

# Sistem Navigasi Cerdas Robot Pengantar Makanan Dengan Pendekatan *Machine Learning*

1<sup>st</sup> Fernando Amanda Nikola

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

fernandoamandanikola@student.telkomuniversity.ac.id

2<sup>nd</sup> Muhammad Tsabit

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

tsabit@student.telkomuniversity.ac.id

3<sup>rd</sup> Nadya Alifia Chairunnisa

Fakultas Teknik Elektro

Telkom University

Bandung, Indonesia

alifianadyaa@student.telkomuniversity.ac.id

4<sup>th</sup> Ahmad Tri Hanuranto

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

athanuranto@telkomuniversity.ac.id

5<sup>th</sup> Sony Sumaryo

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

sonysumaryo@telkomuniversity.ac.id

**Abstrak** — Perkembangan teknologi otomasi telah membawa dampak signifikan di berbagai sektor, termasuk layanan restoran. Salah satu inovasi yang mulai banyak diterapkan adalah robot pengantar makanan untuk meningkatkan efisiensi dan ketepatan pelayanan. Namun, penerapan robot dalam lingkungan restoran yang dinamis menghadirkan tantangan tersendiri, seperti perubahan posisi meja, pergerakan pelanggan, serta tata ruang yang bervariasi. Untuk mengatasi kendala tersebut, dikembangkan sistem navigasi cerdas berbasis Machine Learning yang mampu melakukan pemetaan lingkungan secara real-time menggunakan sensor RPLiDAR dan metode SLAM (Simultaneous Localization and Mapping). Proses pemetaan dan pengolahan data dilakukan oleh Raspberry Pi, sementara perencanaan jalur dilakukan dengan algoritma K-Nearest Neighbors (KNN) untuk pengenalan lingkungan dan Breadth-First Search (BFS) untuk pencarian rute terbaik. Jalur kemudian disempurnakan dengan teknik smoothing untuk memastikan pergerakan robot yang mulus. Hasil pengujian menunjukkan bahwa sistem mampu memetakan dua lingkungan berbeda dengan akurasi 99,55% dan akurasi lokalisasi sebesar 99,29% dari 30 titik uji. Rata-rata deviasi pengukuran kurang dari 5 cm, dan robot mampu bergerak stabil pada kecepatan  $\pm 0,3$  m/s serta menghindari rintangan tanpa tabrakan. Sistem ini terbukti meningkatkan efektivitas dan keandalan robot pengantar makanan dalam menghadapi dinamika lingkungan restoran.

**Kata kunci** — navigasi robot, machine learning, SLAM, RPLIDAR, robot pengantar makanan.

tugas-tugas rutin seperti mengantarkan pesanan. Menurut data yang ada, penggunaan robot pengantar makanan mampu mengurangi biaya tenaga kerja sebesar 30-50% dan kesalahan pengiriman menurun sebesar 90%[1]. Selain itu, robot pengantar makanan juga mampu meningkatkan kepuasan pelanggan sampai 70% [2]. Hal ini menunjukkan bahwa sistem otomatis memiliki potensi besar dalam meningkatkan efisiensi dan kualitas layanan.

Pada tahun lalu, Universitas Telkom telah membuat sebuah robot pengantar makanan yang mampu bergerak dengan baik. Tapi robot ini masih dikendalikan secara manual oleh penggunanya untuk mencapai tujuan [3]. Hal ini tidak memenuhi harapan yang diinginkan untuk mendapatkan efisiensi yang sesungguhnya dari robot pengantar makanan. Robot ini membutuhkan pengembangan lanjutan untuk membuat sistem navigasi yang baik dan mampu dijalankan secara otonom. Pada sistem ini, komunikasi data menjadi inti dari seluruh proses operasional. Pertukaran informasi antara robot, sensor, dan infrastruktur jaringan menjadi komponen utama untuk memastikan sistem dapat bekerja dengan sangat efisien dan *real-time*. Robot pengantar makanan yang dilengkapi dengan berbagai sensor seperti kamera, LiDAR, dan ultrasonik untuk memetakan lingkungan sekitarnya. Data yang dihasilkan dari sensor-sensor ini kemudian dikirimkan melalui jaringan untuk diproses oleh algoritma *Machine Learning* yang ada di server pusat.

## I. PENDAHULUAN

Perkembangan teknologi yang semakin maju setiap harinya, sistem otomatis menjadi penggunaan untuk meningkatkan efisiensi berbagai industri seperti pelayanan restoran. Implementasi sistem otomatis pada pelayanan restoran seperti robot pengantar makanan merupakan ide inovatif yang diharapkan mampu membantu manusia dalam

Pada akhirnya, tantangan utama yang dihadapi oleh robot pengantar makanan adalah bagaimana mengintegrasikan komunikasi data yang cepat dan andal dengan sistem kecerdasan buatan yang mampu memproses informasi lingkungan secara *real-time*. Masalah inilah yang menjadi dasar dari proyek ini, yang bertujuan untuk mengembangkan sistem navigasi cerdas berbasis *Machine Learning* yang didukung oleh infrastruktur telekomunikasi

berlatensi rendah dan mempunyai *bandwidth* tinggi, sehingga dapat meningkatkan kinerja robot dalam pengantaran makanan di lingkungan dinamis.

## II. KAJIAN TEORI

### A. Robot Pengantar Makanan

Robot pengantar makanan adalah salah satu jenis robot yang dirancang untuk mengantar makanan atau pesanan menuju pelanggan [4]. Robot ini membutuhkan berbagai bantuan teknologi seperti sensor dan mikrokontroler yang saling berkerja satu sama lain untuk menciptakan navigasi robot yang baik sehingga robot mampu beroperasi secara otonom. Tujuan dari robot ini adalah untuk meningkatkan efisiensi tenaga dalam pengiriman makanan, mengurangi biaya tenaga kerja, dan meningkatkan kepuasan pelanggan [5].

### B. SLAM (Simultaneous Localization and Mapping)

SLAM adalah sebuah metode yang digunakan untuk membuat peta lingkungan sekitar dan menentukan posisinya berdasarkan peta yang telah dibuat secara bersamaan dengan menggunakan data sensor yang mendukung [6]. SLAM merupakan metode yang sering digunakan untuk mendukung navigasi robot sehingga robot dapat beroperasi secara otonom. SLAM memiliki berbagai algoritma untuk beroperasi seperti GMapping dan Hector SLAM. Algoritma ini memungkinkan robot untuk berjalan secara otonom di lingkungan yang kompleks dan dinamis [6].

### C. Navigasi

Navigasi adalah proses yang memungkinkan robot untuk secara otonom menentukan jalur dari titik awal menuju tujuan yang diinginkan sambil menghindari rintangan di sepanjang perjalanannya [6]. Sistem navigasi robot modern terdiri dari tiga komponen utama, yaitu lokalisasi, perencanaan lintasan (*path planning*), dan kontrol gerakan. Lokalisasi merujuk pada kemampuan robot untuk mengetahui posisi dan orientasinya di dalam peta lingkungan, yang sering kali dihasilkan melalui metode SLAM (Simultaneous Localization and Mapping).

### D. Machine Learning

Machine Learning (ML) merupakan cabang dari kecerdasan buatan (AI) yang memungkinkan sistem untuk belajar dari data secara otomatis [8]. Dengan sistem ML bekerja dengan membangun model matematis berdasarkan pola dalam data, yang kemudian dapat digunakan untuk melakukan prediksi, klasifikasi, atau pengambilan keputusan secara otomatis.

### E. ROS (Robotic Operating System)

Robot Operating System (ROS) bukanlah sistem operasi dalam pengertian tradisional, melainkan sebuah kerangka kerja perangkat lunak berbasis *middleware* yang dirancang untuk mendukung pengembangan aplikasi robotika secara modular dan terdistribusi [7]. ROS menyediakan serangkaian alat bantu (*tools*), pustaka (*libraries*), dan konvensi yang memungkinkan para pengembang untuk merancang dan mengelola sistem perangkat lunak robot secara efisien.

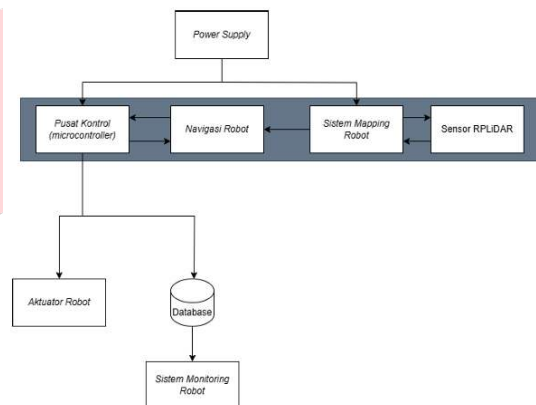
## III. METODE

### F. DBScan

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) adalah algoritma *machine learning* dengan pembelajaran *Unsupervised Learning* [9].

### G. Blok Diagram Sistem

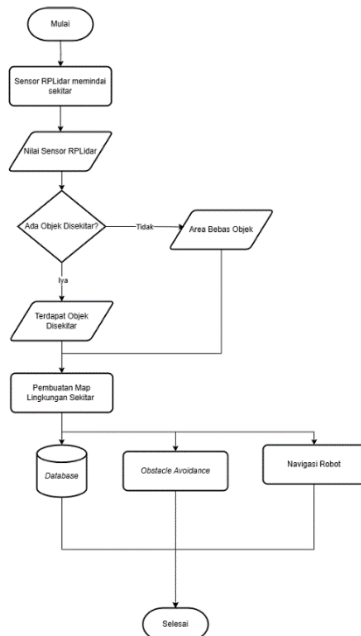
*Flowchart* di bawah ini menggambarkan alur kerja lengkap dari sistem navigasi robot pengantar makanan. Adapun fokus penelitian dalam jurnal ini dibatasi pada bagian yang disorot dengan warna berbeda.



GAMBAR 1  
Blok Diagram Sistem

Sistem *mapping* robot akan mendapatkan daya melalui power supply untuk mendapatkan data terkait objek disekitar robot melalui sensor RPLiDAR. Data tersebut akan dikumpulkan menjadi sebuah peta secara menyeluruh. peta yang terbentuk akan menjadi acuan robot untuk bergerak dan menghindari objek disekitar, sehingga hal ini akan membentuk sistem navigasi robot. Saat sistem ini terbentuk, data yang didapat akan dikirimkan menuju pusat kontrol untuk menggerakkan aktuator robot.

## H. Flowchart Mapping System



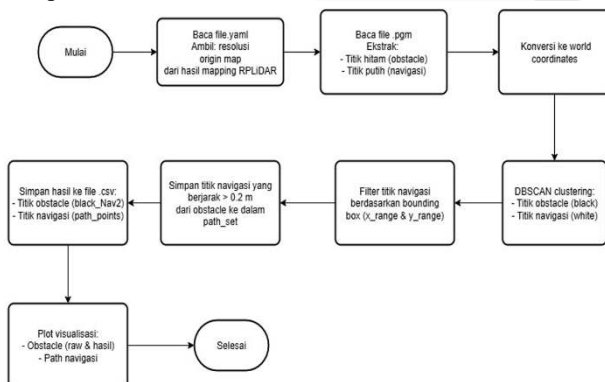
GAMBAR 2

Flowchart Sistem Mapping

Proses dimulai dengan inisialisasi sensor RPLiDAR untuk memindai area sekitar. Data dari sensor kemudian diterima oleh sistem untuk dilakukan deteksi objek. Jika terdeteksi ada objek, area sekitar diolah menjadi peta dengan memperhatikan kondisi lingkungan dan hambatan yang ada. Jika tidak ada objek, sistem tetap memperbarui peta berdasarkan kondisi kosong. Selanjutnya, hasil pemetaan dikirimkan ke *database* dan digunakan untuk perencanaan navigasi robot. Robot kemudian dapat bergerak berdasarkan peta yang telah diperoleh.

## I. Flowchart DBScan

DBSCAN adalah algoritma *unsupervised learning* yang mengelompokkan data berdasarkan kepadatan[9]. Dalam bagian ini, DBSCAN digunakan untuk mengurangi *noise* dalam peta.



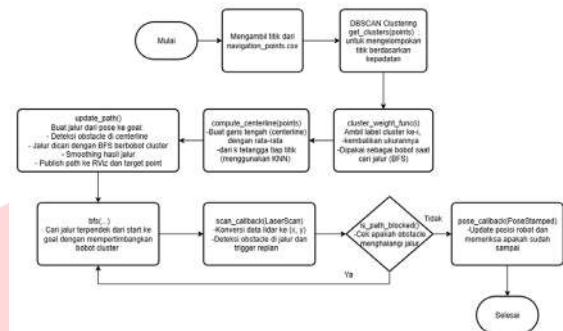
GAMBAR 3

Flowchart DBSCAN

Flowchart diatas dimulai dengan pemrosesan peta lingkungan dalam format .pgm dan metadata .yaml untuk mengubah citra ke koordinat dunia. *Obstacle* diidentifikasi sebagai piksel hitam, dan area navigasi sebagai piksel putih. Data kemudian dikluster menggunakan DBSCAN untuk

mengelompokkan area dan mengurangi *noise*. *Filtering* dilakukan untuk memilih titik navigasi yang berada dalam batas area tertentu dan memiliki jarak aman ( $>0,2$  m) dari *obstacle*. Titik-titik hasilnya disimpan dalam format .csv dan divisualisasikan untuk mendukung perencanaan jalur robot.

## J. Flowchart Navigasi

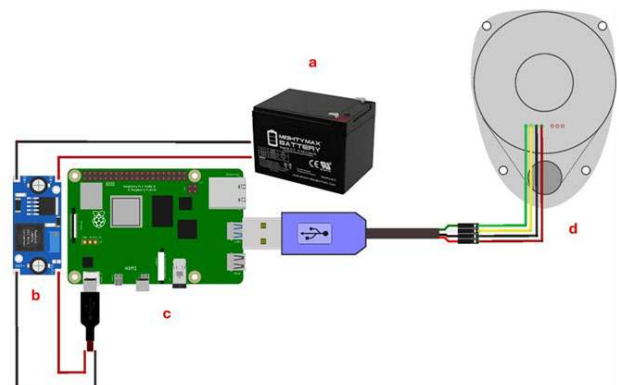


GAMBAR 4

Flowchart Navigasi

Proses perencanaan jalur dimulai dengan pengambilan titik navigasi dari file *navigation\_points.csv*. Titik-titik ini kemudian dikelompokkan menggunakan algoritma DBSCAN untuk membentuk *cluster* berdasarkan kepadatan. Hasil *clustering* diproses untuk menghitung bobot setiap *cluster*, yang dihitung dari ukuran masing-masing *cluster* dan digunakan sebagai pertimbangan dalam pencarian jalur. Selanjutnya, dilakukan pembentukan *centerline* atau garis tengah dari titik-titik navigasi menggunakan metode K-Nearest Neighbors (KNN), dengan mengambil rata-rata posisi dari tetangga terdekat tiap titik. Jalur dari posisi awal ke tujuan kemudian direncanakan menggunakan algoritma Breadth-First Search (BFS) yang mempertimbangkan bobot *cluster*, serta dilakukan smoothing untuk memperhalus hasil jalur. Sistem juga menerima data dari sensor LiDAR yang dikonversi ke dalam koordinat (x, y) dan digunakan untuk mendeteksi *obstacle*. Jika ditemukan rintangan yang menghalangi jalur, sistem akan memicu *replanning* untuk mencari jalur alternatif. Posisi robot diperbarui secara berkala untuk mengecek apakah robot telah sampai ke tujuan.

## K. Desain Perangkat Keras Sistem Navigasi



GAMBAR 5

Desain Hardware Navigasi

Sistem ini menggunakan DC Power Supply 12V sebagai sumber daya utama. Tegangan 12V kemudian diturunkan menjadi 5V menggunakan step down converter, yang selanjutnya digunakan untuk memberi daya ke Raspberry Pi. Raspberry Pi berperan sebagai pusat pengolahan data dalam sistem. Perangkat ini terhubung dengan sensor RPLiDAR melalui sambungan pin TX dan RX untuk komunikasi serial, serta pin GND dan 5V untuk suplai daya ke sensor.

#### IV. HASIL DAN PEMBAHASAN

##### A. Pengukuran Jarak dengan RPLiDAR



GAMBAR 6  
Pengukuran Jarak RPLiDAR

Pengujian ini dilakukan untuk mengevaluasi akurasi pengukuran jarak oleh sensor RPLiDAR di 3 jarak referensi, yaitu pada jarak 30 cm, 60 cm dan 2 m, dengan jumlah pengambilan 30 data dari setiap 45 ° sudut pemindaian. Tujuan dari pengujian ini adalah untuk memastikan bahwa RPLiDAR mampu mengukur jarak secara konsisten dan akurat dalam kondisi lingkungan yang dikendalikan.

Pengujian dilakukan dengan menempatkan objek pantul/target pada jarak tertentu dari sensor dan mencatat hasil pengukuran pada berbagai sudut terhadap sumbu depan sensor. Berikut adalah hasil pengambilan data jarak menggunakan RPLiDAR.

|                    |                |          |
|--------------------|----------------|----------|
| 1747542568.111511] | Jarak pada 0°: | 31.10 cm |
| 1747542568.187430] | Jarak pada 0°: | 31.00 cm |
| 1747542568.263388] | Jarak pada 0°: | 31.00 cm |
| 1747542568.339313] | Jarak pada 0°: | 31.10 cm |
| 1747542568.402875] | Jarak pada 0°: | 31.00 cm |
| 1747542568.478924] | Jarak pada 0°: | 30.90 cm |
| 1747542568.555289] | Jarak pada 0°: | 31.10 cm |
| 1747542568.618954] | Jarak pada 0°: | 31.00 cm |
| 1747542568.694782] | Jarak pada 0°: | 30.90 cm |
| 1747542568.770513] | Jarak pada 0°: | 31.00 cm |
| 1747542568.846709] | Jarak pada 0°: | 31.00 cm |
| 1747542568.910261] | Jarak pada 0°: | 31.00 cm |
| 1747542568.986281] | Jarak pada 0°: | 31.00 cm |
| 1747542569.062517] | Jarak pada 0°: | 31.00 cm |
| 1747542569.125386] | Jarak pada 0°: | 31.00 cm |
| 1747542569.202144] | Jarak pada 0°: | 30.90 cm |
| 1747542569.278165] | Jarak pada 0°: | 31.00 cm |
| 1747542569.353950] | Jarak pada 0°: | 31.00 cm |
| 1747542569.417829] | Jarak pada 0°: | 31.00 cm |
| 1747542569.493553] | Jarak pada 0°: | 31.00 cm |
| 1747542569.569908] | Jarak pada 0°: | 31.00 cm |
| 1747542569.633446] | Jarak pada 0°: | 31.00 cm |
| 1747542569.709482] | Jarak pada 0°: | 31.00 cm |

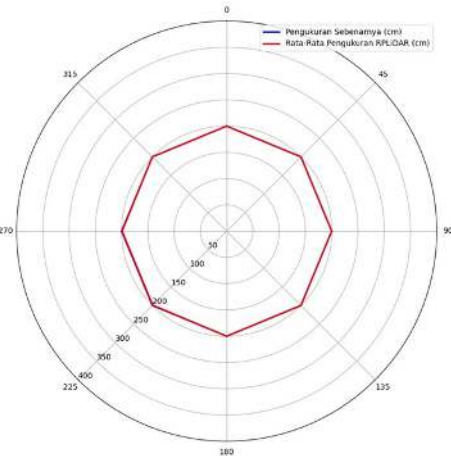
GAMBAR 7  
Timestamp ROS

Untuk mendapatkan berapa waktu 1 rotasi rplidar digunakan rumus berikut yang didapat dari *timestamp* ros.

(5.1)

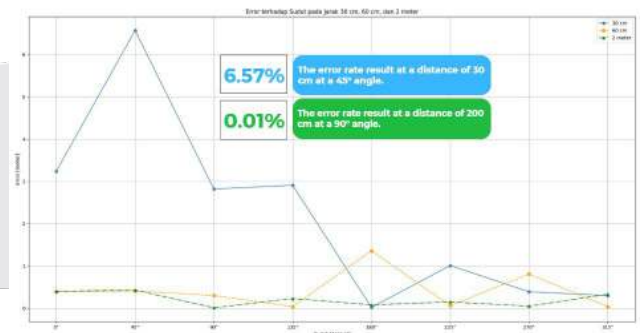
$$Delay = \sum_{n=0}^{\infty} timestamp_{n+1} - timestamp_n$$

Untuk mengetahui berapa nilai *timestamp* dapat dilihat pada gambar diatas yang menggunakan ros untuk mengetahui berapa *timestamp* yang didapat.



GAMBAR 8  
Visualisasi Perbandingan Pengukuran

Gambar 8 ini menunjukkan perbandingan antara jarak sebenarnya dan rata-rata hasil pengukuran RPLiDAR pada berbagai sudut. Garis biru mewakili jarak sebenarnya, sedangkan garis merah menunjukkan hasil dari RPLiDAR. Kedua garis tampak berdekatan, menandakan bahwa RPLiDAR memiliki akurasi tinggi dan konsistensi yang baik dalam mendeteksi jarak di segala arah



GAMBAR 9  
Grafik Error Pengukuran

Gambar 9 menunjukkan hubungan antara sudut pengukuran dan tingkat kesalahan /eror RPLiDAR. Sumbu X menunjukkan sudut dalam derajat, sedangkan sumbu Y menunjukkan besar eror dalam meter. Terlihat bahwa error cenderung meningkat pada sudut sekitar 0°, 45°, dan 315°, namun sebagian besar tetap di bawah 0,2 meter. Hal ini menunjukkan bahwa RPLiDAR memiliki tingkat akurasi yang cukup baik meskipun terdapat fluktuasi eror pada sudut tertentu.

Untuk Selisih, Presentase Error, dan Akurasi dapat dihitung menggunakan ini:

Selisih dapat dihitung dengan menggunakan rumus di bawah ini untuk mengetahui seberapa jauh perbedaan antara jarak hasil yang didapat dengan jarak sebenarnya.

(5.2)

$$\text{Selisih} = \text{Jarak Hasil} - \text{Jarak Sebenarnya}$$

Persentase Error dapat dihitung dengan rumus di bawah ini untuk mengetahui seberapa besar kesalahan dalam bentuk persentase dari nilai sebenarnya.

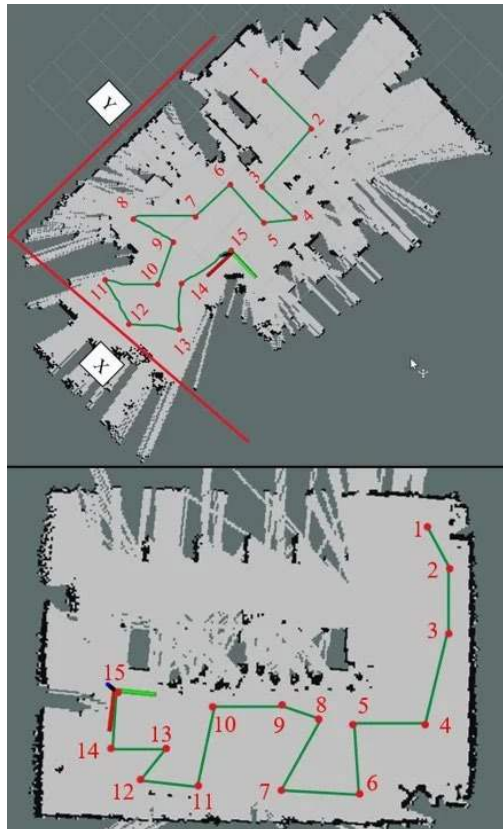
(5.3)

$$\text{presentase error}(\%) = \frac{\text{Selisih}}{\text{Jarak Sebenarnya}} \times 100$$

Persentase Akurasi dapat dihitung dengan rumus di bawah ini untuk memberikan angka seberapa tepat hasilnya dibandingkan nilai sebenarnya, jika error 0% berarti akurasi 100% (hasil sempurna) dan jika error besar maka akurasi akan rendah.

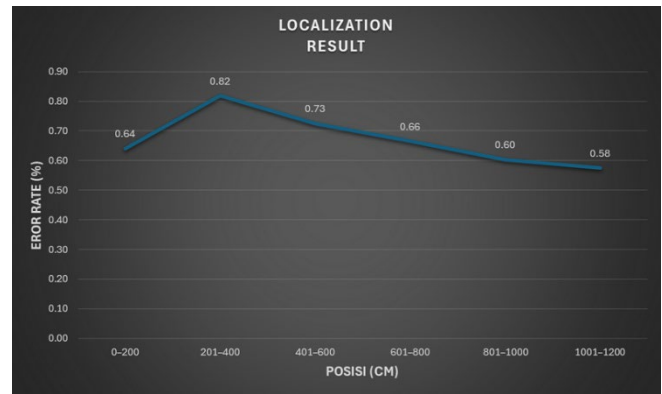
(5.4)

$$\text{akurasi}(\%) = 100\% - \text{presentase error}(\%)$$



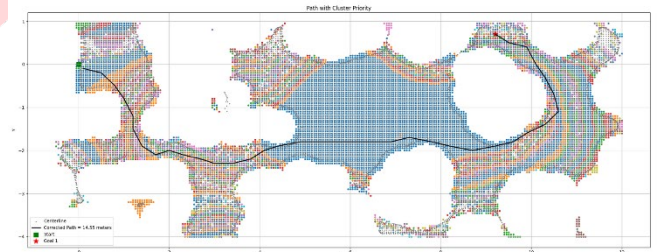
GAMBAR 10  
Pengujian Lokalisasi

Pengujian dilakukan untuk membandingkan posisi estimasi dari SLAM dengan posisi aktual robot yang diukur secara manual. Sebanyak 30 titik diuji untuk menilai akurasi sistem berdasarkan selisih kedua posisi tersebut.



GAMBAR 11  
Hasil Pengujian Lokalisasi

Hasil menunjukkan bahwa estimasi posisi robot memiliki rata-rata error sebesar 0,71% dibandingkan dengan posisi aktual. Nilai error yang rendah ini menunjukkan bahwa metode SLAM mampu menentukan posisi robot secara presisi dan mendukung navigasi.



GAMBAR 12  
Hasil pengujian Mapping

Hasil menunjukkan bahwa sistem menghasilkan jalur sepanjang 14,55 meter dari titik awal ke tujuan dengan mempertimbangkan prioritas kluster. Jalur cenderung melewati area dengan kepadatan titik tinggi, khususnya pada kluster berukuran besar. Hal ini menunjukkan bahwa algoritma berhasil meminimalkan risiko tabrakan dengan tetap menjaga efisiensi jarak tempuh. Untuk mengetahui panjang jalur dapat menggunakan euclidean distance dengan:

(5.5)

$$\text{dist} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

| Scenario | Average Path Length (m) | Percentage of distance to obstacle (%) |               |              |
|----------|-------------------------|----------------------------------------|---------------|--------------|
|          |                         | 0.0 - 0.40 m                           | 0.41 - 0.80 m | 0.81 - 1.0 m |
| 1        | 5.32                    | 0%                                     | 21%           | 21%          |
| 2        | 7.86                    | 0%                                     | 18%           | 28%          |
| 3        | 3.26                    | 0%                                     | 10%           | 24%          |

GAMBAR 13  
Persentase Jarak Obstacle

Pengujian pada tiga skenario dengan masing-masing 10 percobaan menunjukkan bahwa tidak ada lintasan yang berada dalam jarak sangat dekat dengan *obstacle* (0.0–0.40 m). Sebagian besar lintasan berada pada jarak aman, yaitu >0.8 m, dengan proporsi tertinggi sebesar 28% pada skenario

2. Hasil ini menunjukkan bahwa algoritma perencanaan jalur mampu menjaga jarak aman dari *obstacle* dan mendukung navigasi yang dinamis.

#### V. KESIMPULAN

Implementasi sistem navigasi cerdas pada robot pengantar makanan membuktikan bahwa teknologi mampu menyederhanakan tugas-tugas harian secara signifikan. Robot ini dirancang agar dapat mengenali lingkungan, menghindari rintangan, dan mencapai tujuan secara mandiri dengan memanfaatkan sensor RPLiDAR A2 dan Raspberry Pi 4B sebagai pusat navigasi. Sistem ini menggabungkan algoritma DBSCAN untuk identifikasi area aman, K-Nearest Neighbor (KNN) dan Breadth-First Search (BFS) untuk perencanaan jalur, serta teknik smoothing untuk menghasilkan pergerakan yang lebih halus dan efisien. Seluruh proses navigasi didukung oleh metode SLAM untuk pemetaan dan pelacakan posisi secara *real-time*.

Serangkaian pengujian dilakukan untuk mengevaluasi kinerja sistem dari aspek pengukuran jarak, pemetaan, lokalisasi, hingga navigasi otomatis. Hasil pengukuran menunjukkan akurasi tinggi dari sensor RPLiDAR dengan deviasi kecil pada jarak 30 cm dan 60 cm. Akurasi pemetaan mencapai 99,55% dan lokalisasi mencapai 99,29% dari 30 titik uji, menunjukkan ketelitian sistem dalam memahami dan menavigasi lingkungan. Navigasi robot terbukti adaptif terhadap kondisi dinamis dengan kecepatan stabil  $\pm 0,3$  m/s. Meskipun masih terdapat kendala seperti kesulitan mengenali objek transparan, sistem secara keseluruhan menunjukkan solusi potensial bagi industri layanan makanan.

#### REFERENSI

- [1] Marche, "Making robot delivery a reality," *Marche Mates*, Oct. 2, 2024. [Online]. Available: <https://marchemates.ca/making-robot-delivery-a-reality>
- [2] ScaleZone Technologies, "The power of restaurant automation." [Online]. Available: <https://www.scalezonetech.com/the-power-of-restaurant-automation>
- [3] F. Rifqy, N. Gebby, and R. Muhammad, *Buku Tugas Akhir Capstone Design: Mobilisasi Robot Pengantar Makanan Berbasis Odometry dan QR Detection*, Bandung, Indonesia: Self-published, Dec. 2023.
- [4] Parenrengi, K. M. (2021). *Rancang Bangun Prototype Alat Pengantar Makanan Pada Rumah Makan.* " *J. MOSFET*, vol. 1, no. 2, 2021.
- [5] B. K. Nugraha, A. S. Santia, Z. Aurellia, P. Pangaribuan, and I. M. Rodiana, "Robot pengantar makanan berbasis line follower dengan sensor warna TCS3200 dan Internet of Things (IoT)," *JiIP – J. Ilm. Ilmu Pendidik.*, vol. 6, no. 11, pp. 8928–8933, 2023, [Online]. Available: <https://doi.org/10.54371/jiip.v6i11.2774>
- [6] M. A. Maulana, et al., "Autonomous mobile robot dengan menggunakan metode Simultaneous Localization and Mapping berbasis LIDAR," in *Proc. Sem. Nas. Inovasi Teknol. Terapan*, Politeknik Manufaktur Negeri Bangka Belitung, 2022.
- [7] Y. Louise, et al., "Mapping dan navigasi untuk robot pengantar makanan di restoran berbasis ROS," *Techné: J. Ilm. Elektroteknika*, vol. 22, no. 1, pp. 111–128, 2023.
- [8] Dicoding Indonesia, "Machine Learning Adalah: Pengertian, Cara Kerja, dan Contohnya," *Dicoding Blog*, Sep. 22, 2022. [Online]. Available: <https://www.dicoding.com/blog/machine-learning-adalah/>
- [9] D. Tamara, "BFS (Breadth First Search): Pengertian, Kekurangan, Kelebihan dan Contohnya," *Medium*, Mar. 17, 2023. [Online]. Available: <https://medium.com/@defytamara2610/bfs-breadth-first-search-pengertian-kekurangan-kelebihan-dan-contohnya-775a7d808fbc>