

Rancang Bangun Sistem Navigasi Robot Otonom Waiter-Bot Berbasis Robot Operating System

Abd. Salam¹, Muh Anshar², Andani Achmad³, dan Muh. Reza Eka Yudha⁴

^{1,2,4}Cognitive Social Robotics and Advanced Artificial Intelligence ³Laboratorium Jaringan Komputer

^{1,2,3,4}Departemen Teknik Elektro, Universitas Hasanuddin, Makassar, Indonesia

abdsalam19d@student.unhas.ac.id¹; anshar@unhas.ac.id²; andani@unhas.ac.id³;
yudhamre21d@student.unhas.ac.id⁴

Abstrak

Perkembangan teknologi robotika semakin masif diterapkan di industri manufaktur seiring memasuki era revolusi industri 5.0 dalam membantu meringankan beban manusia serta meningkatkan efisiensi waktu dan tenaga. Salah satu bentuk penerapannya pada penelitian robot otonom Waiter-Bot sebagai robot cerdas berbasis Robot Operating System (ROS) dengan misi pengantaran logistik secara autonomus. Penelitian ini menggunakan metode kuantitatif dengan pengujian robot melakukan perencanaan jalur navigasi dalam mencapai misi. Metode perencanaan jalur navigasi terdiri atas perencanaan jalur lokal (local planner) dan jalur global (global planner) untuk memperoleh jalur trayektori lintasan paling terdekat dan teraman. Pengujian global planner dilakukan dengan menggunakan dua skenario. Skenario 1, robot melakukan navigasi dengan kemampuan mengenali batas jalur lintasan. Adapun skenario 2, robot diberikan perlakuan penambahan objek diluar hasil pemetaan untuk mengenali dan menghindari obstacle tersebut. Pengujian local planner menggunakan algoritma Dynamic Window Approach (DWA) untuk mengetahui kemampuan robot merencanakan opsi jalur baru dalam kondisi situasional agar tetap bergerak otonom untuk mencapai target misi. Hasil yang diperoleh dari penelitian ini yakni pada pengujian global planner, waktu tempuh rata-rata saat robot melewati misi A,B,C dengan jarak tempuh 12,02 meter pada skenario 1 yakni 3,81 menit serta skenario 2 dengan waktu tempuh rata-rata 4 menit 12 detik. Sedangkan pengujian local planner, diperoleh waktu transisi pivot dengan rata-rata 73 detik. Dalam sistem navigasi robot berbasis ROS pada penelitian ini diketahui jumlah node yang aktif yakni 14 node dalam 8 topik. Dengan demikian navigasi Waiter-Bot dapat menjalankan misi dengan efektif.

Kata Kunci — robot, navigasi, otonom, ROS, local planner, global planner

Abstract

The development of robotics technology is increasingly being applied in the manufacturing industry as we enter the era of the industrial revolution 5.0 in helping to ease human burdens and improve time and energy efficiency. One form of its application to the research of the autonomous robot Waiter-Bot as the intelligent robot bearing the Robot Operating System (ROS) with a logistical delivery mission autonomically. This research uses quantitative methods with testing robots performing navigation route planning in reaching the mission. The navigation path planning method consists of local path planning and global path planning to obtain the nearest and safest trajectory path. Global planner testing is done using two scenarios. Scenario 1, the robot does navigation with the ability to recognize the boundaries of the trajectory. As for scenario 2, the robot is given the treatment of adding objects outside the mapping result to recognize and avoid the obstacle. Local planner testing uses the Dynamic Window Approach (DWA) algorithm to determine the robot's ability to plan new route options in situational conditions in order to remain autonomously moving to reach mission targets. The results obtained from this study were from global planner tests, the average time spent by robots passing missions A, B, C with a distance of 12,02 meters in scenario 1 of 3.81 minutes and scenario 2 with an average time of 4 minutes and 12 seconds. In the ROS-based robotic navigation system, the number of active nodes is 14 in eight subjects. Thus the navigation of the Waiter-Bot can execute the mission effectively

Keywords — autonomous, global planner, local planner, navigation, robot, ROS.



I. PENDAHULUAN

Perkembangan teknologi seiring dengan tuntutan kebutuhan hidup akan pelayanan berubah, kebutuhan serta kesibukan juga meningkat. Pekerjaan manusia dituntut lebih efisien dalam menggunakan waktu untuk memenuhi kebutuhan hidupnya sehingga dapat lebih produktif untuk bekerja. Dalam sektor industri tercatat sekitar 422.000 unit robot di seluruh dunia dan diperkirakan bertambah sebanyak 12 persen per tahun hingga tahun 2022 (IFR, 2018). Autonomous mobile robot merupakan salah satu jenis robot yang berkembang saat ini. Teknologi yang dapat diterapkan untuk membantu dan meringankan pekerjaan pelayanan restoran adalah dengan memanfaatkan kecanggihan robot cerdas. Robot ini harus memiliki kemampuan untuk membawa logistik, seperti mengantar makanan, barang, baju, dan/atau obat-obatan (Sierra Marín dkk., 2021). Permasalahan yang dihadapi sebuah autonomous mobile robot adalah masalah navigasi karena robot otonom harus mengenali lingkungannya. Oleh karena itu, penelitian ini mengangkat studi rancang bangun Smart Waiter-Bot sebagai robot cerdas yang dirancang untuk dapat melakukan pengantaran makanan secara otonom pada titik tujuan dalam lingkungan dalam ruangan. Sistem robot yang dibangun terdiri dari beberapa komponen diantaranya kamera kinect, sensor encoder, MPU-90, dan HCSR yang dijalankan berbasis middleware Robot Operating System (ROS). ROS digunakan karena setiap grup riset robotika umumnya berfokus pada topik-topik riset robot tertentu sehingga ROS hadir dalam memudahkan pengembangan suatu topik riset tertentu tanpa harus memikirkan topik riset lainnya. ROS mencakup tools, libraries dan convention yang bertujuan untuk menciptakan serta mengembangkan robot yang kompleks. Dengan begitu grup riset yang mengembangkan sistem navigasi tidak perlu harus memikirkan pula sistem biomekanik dari robot tersebut melainkan hanya mengembangkan salah satu bagian dari sistem navigasi tersebut tanpa harus membuat sistem navigasi dari dasar. Dalam pengembangan sistem navigasi Waiter-Bot secara otonom digunakan metode perencanaan jalur salah satunya dengan metode algoritma Dynamic Window Approach (DWA). Selain itu, diperlukan pula sebuah peta yang presisi agar dapat melakukan navigasi, sehingga robot perlu membangun peta dan lokalisasi lingkungannya terlebih dahulu.

II. PENELITIAN TERKAIT

A. Implementasi Sistem Robot Otonom dengan Sensor Kinect menggunakan Algoritma Gmapping dan Timed Elastic Band.

Jurnal. Hamdan Zuhdi Dewanul Arifin, Rizal Maulana, Hurriyatul Fitriyah, Universitas Brawijaya, Indonesia (2022)

Penelitian ini adalah bentuk dari purwarupa robot yang memiliki ukuran yang besar. Untuk menciptakan navigasi secara otonom diperlukan metode perencanaan jalur, maka dari itu digunakan algoritma Timed Elastic Band (TEB).

B. An Autonomous Robot for Waiter Service in Restaurants. Thesis. Dr. Md. Khalilur Rahman. BRAC University, Dhaka, Bangladesh, India (2015)

Penelitian ini berfokus dalam meningkatkan produktifitas suatu restoran dengan inovasi pengantar makanan otomatis. Dengan kemampuan untuk mengikuti jalur yang ditentukan dengan bantuan susunan sensor IR berbasis arduino dalam mencapai tujuan yang presisi dan akurasi.

C. Wireless Waiter Robot. Article. Arwa Hamid Salih Hamdany, Lubab H. Albak, Raid Rafi Omar Al-Nima. Technical Engineering College, Northern Technical University, Mosul, Iraq (2019)

Penelitian ini terdiri atas fitur E-Menu, Keypad, layar LCD, Bluetooth, Arduino, dan perangkat kartu kredit. difasilitasi oleh perangkat keras utama. Robot pelayan dapat mengikuti jalur yang ditentukan untuk mengambil pesanan pelanggan.

III. PENELITIAN TERKAIT

A. Perancangan Umum

Pada tahapan perancangan umum implementasi sistem navigasi robot otonom Waiter-Bot menggunakan modul framework Robot Operating System (ROS) sebagai robot middleware. Tahapan perancangan sistem navigasi robot otonom Waiter-Bot meliputi aspek perangkat keras dan perangkat lunak di mana blok diagram secara keseluruhan dapat dilihat pada figure 1. Sistem kendali yang digunakan merupakan kendali close loop dengan mempertimbangkan rintangan terhadap kecepatan roda robot dalam kemampuan merencanakan jalur terdekat dan teraman dalam sistem navigasinya.

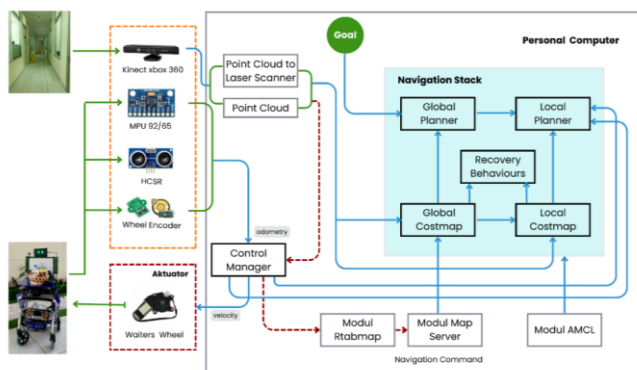


Fig. 1. Sistem Navigasi Waiter-Bot Berbasis ROS

B. Perancangan Perangkat Keras

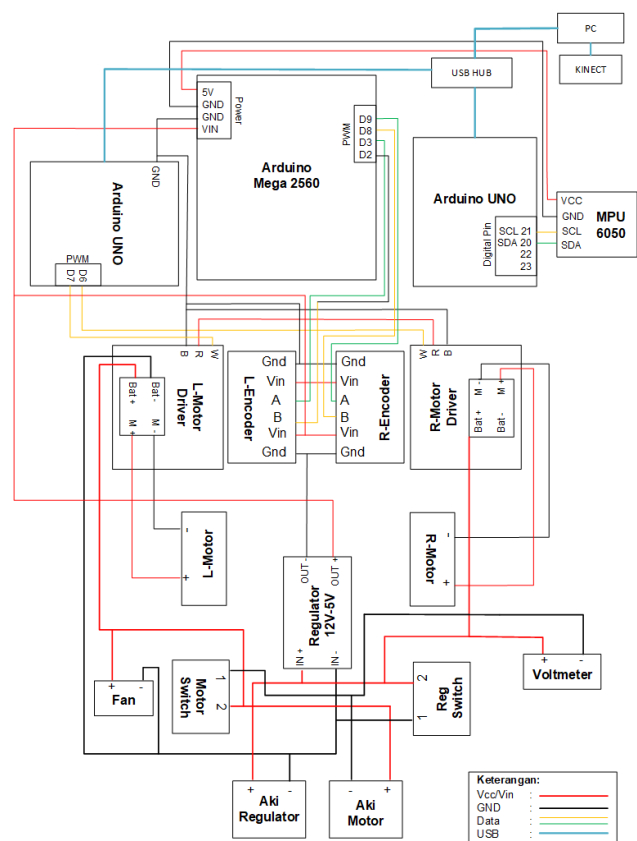


Fig. 2. Skematik Diagram Waiter-Bot

Sensor yang digunakan dalam proses lokalisasi yakni sensor visual berupa sensor kinect dan sensor pergerakan seperti IMU. Sensor Kinect bertindak sebagai persepsi robot yang dapat memindai area di depan robot sehingga dapat dilakukan pemetaan dan navigasi. Sedangkan sensor IMU

dan odometer digunakan untuk lokalisasi agar pergerakan robot dapat selalu dipantau dan membantu melakukan pemetaan dan navigasi. Dalam mekanisasi robot dimulai dengan merancang chassis atau badan robot sehingga perangkat dapat diletakkan dan saling terintegrasi. Chassis dirancang terdiri atas bagian bawah untuk peletakan komponen meliputi sensor, rangkaian daya, aktuator dan miniPC. Kemudian bagian atas meliputi talangan dalam meletakkan sajian makanan serta bagian kepala terdiri atas kamera kinect dan touch screen display sebagai media inputan instruksi robot.

C. Perancangan Umum

Perangkat lunak dibuat dibagi menjadi beberapa bagian sesuai dengan tahapan proses dari awal akuisisi data dari low level kemudian diproses pada sistem high level untuk memperoleh sistem navigasi robot otonom pada ROS. Struktur ROS bersifat modular sehingga memudahkan proses integrasi antara program yang telah dikembangkan sebelumnya dengan program yang sedang dikembangkan. Komunikasi antar proses dalam ROS dilakukan dengan menghubungkan program atau yang disebut dengan node yang saling berkomunikasi via topic. Node merupakan program yang ditulis untuk melakukan berbagai fungsi atau tugas tertentu. Sedangkan topic adalah bus yang digunakan dalam transmisi pesan (message) antar node.



Fig. 3. Tampilan Robot Waiter-Bot

Node dapat mempublish data (misanya data laser scanner) yang disebut dengan publisher dan node dapat pula berfungsi menerima dan menampilkan data yang disebut dengan subscriber. Streaming data (message) via topic tersebut dilakukan dalam jaringan lokal TCP yang disebut dengan jaringan ROS atau ROS network.

Proses secara keseluruhan dari sistem perangkat lunak yang harus dibuat secara umum adalah:

1. Integrasi sensor low level pada komputasi Arduino Mega 2560
2. Sistem komunikasi komponen low level dengan sistem high level
3. Proses pemetaan dan lokalisasi
4. Pembuatan topik dan node pada ROS
5. Perencanaan jalur navigasi robot otonom
6. Sistem Navigasi Pada ROS

Terdapat beberapa hal yang perlu diperhatikan dalam proses navigasi mulai dari perencanaan sensor, gmapping, perencanaan jalur lokal dan perencanaan jalur global. Adapun navigation stack pada Waiters-Bot seperti Figure 4.

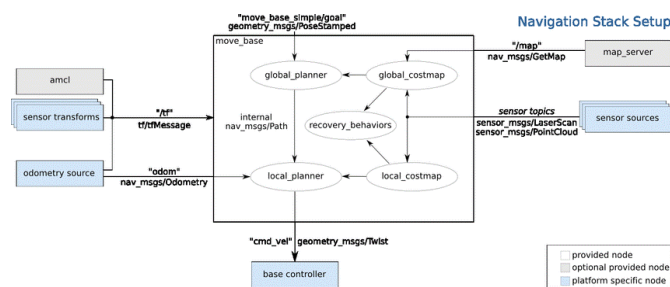


Fig. 4. Navigation stack setup

Sensor transform serta odometry source digunakan sebagai lokalisasi dari posisi robot terhadap lingkungan dan informasi dari odometry digunakan saat melakukan pencarian jalur lokal. Global planner didapat dari informasi global costmap yang merupakan proses dari rtabmap yang telah dibuat sebelumnya oleh proses mapping. Selain itu dalam mencari global costmap dan local costmap diperlukan informasi dari sensor-sensor yang terdapat pada robot. Informasi dari global planner ini akan dipecah-pecah dan menjadi informasi pada local planner yang nantinya akan menentukan gerak dari sistem navigasi Waiter-Bot.

D. Perancangan Umum

1) *Perencanaan Jalur Lokal*: Perencanaan local planner dilakukan untuk mengetahui kemampuan Waiters-Bot dalam mengenali dan menghindari rintangan dalam menjalankan misi. Dalam melakukan perencanaan jalur terpendek dalam

cakupan lokal menggunakan metode yang telah ada di library ROS yakni Dynamic Window Approach (DWA) yang membuat lintasan berdasarkan peta yang telah dibuat sebelumnya [6]. DWA akan membuat beberapa macam lintasan dan membuat suatu fungsi nilai untuk menghitung jarak terdekat serta menghindari adanya penghalang baik itu penghalang yang statis maupun yang dinamis. Nilai dari jarak yang dibuat berdasarkan grid yang dibentuk oleh DWA. Fungsi nilai ini berupa dx , dy dan $dtheta$ yang akan secara simultan dikirimkan ke robot.

2) *Perencanaan Jalur Global*: Setelah melakukan lokalisasi, Waiters-Bot akan melakukan perencanaan jalur navigasi yang akan ditempuh menuju ke titik tujuan. Dalam melakukan perencanaan jalur sebelumnya diubah menjadi model diskrit. Terdapat tiga pendekatan dalam penentuan jalur navigasi ini yaitu pendekatan road map, pendekatan cell decomposition dan pendekatan potential field[5].

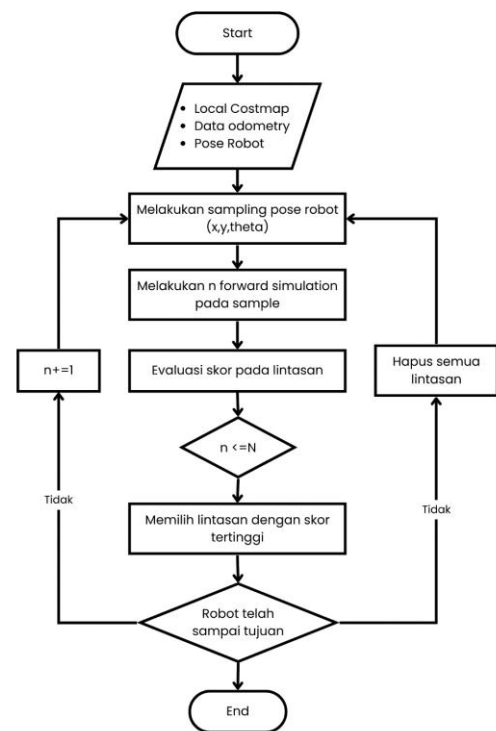


Fig. 5. Flow chart perencanaan jalur lokal

Pendekatan Cell Decomposition akan membedakan antara area geometri yang nantinya disebut dengan sel yang bebas dan yang terisi oleh objek. Algoritma dari pendekatan cell decomposition yaitu:

- a. Langkah pertama adalah membagi peta menjadi daerah daerah kecil dan terhubung yang disebut dengan sel.

b. Tentukan sel-sel yang bebas yang berdekatan dan buat konektifitas antara sel-sel yang bebas tersebut. Sel-sel yang tidak bebas disebut dengan sel yang terdekomposisi.

c. Tentukan sel konfigurasi awal dan tujuan dan buat grafik konektifitas antara kedua sel tersebut.

d. Dari urutan sel tersebut hitung jalur di setiap sel yang dilewati serta buat urutan gerakan untuk urutan terbaik. Pendekatan berikutnya adalah pendekatan potential field.

Pendekatan ini memodelkan robot menjadi suatu partikel yang bergerak dibawah pengaruh medan potensial yang ditentukan oleh tujuan serta beberapa rintangan [7]. Setiap pergerakan robot akan ditentukan oleh medan potensial yang berada di lokasinya. Kelebihan dari metode ini adalah komputasi yang dihasilkan tidak terlalu besar karena setiap pergerakan robot hanya tergantung pada pendekatan medan potensialnya. Selain itu pada pendekatan ini juga memiliki respon yang baik pada halang rintang yang dinamis karena setiap adanya halang rintang maka medan potensial akan selalu berubah.

E. Skenario Pengujian Sistem Navigasi

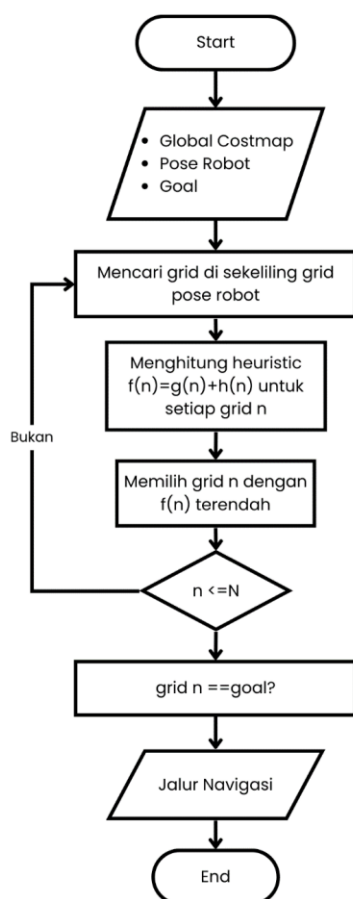


Fig. 6. Visualisasi algoritma DWA

Pengujian navigasi Waiter-Bot Secara keseluruhan Pengujian sistem Waiter-Bot untuk mengetahui kinerja fungsional komponen keseluruhan. Fokus pada pengujian navigasi ini untuk kondisi lingkungan statis dan dinamis. Dalam simulasi pengoperasian Waiter-Bot akan dilakukan pengujian pada saat kondisi dengan rintangan yang statis dan dinamis. Lingkungan statis pada pengujian ini merupakan pengujian dengan mengetahui kemampuan robot melakukan navigasi berdasarkan peta yang telah berhasil dibangun pada tahap pemetaan. Sementara pengujian lingkungan dinamis adalah pengujian sistem navigasi robot untuk mengetahui kemampuannya dalam mendeteksi dan menghindari objek tambahan diluar hasil pemetaan yang berperan sebagai penghalang.

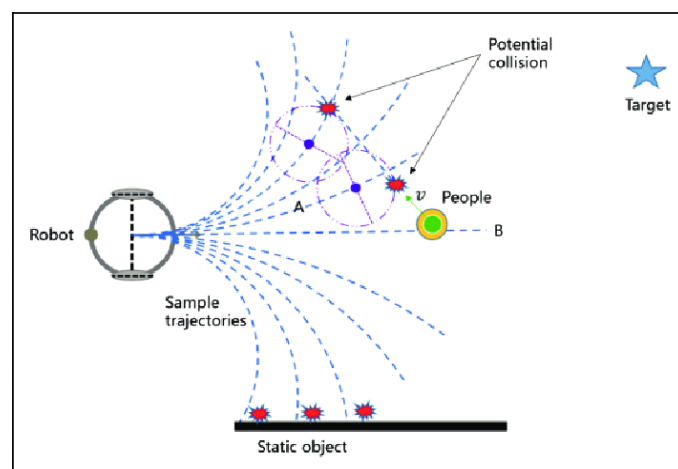


Fig. 7. Flow chart perencanaan jalur global

Kemudian dilakukan juga pengamatan atas perhitungan nilai kecepatan linear (m/s) dan kecepatan angular (rad/s) pada saat robot melakukan navigasi. Luaran yang diharapkan pada pengu bahwa robot dapat mengenali dan adaptif dengan kondisi lingkungan tersebut dalam menjalankan misinya. Berikut desain denah skenario navigasi robot pada lingkungan statis dan dinamis. Terdapat tiga misi tujuan yang akan ditempuh oleh robot otonom Waiter-Bot yakni misi A, B dan C. Robot akan melakukan perjalanan secara otonom dengan terlebih dahulu melakukan perencanaan jalur global dan lokal pada sistem navigasi. Waiter-Bot melakukan perjalanan secara otonom dari titik start ke titik misi A, B dan C yang memiliki jarak tempuh yang berbeda beda tanpa ada objek tambahan diluar hasil pemetaan.



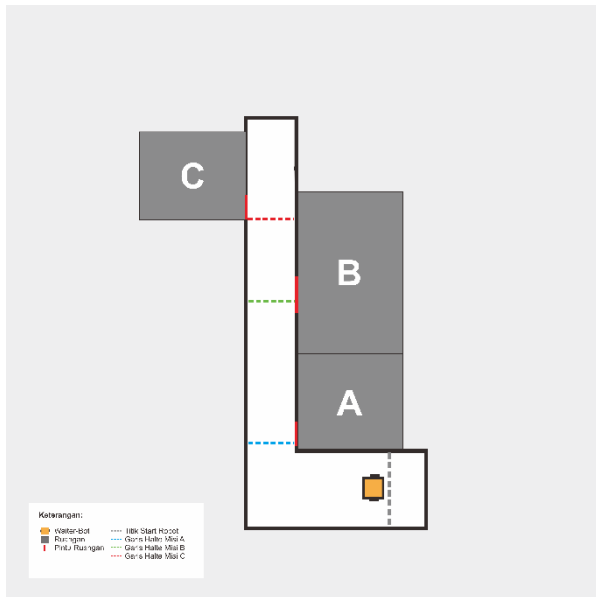


Fig. 8. Denah pengujian navigasi robot pada lingkungan statis

Selanjutnya pada pengujian skenario navigasi lingkungan dinamis oleh Waiker-Bot melakukan perjalanan secara otonom dari titik start ke titik misi A, B dan C yang memiliki jarak tempuh yang berbeda beda dengan perlakuan tambahan objek sebagai obstacle diluar hasil pemetaan yang telah dilakukan.

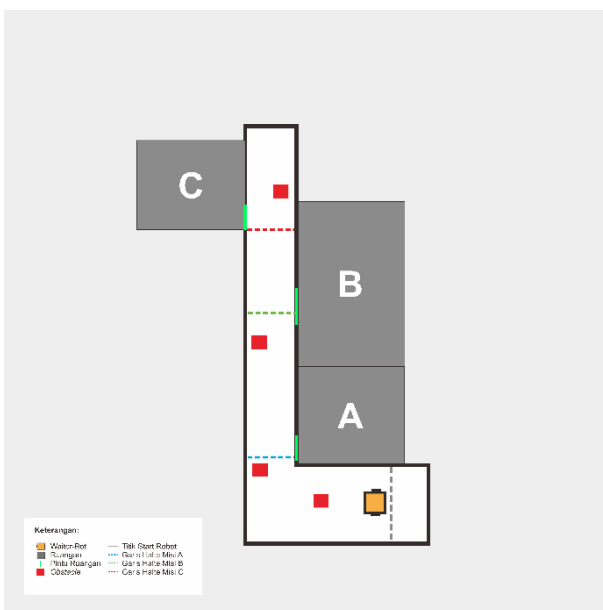


Fig. 9. Denah pengujian navigasi robot pada lingkungan Dinamis

maka akan menggabungkan kecepatan linear dan kecepatan angular dapt digunakan dalam mengetahui kecepatan resultan robot.

IV. HASIL DAN PEMBAHASAN

A. Pengujian Deviasi Kecepatan Linear Robot

Deviasi kecepatan pada robot merujuk pada perbedaan antara kecepatan yang diinginkan dalam hal ini nilai kecepatan yang doatur dalam sistem navigasi ROS terhadap kecepatan eksisting yang terjadi pada robot. Sebelum melakukan pengujian navigasi robot secara otonom, terlebih dahulu dilakukan pengujian deviasi robot dalam mengeksekusi perintah nilai kecepatan dari pengaturan ROS untuk memperoleh deviasi nilai kecepatan sehingga dapat menjadi acuan dalam pengaturan nilai kecepatan linear robot. Dari tabel di atas menunjukkan bahwa pengujian gerak maju robot diberikan perlakuan pengaturan nilai kecepatan pada ROS lalu kemudian dieksekusi oleh robot. Nilai kecepatan diberikan dari nilai 0,21 m/s hingga 1,50 m/s dengan skenario pengujian pada jarak tempuh konstan 1 meter dengan mengabaikan node navigation stack dalam mendekteksi objek penghalang. Adapun grafik deviasi antara perintah nilai kecepatan dan nilai eksekusi kecepatan robot sebagai berikut. Dari grafik kecepatan linear di atas, robot mengeksekusi perintah nilai kecepatan pada ROS secara linear dan berbanding lurus antara nilai kecepatan terhadap waktu tempuh. Didapatkan persentase error paling tinggi 0,55 persen yakni pada pengujian keempat. Adapun nilai perintah kecepatan yang lebih presisi yakni dengan nilai kecepatan 0,21-0,23 m/s dengan persentase error 0,09 persen.

B. Pengujian Skenario Navigasi Robot Pada Lingkungan Statis dan Dinamis

1) *Skenario Pengujian Navigasi Pada Lingkungan Statis:*
 Pada pengujian sistem navigasi robot otonom skenario 1, Waiker-Bot menjalankan misi A dengan rata-rata waktu tempuh 1 menit 37 detik pada jarak tempuh 530 cm.

Kombinasi dari kecepatan linear dan kecepatan angular digunakan untuk mengontrol pergerakan robot dalam navigasi otonom. Misalnya jika robot perlu mengikuti jalur lengkung,

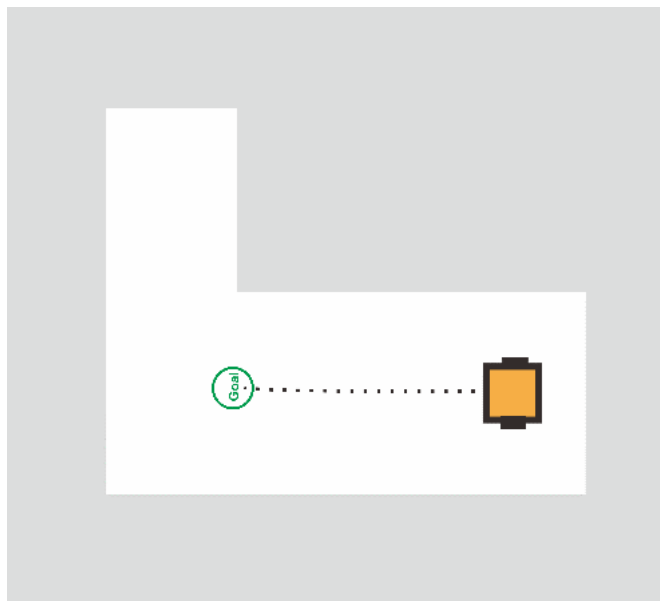


Fig. 10. Pengujian deviasi kecepatan robot

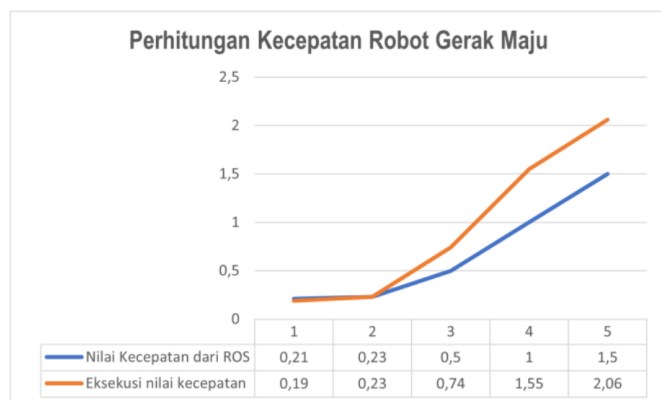


Fig. 11. Grafik deviasi kecepatan linear robot

Dalam menjalankan misi B, diperoleh waktu tempuh rata-rata pada jarak 920 cm yakni 2 menit 53 detik. Kemudian saat menempuh misi C dengan jarak 1202 cm, diperoleh waktu 3 menit 81 detik. Seluruh pengujian navigasi robot dalam skenario perencanaan global berhasil mencapai goal sesuai trayektori yang dibangun dalam ROS.

2) *Skenario Pengujian Navigasi Pada Lingkungan Dinamis:* Pengujian navigasi Waiter-Bot dalam skenario 2, robot menjalankan misi A dengan jarak 530 cm dengan waktu tempuh rata-rata misi A yakni 1 menit 48 detik. Misi B dengan jarak tempuh 920, diperoleh waktu tempuh rata-rata 3 menit 14 detik. Serta misi C dengan jarak tempuh 1202 cm, diperoleh waktu tempuh rata-rata 4 menit 12 detik. Sehingga pada pengujian sekanrio 1 dan 2 dinyatakan berhasil karena

robot dapat mengenali objek dan merencanakan jalur trayektori terdekat dan teraman.

C. Sistem Komunikasi ROS Pada Navigation Stack

1) *Node Aktif Pada Saat Robot Tidak Bergerak:* Berikut tabel keterangan topik yang digunakan dalam sistem navigasi robot berbasis ROS.

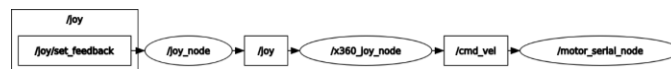


Fig. 12. Komunikasi node saat robot tidak bergerak

2) *Node Aktif Pada Saat Robot Bergerak:* Dari gambar tersebut, diperoleh data aktivasi status node sebagai berikut.

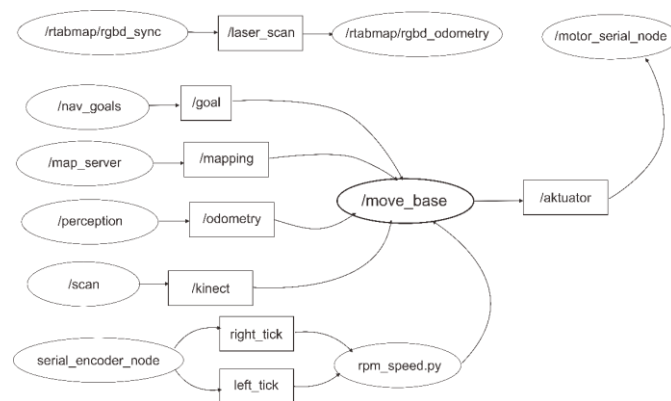


Fig. 13. Komunikasi node saat robot bergerak

V. KESIMPULAN

Dari penelitian ini dapat diperoleh kesimpulan sebagai berikut:

a. Deviasi kecepatan pada robot merujuk pada perbedaan antara kecepatan yang diinginkan dalam hal ini nilai kecepatan yang diatur dalam sistem navigasi ROS. Pada pengujian yang dilakukan diperoleh bahwa perintah nilai kecepatan linear yang ideal untuk digunakan dalam sistem navigasi robot otonom yakni interval 0 – 0.23 m/s dengan persentase error tertinggi 0.09 persen. Adapun pengujian respon kecepatan linear robot terhadap objek penghalang, diperoleh data kecepatan linear robot yakni interval 0.12-0.22 m/s serta kecepatan angluer robot diperoleh nilai interval - 0.13 hingga 0.17 rad/s. Apabila robot mendeteksi objek penghalang maka sistem navigasi pada based local planner ROS akan memberikan instruksi nilai kecepatan linear lebih kecil. Begitupun nilai angular, apabila bernilai negatif atau bahkan positif maka roda robot menunjukkan bahwa robot



sedang melakukan penyesuaian navigasi dengan kompensasi adanya manuver.

b. Dalam pengujian skenario 1 lingkungan statis, robot melakukan navigasi menuju misi A dengan rata-rata waktu tempuh 1 menit 37 detik pada jarak tempuh 530 cm. Dalam menjalankan misi B, diperoleh waktu tempuh rata-rata pada jarak 920 cm yakni 2 menit 53 detik. Kemudian saat menempuh misi C dengan jarak 1202 cm, diperoleh waktu 3 menit 81 detik. Seluruh pengujian navigasi robot dalam skenario perencanaan global berhasil mencapai goal sesuai trayektori yang dibangun dalam ROS. Adapun pada skenario navigasi lingkungan dinamis, robot menjalankan misi A dengan jarak 530 cm dengan waktu tempuh rata-rata 1 menit 48 detik. Misi B dengan jarak tempuh 920, diperoleh waktu tempuh rata-rata 3 menit 14 detik. Serta misi C dengan jarak tempuh 1202 cm, diperoleh waktu tempuh rata-rata 4 menit 12 detik. Sehingga pada pengujian skenario 1 dapat dinyatakan berhasil karena robot dapat mengenali objek dan merencanakan jalur trayektori terdekat dan teraman.

c. Pada pengujian skenario 2 navigasi lingkungan statis diperoleh data grafik dari pengujian robot bergerak lurus tanpa penghalang yakni kecepatan linear dengan interval nilai 0-0.21 m/s. Serta kecepatan angular dengan interval -0.31 hingga 0.22. Adapun pada pengujian skenario lingkungan dinamis, diperoleh data kecepatan linear robot yakni interval 0.12-0.22 m/s serta kecepatan angular robot diperoleh instruksi nilai interval -0.13 hingga 0.17 rad/s sepanjang robot melakukan navigasi mencapai goal. Apabila robot mendeteksi objek penghalang maka sistem navigasi pada based local planner ROS akan memberikan instruksi nilai kecepatan linear lebih kecil begitupun nilai angular, apabila bernilai negatif maka roda robot menunjukkan bahwa robot sedang melakukan pembelokan atau manuver sehingga pergerakan robot akan lebih lambat hingga berhasil melewati objek penghalang lalu kemudian mengikuti trayektori berdasarkan perencanaan jalur navigasi dari sistem ROS.

d. Dalam pengamatan sistem komunikasi antar node dalam ROS Navigation Stack, diketahui jumlah node yang aktif saat robot tidak bergerak yakni 4 node. Sedangkan pada saat robot mulai bergerak secara otonom, diketahui jumlah node yang aktif yakni 14 node pada 7 topik pendukung navigasi.

2AIR) Departemen Teknik Elektro Fakultas Teknik Universitas Hasanuddin.

REFERENSI

- [1] B. Anggoro. Desain Pemodelan Kinematik dan Dinamik Humanoid Robot. Skripsi. Universitas Diponegoro, Bandung, 2018.
- [2] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram. (2012). "Trajectory Modification Considering Dynamic Constraints of Autonomous Robots," 7th Ger. Conf. Robot. Robot. 2012, pp. 74–79, 2012.
- [3] C. Wang and D. Du, "Research on Logistics Autonomous Mobile Robot System," 2016 IEEE Int. Conf. Mechatronics Autom. IEEE ICMA 2016, pp. 275–280, 2016.
- [4] E. Dwiky, D. Endang, H.S Rosalia, Sunarto, T. S Kuart, G. Ferrianto. Sistem Navigasi Mobile Robot Dalam Ruangan Berbasis Autonomous Navigation. Journal of Mechanical Engineering and Mechatronics, pp 78-86, 2019.
- [5] F. Dieter, B. Wolfram, and T. Sebastian. The Dynamic Window Approach to Collision Avoidance. pp. 137–146, 1997.
- [6] G. Dudek and M. Jenkin, Computational Principles of Mobile Robotics, second ed. New York: Cambridge University Press, 2010.
- [7] Gupta, K.V.N., Prasthech, K., Lakshmi, K.R.V., Kadam, S.S., Bailey, K. 2021. Autonomous Navigation in Dynamic Environment. International Research Journal of Engineering and Technology (IRJET). Vol. 08 (6). BMS College of Engineering, Karnataka, India.
- [8] Ilham. Navigasi Mobile Robot Darat Menggunakan Odometri Visual Berbasis Citra Stereo. Tesis. Institut Teknologi Sepuluh Nopember Surabaya, 2019.
- [9] Joseph, Lentin. 2018. Robot Operating System for Absolute Beginners. Apress. India.
- [10] Muhammad, F., Wicaksana, C. A. 2021. Literature Review Sistem Navigasi Autonomous Mobile Robot Berbasis ROS (Robot Operating System). Jurnal Ilmiah Setrum. Vol.10 No.1 (2021) 103-112.
- [11] N. America, "Executive Summary World Robotics Twenty Nineteen Industrial Robots," pp. 13–16, 2019. <https://ifr.org/webpkgcache.com/doc/s/ifr.org/downloads/press2018/Executive%20Summary%20WR%202019%20Industrial%20Robots>.
- [12] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," The International Journal of Robotics Research, vol. 5, 1986, p. 90.
- [13] Y. Sun, L. Guan, Z. Chang et al., "Design of a low-cost indoor navigation system for food delivery robot based on multi-sensor information fusion," Sensors, vol. 19, no. 22, 4980 pages, 2019.
- [14] Z. Jianwei, L. Shengyi, L. Jinyu. 2022. Research and Implementation of Autonomous Navigation for Mobile Robots Based on SLAM Algorithm under ROS.
- [15] Z. Meng, H. Sun, H. Qin, Z. Chen, C. Zhou, and M. H. Ang. 2018. Intelligent Robotic System for Autonomous Exploration and Active SLAM in Unknown Environments. SII 2017 - 2017 IEEE/SICE Int. Symp. Syst. Integr., Vol. 2018-Janua, pp. 651–656.

Penulis mengucapkan terimakasih kepada Dosen Pembimbing dan rekan Grup Riset Cognitive Social Robotics and Advanced Artificial Intelligence Research Centre (CSR-