Slam_Toolbox Implementation for Autonomous Indoor Delivery Robot: A Case Study

Bobby Anggra Putra[1], Anugerah Wibisana[2]

[1]Jurusan Teknik Elektro, Prodi Teknik Robotika, Politeknik Negeri Batam, Batam, Indonesia

*Email: bobby.lumbangaol27@gmail.com

Received on dd-mm-yyyy | *Revised on dd-mm-yyyy* | *Accepted on dd-mm-yyyy*

***Abstract*—Autonomous mobile robots are increasingly deployed for indoor applications, such as delivery services. This paper presents the implementation and analysis of the SLAM_Toolbox and Nav2 framework within the Robot Operating System (ROS) 2 Humble for an autonomous indoor delivery robot. Conducted at the Main Building of Batam State Polytechnic, this study aims to develop a robust navigation system capable of accurate mapping and reliable localization in dynamic environments. The methodology involves sensor integration, occupancy grid mapping using SLAM_Toolbox, and autonomous path planning. The results demonstrate that the system effectively generated consistent maps and achieved stable localization. In navigation tests, the robot successfully reached predefined goals with a final precision of 0.36 meters and an average positional accuracy of 0.1 meters. Despite encountering environmental challenges that triggered four recovery behaviors, the robot completed the mission in 96 seconds without manual intervention. This performance confirms the viability of using open-source ROS 2 modules for cost-effective and resilient autonomous delivery robots in structured indoor settings.**

Keywords: Autonomous Robots, Delivery System, Indoor Navigation, Robot Operating System (ROS), SLAM_Toolbox.

I. INTRODUCTION

The demand for automation in logistics and internal supply chain management has driven a great deal of interest in Autonomous Mobile Robots (AMR). In indoor environments such as offices, hospitals, and educational institutions, AMR can handle the transportation of materials and delivery tasks efficiently, reducing human labor and improving operational efficiency. A critical enabler technology for AMR is Simultaneous Localization and Mapping (SLAM) [1], which allows a robot to construct a map of an unknown environment while tracking its location within the map.

Robot Operating Systems (ROS)[2]. have emerged as the de facto standard for developing robotics applications, providing a flexible framework and a rich package ecosystem. Among them, the SLAM_Toolbox package offers a modern and efficient 2D SLAM solution, integrating advanced algorithms with real-time performance suitable for practical applications. While many studies have evaluated SLAM algorithms in controlled laboratory settings, more case studies are needed in real-world operational environments to assess their performance and practical limitations.

This study answers this gap by implementing and analyzing SLAM_Toolbox on custom delivery robots built in the Main Building of the Batam State Polytechnic. The objectives of this study were: to implement SLAM_Toolbox [3], [4]. for mapping and navigation [5],[6],in complex indoor environments, to evaluate the performance of the system in terms of mapping accuracy, localization stability, and task completion success rate, and to identify practical challenges and propose potential solutions. The findings from this case study are expected to provide valuable insights for the development of reliable and cost-effective autonomous delivery systems for similar institutional environments.

II. METHOD

2.1 System Architecture

The autonomous delivery robot used in this study is a differential-drive mobile robot platform. The system architecture consists of hardware and software components integrated within the ROS 2 Humble framework. The robot is equipped with a 2D LiDAR sensor for environment perception and wheel encoders for odometry estimation. All sensor data are processed onboard using a Raspberry Pi 5.

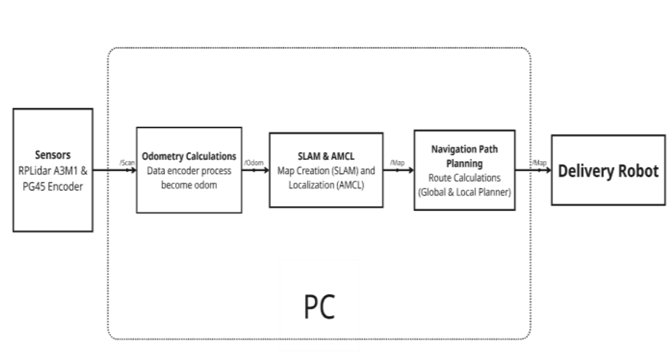

Figure 1. illustrates the overall system architecture of the robot, including sensor inputs, SLAM processing, localization, and navigation modules.

The software and hardware coordination is managed through a modular architecture. Figure 1 illustrates how the LiDAR A3M1 and PG45 Encoder provide raw data to the SLAM and AMCL modules. This data flow is critical for the Navigation Path Planning unit to send precise velocity commands back to the delivery robot platform

Component	Specification
Robot Type	Differential-drive Mobile Robot
Chassis Material	ACP and aluminium 6061
Dimensions (L x W x H)	53 cm x 50 cm x 40 cm
Total Weight	8 kg
Drive Motors	2x DC geared motors
Motor Rated Voltage	12V
Wheel Diameter	40 mm
Lidar Sensor	RPLidar A3M1
Onboard Computer	Computer
Operating System	Ubuntu 22.04 LTS
Robot Middleware	ROS 2 Humble Hawksbill

Tabel I Main hardware specifications of the robot platform.

The physical construction of the robot is designed to handle indoor delivery tasks. As detailed in Table I, the robot is a differential-drive platform with a total weight of 8 kg and dimensions of 53 cm x 50 cm x 40 cm. The processing power is driven by an onboard computer running Ubuntu 22.04 LTS and ROS 2 Humble to ensure real-time sensor data integration.

2.2 Sensor Configuration and Odometry

The sensor configuration on the introductory robot system is focused on precise hardware integration to support real-time environmental perception and accurate position calculations. The main perception instrument used is the RPLidar A3 2D LiDAR, a laser-based sensor configured to scan a 360-degree area around the robot to detect both static and dynamic obstacles. Technically, these sensors are set to operate at high sampling rates in order to publish laser scan data to topics /scans in the ROS 2 ecosystem continuously. The placement of the sensor is physically carried out at the highest coordinate point of the robot's chassis structure; This aims to minimize blind spots or blind areas that may be caused by the robot body itself, thus ensuring maximum detection range in the indoor environment of the Polibatam Main Building.

To support the localization system, the odometric configuration was constructed using data from a pair of PG45 drive motors that had been equipped with a hall-effect quadrature encoder. The robot's odometry is configured by integrating the physical parameters of the wheel, such as the wheel's radius of 0.04 m and the wheelbase, to convert each pulse (tick) generated by the encoder into a linear distance unit. This calculation process involves the mathematical transformation of the rotational movement of the motor into linear (v) and angular velocity (ω) estimates which are then published to the topic /odom. In accordance with the system architecture, this odometric data serves as a crucial initial kinematic input for the SLAM_Toolbox algorithm, which provides an estimate of the robot's movement before being further corrected by LiDAR scan data to produce precise positioning.

2.3 SLAM_Toolbox Configuration

The implementation of the mapping system on this robot uses the SLAM_Toolbox package integrated into the ROS 2 Humble ecosystem. In its configuration, the slam_toolbox node operates in synchronous mode and subscribes to the /scan topic from the LiDAR sensor and the /odom topic for wheel odometry data to perform simultaneous localization and mapping. The main parameters used in this study are summarized in Table II. As shown in the table, the map resolution is set at 0.05 m/pixel to ensure that the details of the main building environment are accurately captured on the occupancy grid map. The selection of the coordinate frame is also configured systematically, where the map functions as a global reference, odom as a local reference, and base_link as the physical center point of the robot.

The map building process began with a manual teleoperation phase to explore the entire ground floor area of the Main Building of the Batam State Polytechnic. During this phase, the robot was directed to traverse corridors, rooms, and intersections to ensure complete area coverage. The SLAM_Toolbox algorithm continuously updates the map and performs a loop closure mechanism to correct accumulated errors in odometry, resulting in a globally consistent map without significant distortion. Once a sufficient map is obtained, the data is stored in occupancy grid format for use by the autonomous navigation module in subsequent delivery tasks.

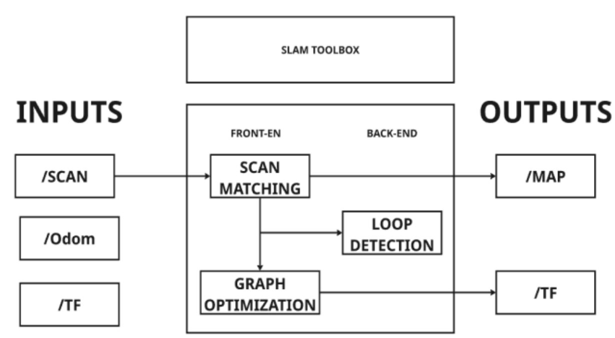

Figure 2. Algoritma Pemprosesan Data pada SLAM_Toolbox.

The implementation SLAM_Toolbox using a laptop as the main processing unit provides an advantage in data processing speed over the use of standard microcomputers. With higher laptop specifications, Back-end processes such as Graph Optimization and Loop Detection can run asynchronously at high frequencies without disrupting the stability of the data flow on the main navigation system. This ensures that the resulting maps are more precise and responsive to changes in the environment in real-time.

It also provides the flexibility for the system to run high-level visualizations simultaneously through RViz without overloading the core navigation process. Thus, the integration between *the performance of the laptop*'s hardware and *the algorithm SLAM_Toolbox* result in a system that is not only resilient in mapping, but also has better operational redundancy in the face of sudden sensor reading interruptions in the robot's work environment.

Parameter	Configuration/value	Description
Slam Mode	Synchronous	Real-time SLAM operation
Map Resolution	0.05 m/pixel	Occupancy grid map resolution
Laser Scan Topic	/scan	Input from 2D Lidar
Odometry Topic	/odom	Wheel odometry input
Odometry Frame	odom	Local Reference frame
Base Frame	Base_link	Robot base reference frame
Map Frame	Map	Global Reference Frame

Table II summarizes the main SLAM_Toolbox parameters used in this study

When configuring the software, the SLAM_Toolbox package was set with specific parameters to suit the conditions of the Polibatam Main Building environment. The details of these operational parameters are summarized in Table II, which includes synchronous mode settings and a map resolution of 0.05 m/pixel. Determining the frame of reference from the map to the base_link listed in Table II is crucial to ensure that the LiDAR sensor and wheel odometry data are synchronized correctly during the data collection process.

2.4 Mapping Procedure

The mapping process was conducted by manually teleoperating the robot throughout the ground floor of the Main Building at Batam State Polytechnic. The robot was guided to explore corridors, rooms, and intersections to ensure complete coverage of the environment. During this process, SLAM_Toolbox continuously updated the map and performed loop closure to correct accumulated odometry errors. Once a satisfactory map was obtained, the final occupancy grid map was saved and used for subsequent navigation experiments.

2.5 Autonomous Navigation Setup

After the environmental map has been successfully created and saved, the autonomous navigation system is activated using the ROS 2 Navigation Stack framework. The main component in this process is Adaptive Monte Carlo Localization (AMCL), which functions to estimate the robot's position relative to the existing occupancy grid map using data from LiDAR sensors and odometry. This position estimate is mathematically calculated based on the differential drive kinematic model to determine the robot's coordinates (x, y) and orientation (θ) as follows:

$$x_t + 1 = x_t + v_t \cos(\theta_t)\,\Delta t \quad (1)$$

This equation (1) is used to calculate the robot's latest x coordinate. The value of x_{t+1} is determined by the previous position (x_t) plus the result of projecting the linear velocity (v_t) onto the robot's orientation angle ($\cos\theta_t$) during a certain duration of time.

$$y_{t+1} = y_t + v_t \sin(\theta_t)\,\Delta t \quad (2)$$

This equation (2) is used to calculate the robot's latest y coordinate. The position y_{t+1} is obtained by projecting the robot's linear velocity onto the sine of its orientation angle ($\sin\theta_t$).

$$\theta_{t+1} = \theta_t + \omega_t\,\Delta t \quad (3)$$

This equation (3) is used to calculate the change in the robot's angle or orientation (θ). The latest angle (θ_{t+1}) is obtained from the previous angle plus the product of the angular velocity (ω_t) and the time interval (ωt).

The next stage involves determining several *navigation points* that represent the delivery location in the building. The robot is instructed to navigate autonomously from the starting position to each such destination point by utilizing global and local route planning. During the trip, the navigation system dynamically processes LiDAR data to detect and avoid static and dynamic obstacles to ensure the safety and success of the delivery task. The performance of this navigation is evaluated based on the stability of localization and the success of the robot in achieving the goal without collation.

III. RESULT AND DISCUSSION

3.1 Mapping Results

SLAM_Toolbox successfully generated a consistent occupancy grid map of the Main Building environment. Figure 3 shows the final map produced during the mapping phase. Visual inspection indicates that the map accurately represents the corridor layout and room boundaries. The loop closure mechanism effectively reduced accumulated odometry drift, resulting in a globally consistent map without noticeable distortions.

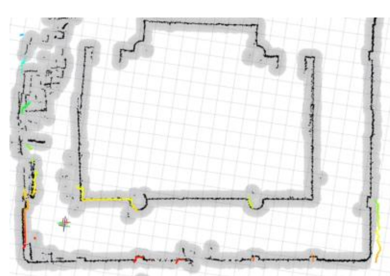

Figure 3. Occupancy grid map of the second floor of the Main Building generated using SLAM_Toolbox.

The final results of the mapping process conducted on the second floor of the Main Building of Batam State Polytechnic are shown in Figure 3. Based on the visualization in the figure, the system successfully identified the corridor structure and room boundaries very clearly. The sharpness of the lines in Figure 3 proves that the loop closure mechanism in SLAM_Toolbox works effectively in reducing odometry error accumulation (drift), resulting in a globally consistent map.

3.2 Localization Performance

Localization accuracy was evaluated by observing the robot's estimated pose relative to known reference points in the environment. The system achieved an average positional error of approximately 0.1 m, with a maximum observed error below 0.2 m.

Metric	Value
Average Position Error	0.10 m
Maximum Position Error	0.18 m
Localization Stability	Stable
Recovery after Disturbance	Succesful
Localization Method	AMCL

Table III presents the localization accuracy results obtained during multiple test runs. The localization remained stable even in the presence of moderate dynamic obstacles such as pedestrians.

A comprehensive evaluation of the robot's autonomous navigation capabilities was carried out by measuring the level of localization precision as the robot moved towards a predetermined target point. The results of these performance tests are presented in detail in Table III, which includes the positional error parameters in various trajectory scenarios. Based on the data recorded in Table III, the system showed excellent consistency with an average position error of 0.10 meters.

This achievement indicates that the integration between the RPLidar A3M1 LiDAR sensor and the Adaptive Monte Carlo Localization (AMCL) algorithm is able to provide rapid position correction against odometric drifts. In addition, the low maximum error value in Table III proves that the robot has sufficient stability to operate in the narrow corridors of the Polibatam Main Building without the risk of crashing into walls or surrounding objects

3.3 Autonomous Navigation and Delivery Performance

The autonomous navigation performance was evaluated by commanding the robot to move between predefined goal locations within the indoor environment. The navigation process utilized the ROS 2 Navigation Stack with AMCL for localization and global and local planners for path generation.

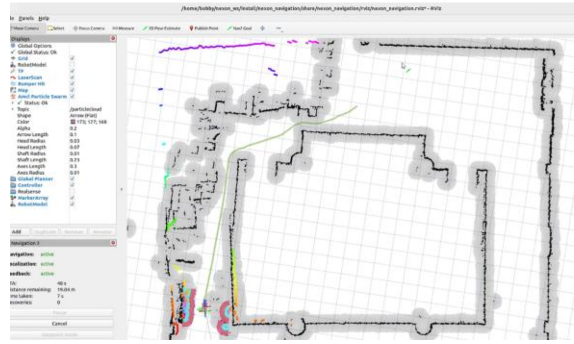

Figure 4. RViz visualization of the autonomous navigation process showing the global path, local trajectory, and robot pose during a delivery task.

Real-time monitoring of robot behavior during the execution of autonomous navigation tasks is visualized through the RViz interface as shown in Figure 4. The image is a representation of the navigation system decision-making process in the ROS 2 ecosystem, where the synchronous coordination between the Global Planner and the Local Planner can be seen. As illustrated in Figure 4, the robot manages to generate a barrier-free global path (marked with a green line) and execute the local trajectory smoothly as it travels through the indoor environment.

The alignment seen in Figure 4 between the laser scan data (red dots) and the wall boundary on the grid occupancy map shows a high level of trust (covariance) in the localization system. The estimated position of the robot remains stable throughout the navigation process, which is evidenced by the consistency between the robot's pose to the static map that has been created beforehand. This visual phenomenon confirms that the robot is not only capable of moving along the planned path, but also has the ability to re-localize independently in the event of an accumulation of errors in the odometry.

Although navigation is smooth, interruptions to the system mainly occur when dynamic obstacles temporarily block narrow corridors. In these conditions, the system dynamically reprocesses sensor data to replan the path to ensure operational safety. Thus, the visualization in Figure 4 validates that the entire implementation of the Navigation Stack has been running optimally to support the task of delivering goods autonomously at the Batam State Polytechnic Main Building.

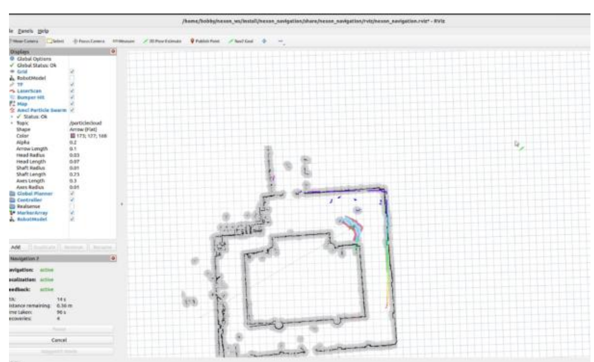

Figure 5. Real-time monitoring showing successful goal attainment, where the robot maintains stable AMCL localization and accurate map alignment upon finishing the trajectory.

Through the visualization of the RViz interface in Figure 5, it can be seen that the navigation system has successfully completed the delivery mission with an optimal level of precision. The robot is able to reach the specified target point with a final distance of only 0.36 meters remaining. This success confirms the effectiveness of the local planner in executing the final approach maneuver in the operational environment of the Batam State Polytechnic Main Building. The efficiency of the robot's movement in taking the path that has been prepared by the global planner is reflected in the total travel time of 96 seconds until the mission is declared complete.

On the way to the destination, the system recorded 4 recovery behaviors. This condition shows that the robot had faced technical challenges, such as navigation in narrow aisles or the presence of dynamic obstacles around the path, but the system was able to utilize the behavior tree algorithm to replan the path independently without the need for manual intervention from the operator. In addition, the stability of AMCL-based localization was monitored to be maintained in the active state, as evidenced by the alignment between the laser scan data and the wall boundary on the occupancy map until the robot stopped perfectly at the target location.

IV. CONCLUSION

This study successfully implemented and analyzed the SLAM_Toolbox and Nav2 framework in an autonomous indoor delivery robot system at the Main Building of Politeknik Negeri Batam. The experimental results demonstrate that SLAM_Toolbox is capable of generating a high-resolution occupancy grid map with consistent loop closure, providing a reliable foundation for indoor navigation tasks.

The autonomous navigation experiments, monitored through RViz, showed that the system could execute complex trajectories while maintaining stable AMCL-based localization. Specifically, the robot demonstrated high precision by reaching predefined goals with a final distance remaining of 0.36 meters and a total mission time of 96 seconds. Although navigation performance was tested in narrow corridors with environmental constraints—triggering four recovery behaviors—the system proved its resilience by autonomously replanning its path without human intervention, ensuring collision-free execution.

While this study confirms the viability of open-source ROS 2 modules for indoor delivery, it is limited to a single indoor environment. Future work will focus on optimizing local planner parameters to enhance navigation smoothness in higher pedestrian density areas and integrating multi-sensor fusion to further increase localization robustness in various structural settings.

ACKNOWLEDGMENT

The author would like to thanks to supervisor and all parties who have provide support during the implementation of this research. Appreciation was also expressed to the Robotics Engineering Study Program for the facilities and academic environment that supported the implementation of this research. Thanks are also given to colleagues who have contributed through discussions, technical assistance, and constructive suggestion during the system development process. This research is part of academic activities.

.

REFERENCES

[1] Simanjuntak, W. P. S., & Wibisana, A. (2024). Depth camera-based human detection using YOLOv5. In Proceedings of the 7th International Conference on Applied Engineering (ICAE 2024) (pp. 150–162).

[2] S. Macenski, T. Moore, D. Lu, A. Merzlyakov, and J. L. Fernandez, “SLAM Toolbox: SLAM for the dynamic world,” *Journal of Open Source Software*, vol. 6, no. 62, p. 2783, 2021.

[3] S. Macenski, F. Martín, R. White, and J. Ginés, “The Marathon 2: A navigation system,” in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2020, pp. 2718–2725.

[4] S. Macenski, M. Spenko, D. Lee, J. Behley, and S. Haeberli, “ROS 2: Design, architecture, and uses in robotics,” *Science Robotics*, vol. 7, no. 66, 2022.

[5] S. Macenski and T. Moore, “Navigation2: A ROS 2 navigation system,” *Journal of Open Source Software*, vol. 5, no. 54, p. 2783, 2020.

[6] Y. K. Tan and S. Huang, “Autonomous indoor mobile robot navigation using LiDAR,” *Sensors*, vol. 20, no. 14, p. 3938, 2020.

[7] A. J. Marconnot, S. Bhadani, and J. P. James, “Analysis of ROS 2 Communication Overhead for Real-Time Robotics,” *IEEE Real-Time Systems Symposium (RTSS)*, pp. 412-415, 2020.

[8] L. Rocha, J. Reis, and M. S. Couceiro, “A Review on SLAM and Navigation for Autonomous Mobile Robots in Indoor Environments,” *Applied Sciences*, vol. 11, no. 11, p. 5032, 2021.

[9] H. S. Kim and J. H. Park, “Optimization of LiDAR-based SLAM for Autonomous Mobile Robots in Dynamic Indoor Environments,” *Journal of Institute of Control, Robotics and Systems*, vol. 27, no. 4, pp. 250-257, 2021.

[10] M. G. Rivera and J. C. Hernandez, “Performance Evaluation of ROS 2 Humble Navigation Stack,” *Journal of Engineering Research*, vol. 9, no. 2, 2021.

[11] R. S. Gupta and P. Kumar, “A Survey of Simultaneous Localization and Mapping (SLAM) Algorithms for Indoor Navigation,” *Procedia Computer Science*, vol. 184, pp. 633-640, 2021.

[12] C. Guimarães, A. P. Moreira, and L. S. Costa, “Evaluation of ROS 2 Navigation Stack for Indoor Mobile Robots,” *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 154-159, 2020.

[13] M. S. Ayas and I. H. Altas, “Autonomous Navigation of a Mobile Robot using LiDAR and SLAM in ROS 2 Environment,” *International Conference on Electrical and Electronics Engineering (ELECO)*, pp. 240-244, 2021.

[14] T. Hiratsuka and K. Ohno, “Performance Evaluation of Local Planners in ROS 2 Navigation Stack for Narrow Corridors,” *Journal of Robotics and Mechatronics*, vol. 33, no. 4, pp. 812-821, 2021.

[15] R. S. Ferreira, J. Lima, and P. Costa, “Integration of RPLidar and Odometry for Robust Indoor SLAM using ROS 2,” *Sensors and Actuators A: Physical*, vol. 320, p. 112560, 2021.