y = \sin(\theta_1 + \frac{\pi}{2})R_1 + \sin(\theta_2 + \frac{\pi}{2})R_2 + \sin(\theta_3 + \frac{\pi}{2})R_3 \quad (5)$$

$$\omega = \frac{R_1}{3L} + \frac{R_2}{3L} + \frac{R_3}{3L} \quad (6)$$

Data yang diperoleh merupakan data lokal dari robot. Oleh karena itu, diperlukan matriks rotasi untuk mengubah data lokal menjadi data global. Dengan menggunakan persamaan (7), maka didapatkan persamaan berikut.

$$\begin{bmatrix} X_* \\ Y_* \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} \quad (7)$$

2.3 Algoritma A*

Algoritma A* merupakan pengembangan dari algoritma Dijkstra yang dikenal lebih efisien dalam menemukan jalur optimal. Perbedaan utamanya terletak pada penggunaan nilai heuristik, yaitu estimasi jarak terpendek dari suatu titik ke tujuan, yang biasanya dihitung sebagai garis lurus. Selama proses pencarian, algoritma A* akan mengevaluasi semua simpul di sekitar simpul aktif dan memilih jalur dengan nilai total biaya terendah. Dengan strategi ini, A* mampu menghasilkan rute terpendek yang efisien dan aman untuk dilalui [15]. Perhitungan jalur pada algoritma ini didasarkan pada rumus berikut:

$$f(n) = g(n) + h(n) \quad (8)$$

Keterangan:

$f(n)$ = Total nilai evaluasi pada simpul n

$g(n)$ = Harga dari titik awal menuju titik ke n

$h(n)$ = Estimasi harga terkecil dari titik ke n menuju titik tujuan

Salah satu metode umum untuk menghitung nilai heuristik $h(n)$ adalah dengan menggunakan jarak Euclidean, karena metode ini cocok untuk memperkirakan jarak dalam arah diagonal.

$$h(n) = \sqrt{(X_{aktual} - X_{tujuan})^2 + (Y_{aktual} - Y_{tujuan})^2} \quad (9)$$

Persamaan (9) menghitung jarak lurus antara dua titik koordinat dalam bidang dua dimensi, yaitu antara posisi saat ini (X_{aktual}, Y_{aktual}) dan posisi tujuan (X_{tujuan}, Y_{tujuan}) .

2.4 Frontier Cells Detection

Frontier cells detection merupakan proses identifikasi batas antara area yang telah dikenal dengan area yang belum diketahui dalam sebuah peta. Eksplorasi berbasis frontier dilakukan secara berulang dengan cara mendeteksi frontier cells, kemudian mengarahkan robot untuk bergerak menuju sel-sel tersebut hingga tidak ditemukan lagi frontier pada peta. Kondisi ini menandakan bahwa seluruh area yang dapat diakses oleh robot telah berhasil dieksplorasi secara menyeluruh [9].

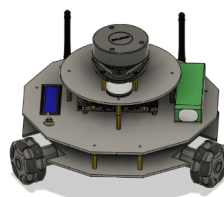
Secara umum, lingkungan direpresentasikan dalam bentuk peta grid dua dimensi $M(x,y)$, di mana setiap sel (x,y) memiliki status $M(x,y) = -1$ menunjukkan area yang belum diketahui (unknown), yaitu sel yang belum terdeteksi atau belum dijangkau oleh sensor sedangkan $M(x,y) = 0$ menunjukkan ruang bebas (free space), yaitu area yang telah dipetakan tanpa hambatan dan dapat dilalui oleh robot serta $M(x,y)=100$ menunjukkan keberadaan hambatan atau rintangan (obstacle) yang tidak dapat dilalui karena merupakan objek penghalang [16] [17].

Suatu sel (x,y) dikategorikan sebagai frontier cell apabila memenuhi dua kondisi secara bersamaan, yaitu sel tersebut merupakan ruang bebas dengan nilai $M(x,y) = 0$, dan memiliki setidaknya satu tetangga langsung yang belum diketahui dengan nilai $M(x,y) = -1$. Tetangga langsung yang dimaksud meliputi sel bagian atas $(x-1,y)$, bawah $(x+1,y)$, kiri $(x,y-1)$, dan kanan $(x,y+1)$. Dengan demikian, frontier cell berada di perbatasan antara area yang diketahui dan area yang belum diketahui.

3. METODE PENELITIAN

3.1 Desain Robot Omnidireksional

Gambar 1 menunjukkan desain dari Robot Omnidirectional



Gambar 1. Desain Robot Omnidirectional

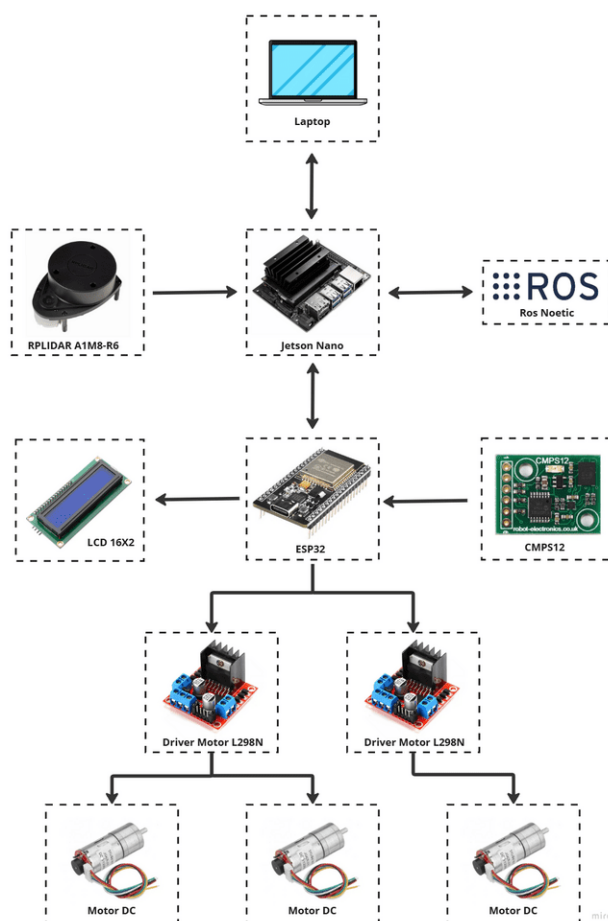
Berdasarkan Gambar 1, robot *omnidireksional* dirancang menggunakan tiga roda yang diposisikan membentuk konfigurasi huruf Y, dengan sudut pemasangan masing-masing roda yaitu $\theta_1 = 30^\circ$, $\theta_2 = 150^\circ$, dan $\theta_3 = 270^\circ$, yang dipasang pada rangka berbahan akrilik. Gambar 1 menunjukkan bentuk fisik robot omnidireksional, sementara Tabel 1 menyajikan spesifikasi dari robot omnidireksional.

Tabel 1. Spesifikasi Robot Omnidirectional

Paramater	Spesifikasi
Nama	Robot Omnidirectional
Sensor	RPLIDAR A1M8-R6
Jangkauan Sensor	0,15 – 12 m
Konektifitas	WIFI
Kapasitas Baterai	1800 mAh
Dimensi Robot	241,48 mm (Diameter) × 250 mm (Tinggi)
Berat	2.06 kg

3.2 Sistem Robot Omnidireksional

Sistem Robot Omnidirectional yang dirancang pada penelitian ini dapat dilihat pada Gambar 2.

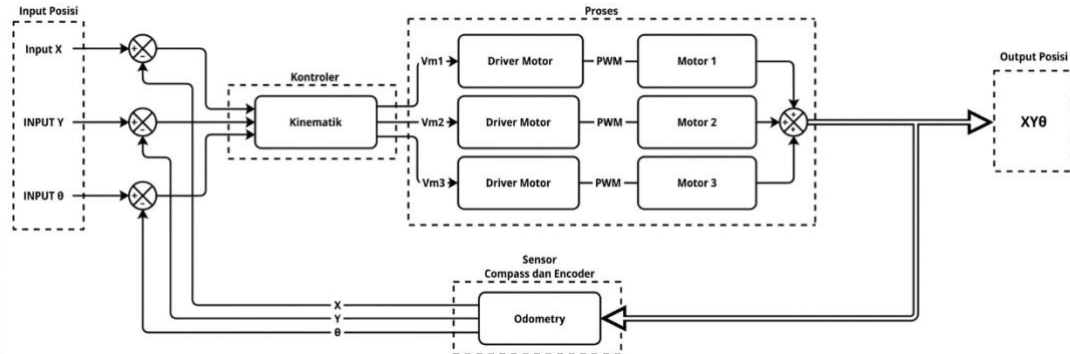
Gambar 2. Sistem Robot *Omnidirectional*

Berdasarkan Gambar 2, sistem *robot omnidirectional* terdiri dari beberapa komponen yang memiliki peran penting dan saling terhubung untuk mendukung kinerja robot secara optimal. Laptop digunakan sebagai antarmuka utama untuk mengontrol dan memantau hasil pemetaan melalui GUI secara nirkabel, melalui jaringan WiFi dan NoMachine, sehingga memudahkan pengoperasian Jetson Nano dari jarak jauh. Jetson Nano berperan sebagai pusat pengendali (*master*) yang memproses data dari sensor RPLIDAR dan ESP32 menggunakan ROS. ESP32 bertindak sebagai *slave*, berfungsi mengendalikan motor DC serta membaca data dari sensor kompas dan encoder. Kompas digunakan untuk mengetahui orientasi robot, sedangkan *encoder* membantu menghitung posisi roda untuk

mendukung sistem *odometri*. ROS berperan sebagai platform integrasi yang memungkinkan komunikasi antar komponen secara efisien melalui sistem *node*.

3.3 Diagram sistem Omnidireksional

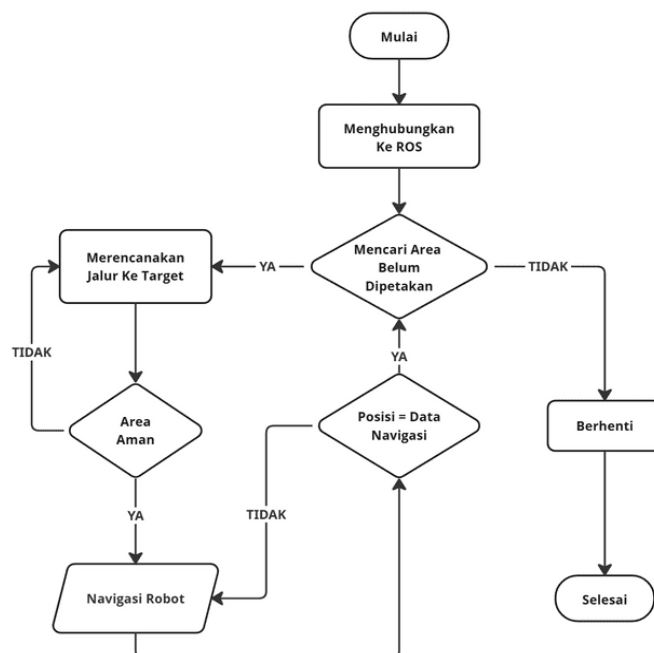
Diagram blok dari sistem omnidireksional dapat dilihat pada gambar 3 berikut:



Gambar 3. Diagram Blok Sistem

Gambar 3 menunjukkan diagram blok sistem robot *omnidirectional* dengan kendali *closed-loop* untuk mengatur pergerakan robot berdasarkan koordinat tujuan. Sistem menerima masukan koordinat posisi dan orientasi (X, Y, θ), di mana θ diatur menjadi 0 untuk menjaga arah robot saat menuju tujuan. Data ini digunakan oleh algoritma *inverse kinematics* untuk menghitung kecepatan masing-masing motor (V_{m1}, V_{m2}, V_{m3}). Kecepatan tersebut dikirim ke modul motor driver untuk menggerakkan roda sehingga robot mencapai posisi target dengan output posisi baru (X, Y, θ).

Selama bergerak, sensor *encoder* dan kompas mengukur perubahan posisi dan orientasi robot. Data ini digunakan dalam odometri untuk memperkirakan posisi robot secara real-time. *Odometri* berfungsi sebagai umpan balik dalam sistem sehingga pergerakan robot dapat disesuaikan dan dikoreksi secara dinamis selama navigasi.



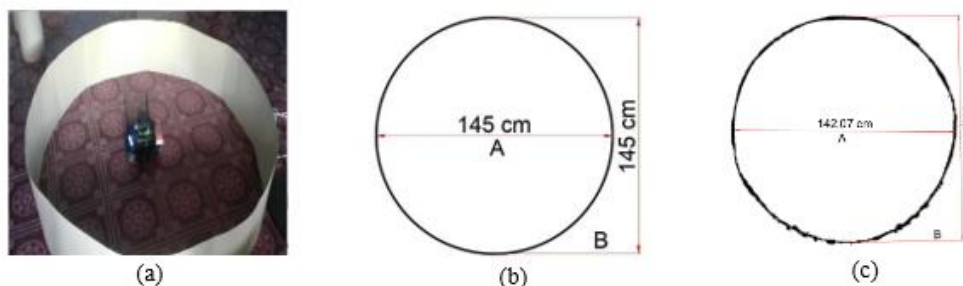
Gambar 4. Diagram Alir Sistem

Pada Gambar 4 proses pemetaan dijelaskan dengan beberapa tahapan. Diawali dengan proses awal “Mulai” sistem kemudian terhubung ke ROS agar dapat menerima data dari ROS master. Setelah data diterima, robot akan mencari area yang belum dipetakan dengan mencari batasan antara area yang telah dikenal dengan area yang belum diketahui dalam sebuah peta. Jika masih terdapat area yang belum dipetakan, robot akan merencanakan jalur aman menggunakan algoritma A* menuju area tersebut, sehingga dapat bergerak dengan aman ke tujuan. Namun, apabila tidak terdapat lagi area yang belum dipetakan, maka robot akan berhenti dan pemetaan selesai.

4. HASIL DAN PEMBAHASAN

4.1 Pengujian Bentuk Lingkaran

Gambar 5 menunjukkan area pengujian ruangan, desain ruangan bentuk lingkaran, dan hasil pemetaan bentuk lingkaran. Pengujian pertama dilakukan pada ruangan berbentuk lingkaran dengan diameter 145 cm. Hasil pemetaan menunjukkan bahwa bentuk ruangan yang dihasilkan sesuai dengan bentuk aslinya. Robot bergerak memutar untuk memetakan area di sekitarnya tanpa mengalami kendala.



Gambar 5 (a) Area Pengujian Ruangan, (b) Desain Ruangan Bentuk Lingkaran (c) Hasil Pemetaan Bentuk Lingkaran

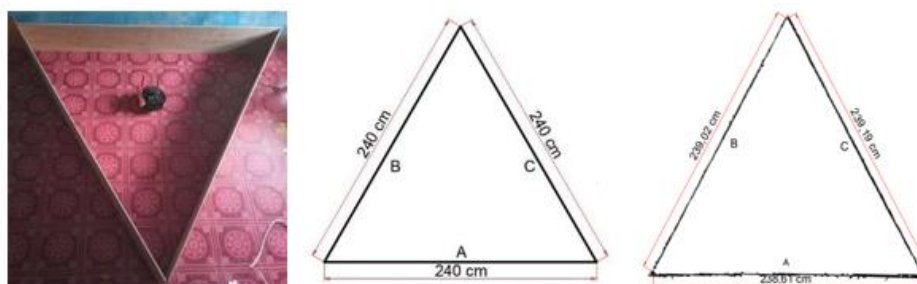
Tabel 2. Data Pengujian Pemetaan Bentuk Lingkaran

Pengujian	Hasil Pemetaan A (cm)	Hasil Pemetaan B (cm)	Error A (%)	Error B (%)
1	142,07	144,38	2,02	0,43
2	141,20	144,76	2,62	0,17
3	142,91	144,97	1,44	0,02
4	143,43	145,12	1,08	0,08
5	142,66	144,71	1,61	0,20
6	141,52	143,38	2,40	1,12
7	141,71	144,72	2,27	0,20
8	144,75	145,85	0,17	0,59
9	142,50	144,35	1,73	0,45
10	142,30	144,51	1,86	0,34
Rata - rata error (%)			1,72	0,36

Berdasarkan Tabel 2, hasil pemetaan ruangan berbentuk lingkaran dengan diameter 145 cm yang dilakukan sebanyak 10 percobaan menunjukkan bahwa rata-rata *error* pada posisi A sebesar 1,72%, dengan *error* tertinggi 2,62% pada pengujian ke-2 dan *error* terendah 0,17 pada pengujian ke-8. Sedangkan pada posisi B sebesar 0,36%, dengan *error* tertinggi 1,12% pada pengujian ke-6 dan *error* terendah 0,02% pada pengujian ke-3.

4.2 Pengujian Bentuk Segitiga

Gambar 6 menunjukkan area pengujian ruangan segitiga, desain ruangan bentuk segitiga, dan hasil pemetaan bentuk segitiga. Pengujian kedua dilakukan pada ruangan berbentuk segitiga dengan panjang sisi-sisi 240 cm, 240 cm, dan 240 cm, membentuk segitiga sama sisi. Robot dapat bergerak dengan baik di dalam ruangan tanpa menabrak dinding, dan hasil pemetaan yang diperoleh menunjukkan kesesuaian dengan bentuk ruangan yang sebenarnya.



Gambar 6 (a) Area pengujian ruangan segitiga, (b) Desain ruangan bentuk segitiga, (c) Hasil pemetaan bentuk segitiga

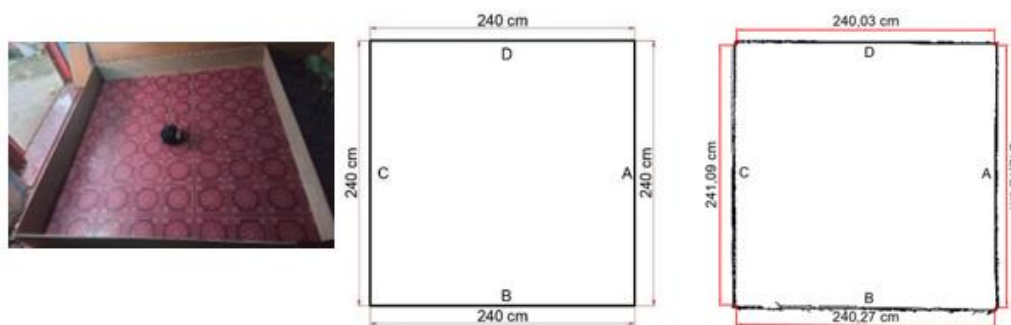
Tabel 3. Data pengujian pemetaan bentuk segitiga

Pengujian	Hasil Pemetaan A (cm)	Hasil Pemetaan B (cm)	Hasil Pemetaan C (cm)	Error A (%)	Error B (%)	Error C (%)
1	238,61	239,02	239,19	0,58	0,41	0,34
2	239,49	239,47	239,51	0,21	0,22	0,21
3	239,41	239,09	239,00	0,24	0,38	0,42
4	239,41	240,05	239,05	0,25	0,02	0,40
5	239,67	240,67	239,11	0,14	0,28	0,37
6	239,25	239,55	239,18	0,31	0,19	0,34
7	239,47	239,67	239,24	0,22	0,14	0,32
8	239,31	239,19	239,13	0,29	0,34	0,36
9	239,57	239,44	239,26	0,18	0,23	0,31
10	239,06	239,94	239,15	0,39	0,03	0,36
Rata - rata error (%)				0,28	0,22	0,34

Berdasarkan Tabel 3, hasil pemetaan ruangan berbentuk segitiga dengan panjang sisi-sisi 240 cm, 240 cm, dan 240 cm, membentuk segitiga sama sisi yang dilakukan sebanyak 10 kali percobaan menunjukkan bahwa rata-rata error pada posisi A 0,28%, dengan error tertinggi 0,58% pada pengujian ke-1 dan error terendah 0,14% pada pengujian ke-5. Pada posisi B, rata-rata error 0,22%, dengan error tertinggi 0,41% pada pengujian ke-1 dan error terendah 0,02% pada pengujian ke-4. Sementara itu, pada posisi C, rata-rata error 0,34%, dengan error tertinggi 0,42% pada pengujian ke-3 dan error terendah 0,21% pada pengujian ke-2.

4.3 Pengujian Bentuk Persegi

Gambar 7 menunjukkan area pengujian ruangan persegi, desain ruangan bentuk persegi, dan hasil pemetaan bentuk persegi. Pengujian ketiga dilakukan pada ruangan berbentuk persegi dengan ukuran 240 cm × 240 cm. Dalam pengujian ini, robot juga mampu bergerak bebas di dalam ruangan tanpa menabrak dinding, serta menghasilkan pemetaan yang sesuai dengan bentuk asli ruangan.



Gambar 7 (a) Area pengujian ruangan persegi, (b) Desain ruangan bentuk persegi, (c) Hasil pemetaan bentuk persegi

Tabel 4. Data Pengujian Pemetaan Bentuk Persegi

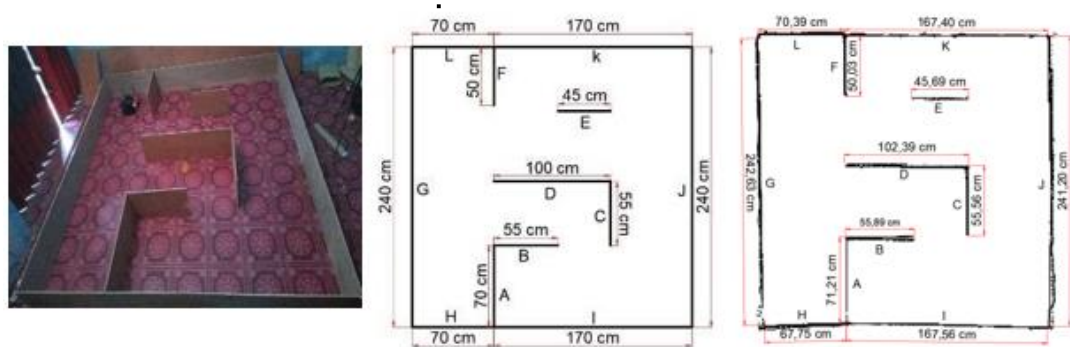
Pengujian	Hasil Pemetaan A (cm)	Hasil Pemetaan B (cm)	Hasil Pemetaan C (cm)	Hasil Pemetaan D (cm)	Error A (%)	Error B (%)	Error C (%)	Error D (%)
1	242,72	240,27	241,09	240,03	1,13	0,11	0,45	0,01
2	241,98	241,13	241,97	240,47	0,83	0,47	0,82	0,19
3	240,44	242,44	240,76	241,33	0,18	1,02	0,32	0,55
4	241,31	241,88	240,25	242,98	0,55	0,78	0,10	1,24
5	240,43	241,88	240,78	242,31	0,18	0,78	0,33	0,96
6	241,65	241,72	241,94	240,67	0,69	0,72	0,81	0,28
7	242,86	241,12	242,84	240,35	1,19	0,47	1,18	0,15
8	242,20	240,41	243,18	240,21	0,92	0,17	1,32	0,09
9	241,90	240,12	242,88	240,23	0,79	0,05	1,20	0,10
10	241,98	240,85	242,94	240,33	0,82	0,35	1,23	0,14
Rata - rata error (%)					0,73	0,49	0,78	0,37

Berdasarkan Tabel 4, hasil pemetaan ruangan berbentuk persegi dengan ukuran 240 cm × 240 cm yang dilakukan sebanyak 10 kali percobaan menunjukkan bahwa rata-rata error pada posisi A sebesar 0,73%, dengan error tertinggi 1,19% pada pengujian ke-7 dan error terendah 0,18% pada pengujian ke-3 dan pengujian ke-5. Pada posisi B, rata-rata error sebesar 0,49%, dengan error tertinggi 1,02% pada pengujian ke-3 dan error terendah 0,05% pada pengujian ke-9. Pada posisi C, rata-rata error 0,78%, dengan error tertinggi 1,32% pada pengujian ke-8 dan error terendah 0,10% pada pengujian ke-4. Adapun pada posisi D, rata-rata error 0,37%, dengan error tertinggi 1,24% pada pengujian ke-4 dan error terendah 0,01% pada pengujian ke-1.

Secara keseluruhan, ketiga pengujian menunjukkan bahwa robot mampu melakukan pemetaan secara akurat pada berbagai bentuk geometri ruangan, serta bergerak dengan aman tanpa mengalami tabrakan. Hal ini mengindikasikan bahwa sistem navigasi dan pemetaan pada robot berfungsi dengan baik.

4.4 Pengujian Ruangan Dengan Halangan

Gambar 8 menunjukkan area pengujian ruangan dengan halangan, desain ruangan dengan halangan, dan hasil pemetaan dengan halangan. Pengujian keempat dilakukan di dalam ruangan buatan berukuran 240 cm x 240 cm. Robot menelusuri seluruh lorong yang terdapat di dalam ruangan. Selama pengujian, robot mampu bergerak bebas tanpa menabrak dinding serta berhasil menghasilkan pemetaan yang sesuai dengan bentuk asli ruangan.



Gambar 8 (a) Area pengujian ruangan dengan halangan, (b) Desain ruangan dengan halangan, (c) Hasil pemetaan dengan halangan

Tabel 5. Data Pengujian Ruang Dengan Halangan A - D

Pengujian	Hasil Pemetaan A (cm)	Hasil Pemetaan B (cm)	Hasil Pemetaan C (cm)	Hasil Pemetaan D (cm)	Error A (%)	Error B (%)	Error C (%)	Error D (%)
1	71,21	55,89	55,56	102,39	1,73	1,61	1,01	2,39
2	70,74	55,23	56,19	102,21	1,06	0,42	2,16	2,21
3	69,92	54,71	55,11	102,26	0,11	0,53	0,19	2,26
4	70,80	55,05	55,29	102,65	1,14	0,09	0,52	2,65
5	70,72	55,61	56,79	101,61	1,03	1,10	3,25	1,61
6	70,58	54,88	56,84	102,77	0,84	0,22	3,34	2,77
7	70,95	55,52	56,96	102,58	1,36	0,95	3,56	2,58
8	69,61	55,97	56,17	101,76	0,56	1,77	2,13	1,76
9	70,74	54,56	55,70	101,88	1,06	0,80	1,27	1,88
10	71,07	54,46	56,15	101,13	1,53	0,99	2,10	1,13
Rata - rata error (%)					1,04	0,85	1,95	2,13

Berdasarkan Tabel 5, hasil pemetaan ruangan dengan halangan yang dilakukan sebanyak 10 kali percobaan menunjukkan bahwa rata-rata error pada posisi A 1,04%, dengan error tertinggi 1,73% pada pengujian ke-1 dan error terendah 0,11% pada pengujian ke-3. Pada posisi B, rata-rata error 0,85%, dengan error tertinggi 1,77% pada pengujian ke-8 dan error terendah 0,09% pada pengujian ke-4. Pada posisi C, rata-rata error 1,95%, dengan error tertinggi 3,56% pada pengujian ke-7 dan error terendah 0,19% pada pengujian ke-3. Adapun pada posisi D, rata-rata error 2,13%, dengan error tertinggi 2,77% pada pengujian ke-6 dan error terendah 1,13% pada pengujian ke-10.

Tabel 6. Data Pengujian Ruang Dengan Halangan E – H

Pengujian	Hasil Pemetaan A (cm)	Hasil Pemetaan B (cm)	Hasil Pemetaan C (cm)	Hasil Pemetaan D (cm)	Error A (%)	Error B (%)	Error C (%)	Error D (%)
1	45,69	50,03	242,63	67,75	1,53	0,07	1,10	3,21
2	46,81	49,64	240,49	68,79	4,02	0,71	0,20	1,73
3	46,09	51,49	240,82	69,54	2,41	2,99	0,34	0,65
4	48,39	50,29	240,53	69,63	7,53	0,58	0,22	0,53
5	48,73	51,75	241,18	69,26	8,29	3,50	0,49	1,06
6	46,60	50,06	239,91	70,38	3,55	0,11	0,04	0,54
7	46,74	50,73	242,20	68,90	3,86	1,45	0,91	1,57
8	48,14	51,10	242,34	69,53	6,97	2,20	0,98	0,68
9	48,04	50,63	242,31	69,19	6,75	1,26	0,96	1,15
10	46,65	50,29	240,89	70,69	3,66	0,58	0,37	0,98
Rata - rata error (%)					4,86	1,35	0,56	1,21

Berdasarkan Tabel 6, hasil pemetaan ruangan dengan halangan yang dilakukan sebanyak 10 kali percobaan menunjukkan bahwa rata-rata error pada posisi E 4,86%, dengan error tertinggi 8,29% pada pengujian ke-5 dan error terendah 1,53% pada pengujian ke-1. Pada posisi F, rata-rata error 1,35%, dengan error tertinggi 3,50% pada pengujian ke-5 dan error terendah 0,07% pada pengujian ke-1. Pada posisi G, rata-rata error 0,56%, dengan error tertinggi 1,10% pada pengujian ke-1 dan error sebesar 0,04% pada pengujian ke-6. Adapun pada posisi H, rata-rata error sebesar 1,32%, dengan error tertinggi 3,21% pada pengujian ke-1 dan error terendah 0,53% pada pengujian ke-4.

Tabel 7. Data Pengujian Ruang Dengan Halangan I – L

Pengujian	Hasil Pemetaan I (cm)	Hasil Pemetaan J (cm)	Hasil Pemetaan K (cm)	Hasil Pemetaan L (cm)	Error I (%)	Error J (%)	Error K (%)	Error L (%)
1	167,56	242,20	167,40	70,39	1,44	0,92	1,53	0,55
2	168,17	241,62	168,55	69,46	1,08	0,67	0,85	0,78
3	168,39	242,24	167,84	69,92	0,95	0,93	1,27	0,12
4	167,15	242,75	168,08	70,64	1,68	1,14	1,13	0,92
5	168,26	243,28	168,81	69,61	1,03	1,37	0,70	0,56
6	167,83	241,85	168,39	69,64	1,28	0,77	0,95	0,52
7	168,61	241,96	168,78	69,57	0,82	0,81	0,72	0,61
8	168,62	241,25	168,67	69,80	0,81	0,52	0,78	0,29
9	168,70	241,51	168,54	69,60	0,77	0,63	0,86	0,57
10	167,35	242,35	168,74	70,04	1,56	0,98	0,74	0,05
Rata - rata error (%)					1,14	0,88	0,95	0,50

Berdasarkan Tabel 7, hasil pemetaan ruangan dengan halangan yang dilakukan sebanyak 10 kali percobaan menunjukkan bahwa rata-rata error pada posisi I 1,14%, dengan error tertinggi 1,68% pada pengujian ke-4 dan error terendah 0,77% pada pengujian ke-9. Pada posisi J, rata-rata error 0,88%, dengan error tertinggi 1,37% pada pengujian ke-5 dan error terendah 0,52% pada pengujian ke-8. Pada posisi K, rata-rata error 0,95%, dengan error tertinggi 1,53% pada pengujian ke-1 dan error terendah 0,70% pada pengujian ke-5. Adapun pada posisi L, rata-rata error 0,50%, dengan error tertinggi 0,92% pada pengujian ke-4 dan error terendah 0,12% pada pengujian ke-3.

5. KESIMPULAN

Berdasarkan hasil pengujian yang dilakukan sebanyak 10 kali pada berbagai bentuk ruangan, dapat disimpulkan bahwa robot mampu melakukan pemetaan dengan tingkat akurasi yang baik. Pada ruangan berbentuk lingkaran dengan diameter 145 cm, rata-rata error pemetaan mencapai 1,72% pada posisi A dan 0,36% pada posisi B, dengan nilai error tertinggi masing-masing 2,62% dan 1,12%. Untuk ruangan berbentuk segitiga sama sisi dengan panjang sisi 240 cm, rata-rata error posisi relatif lebih kecil, yakni 0,28 % pada posisi A, 0,22% pada posisi B, dan 0,34% pada posisi C. Pada ruangan persegi berukuran 240 cm x 240 cm, rata-rata error posisi berkisar antara 0,37% hingga 0,78% pada beberapa titik pengujian, dengan error tertinggi mencapai 1,32%.

Pengujian pada ruangan dengan halangan menunjukkan nilai error yang sedikit lebih tinggi dibandingkan bentuk ruang tanpa halangan, dengan rata-rata error pada berbagai posisi berkisar antara 0,5% hingga 4,86%. Error tertinggi tercatat sebesar 8,29% pada salah satu posisi, yang menunjukkan adanya

pengaruh kompleksitas lingkungan terhadap akurasi pemetaan. Meskipun demikian, robot tetap mampu bergerak secara aman dan memetakan seluruh area tanpa mengalami tabrakan.

Secara keseluruhan, data menunjukkan bahwa sistem navigasi yang menggabungkan algoritma Frontier cells detection untuk deteksi area yang belum dipetakan dan algoritma A* untuk perencanaan jalur, didukung oleh kinematik dan odometri, mampu menghasilkan pemetaan yang akurat dan konsisten di berbagai kondisi lingkungan.

REFERENSI

- [1] [1] F. W. Parahyangan, C. Zaskie, and B. M. Arthaya, "Kendali Kecepatan Robot Beroda Omni dengan Kemampuan Menuju Posisi dan Orientasi yang Diinginkan Berbasis Pengendali PID," *JTERA (Jurnal Teknologi Rekayasa)*, vol. 6, no. 2, pp. 273–284, 2021, doi: 10.31544/jtera.v6.i2.2021.273-284.
- [2] Rendyansyah, "Navigasi Pada Mobile Robot Omnidirectional Dalam Menjelajah Ruang Menggunakan Support Vector Machine," *Setrum: Sistem Kendali-Tenaga-elektronika-telekomunikasi-komputer*, vol. 10, no. 1, pp.15-23, 2021, doi: 10.36055/setrum.v10i1.10893.
- [3] A. Prayudi, E. D. Marindani, and F. T. P. W., "Implementasi Potential Field pada Robot Omnidirectional Berbasis LiDAR Menggunakan ROS," *Action Research Literate*, vol. 9, no. 7, pp. 1861-1875, 2025, doi: 10.46799/ar.v9i7.2996.
- [4] A. A. Fikri and L. Anifah, "Mapping and Localization System pada Mobile Robot Menggunakan Metode SLAM Berbasis LiDAR," *JIEET (Journal of Information Engineering and Educational Technology)*, vol. 5, no. 1, pp. 27–33, 2021, doi: 10.26740/jieet.v5n1.p27-33.
- [5] D. Erlangga, E. D., R. H. S., S. Sunarto, K. R. T. S., and F. G., "Sistem Navigasi Mobile Robot dalam Ruang Berbasis Autonomous Navigation," *Journal of Mechanical Engineering and Mechatronics*, vol. 4, no. 2, pp. 78-86, 2019, doi: 10.33021/jmem.v4i2.823.
- [6] F. I. Firmansyah, M. S. Zuhrie, M. Rohman, E. Endryansyah, and M. Aulia, "Sistem Obstacle Avoidance pada Omnidirectional Mobile Robot dengan Metode Artificial Potential Field (APF) Berbasis Fuzzy Logic Controller (FLC)," *Jurnal Teknik Elektro*, vol. 12, no. 2, pp. 6–13, 2023, doi: 10.26740/jte.v12n2.p6-13.
- [7] M. B. Alatise and G. P. Hancke, "A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods," *IEEE Access*, vol. 8, pp. 39830–39846, Mar. 2020, doi: 10.1109/ACCESS.2020.2975643.
- [8] S. Prayoga and D. Sahidan, "Implementasi Pemetaan Robot Roda Mecanum Otonom Berbasis LIDAR dengan SLAM," *Jurnal Integrasi*, vol. 17, no. 1, 2025, doi: 10.30871/ji.v17i1.8575.
- [9] H. Soebhakti and R. H. Pangantar, "Simulation of Mobile Robot Navigation System using Hector SLAM on ROS," *Jurnal Integrasi*, vol. 16, no. 1, pp. 11-20, 2024, doi: 10.30871/ji.v16i1.5755.
- [10] P. Quin, D. D. K. Nguyen, T. L. Vu, A. Alempijevic, and G. Paul, "Approaches for Efficiently Detecting Frontier Cells in Robotics Exploration," *Front Robot AI*, vol. 8, Feb. 2021, doi: 10.3389/frobt.2021.616470.
- [11] M. A. Yaqin, "Implementasi Path Planning A* Untuk Menghindari Robot lawan Menggunakan Robot Operating System pada KRSBI-Beroda," *Jurnal Elkolind*, vol. 12, no. 1, pp. 98–111, May 2025, doi: 10.33795/elkolind.v12i1/7268.
- [12] D. U. Rijalusalam and I. Iswanto, "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)*, vol. 2, no. 5, pp. 448–455, Sep. 2021, doi: 10.18196/jrc.25121.
- [13] F. Wahab, "Metode Potential Field sebagai Kendali Robot Roda Omni untuk Menuju Target dan Menghindari Rintangan," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 10, no. 1, p. 177, Jan. 2022, doi: 10.26760/elkomika.v10i1.177.
- [14] S. Setiawan, E. D. Marindani, and B. Wibowo Sanjaya, "Navigasi Berbasis Koordinat dengan Penghindaran Rintangan untuk Robot Omnidireksional," *Jurnal Teknik Elektro dan Komputasi (ELKOM)*, vol. 6, no. 2, pp. 191–200, Aug. 2024, doi: 10.32528/elkom.v6i2.22493.
- [15] R. F. Syihabuddin, M. N. Jauhari, M. Khudzaifah, and H. Fahmi, "Implementasi Algoritma A-Star dalam Menentukan Rute Terpendek Destinasi Wisata Kota Malang," *Jurnal Riset Mahasiswa Matematika*, vol. 1, no. 5, pp. 236–245, Jun. 2022, doi: 10.18860/jrmm.v1i5.14497.
- [16] D. Herath and D. St-Onge Eds, *Foundations of Robotics A Multidisciplinary Approach with Python and ROS A Multidisciplinary Approach with Python and ROS*. 2022. doi: 10.1007/978-981-19-1983-1.

- [17] A. Rahman, "Penerapan SLAM Gmapping dengan Robot Operating System Menggunakan Laser Scanner pada Turtlebot," *Jurnal Rekayasa Elektrika*, vol. 16, no. 2, pp. 103–109, Aug. 2020, doi: 10.17529/jre.v16i2.16491.