

Perencanaan Jalur *TurtleBot3* Berbasis A* dengan LiDAR dalam Simulasi Gazebo

Zindhu Maulana Ahmad Putra¹, Muhammad Ardi Kurniawan², Lilik Subiyanto³, Dimas Pristovani Riananda⁴, Rocky Andiana⁵

^{1,4} Program Studi Teknik Kelistrikan Kapal, Jurusan Teknik Kelistrikan Kapal, Politeknik Perkapalan Negeri Surabaya Kampus ITS Sukolilo Jl. Teknik Kimia, Keputih Sukolilo, Surabaya, Indonesia

^{2,3} Program Studi Teknik Otomasi, Jurusan Teknik Kelistrikan Kapal, Politeknik Perkapalan Negeri Surabaya Kampus ITS Sukolilo Jl. Teknik Kimia, Keputih Sukolilo, Surabaya, Indonesia

⁵ Program Studi Teknik Pengelasan, Jurusan Teknik Bangunan Kapal, Politeknik Perkapalan Negeri Surabaya Kampus ITS Sukolilo Jl. Teknik Kimia, Keputih Sukolilo, Surabaya, Indonesia

zindhu@ppns.ac.id¹, mardi20@student.ppns.ac.id², llksubiyanto@ppns.ac.id³, dimaspristovani@ppns.ac.id⁴, rockyandiana@ppns.ac.id⁵

Abstrak

Navigasi merupakan komponen krusial dalam pengembangan robot otonom, khususnya ketika robot diharuskan bergerak dalam lingkungan yang kompleks dan penuh hambatan. Penelitian ini mengimplementasikan sistem navigasi berbasis algoritma A* dengan dukungan sensor LiDAR, rotary encoder, dan scanner GM65 untuk meningkatkan kemampuan pemetaan dan perencanaan jalur. Sensor LiDAR digunakan untuk membentuk peta lingkungan dalam bentuk occupancy grid map, sementara rotary encoder dan sensor MPU9250 memberikan data odometri untuk pelacakan posisi robot. Data sensor diproses secara terintegrasi oleh ESP32 dan Jetson Nano menggunakan Robot Operating System (ROS). Simulasi dilakukan menggunakan platform Gazebo untuk menguji kinerja sistem dalam lingkungan virtual. Hasil visualisasi melalui *rqt_graph* menunjukkan interaksi yang efisien antar node ROS. Algoritma A* berhasil merancang jalur optimal dari titik awal menuju tujuan dengan mempertimbangkan hambatan lingkungan. Jalur tersebut kemudian diikuti robot menggunakan kendali kecepatan linear dan sudut yang disesuaikan berdasarkan arah menuju titik sasaran. Hasil simulasi menunjukkan bahwa sistem mampu mengarahkan robot secara akurat menuju target dengan perubahan posisi yang konsisten terhadap sumbu koordinat. Penggunaan kombinasi ROS, algoritma A*, dan simulasi Gazebo terbukti efektif sebagai kerangka uji sebelum sistem diterapkan secara nyata. Dengan demikian, penelitian ini memberikan solusi navigasi yang adaptif, efisien, dan dapat diandalkan untuk aplikasi robotika di berbagai bidang.

Kata kunci: Robot Otonom, Sistem Navigasi, Algoritma A*, LiDAR, ROS

Abstract

Navigation is a crucial component in the development of autonomous robots, especially when operating in complex and obstacle-rich environments. This research implements a navigation system based on the A* algorithm supported by LiDAR sensors, rotary encoders, and a GM65 scanner to enhance mapping and path-planning capabilities. The LiDAR sensor is used to generate an occupancy grid map of the environment, while the rotary encoder and MPU9250 provide odometry data for accurate position tracking. Sensor data is processed in an integrated manner by an ESP32 microcontroller and a Jetson Nano using the Robot Operating System (ROS). Simulations were conducted using the Gazebo platform to evaluate system performance in a virtual environment. Visualization through *rqt_graph* illustrates efficient interaction between ROS nodes. The A* algorithm successfully plans the optimal path from the start to the goal point by considering environmental obstacles. This path is then followed by the robot using a simple control method, adjusting linear and angular velocities based on the direction toward the target point. Simulation results demonstrate that the system can accurately guide the robot toward the destination, with consistent position updates along the coordinate axis. The integration of ROS, the A* algorithm, and the Gazebo simulator proves to be an effective testing framework before real-world implementation. Thus, this research offers an adaptive, efficient, and reliable navigation solution for robotics applications across various domains.

Keywords: Autonomous Robot, Navigation System, A* Algorithm, LiDAR, ROS

1. Pendahuluan

Navigasi adalah elemen penting dalam pengembangan robot otonom, terutama ketika robot harus bergerak di lingkungan yang rumit dan tidak teratur. Agar dapat beroperasi secara mandiri, robot perlu memiliki kemampuan untuk merencanakan jalur pergerakan dan menghindari hambatan di sekitarnya. Kemampuan ini sangat penting dalam berbagai bidang, mulai dari industri manufaktur dan sistem transportasi otomatis, hingga misi eksplorasi di area berbahaya. Untuk mewujudkan navigasi yang efektif dan aman, dibutuhkan sinergi antara sistem yang mampu memahami kondisi lingkungan serta algoritma cerdas yang dapat menentukan jalur terbaik.

Sensor *Light Detection and Ranging* (LiDAR) merupakan salah satu perangkat utama dalam sistem persepsi robotik karena kemampuannya dalam memetakan lingkungan sekitar secara detail, cepat, dan akurat. LiDAR bekerja dengan memancarkan pulsa laser dan mengukur waktu pantulan untuk menentukan jarak objek di sekitarnya, sehingga menghasilkan data spasial tiga dimensi yang sangat berguna untuk navigasi. (Lee et al., 2024) Data mentah dari LiDAR kemudian dikonversi menjadi representasi peta lingkungan seperti occupancy grid map, yang memetakan area bebas dan terhalang dalam bentuk sel-sel grid. (Fikri & Anifah, 2021) Peta ini menjadi landasan penting bagi algoritma perencanaan jalur. Salah satu algoritma yang paling banyak diterapkan dalam konteks ini adalah A* (A-star), yakni algoritma pencarian jalur berbasis graf yang memanfaatkan fungsi heuristik untuk memperkirakan biaya menuju tujuan. Dengan menggabungkan biaya aktual dan estimasi biaya ke tujuan, A* mampu menghitung jalur optimal secara efisien dalam lingkungan yang diketahui (RIYAN FAHMI SYIHABUDDIN, n.d.).

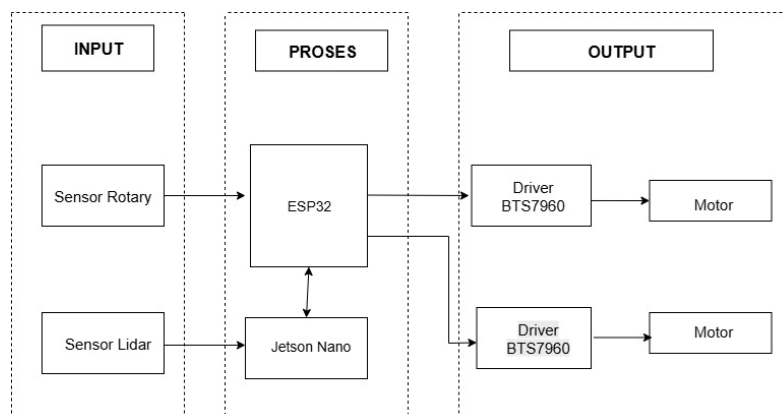
Platform Gazebo, yang secara native terintegrasi dengan Robot Operating System (ROS) (Rosadi et al., 2024), telah menjadi standar dalam pengembangan dan pengujian algoritma robotika secara virtual. Gazebo menyediakan lingkungan simulasi tiga dimensi yang realistis, lengkap dengan dukungan untuk dinamika fisika, pencahayaan, sensor, dan interaksi antar objek, sehingga memungkinkan pengujian sistem robotik dengan kondisi yang menyerupai dunia nyata. Simulasi pada Gazebo memungkinkan pengguna untuk mensimulasikan berbagai jenis robot, mulai dari robot roda hingga lengan robotik, serta menguji berbagai skenario lingkungan seperti rintangan dinamis, permukaan yang tidak rata, hingga variasi kondisi pencahayaan. Keunggulan utama dari simulasi ini adalah kemampuannya dalam menguji dan memverifikasi algoritma seperti navigasi, pelacakan jalur, dan penghindaran rintangan secara aman tanpa risiko merusak perangkat keras. Selain itu, integrasi erat antara Gazebo dan ROS memungkinkan pertukaran data secara real-time antara simulasi dan sistem kontrol robot, sehingga pengujian dalam simulasi dapat langsung diterapkan ke robot fisik dengan sedikit atau tanpa perubahan kode. (Susanthi, n.d.)

2. Metode Penelitian

Pada bagian ini menjelaskan konsep penelitian yang akan menjadi dasar penelitian. Konsep penelitian digunakan sebagai acuan atau landasan bagi desain dan pengembangan sistem yang akan dilakukan oleh peneliti.

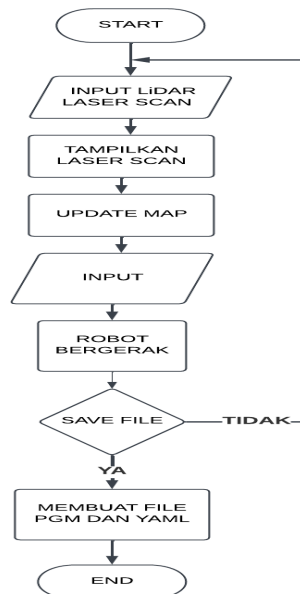
2.1 Perancangan Sistem

Pada sub bab perancangan sistem meliputi diagram blok dan flowchart sistem.



Gambar 1. Diagram Blok Sistem

- **INPUT** terdapat 2 input sensor terdiri dari:
 1. Sensor LiDAR Laki Beam: Berfungsi untuk mendeteksi objek di sekitar sekaligus membuat peta lingkungan. Data hasil pemindaian dikirimkan ke laptop melalui USB serial untuk diproses menjadi peta visual.
 2. Rotary Encoder: Digunakan untuk menghitung jarak tempuh dan menentukan orientasi robot berdasarkan perhitungan wheel odometry yang dilakukan di ESP32. Data ini kemudian dikirim ke Jetson Nano untuk pemrosesan lebih lanjut menggunakan ROS.
- **PROSES** menggunakan Mikrokontroler esp32 dan Jetson Nano
 4. ESP32: Memproses data dari rotary encoder untuk menghasilkan informasi jarak dan orientasi. Dan Mengirimkan data tersebut ke Jetson Nano untuk integrasi dengan data sensor lainnya.
 5. Jetson Nano: Mengolah data dari ROS, termasuk informasi dari sensor LiDAR Melakukan perencanaan jalur (path planning) untuk menentukan kecepatan gerak robot. Dan Mengirimkan instruksi kecepatan (speed) ke sistem penggerak pada blok output.
- **OUTPUT** terdiri dari empat driver IBT2, Empat motor DC 45
 6. Motor DC PG45: Digerakkan oleh driver motor berdasarkan instruksi kecepatan dari ROS untuk menggerakkan robot sesuai rute yang direncanakan.



Gambar 2. Flowchart Proses Mapping

pada Gambar 1. menunjukkan diagram blok di mana sensor LiDAR berfungsi untuk mendeteksi objek dan memetakan lingkungan. Data hasil pemindaian dikirim ke Jetson Nano untuk diproses. Proses kerja LiDAR saat memulai pemindaian ditampilkan pada Gambar 2.

2.2 Algoritma A* (A-star)

Algoritma A* sebagai metode pencarian jalur untuk menentukan lintasan terpendek secara efisien dalam lingkungan yang telah dipetakan. Algoritma ini menggabungkan dua pendekatan, yaitu *Uniform Cost Search*, yang memperhitungkan biaya aktual dari titik awal ke simpul saat ini ($g(n)$), serta *Greedy Best-First Search*, yang memanfaatkan estimasi jarak ke tujuan ($h(n)$). (Hermawan & Setiyani, 2019) Total biaya dievaluasi berdasarkan

fungsi $f(n) = g(n) + h(n)$, dan simpul dengan nilai $f(n)$ terkecil diprioritaskan untuk dieksplorasi terlebih dahulu, sehingga pencarian jalur menjadi lebih terarah dan optimal.

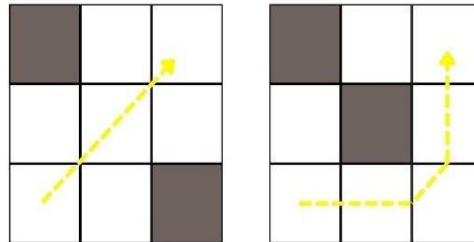
Contoh persamaan matematika:

$$f(n) = g(n) + h(n) \quad (1)$$

$F(n)$ = biaya estimasi

$g(n)$ = biaya yang sudah dikeluarkan dari titik awal sampai keadaan n

$h(n)$ = estimasi biaya untuk sampai pada titik tujuan dimulai dari keadaan n



Gambar 1. Simulasi Path planning A*

2.3 Fungsi Heuristic

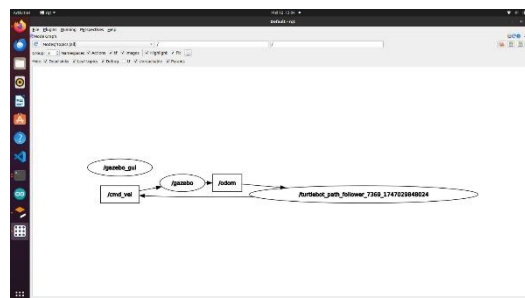
fungsi heuristic untuk ‘menuntun’ pencarian rute, khususnya dalam hal pengembangan dan pemeriksaan node-node pada peta. Dalam aplikasi ini, fungsi heuristic yang dipakai untuk pencarian rute mengisi nilai/notasi h pada algoritma A*. Ada beberapa fungsi heuristic umum yang bisa dipakai untuk algoritma BFS dan A* ini. Salah satunya adalah yang dikenal dengan istilah ‘Manhattan Distance’. Fungsi heuristic ini digunakan untuk kasus dimana pergerakan pada peta hanya lurus (horizontal atau vertikal), tidak diperbolehkan pergerakan diagonal. Perhitungan nilai heuristic untuk node ke- n menggunakan Manhattan Distance adalah sebagai berikut :

$$h(n) = D * (abs(n.x-goal.x) + abs(n.y-goal.y)) \quad (2)$$

Dimana $h(n)$ adalah nilai heuristic untuk node n , D adalah nilai/cost perpindahan dari satu node ke node lain (secara vertikal/horizontal), dan goal adalah node tujuan. D dapat berupa bilangan real ataupun bilangan bulat, positif maupun negatif, selama tidak sama dengan nol. Sebab jika D bernilai nol, maka $h(n)$ juga bernilai nol (Fernando et al., 2020)

3 Hasil dan Diskusi

Pada bagian ini menjelaskan hasil dari penelitian yang berupa hasil dari Grafik RQT yang merupakan hasil visualisasi dari komunikasi antar node dan topik dalam sistem ROS menggunakan plugin `rqt_graph`. Gambar ini menggambarkan struktur interaksi selama proses simulasi robot Turtlebot yang berjalan di lingkungan virtual Gazebo. Terdapat beberapa node penting yang ditampilkan dalam grafik ini. Node `/gazebo` dan `/gazebo_gui` berasal dari simulator Gazebo, di mana `/gazebo` berfungsi menjalankan simulasi fisika, sedangkan `/gazebo_gui` menyediakan antarmuka grafis untuk interaksi pengguna dengan simulasi.



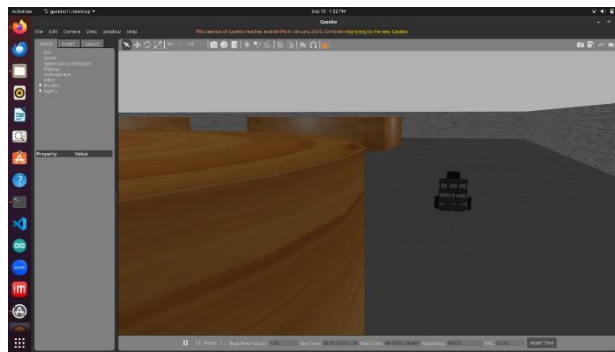
Gambar 2. Grafik RQT pada node ROS

Node ini menerima data posisi dari topik `/odom`, yang dikirim oleh node `/gazebo`, dan menggunakan informasi tersebut untuk menghitung serta mengirimkan perintah gerakan melalui topik `/cmd_vel`. Topik `/odom` menyediakan informasi terkait posisi, orientasi, serta kecepatan robot secara kontinu. Informasi ini digunakan oleh sistem pengendali untuk menyesuaikan pergerakan robot dengan lintasan yang telah ditentukan. Selanjutnya, perintah kecepatan dikirim melalui topik `/cmd_vel`, yang digunakan oleh pengendali robot dalam simulasi untuk menggerakkan robot.

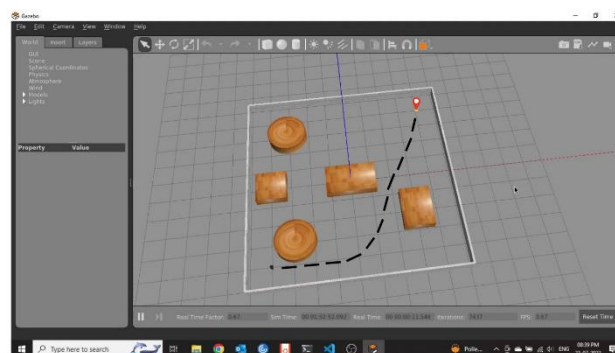
Struktur komunikasi dalam sistem ini menunjukkan bahwa komponen navigasi, kontrol, dan simulasi telah terhubung dengan baik. Node pengikut lintasan dapat memproses data odometri secara real time dan mengeluarkan perintah gerakan yang sesuai. Algoritma A* digunakan untuk merancang jalur optimal dari posisi awal menuju titik tujuan dengan mempertimbangkan lingkungan sekitar. Jalur hasil perencanaan kemudian dikonversi menjadi serangkaian koordinat (x, y, θ) yang disesuaikan dengan sistem koordinat simulasi, dan digunakan oleh Turtlebot3 untuk mengikuti lintasan tersebut. Robot bergerak secara berurutan menuju setiap titik pada jalur dengan metode kontrol sederhana, di mana kecepatan linear dijaga tetap konstan, sedangkan kecepatan sudut dikendalikan secara proporsional terhadap perbedaan sudut antara orientasi saat ini dan arah menuju titik yang dituju. Ketika jarak robot terhadap titik yang sedang dituju kurang dari 0,1 meter, robot secara otomatis berpindah ke titik berikutnya dalam lintasan.

3.1. Simulasi pada Gazebo

Gambar tersebut memperlihatkan sebuah simulasi robotik yang dijalankan menggunakan **Gazebo** pada sistem operasi **Ubuntu**. Gazebo merupakan simulator yang umum dipakai bersama **ROS (Robot Operating System)** untuk menguji berbagai algoritma navigasi dan kontrol secara virtual sebelum diterapkan di lingkungan nyata. Dalam simulasi ini, robot tampak berada dalam area tertutup yang dilengkapi dengan beberapa balok kayu, yang kemungkinan berfungsi sebagai rintangan atau titik acuan. Sumbu koordinat (X, Y, Z) divisualisasikan untuk menunjukkan arah dan orientasi robot dalam lingkungan simulasi.



Gambar 4. Simulasi teleoperasi di GAZEBO



Gambar 3. Pola pergerakan robot metode A*

Tampilan terminal yang berada di depan jendela Gazebo menunjukkan output dari sebuah node ROS yang mencetak data posisi robot secara langsung. Informasi yang ditampilkan meliputi waktu simulasi, posisi robot saat ini dalam koordinat X dan Y, serta jarak antara robot dengan targetnya. Salah satu baris output juga mencantumkan posisi target dan nilai sudut orientasi (θ). Sebagai contoh, θ sebesar 90 derajat mengindikasikan bahwa arah hadap robot seharusnya mengarah ke sumbu Y positif atau ke utara. Berdasarkan data yang ditampilkan, terlihat bahwa posisi X robot tetap berada di sekitar angka 4.5, sementara posisi Y terus meningkat dari -3.41 ke -3.06. Ini menunjukkan bahwa robot sedang bergerak ke arah Vertikal menuju targetnya. Jarak ke target pun semakin kecil, yang menunjukkan bahwa robot

bergerak dengan tepat sesuai perhitungan. Hal ini mengindikasikan penggunaan algoritma kontrol posisi, seperti PID atau metode navigasi berbasis koordinat dan jarak.

4 Kesimpulan

Berdasarkan hasil simulasi yang dijalankan pada Gazebo serta analisis data dari output ROS, dapat disimpulkan bahwa sistem navigasi robot menunjukkan kinerja yang baik dalam melacak posisi dan mengarahkan pergerakan menuju target secara akurat, meskipun berada dalam lingkungan virtual yang mengandung berbagai hambatan. Perubahan koordinat, terutama pada sumbu Y, mengindikasikan bahwa metode navigasi berbasis koordinat dan orientasi bekerja secara efektif dalam mencapai sasaran yang ditetapkan. Selain itu, penerapan algoritma A* dalam sistem navigasi terbukti berperan penting dalam proses perencanaan jalur. Algoritma ini memungkinkan robot untuk memetakan lingkungan dengan terlebih dahulu mengidentifikasi dan menginisialisasi lokasi hambatan, seperti dinding atau rintangan pada setiap sel dalam arena simulasi. Setelah proses inisialisasi, robot merancang lintasan terpendek menuju titik tujuan dengan memberikan indeks berurutan pada sel-sel yang akan dilalui. Pendekatan ini tidak hanya meningkatkan efisiensi pergerakan robot, tetapi juga memastikan jalur yang diambil merupakan rute optimal berdasarkan kondisi lingkungan yang ada. Dengan demikian, kombinasi antara algoritma A* dan simulasi berbasis ROS-Gazebo dapat dijadikan sebagai kerangka pengujian yang andal sebelum sistem diterapkan pada robot nyata.

5 Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada kedua orang tua yang selalu mendoakan dan mendukung selama ini. Juga kepada Dosen pembimbing yang telah memberikan arahan serta dukungan yang tak ternilai sehingga penelitian ini dapat terselesaikan. Terima kasih juga kepada semua teman-teman atas doa dan dukungan yang telah memberikan semangat dan inspirasi selama perjalanan penelitian ini.

Daftar Pustaka

- Fernando, Y., Mustaqov, M. A., & Megawaty, D. A. (2020). PENERAPAN ALGORITMA A-STAR PADA APLIKASI PENCARIAN LOKASI FOTOGRAFI DI BANDAR LAMPUNG BERBASIS ANDROID. *Jurnal Teknoinfo*, 14(1), 27. <https://doi.org/10.33365/jti.v14i1.509>
- Fikri, A. A., & Anifah, L. (2021). Mapping And Localization System Pada Mobile Robot Menggunakan Metode SLAM Berbasis LiDAR. *Journal of Information Engineering and Educational Technology*, 5(1), 27–33. <https://doi.org/10.26740/jieet.v5n1.p27-33>
- Hermawan, H., & Setiyani, H. (2019). IMPLEMENTASI ALGORITMA A-STAR PADA PERMAINAN KOMPUTER ROGUELIKE BERBASIS UNITY. *Jurnal Algoritma, Logika Dan Komputasi*, 2(1). <https://doi.org/10.30813/j-alu.v2i1.1571>
- Lee, D., Jung, M., Yang, W., & Kim, A. (2024). LiDAR odometry survey: recent advancements and remaining challenges. In *Intelligent Service Robotics* (Vol. 17, Issue 2, pp. 95–118). Springer Science and Business Media Deutschland GmbH. <https://doi.org/10.1007/s11370-024-00515-8>
- RIYAN FAHMI SYIHABUDDIN. (n.d.). *IMPLEMENTASI ALGORITMA A-STAR DALAM MENENTUKAN RUTE TERPENDEK DESTINASI WISATA KOTA MALANG SKRIPSI OLEH RIYAN FAHMI SYIHABUDDIN NIM.17610066 PROGRAM STUDI METEMATIKA FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG 2021*.
- Rosadi, A., Manurung, P., Yulianti, Y., Marjunus, R., Junaidi, dan, Fisika FMIPA Universitas Lampung Jl Soemantri Brodjonegoro, J., & Lampung, B. (2024). Desain dan Realisasi Sistem Kontrol Kecepatan dan Ketinggian Motor Menggunakan Sensor Optocoupler dan Sensor Ultrasonik Berbasis Arduino untuk Aplikasi Pengaduk Otomatis. In *Jurnal Teori dan Aplikasi Fisika* (Vol. 12, Issue 01).
- Susanthi, Y. (n.d.). *Mapping dan Navigasi untuk Robot Pengantar Makanan di Restoran Berbasis ROS*.