

SISTEM NAVIGASI OTONOM ROBOT MOBIL BERBASIS ROS PADA ROBOT PENGGERAK DIFERENSIAL

Irma Nirmala¹, Rahmi Hidayati²

^{1,2}Rekayasa Sistem Komputer, FMIPA, Universitas Tanjungpura Pontianak

Email: irma.nirmala@siskom.untan.ac.id

ABSTRACT

This study developed an autonomous navigation simulation for the TurtleBot3 using the Robot Operating System (ROS) and Gazebo. Two scenarios were tested: environments with and without obstacles. The SLAM algorithm was employed for environmental mapping, while AMCL was used for dynamic robot localization. The results demonstrated that TurtleBot3 was capable of navigating efficiently, detecting obstacles, and planning paths effectively. However, some failures occurred in narrow areas, causing the robot to get stuck. To address this issue, adjustments were made to speed parameters and local path planning, which significantly improved the robot's performance. This simulation highlights the potential for mobile robot technology in various sectors, such as industry and services, offering higher efficiency and lower risks compared to physical testing.

Keywords: ROS, Gazebo, Turtlebot3, Navigation, Differential Drive Robot

ABSTRAK

Penelitian ini mengembangkan simulasi navigasi otonom pada TurtleBot3 menggunakan Robot Operating System (ROS) dan Gazebo. Dua skenario diuji: lingkungan tanpa hambatan dan dengan hambatan. Algoritma SLAM digunakan untuk pemetaan lingkungan, sedangkan AMCL digunakan untuk melokalisasi posisi robot secara dinamis. Hasilnya menunjukkan bahwa TurtleBot3 dapat bernavigasi dengan baik, mendeteksi hambatan, serta merencanakan jalur secara efisien. Namun, beberapa kegagalan terjadi di area sempit, yang mengakibatkan robot terjebak. Untuk mengatasi masalah ini, dilakukan penyesuaian pada parameter kecepatan dan perencanaan jalur lokal, yang terbukti meningkatkan kinerja robot. Simulasi ini menunjukkan potensi besar penggunaan teknologi robot mobile dalam berbagai sektor, seperti industri dan layanan, dengan efisiensi yang lebih tinggi dan risiko yang lebih rendah dibandingkan pengujian fisik.

Kata Kunci: ROS, Gazebo, Turtlebot3, Navigasi, Robot Penggerak Diferensial

Riwayat Artikel :

Tanggal diterima : 08-09-2024

Tanggal revisi : 18-09-2024

Tanggal terbit : 26-09-2024

DOI :

<https://doi.org/10.31949/infotech.v10i2.11219>

INFOTECH journal by Informatika UNMA is licensed under CC BY-SA 4.0

Copyright © 2024 By Author



1. PENDAHULUAN

1.1. Latar Belakang

Kemajuan pesat dalam teknologi robotika layanan telah membawa perubahan signifikan di berbagai sektor kehidupan dan industri. Dalam dunia medis, misalnya, robot telah berperan penting dalam mempercepat pemulihan pasien melalui penggunaan robot bedah yang presisi (Aprianti, 2023). Begitu juga di sektor industri, di mana robot berperan dalam meningkatkan produktivitas dan efisiensi operasional melalui otomatisasi tugas-tugas berulang dan berbahaya, yang sebelumnya dilakukan oleh pekerja manusia (Supriandi, 2023). Selain di industri dan medis, penggunaan robot juga mulai berkembang di sektor hiburan, seperti robot humanoid yang dapat berinteraksi dengan pengunjung atau melakukan pertunjukan dalam taman hiburan (Aprianti, 2023).

Peningkatan kebutuhan akan robot yang dapat bergerak secara otonom di berbagai lingkungan semakin mendorong pengembangan teknologi robot *mobile*. Banyak robot layanan, khususnya yang dirancang untuk keperluan industri, medis, pendidikan, hingga hiburan, mengandalkan mobilitas untuk menjalankan tugasnya secara efisien. Robot *mobile* dengan kemampuan navigasi otonom menjadi solusi bagi berbagai permasalahan seperti navigasi di area dengan rintangan kompleks serta lingkungan yang dinamis (Kurnia, 2023). Navigasi otonom ini sangat krusial, karena robot perlu bergerak secara mandiri tanpa campur tangan manusia, yang berarti membutuhkan sistem navigasi yang canggih untuk memetakan lingkungan dan merencanakan jalur dengan akurat.

Simulasi menjadi bagian penting dalam pengujian robot *mobile*. Teknologi simulasi memungkinkan pengembang untuk merencanakan dan menguji berbagai aspek dari pergerakan robot secara virtual sebelum diterapkan secara fisik. Melalui simulasi, risiko kerusakan baik pada robot maupun lingkungan pengujian dapat diminimalkan. Pengujian melalui simulasi juga membantu dalam percepatan proses pengembangan dan evaluasi, tanpa harus bergantung pada perangkat keras yang mahal dan memakan waktu (Wijaya, 2024). Penggunaan simulasi terbukti lebih efisien dalam hal waktu dan biaya, sekaligus memberikan fleksibilitas dalam pengujian berbagai skenario yang mungkin sulit dilakukan di dunia nyata.

Dalam perkembangan teknologi robotika, khususnya robot *mobile*, ada beberapa masalah yang perlu diselesaikan. Pertama, bagaimana cara memastikan robot *mobile* dapat bergerak secara otonom di berbagai lingkungan tanpa campur tangan manual? Ini menjadi tantangan besar karena setiap lingkungan memiliki karakteristik unik yang harus diidentifikasi oleh robot secara *real-time*. Kedua, simulasi sebagai alat pengujian efektif menjadi pertanyaan penting, yaitu bagaimana simulasi dapat digunakan sebagai alternatif yang aman dan efisien untuk menguji navigasi dan pergerakan robot *mobile*

tanpa risiko pada manusia atau lingkungan? Simulasi dapat membantu mengatasi risiko tersebut, tetapi bagaimana mengoptimalkan proses ini agar lebih dekat dengan kondisi dunia nyata?

Masalah berikutnya adalah keterbatasan pengujian fisik, di mana perangkat keras yang mahal dan waktu yang dibutuhkan untuk persiapan pengujian menjadi kendala utama. Bagaimana cara mengatasi hal ini melalui simulasi? Pengujian fisik cenderung memerlukan investasi yang besar dalam hal waktu dan biaya, yang dapat dipangkas secara signifikan melalui pengujian simulasi yang efisien. Selain itu, parameter navigasi robot yang sering kali tidak optimal menjadi hambatan dalam pergerakan robot *mobile* di lingkungan yang penuh rintangan. Bagaimana cara mengoptimalkan parameter tersebut agar robot dapat mencapai tujuan dengan akurat dan efisien?

Penelitian ini bertujuan untuk mengembangkan model simulasi robot *mobile* menggunakan penggerak diferensial dalam lingkungan simulasi berbasis Robot Operating System (ROS) dan Gazebo. Tujuan utama adalah menciptakan model yang mampu bergerak secara otonom, serta menavigasi lingkungan dengan berbagai rintangan. Penelitian ini juga bertujuan untuk mengonfigurasi sistem navigasi otonom menggunakan ROS navigation stack, yang meliputi pemetaan, lokalisasi, dan perencanaan jalur. Dengan konfigurasi ini, diharapkan robot mampu beradaptasi dengan lingkungan yang berubah-ubah secara *real-time*.

Selain itu, penelitian ini akan melakukan eksperimen untuk mengevaluasi performa robot dalam dua skenario: lingkungan dengan dan tanpa hambatan. Penelitian juga akan fokus pada optimalisasi parameter sistem seperti *base_local_planner* dan *local_costmap* untuk meningkatkan akurasi navigasi robot. Simulasi ini memberikan pandangan lebih dalam mengenai bagaimana parameter-parameter tersebut berinteraksi dan mempengaruhi kinerja robot secara keseluruhan, serta bagaimana cara mencapai keseimbangan optimal antara kecepatan, akurasi, dan efisiensi energi.

Dengan menggunakan simulasi, penelitian ini bertujuan untuk menjawab beberapa pertanyaan utama dalam pengembangan robot *mobile* otonom, seperti bagaimana cara meningkatkan efisiensi pengujian robot dan mengoptimalkan navigasi otonom untuk berbagai skenario. Hasil dari penelitian ini diharapkan dapat menjadi langkah maju dalam penerapan robot *mobile* di berbagai sektor kehidupan, sekaligus meminimalkan risiko dan biaya yang terkait dengan pengembangan dan pengujian robot *mobile* secara fisik.

1.2. Tinjauan Pustaka

Ada berbagai pendekatan dan teknologi yang digunakan untuk menjawab permasalahan terkait pergerakan otonom robot *mobile*, simulasi sebagai alat pengujian, keterbatasan pengujian fisik, dan

optimasi parameter navigasi dalam sistem robot *mobile*.

Pergerakan otonom robot *mobile* memerlukan integrasi yang kuat antara teknologi sensor, algoritma navigasi, dan teknik fusi data yang efektif. Sebagai contoh, penggunaan berbagai sensor seperti ultrasonik, LiDAR, kamera, dan IMU memungkinkan robot untuk mengumpulkan data lingkungan yang lebih komprehensif (Zhao et al., 2021). Penggabungan sensor ini sangat penting untuk menghindari rintangan dan memastikan navigasi yang aman (Liu, 2024). Fusi data dari berbagai sensor memungkinkan penciptaan peta lingkungan yang akurat dan perencanaan jalur tanpa tabrakan, sehingga meningkatkan robustitas sistem navigasi (Li et al., 2022; Singh & Nagla, 2019).

Simulasi telah menjadi alat yang penting dalam pengembangan robot *mobile*, terutama untuk pengujian navigasi. Simulasi menciptakan lingkungan virtual yang menyerupai skenario dunia nyata, memungkinkan pengujian tanpa risiko terhadap manusia atau lingkungan. Penggunaan ROS dan Gazebo sebagai alat simulasi membantu mengurangi risiko pengujian dan mempercepat iterasi pengembangan algoritma. Simulasi ini juga memungkinkan uji coba algoritma navigasi dalam skenario yang kompleks dan sulit direplikasi di dunia nyata, seperti yang dicontohkan dalam penelitian yang menggunakan simulasi untuk pengujian algoritma berbasis pembelajaran mendalam (Chen et al., 2021).

Keterbatasan pengujian fisik, seperti biaya perangkat keras yang mahal dan waktu persiapan yang lama, dapat diatasi dengan teknologi simulasi yang lebih efisien. Simulasi tidak hanya mengurangi kebutuhan perangkat keras tetapi juga memungkinkan pengujian dalam berbagai skenario yang sulit dicapai melalui pengujian fisik (Xu, 2019). Hal ini dapat mempercepat proses pengembangan dan memungkinkan pengujian lebih aman pada sistem robot yang kompleks, seperti yang dibahas dalam penelitian tentang simulasi berbasis sensor LiDAR (Iqbal et al., 2020).

Optimalisasi parameter navigasi dalam robot *mobile* juga menjadi tantangan tersendiri, terutama dalam lingkungan yang penuh rintangan. Algoritma seperti Simultaneous Localization and Mapping (SLAM) digunakan untuk memperbarui posisi robot secara berkelanjutan dan memungkinkan robot untuk menghindari rintangan secara efisien (Fu, 2023). Perencanaan jalur yang efektif, yang memungkinkan robot menghasilkan jalur optimal dan adaptif, juga memegang peranan penting dalam memastikan navigasi yang akurat di berbagai kondisi lingkungan (Myint, 2018).

Dalam kesimpulan, tantangan pergerakan otonom robot *mobile*, keterbatasan pengujian fisik, dan parameter navigasi dapat diatasi melalui integrasi teknologi sensor yang canggih, fusi data yang efektif, dan simulasi sebagai alat pengujian yang aman dan efisien. Dengan memanfaatkan

pendekatan ini, robot *mobile* dapat mencapai pergerakan otonom yang andal dan efisien di berbagai lingkungan.

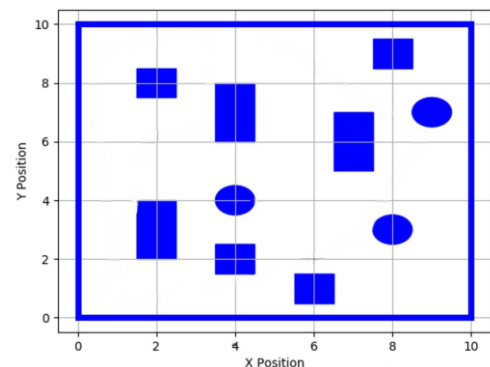
1.3. Metodologi Penelitian

Penelitian ini menggunakan metode simulasi berbasis Robot Operating System (ROS) dengan platform simulasi Gazebo untuk menguji kemampuan navigasi otonom TurtleBot3 dalam skenario lingkungan yang bervariasi. Fokus utama dari penelitian ini adalah pada pemetaan, lokalisasi, dan navigasi otonom yang menggunakan sensor LiDAR, SLAM (Simultaneous Localization and Mapping), dan AMCL (Adaptive Monte Carlo Localization). Tahapan penelitian ini meliputi persiapan lingkungan simulasi, pelaksanaan simulasi pemetaan dan lokalisasi, serta uji eksperimen navigasi otonom.

a. Persiapan lingkungan simulasi

Pada tahap awal, dilakukan konfigurasi lingkungan simulasi di Gazebo. Gazebo dipilih karena merupakan simulator *open-source* yang kompatibel dengan ROS dan mendukung simulasi fisika realistis serta rendering 3D yang memungkinkan peneliti untuk menguji algoritma dalam lingkungan yang mendekati dunia nyata tanpa risiko merusak perangkat keras.

Lingkungan simulasi dibuat dengan memanfaatkan fitur-fitur yang tersedia dalam ROS dan Gazebo. Peta dunia virtual disiapkan untuk merepresentasikan skenario nyata, yang melibatkan berbagai jenis hambatan seperti dinding, kotak, dan benda statis lainnya. Peta ini bertujuan untuk menguji kemampuan robot TurtleBot3 dalam mengenali, memetakan, dan menavigasi di sekitar hambatan tersebut. TurtleBot3, yang dilengkapi dengan sensor LiDAR, digunakan sebagai platform robotik dalam simulasi ini. Pada lingkungan dengan rintangan, rintangan akan diatur sedemikian rupa pada posisi-posisi yang tersebar di dalam ruangan seperti yang diperlihatkan oleh Gambar 1. Rintangan ini memiliki bentuk yang bervariasi.



Gambar 1. Posisi Rintangan dalam Lingkungan

TurtleBot3 dipilih karena merupakan platform robotik yang kompatibel dengan ROS dan dilengkapi dengan perangkat keras yang mendukung simulasi pergerakan otonom, seperti sensor LIDAR dan IMU (Inertial Measurement Unit). Lingkungan simulasi yang dirancang di Gazebo mengandung berbagai hambatan yang ditujukan untuk menguji kemampuan TurtleBot3 dalam mendeteksi dan menghindari hambatan selama navigasi.

b. Simulasi Pemetaan dan Lokalisasi

Setelah lingkungan simulasi dikonfigurasi, tahap selanjutnya adalah pelaksanaan pemetaan dan lokalisasi menggunakan algoritma SLAM dan AMCL.

Pemetaan dilakukan dengan menjalankan node `slam_gmapping` di ROS, yang memungkinkan TurtleBot3 untuk memetakan lingkungan secara simultan menggunakan data dari sensor LIDAR. Dalam proses ini, TurtleBot3 bergerak secara otomatis di dalam lingkungan simulasi, baik melalui pergerakan manual yang dikendalikan oleh peneliti atau menggunakan navigasi otonom. Data yang diperoleh dari sensor LIDAR dikombinasikan secara real-time untuk membentuk peta lingkungan. Node `slam_gmapping` menggunakan teknik probabilistik untuk memastikan akurasi peta meskipun terdapat ketidakpastian pada data sensor.

Setelah peta lingkungan terbentuk, proses selanjutnya adalah lokalisasi menggunakan Adaptive Monte Carlo Localization (AMCL). AMCL adalah algoritma berbasis partikel yang digunakan untuk melacak posisi TurtleBot3 dalam peta yang telah terbentuk. AMCL bekerja dengan memproses data sensor LIDAR untuk memperbarui posisi robot secara dinamis saat TurtleBot3 bergerak di dalam lingkungan. Pada tahap ini, penting untuk memastikan bahwa AMCL menerima data sensor yang valid dan dapat memberikan estimasi posisi yang akurat.

c. Uji Eksperimen Navigasi Otonom

Uji eksperimen dilakukan untuk mengevaluasi kemampuan TurtleBot3 dalam menavigasi lingkungan dengan dan tanpa hambatan.

Pada eksperimen pertama, lingkungan simulasi dirancang tanpa hambatan. Node `move_base` digunakan untuk merencanakan jalur TurtleBot3 dari titik awal ke titik tujuan yang telah ditentukan. Algoritma perencanaan jalur bekerja dengan memanfaatkan peta yang telah dibuat sebelumnya melalui SLAM. Dalam eksperimen ini, keberhasilan TurtleBot3 diukur berdasarkan kemampuan robot untuk mencapai tujuan tanpa deviasi signifikan dari jalur yang direncanakan. Hal ini penting untuk mengukur efektivitas algoritma perencanaan global (`global_planner`) yang digunakan oleh `move_base`.

Pada eksperimen kedua, hambatan ditambahkan ke dalam lingkungan simulasi.

Hambatan tersebut berfungsi sebagai rintangan yang harus dihindari oleh TurtleBot3 selama navigasi menuju tujuan. Dalam eksperimen ini, kemampuan TurtleBot3 diuji menggunakan kombinasi perencanaan jalur global dan lokal. `Global_planner` merencanakan rute keseluruhan, sementara `local_planner` menyesuaikan jalur saat TurtleBot3 mendekati hambatan, sehingga robot dapat menghindarinya secara dinamis. Kinerja sistem dievaluasi berdasarkan seberapa efektif TurtleBot3 dapat menghindari hambatan dan tetap mencapai tujuan.

Data yang dikumpulkan selama simulasi mencakup waktu tempuh, akurasi posisi, jumlah deviasi dari jalur yang direncanakan, serta jumlah tabrakan atau hampir tabrakan dengan hambatan. Analisis dilakukan untuk mengevaluasi efisiensi algoritma yang digunakan, baik dalam skenario tanpa hambatan maupun dengan hambatan. Hasil ini akan dianalisis secara kuantitatif untuk menentukan efektivitas sistem navigasi yang dikembangkan.

Dengan metodologi ini, penelitian ini bertujuan untuk mengevaluasi kemampuan TurtleBot3 dalam menavigasi lingkungan yang kompleks dan memvalidasi kehandalan sistem pemetaan, lokalisasi, serta perencanaan jalur yang digunakan.



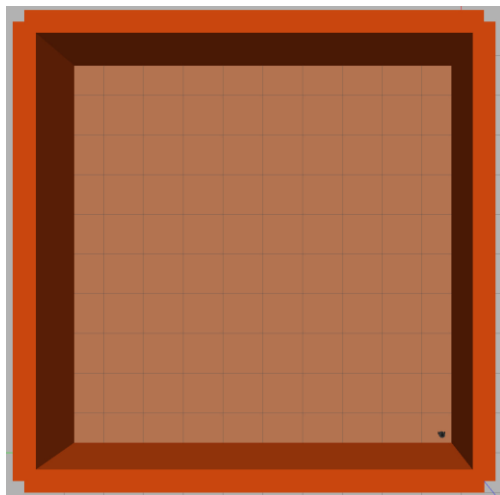
Gambar 2. Metodologi Penelitian

2. PEMBAHASAN

Tahapan dimulai dengan persiapan lingkungan simulasi dan pemahaman spesifikasi TurtleBot3 yang akan digunakan. Langkah awal ini sangat penting untuk memastikan bahwa simulasi berjalan lancar dan menghasilkan data yang valid. Simulasi dilakukan dalam dua jenis ruangan: ruangan kosong dan ruangan berhalangan, yang masing-masing berukuran 10 meter x 10 meter. Ruangan kosong memberikan kondisi sederhana untuk menguji kemampuan dasar navigasi TurtleBot3, sementara ruangan berhalangan, dengan rintangan berupa kubus, balok, dan silinder, menambah kompleksitas simulasi untuk menguji adaptabilitas robot terhadap tantangan yang lebih menantang. Lingkungan simulasi berupa ruang kosong diperlihatkan oleh Gambar 3.

Penelitian ini bertujuan untuk membandingkan performa TurtleBot3 di dua lingkungan berbeda. Ruangan kosong menyediakan data dasar tentang gerakan robot, sedangkan ruangan berhalangan memberikan wawasan tentang bagaimana robot menghadapi lingkungan yang lebih sulit. Ini penting dalam menilai kemampuan navigasi TurtleBot3, terutama dalam menghadapi rintangan dan

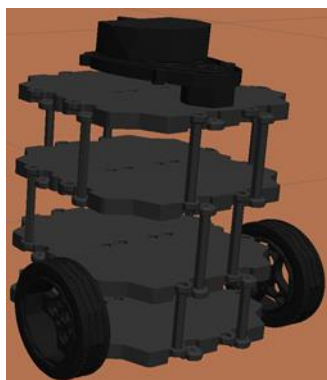
memetakan strategi untuk bergerak dengan efisien di lingkungan yang lebih kompleks.



Gambar 3. Lingkungan Simulasi Ruang Kosong

TurtleBot3 Burger, yang digunakan dalam penelitian ini, adalah robot mobile kecil dengan platform dasar sistem operasi robotik (ROS). Dengan diameter 138 mm, tinggi 192 mm, dan berat sekitar 0,9 kg, robot ini memiliki ukuran yang kompak, memungkinkannya bergerak lincah dalam ruang terbatas. Sistem penggerak diferensial pada dua roda utama memungkinkan TurtleBot3 berbelok dan bergerak dengan fleksibilitas tinggi, menyesuaikan kecepatan tiap roda secara independen.

Selain itu, TurtleBot3 dilengkapi dengan sensor-sensor penting untuk mendukung fungsionalitasnya. Sensor Lidar berperan krusial dalam mendeteksi jarak dan memetakan lingkungan dalam dua dimensi. Lidar terletak di bagian atas robot dan memungkinkan TurtleBot3 mendeteksi objek di sekitarnya, yang sangat penting dalam simulasi ruangan berhalangan. Sensor IMU (Inertial Measurement Unit) membantu dalam mempertahankan stabilitas gerakan dan orientasi, terutama saat robot bergerak melalui jalur yang berubah atau tidak simetris. Tampilan fisik dari Turtlebot3 Burger ditunjukkan Gambar 4.



Gambar 4. Turtlebot3 Burger di Gazebo

TurtleBot3 juga memiliki sensor encoder pada roda penggeraknya untuk mengukur jarak yang ditempuh

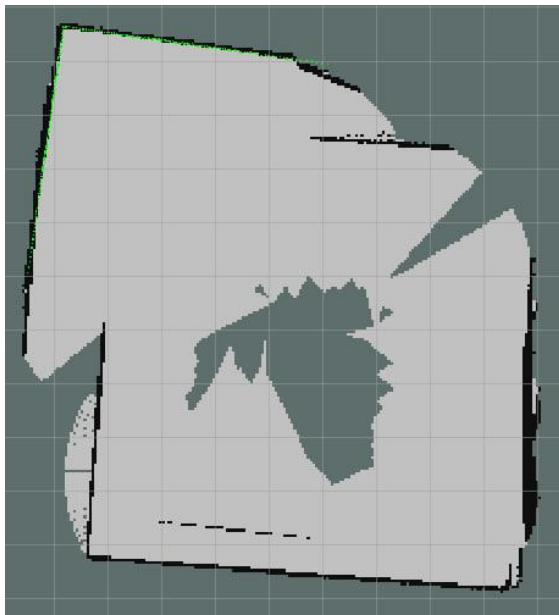
dan kecepatan gerakan, yang digunakan untuk mengevaluasi performa selama simulasi. Data ini memberikan informasi tentang efisiensi dan akurasi navigasi robot. Meskipun sensor bumper, yang mendeteksi benturan, tidak digunakan dalam penelitian ini, sensor tersebut dapat menambah lapisan keamanan pada navigasi, terutama di lingkungan yang lebih dinamis.

Tahapan berikutnya Simulasi pemetaan dan lokalisasi merupakan salah satu elemen penting dalam penelitian robotika, khususnya dalam konteks pengembangan sistem navigasi otonom. Pada bagian ini, dilakukan pemetaan ruangan serta estimasi posisi robot dalam lingkungan menggunakan teknologi SLAM (*Simultaneous Localization and Mapping*). Proses ini melibatkan beberapa langkah penting, mulai dari pengumpulan data sensor hingga pemrosesan informasi untuk menghasilkan peta yang akurat dan memungkinkan robot memahami posisinya di dalam ruangan.

Proses pemetaan ruangan dilakukan dengan menggerakkan TurtleBot3 secara bebas di dalam ruangan. Pada simulasi ini, TurtleBot3 menggunakan sistem SLAM untuk memetakan lingkungannya sambil memperkirakan posisinya secara simultan. SLAM bekerja dengan memanfaatkan data sensor yang dikumpulkan selama pergerakan robot, terutama dari odometri dan pemindai laser (Lidar). Data ini kemudian diproses untuk membentuk representasi dua dimensi dari ruangan tempat robot berada.

Dalam ruangan kosong, di mana tidak terdapat banyak objek selain dinding, sistem SLAM menghadapi tantangan tersendiri. Peta yang dihasilkan oleh TurtleBot3 pada kondisi ini cenderung tidak lengkap. Hal ini disebabkan oleh keterbatasan jumlah fitur lingkungan yang dapat dikenali oleh sensor. SLAM bergantung pada fitur-fitur seperti sudut, dinding, dan objek statis lainnya sebagai titik referensi untuk melakukan pemetaan. Di ruangan kosong, fitur yang bisa dipetakan terbatas hanya pada dinding ruangan, tanpa ada variasi lain yang dapat diidentifikasi oleh sensor. Gambar 5 menunjukkan hasil pemetaan yang dilakukan di ruangan kosong ini, di mana terlihat bahwa peta yang dihasilkan tidak sepenuhnya sempurna.

Sebaliknya, ketika TurtleBot3 bergerak di ruangan yang berhalangan, peta yang dihasilkan oleh SLAM lebih lengkap dan akurat. Hal ini disebabkan oleh adanya rintangan seperti kubus, balok, dan silinder yang memberikan lebih banyak titik referensi bagi sistem SLAM untuk mengidentifikasi posisi robot dan lingkungan sekitar. Namun, terdapat kesulitan pada area rintangan yang sangat dekat dengan dinding, di mana sensor Lidar terkadang tidak dapat mendeteksi objek secara akurat karena gangguan dari pantulan sinyal. Secara keseluruhan, proses pemetaan di ruangan berhalangan memberikan hasil yang lebih baik dibandingkan dengan ruangan kosong.



Gambar 5. Hasil Pemetaan Ruang Tanpa Hambatan

Pemetaan dilakukan menggunakan teknologi SLAM yang memanfaatkan data dari odometri dan pemindai laser. Sistem SLAM dalam penelitian ini dijalankan melalui platform ROS (Robot Operating System) yang memfasilitasi komunikasi antara berbagai node dalam sistem. Data sensor yang dikumpulkan oleh TurtleBot3 diproses oleh algoritma SLAM untuk menghasilkan peta yang kemudian divisualisasikan menggunakan interface RViz. Hasil pemetaan tersebut dapat disimpan untuk digunakan dalam proses navigasi selanjutnya.

Proses pemetaan memerlukan penyesuaian parameter seperti kecepatan robot dan frekuensi pembaruan data sensor untuk memastikan hasil yang optimal. Kecepatan TurtleBot3 diatur secara bervariasi selama pemetaan, tergantung pada posisi relatifnya terhadap rintangan di sekitarnya. Hal ini bertujuan untuk meminimalkan kesalahan dalam pengukuran dan memastikan bahwa robot dapat menghindari tabrakan dengan objek di sekitarnya sambil tetap mengumpulkan data yang akurat.

Setelah peta lingkungan berhasil dibuat, langkah selanjutnya adalah melakukan lokalisasi. Lokalisasi adalah proses di mana robot memperkirakan posisinya dalam peta yang telah dibuat. Dalam penelitian ini, TurtleBot3 menggunakan paket AMCL (Adaptive Monte Carlo Localization) yang berfungsi untuk memperkirakan posisi robot berdasarkan data sensor dan peta yang sudah tersedia. AMCL menggunakan metode probabilistik untuk menentukan posisi robot dengan memanfaatkan informasi dari pemindai laser dan data odometri.

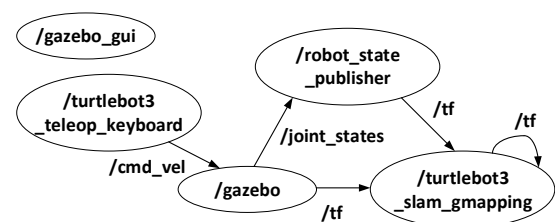
Selama proses lokalisasi, TurtleBot3 terus memperbarui perkiraan posisinya seiring dengan pergerakannya di dalam ruangan. AMCL menghitung kemungkinan posisi robot dengan membandingkan data sensor saat ini dengan peta

yang telah dibuat sebelumnya. Dengan cara ini, robot dapat mengetahui posisinya dengan cukup akurat meskipun terdapat variasi lingkungan atau perubahan posisi yang tidak terduga. Lokalisasi yang tepat sangat penting dalam konteks navigasi, karena memungkinkan robot untuk mengambil keputusan yang benar dalam menentukan jalur pergerakannya.

Hasil pemetaan dan lokalisasi dipengaruhi oleh berbagai parameter yang diatur selama simulasi. Parameter seperti kecepatan rotasi robot, kecepatan maksimum, dan batas akselerasi pada sumbu x, y, dan z disesuaikan dengan kebutuhan platform robot untuk mendapatkan hasil yang optimal. Kecepatan pembaruan data dari sensor Lidar dan odometri juga merupakan faktor penting yang mempengaruhi akurasi peta dan estimasi posisi robot.

Pengaturan parameter planner global dan lokal dalam sistem navigasi TurtleBot3 juga berperan penting dalam meningkatkan performa robot selama simulasi. Planner global bertanggung jawab untuk menentukan jalur optimal yang harus diambil robot berdasarkan peta lingkungan, sementara planner lokal memastikan bahwa robot dapat mengikuti jalur tersebut sambil menghindari rintangan yang mungkin muncul secara tiba-tiba. Penyesuaian parameter ini dilakukan secara iteratif selama simulasi untuk memastikan bahwa TurtleBot3 dapat bergerak dengan lancar dan efisien di dalam ruangan.

Setelah proses pemetaan dan lokalisasi selesai, dilakukan validasi untuk memastikan bahwa hasil yang diperoleh akurat dan dapat diandalkan. Validasi dilakukan dengan memeriksa topik dan node yang aktif dalam sistem ROS menggunakan alat bantu seperti rqt_graph. Diagram graf yang dihasilkan oleh rqt_graph memungkinkan peneliti untuk memverifikasi apakah semua node dalam sistem berfungsi dengan benar dan apakah ada link komunikasi antar-node yang hilang.



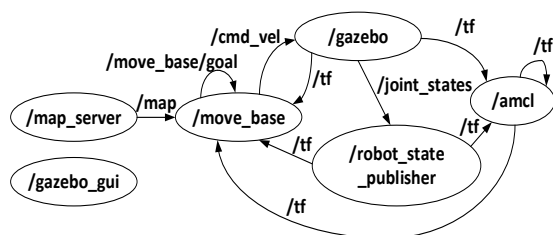
Gambar 6. Diagram Graf Node yang Aktif pada Lokalisasi dan Pemetaan

Gambar 6 adalah gambar yang diadaptasi dari gambar yang dihasilkan oleh rqt_graph menunjukkan aliran data dalam sistem simulasi, termasuk komunikasi antara node seperti /turtlebot3_teleop_keyboard, /cmd_vel, dan /gazebo. Node /turtlebot3_teleop_keyboard bertanggung jawab untuk mengirimkan perintah kecepatan ke robot melalui topik /cmd_vel, yang kemudian diproses oleh simulator gazebo untuk menghasilkan data /joint_states. Data ini dipublikasikan ke node /robot_state_publisher,

yang kemudian mempublikasikan transformasi robot melalui `/tf`. Node `/turtlebot3_slam_gmapping` memproses data `/tf` untuk melakukan pemetaan dengan menggunakan SLAM. Visualisasi hasil pemetaan dilakukan melalui `/gazebo_gui`, yang menyediakan tampilan visual dari lingkungan simulasi dan pergerakan TurtleBot3 di dalamnya.

Validasi ini memastikan bahwa semua komponen sistem bekerja secara sinergis untuk menghasilkan hasil simulasi yang akurat dan konsisten. Jika ditemukan adanya link komunikasi yang hilang atau kesalahan pada salah satu node, langkah-langkah perbaikan dapat segera diambil untuk mengatasi masalah tersebut.

Setelah dilakukan tahapan pemetaan dan lokalisasi, berikutnya dilakukan pengujian navigasi otonom. Setelah semua node diaktifkan, tahapan ini diawali lagi dengan validasi node yang aktif untuk tahapan ini. Validasi ini dilakukan dengan memantau aliran data antara node-node dalam ROS menggunakan alat bantu visualisasi `rqt_graph` kembali. Penggunaan `rqt_graph` memungkinkan untuk memeriksa apakah koneksi antar-node dalam ROS berjalan dengan benar serta mendeteksi adanya link yang hilang atau tidak berfungsi, yang dapat memengaruhi proses navigasi otonom. Graf tersebut ditampilkan oleh gambar 7.



Gambar 7. Diagram Graf Node yang Aktif pada Pengujian Navigasi Otonom

Secara umum, diagram aliran data menunjukkan bagaimana berbagai node dalam ROS berinteraksi satu sama lain. Sistem ini melibatkan beberapa komponen kunci seperti `move_base` untuk navigasi, `AMCL` (Adaptive Monte Carlo Localization) untuk lokalisasi, serta `Gazebo` untuk simulasi fisik robot. Di dalam sistem ini, `map_server` mempublikasikan peta melalui topik `/map` ke `move_base`, yang kemudian digunakan oleh robot untuk menentukan jalur. Selanjutnya, `move_base` mengatur perintah tujuan dan mengirimkan perintah kecepatan melalui topik `/cmd_vel` ke `Gazebo`, yang mensimulasikan pergerakan robot.

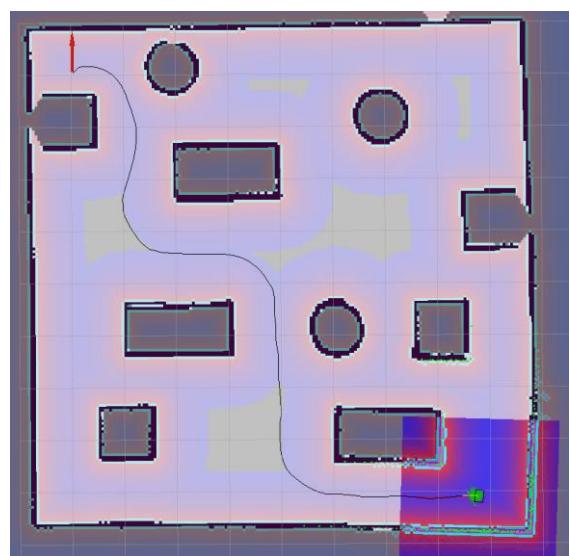
Pada sistem ini, validasi navigasi otonom juga melibatkan penggunaan `AMCL` yang bertugas memproses transformasi dari data `tf` serta informasi dari peta untuk menentukan posisi robot. Transformasi data dilakukan oleh `robot_state_publisher`, yang bertugas mengonversi data keadaan sendi robot menjadi transformasi `tf`. Secara keseluruhan, aliran data ini membentuk

fondasi utama dalam sistem navigasi otonom yang digunakan dalam penelitian ini, dan dengan bantuan `rqt_graph`, aliran data dapat divalidasi untuk memastikan bahwa setiap komponen bekerja dengan baik.

Dalam eksperimen ini, dua skenario utama diujikan untuk memvalidasi performa navigasi otonom robot. Skenario pertama adalah pergerakan robot di ruangan tanpa hambatan, sementara skenario kedua melibatkan penambahan hambatan setelah peta awal dibuat. Kedua eksperimen ini dilakukan untuk menguji kemampuan sistem dalam menavigasi ruangan yang berbeda kompleksitasnya serta dalam menghadapi perubahan lingkungan yang tidak terduga.

Pada eksperimen pertama, robot bergerak di dalam ruangan yang luas tanpa hambatan fisik. Hasil eksperimen menunjukkan bahwa, untuk pengujian berbagai jalur dan berbagai lokasi tujuan, robot selalu berhasil mencapai tujuan yang telah ditetapkan. Hal ini menunjukkan bahwa sistem navigasi bekerja secara optimal pada ruangan yang tidak memiliki kendala, meskipun hasil pemetaan mungkin tidak sempurna. Kondisi ini memungkinkan `move_base` untuk merencanakan jalur yang sederhana dan mudah diikuti oleh robot, tanpa perlu melakukan banyak perhitungan ulang atau penyesuaian jalur.

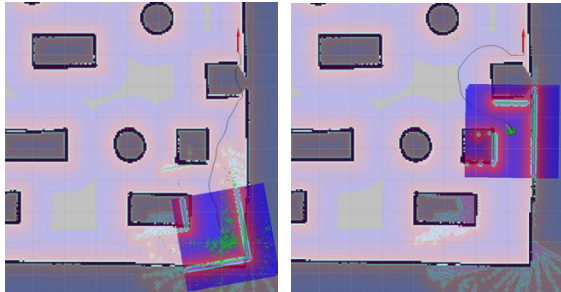
Eksperimen kedua melibatkan skenario yang lebih kompleks, di mana hambatan ditambahkan dalam lingkungan. Pada beberapa skenario percobaan dengan berbagai titik tujuan dan jalur, robot berhasil mengikuti jalur dan mencapai tujuan yang telah ditetapkan. Ini adalah bukti bahwa sistem `AMCL` dan `move_base` bekerja dengan baik. Hasil pengujian pada lingkungan dengan hambatan diperlihatkan pada Gambar 8.



Gambar 8. Pengujian Pada Ruang dengan Hambatan

Namun, dalam beberapa kasus, robot harus mengganti jalur beberapa kali, terutama jika jalur

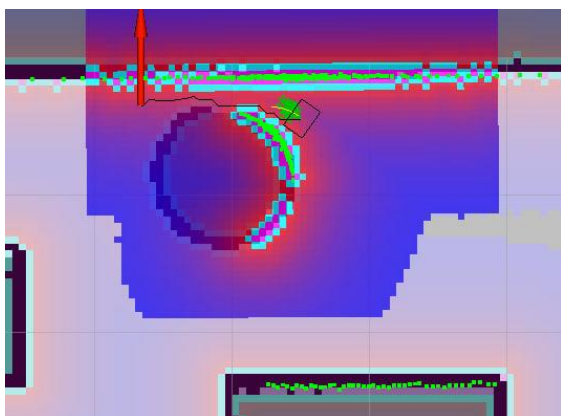
yang telah ditentukan melewati lorong sempit. Meskipun demikian, sistem navigasi mampu melakukan *replanning* atau perencanaan ulang dengan baik, sehingga robot tetap dapat mencapai tujuan. Keberhasilan ini menunjukkan fleksibilitas sistem dalam menghadapi lingkungan yang dinamis dan kompleks, meskipun robot dihadapkan pada tantangan berupa hambatan fisik yang tidak terduga.



Gambar 9. Replanning pada Lokasi yang Sempit

Meskipun sistem navigasi menunjukkan hasil yang positif, ada beberapa kasus di mana robot gagal mencapai tujuan. Kegagalan ini umumnya terjadi ketika robot terjebak di zona tabrakan atau di pojok ruangan yang sempit. Pada situasi ini, robot tidak mampu keluar dari kondisi terjebak, yang mengindikasikan adanya kelemahan dalam sistem kontrol kecepatan dan pemetaan lokal (*local_costmap*).

Salah satu penyebab utama kegagalan ini adalah kurang tanggunya pengontrol kecepatan pada robot serta lambatnya kecepatan refresh dari *local_costmap*. *Local_costmap* bertugas untuk memperbarui peta lokal di sekitar robot dan menentukan jalur terbaik yang harus diambil. Ketika kecepatan refresh lambat, robot mungkin tidak mendapatkan informasi terbaru dengan cukup cepat untuk menghindari hambatan yang baru muncul. Selain itu, kelemahan dalam pengaturan kecepatan robot juga menyebabkan robot tidak mampu bermanuver dengan baik di ruangan yang sempit.



Gambar 10. Robot Terjebak di Ruang Sempit

Untuk mengatasi masalah yang terjadi pada eksperimen, dilakukan beberapa perbaikan terhadap parameter kecepatan robot dan pengaturan *base_local_planner*. Pengaturan awal untuk parameter kecepatan, seperti *max_vel_x*, *min_vel_x*,

max_vel_theta, dan *min_vel_theta*, diuji ulang dan disesuaikan. Pada awalnya, *max_vel_x* diatur pada 0,22 dan *min_vel_x* pada -0,22, namun setelah ditemukan masalah pada ruang sempit, parameter ini diubah menjadi *max_vel_x* 0,15 dan *min_vel_x* -0,1.

Perubahan pada parameter kecepatan rotasi juga dilakukan, dengan mengurangi *max_vel_theta* menjadi 0,5 dan *min_vel_theta* menjadi 0,1. Hasil dari penyesuaian ini menunjukkan peningkatan yang signifikan, di mana robot berhasil keluar dari kondisi terjebak di pojokan dan mampu melanjutkan navigasi hingga mencapai tujuan. Ini menunjukkan bahwa pengaturan parameter kecepatan yang tepat sangat penting untuk memastikan bahwa robot dapat bermanuver dengan baik, terutama dalam kondisi lingkungan yang sempit atau rumit.

Dari hasil eksperimen yang telah dilakukan, didapatkan bahwa sistem navigasi otonom yang diuji mampu memberikan hasil yang memuaskan dalam kondisi yang ideal maupun dalam kondisi dengan hambatan yang tidak terduga. Keberhasilan robot dalam mendeteksi hambatan baru dan menyesuaikan jalur yang direncanakan secara otomatis menunjukkan bahwa sistem AMCL dan *move_base* bekerja dengan baik dalam merespons perubahan lingkungan.

Namun, beberapa kegagalan yang terjadi selama eksperimen menunjukkan bahwa masih ada ruang untuk perbaikan, terutama dalam hal kecepatan pemetaan lokal dan kontrol kecepatan robot. Penyesuaian terhadap parameter kecepatan dan pengaturan *base_local_planner* telah terbukti efektif dalam mengatasi beberapa masalah yang dihadapi, namun perlu dilakukan pengembangan lebih lanjut untuk meningkatkan performa sistem secara keseluruhan.

3. KESIMPULAN

Penelitian ini menunjukkan bahwa TurtleBot3 mampu melakukan navigasi otonom dengan cukup baik dalam lingkungan simulasi berbasis ROS, baik di ruangan tanpa hambatan maupun dengan hambatan. Pemetaan menggunakan SLAM bekerja dengan efektif di ruangan yang memiliki banyak fitur, namun kurang optimal di ruangan kosong karena minimnya fitur lingkungan. Lokalisasi menggunakan AMCL juga berhasil melacak posisi robot dengan akurat. Uji coba navigasi di ruangan berhambatan menunjukkan bahwa TurtleBot3 dapat mendeteksi dan menghindari rintangan secara efisien, meskipun ada beberapa kegagalan saat robot terjebak di area sempit atau zona tabrakan.

Perbaikan pada parameter kecepatan dan akselerasi robot berhasil meningkatkan kinerja dalam menghadapi situasi sulit. Namun, penyesuaian lebih lanjut pada pengontrol kecepatan dan kecepatan refresh peta lokal diperlukan untuk mengoptimalkan akurasi dan responsivitas. Dari sudut pandang keberlanjutan, simulasi ini menunjukkan potensi aplikasi robotik dalam lingkungan nyata yang lebih

kompleks, mendukung pengembangan teknologi robotik untuk otomatisasi tugas di berbagai sektor industri dan layanan. Penggunaan sistem robot yang lebih efisien dan adaptif juga dapat berkontribusi pada peningkatan produktivitas dan pengurangan penggunaan sumber daya manusia di berbagai bidang.

PUSTAKA

- Aprianti, N. A. (2023). Robotika Kesehatan: Tren Terkini Dalam Layanan Medis Dan Rehabilitasi. *Jurnal Multidisiplin West Science*, 2(08). <https://doi.org/10.58812/jmws.v2i08.591>
- Chen, G., Pan, L., Chen, Y., Xu, P., Wang, Z., Wu, P., Ji, J., & Chen, X. (2021). Deep Reinforcement Learning of Map-Based Obstacle Avoidance for Mobile Robot Navigation. *Sn Computer Science*, 2(6). <https://doi.org/10.1007/s42979-021-00817-z>
- Fu, Y. (2023). Analysis and Application Research of Mobile Robot Navigation Related Technologies. *Applied and Computational Engineering*, 9(1), 92–96. <https://doi.org/10.54254/2755-2721/9/20230055>
- Iqbal, J., Xu, R., Sun, S., & Li, C. (2020). Simulation of an Autonomous Mobile Robot for LiDAR-Based in-Field Phenotyping and Navigation. *Robotics*, 9(2), 46. <https://doi.org/10.3390/robotics9020046>
- Kurnia, M. S. (2023). Kontrol PID Untuk Navigasi Robot Berkaki Berbasis Sensor TOF Dan BNO055. *Jurnal Elektronika Dan Otomasi Industri*, 10(3), 462–470. <https://doi.org/10.33795/elkolind.v10i3.1952>
- Li, A., Cao, J., Li, S., Huang, Z., Wang, J., & Liu, G. (2022). Map Construction and Path Planning Method for a Mobile Robot Based on Multi-Sensor Information Fusion. *Applied Sciences*, 12(6), 2913. <https://doi.org/10.3390/app12062913>
- Liu, Y. (2024). Dynamic Validation of Calibration Accuracy and Structural Robustness of a Multi-Sensor Mobile Robot. *Sensors*, 24(12), 3896. <https://doi.org/10.3390/s24123896>
- Myint, H. (2018). Development of Robot Navigation System With Collision Free Path Planning Algorithm. *Machine Learning Research*, 3(3), 60. <https://doi.org/10.11648/j.mlr.20180303.12>
- Singh, R. P., & Nagla, K. S. (2019). Comparative Analysis of Range Sensors for the Robust Autonomous Navigation – A Review. *Sensor Review*, 40(1), 17–41. <https://doi.org/10.1108/sr-01-2019-0029>
- Supriandi, N. (2023). Desain Adaptif Dan Fleksibel Pada Robotika Industri: Membuka Jalan Untuk Produksi Berkelanjutan Dan Otomatisasi Yang Efisien. *Jurnal Multidisiplin West Science*, 2(06). <https://doi.org/10.58812/jmws.v2i6.435>
- Wijaya, A. R. (2024). Analisis Dan Optimasi Sistem Kendali Robot Falcon Millenium: Automatic Robot Palletizer Menggunakan PLC Omron. *Faktor Exacta*, 17(1). <https://doi.org/10.30998/faktorexacta.v17i1.19593>
- Xu, R. (2019). Path Planning of Mobile Robot Based on Multi-Sensor Information Fusion. *Eurasip Journal on Wireless Communications and Networking*, 2019(1). <https://doi.org/10.1186/s13638-019-1352-1>
- Zhao, J., Fang, J., Wang, S., Wang, K., Li, C., & Han, T. (2021). Obstacle Avoidance of Multi-Sensor Intelligent Robot Based on Road Sign Detection. *Sensors*, 21(20), 6777. <https://doi.org/10.3390/s21206777>