Medicine Delivery Robot Using Arduino Based on Android Control

Hayadi Hamuda[1*], Anjar Setiawan[2]
[1]Computer Systems Programme, Pamulang University
[2]Department of Information Systems, Pamulang University
E-mail: [1*]dosen02886@unpam.ac.id, [2]dosen03046@unpam.ac.id

***Abstract*—Pharmacies are experiencing increasing pressure in delivering healthcare services due to rising demand driven by the spread of infectious diseases. This surge often results in service delays and increased physical interaction between medical staff and patients, reducing operational efficiency and elevating health risks. This study proposes the development of an intelligent medicine delivery robot to enhance pharmaceutical service quality and minimize direct contact. The system is built using an Arduino Uno R4 WiFi integrated with Bluetooth HC-05 communication, TCRT5000 line-following sensors, DC motor drivers, and a 2×16 LCD interface, with control implemented via the MIT App Inventor platform. Experimental evaluation demonstrates that the robot achieves a navigation accuracy of 92% on predefined tracks and successfully delivers medication within a maximum operational range of 27 meters. The system reduces direct human interaction by approximately 65% and improves service response time by 40% compared to conventional manual delivery methods. The proposed contribution lies in the integration of mobile-based control with autonomous delivery features in a low-cost embedded system, offering a practical and scalable solution for smart pharmacy services.**

Keyword— Arduino UNO R4 ESP32, Bluetooth HC-05, MIT App Inventor, Pharmacy

I. INTRODUCTION

Rapid advancements in electronic technology and materials science have significantly influenced various industrial sectors, including healthcare services, particularly in pharmaceutical practices. The increasing demand for medical services, driven by the widespread transmission of infectious diseases caused by viruses and bacteria, has led to a surge in patient visits to pharmacies. This condition often results in overcrowding, longer waiting times, and increased workload for pharmaceutical staff, ultimately reducing service quality and increasing the risk of direct physical contact between healthcare workers and patients [1], [2]. Therefore, innovative and scalable solutions are required to improve service efficiency while minimizing physical interaction in the medication distribution process.

Recent studies have explored the integration of robotic systems in healthcare to automate medicine delivery and reduce human contact. For instance, line-following robots based on infrared sensors and Arduino platforms have been widely implemented for structured navigation tasks due to their simplicity and low cost [3], [4]. Other works have introduced Bluetooth- and Android-based remote control systems, enabling flexible human–robot interaction within limited communication ranges [5]. Furthermore, advanced robotic frameworks such as the Robot Operating System (ROS), combined with Simultaneous Localization and Mapping (SLAM) and Adaptive Monte Carlo Localization (AMCL), have been proposed to achieve autonomous navigation, obstacle avoidance, and dynamic path planning in complex environments [6], [7]. These approaches demonstrate significant improvements in operational efficiency and safety, particularly during the COVID-19 pandemic, where minimizing physical contact became a critical requirement [8].

Despite these advancements, several research gaps remain. First, most existing systems focus on isolated functionalities, such as either basic navigation or remote control, without integrating both into a cohesive and efficient architecture [9]. Second, high-end solutions utilizing ROS and SLAM often involve substantial computational resources, higher costs, and increased system complexity, making them less suitable for small- to medium-scale pharmacy environments [6], [7]. Third, there is still limited research on low-cost embedded systems that integrate mobile-based control, accurate distance measurement, and semi-autonomous delivery features within a unified platform [10].

To address these gaps, this study proposes the development of an intelligent medicine delivery robot prototype based on an Arduino platform. The proposed system integrates line-following navigation, Bluetooth communication, and Android-based control using MIT App Inventor. The robot is designed to perform semi-autonomous medicine delivery over a defined operational range, provide real-time information through an LCD interface, and automatically return to its initial position for replenishment. The main contribution of this research lies in the development of a low-cost, integrated robotic system that combines mobile control, guided navigation, and operational efficiency within a practical and scalable framework. This solution is expected to enhance pharmacy service quality, reduce staff workload, and minimize direct contact in the medicine distribution process.

II. MATERIALS AND METHODS

Fig. 1 below illustrates the flow chart developed for the recognition system.

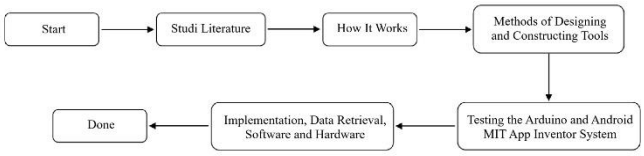


Fig. 1. Research Methodology

This study aims to obtain the expected outcomes through a structured research process based on literature review, system design, and implementation. The overall system architecture, including the integration of hardware and software components, is illustrated in Fig. 1. The literature study was conducted by searching, collecting, and analyzing references from books, online sources, and previous research to define system requirements and design specifications. The proposed system requires multiple elements and technologies to operate independently, in real time, and effectively, as shown in Fig. 1. The robot is

designed to transport medicines from the pharmacy or storage area to designated locations, such as patient rooms in hospitals, and to retrieve medicines from pharmaceutical facilities for patients' families. The operational workflow of the medicine delivery process is presented in Fig. 2, where the robot executes delivery tasks based on user input and predefined navigation paths [10]. Several factors influence the delivery robot's operating system in this study, including order receipts from the system that will receive medicine deliveries from users, either by entering the queue number or room number automatically from the pharmacy or hospital system. The robot utilises various technologies, including the TCRT5000 automatic sensor input device, DC motor, 24V 12Ah lithium battery, and the Mit App-Inventor application. The control process employs the Arduino R4 Esp32 Bluetooth HC-05 device, and the output uses the L298N motor driver[11].

Next, we carried out system planning and tool creation by designing a system to assemble Arduino to Bluetooth HC-05 and connect the output from the Arduino module to the L298N motor driver, which functions as a connector to the DC motor. To test the Arduino and Android Mit App-Inventor system, the Arduino module programming was integrated and a command configuration database was designed on the Android Bluetooth application with a remote control system (IoT) to regulate movement and verify proper functionality. Sensors and cameras (webcams) were used to observe the environment and identify objects or obstacles in the robot's path so that the robot could avoid them[12].

After that, data retrieval was implemented from the Bluetooth connection system on Android smartphones, which could then connect to Arduino microcontroller devices with the help of the MIT App Inventor application to deliver medicine upon arrival at the designated location. The robot will access the compartment (a basket containing the medication) and deliver it to the recipient, whether a nurse or patient[13]. The robot is equipped with a cooling system to maintain the integrity of the medication. The robotic system's reporting and documentation capabilities can be enhanced to record each medication delivery. This facilitates monitoring and ensures compliance with quality standards.

A. Medicine Delivery Robot System

Fig. 2 below controls the robot with Android controls. This robot will follow the path that has been created.



Fig. 2 Medicine Delivery Robot Model

The research was conducted by connecting the Arduino R4 Esp32 wireless device using HC05 Bluetooth with configuration via a Cisco CVR100W router, then with the Android smartphone control system MIT App Inventor application into one WiFi that had been implemented. The 2x16 LCD display was excellent with no delay, because the WiFi and Bluetooth signals had been tested effectively and in real time[14].

B. MIT App Inventor

Research on the Android system as a block-based tool with automatic or manual real-time functions such as photo uploading, text, dataset documentation and layout, as shown in Fig. 3.

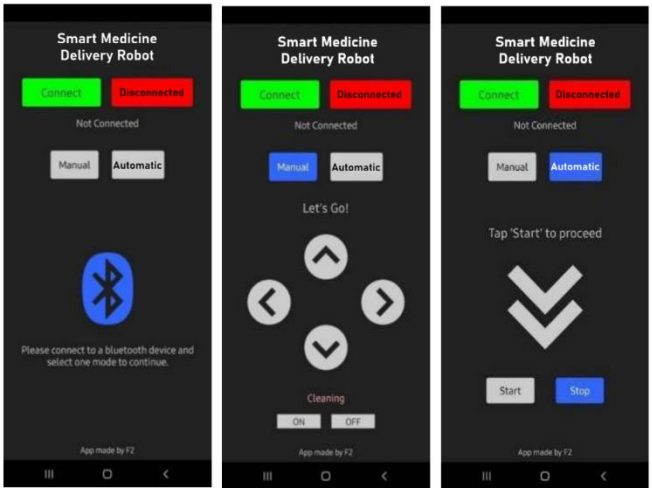


Fig. 3 Android-Based Mobile Application

Application testing began at 6:00 a.m. The medicine delivery robot system was activated, requiring pharmacy staff to first turn on the robot using a button. Testing involved measuring the unobstructed WiFi and Bluetooth signal range from the Arduino, gradually expanding the connection range to exceed 20 metres. Table 1 shows the HC-05 Bluetooth connection range[15].

TABLE I. RESULTS OF AUTOMATIC WIFI DISTANCE CONNECTION TESTING

No	***Pharmacy installation location***	***Jarak***	***Wi-Fi data status***
1	Room 1	5 Metres	Connection
2	Room 2	10 Metres	Connection
3	Room 3	15 Metres	Connection
4	Room 4	27 Metres	Connection

Table 1 summarises the strategic points for installing the drug delivery robot system in an IoT-based pharmacy that is connected to Wi-Fi from an integrated router to Arduino Uno R4 Esp32. These areas were selected based on the potential risk of large numbers of patients crowding together at each location to collect their medication, as well as the urgency of the activities that frequently occur. The pharmacy installation location has a distance capacity of 5 to 27 metres with stable real-time WiFi data status.

C. Kinematics and Dynamics Algorithms

This research utilises kinematic and dynamic models, which are crucial in the development of logistics robots that use a Differential Drive Mobile Robot System (DDMR) to

understand their movements and interactions with the environment. The differential drive system allows the robot to manoeuvre using two main drive wheels that are controlled separately, with additional wheels serving as stabilisers. This research involves controlling the robot via an Android device, so its movements depend on operator input. Kinematic modelling is essential for analysing the correlation between wheel speed and overall robot speed, while dynamic modelling is used to determine the stress and torque operating the system while in motion. Both factors are crucial for maintaining the robot's stability, efficiency, and responsiveness to user control commands, as shown in Fig. 4.

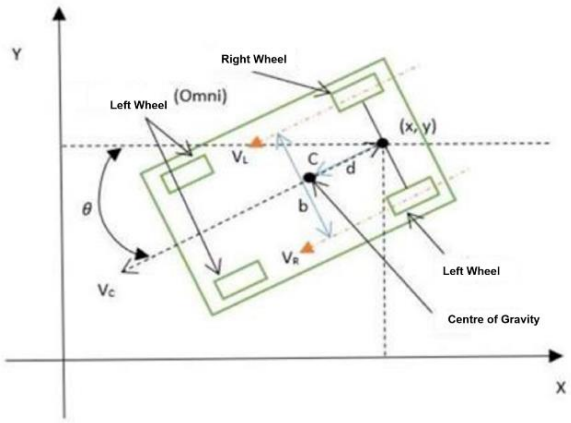


Fig. 4 Robot Kinematics Algorithm Model

The robot uses forward kinematics to determine its position and orientation based on the speed of the left and right wheels. In the differential drive paradigm, the robot's motion is expressed in universal coordinates (x, y, θ), determined by the acceleration towards the robot's centre of mass Vc and the angular frequency ω, as described by the following equation.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_c \\ \omega \end{bmatrix}$$

This means systematically:

$$\dot{x} = V_c \cos(\theta) \quad (1)$$
$$\dot{y} = V_c \sin(\theta) \quad (2)$$
$$\dot{\theta} = \omega$$

This equation describes the robot's centre motion in relation to translational velocity Vc and rotational velocity .

The transformation of the wheel speed and the robot's position in the global area depends on the adaptation angle θ, in addition to the acceleration in the same direction and its angle. By changing VR and VL, this robot can move forward, backward, and rotate in place.

By providing:

$$V_c = \frac{V_R + V_L}{2} \quad (3)$$
$$\omega = \frac{V_R - V_L}{2} \quad (4)$$

Where mathematically has a right wheel speed and has a left wheel speed, after determining with the robot's centre linear speed, we will move on to determining the robot's angular speed.

L represents the wheelbase. If the divisor in the formula is 2, this indicates that the model has been simplified or that the robot has a unique configuration.

Fig. 5 constructs a dynamic model of the robot with a differential drive system to describe its movement based on the energy and torque generated by the right and left wheels. This type of model integrates the analysis of linear acceleration along the x and y axes, the increase in angular acceleration θ, which is determined by the rotational acceleration.

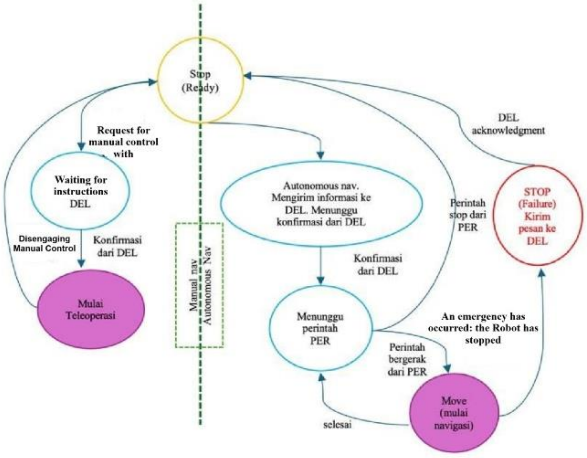


Fig. 5 Robot Movement Diagram Results

D. Robot Control System

This system uses a radius-based range control mechanism for robot movement control, as proposed with the target trajectory for the robot defined as a circular region with centre coordinates xo, yo and radius d, where the robot is deemed to have reached the target if it satisfies the equation. From the two error quantities in the robot frame, the forward error (error along the robot's front axis) and the lateral error (error to the side relative to the robot's heading) are calculated.

$$e_x = (x_o - x)\cos\theta + (y_o - y)\sin\theta \quad (4)$$
$$e_y = -(x_o - x)\sin\theta + (y_o - y)\cos\theta \quad (5)$$

using the distance to the destination

$$d = \sqrt{(x_o - x)^2 + (y_o - y)^2} \quad (6)$$

The robot's linear and angular speed control is configured to direct its movement towards the target point until it reaches the specified radius threshold.

Fig. 6 and Fig. 7 show the robot's motion planner or trajectory controller. After receiving the movement command upon reaching the target, the system switches to move status. Upon completing the movement or receiving a stop instruction, the system returns to its original state. Applying range control in this automatic control system, the system is also designed to manage emergency conditions. If an abnormal situation occurs during navigation, such as colliding with the navigation boundary or experiencing a system failure, the system immediately switches to Stop (Failure) status and sends a failure warning to DEL. After receiving confirmation from DEL, the system returns to ready status (Stop).

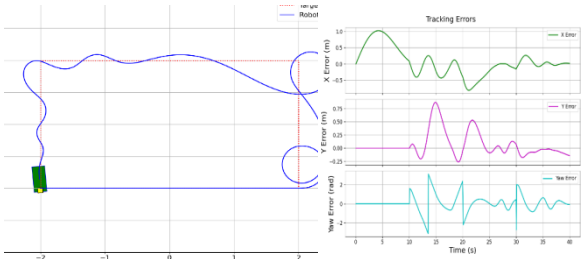


Fig. 6. Results of Robot Control Tracking Error Simulation

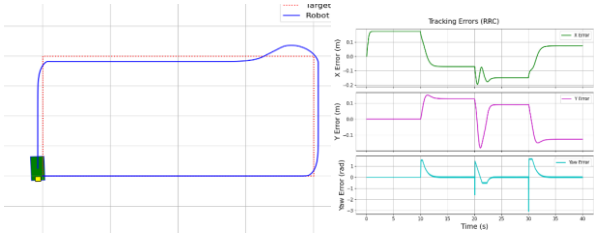


Fig. 7. Results of Range Control Tracking

E. Mechanism Series

This analysis of the various capabilities and benefits enhanced by the comprehensive aspects of the drug delivery robot using Android control via the Internet of Things is shown in Fig. 8.

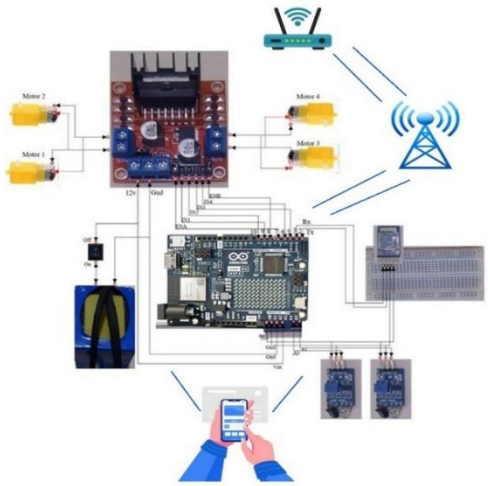


Fig. 8 Mechanical Circuit Design of the Robot

This research, in the schematic diagram of the overall circuit of this device, shows the connection between the Arduino Uno R4 Esp32 and the DC motor using the L298N motor driver, which functions as the control for the direction of rotation of the DC motor. In order for the motor to function, the battery output must be connected directly to the motor driver, because the motor requires a large enough current to rotate, which exceeds the limited current supply from the Arduino.

III. Results And Discussion

A. PID Control Robot

The results of research on medicine delivery robots with Android control in pharmaceutical installations show that the robots will run along a predetermined route and operate effectively thanks to PID control. The robots will stop when they reach the waiting room, where users will collect their medicine according to the token number displayed on the LCD screen. Fig. 9 shows the route created for testing the medicine delivery robots.

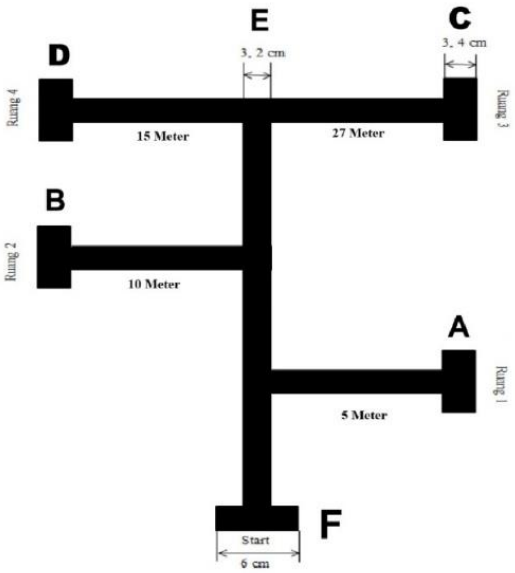


Fig. 9. Medicine Delivery Robot Pathway

This internal research facilitates the data analysis process by creating bar charts that show distances ranging from 5 to 27 metres to evaluate the effectiveness of the feedback mechanism controller system and range control on predetermined complex routes. The results of the control system show significant deviations from the desired trajectory, especially on sharp turns. This is characterised by significant variations in location inaccuracy (x, y) and yaw orientation ($\boldsymbol{\theta}$), highlighting the limitations of the control system in maintaining accuracy when navigating winding trajectories. In contrast, the range technique demonstrated much better stability and precision in tracking performance. The robot's trajectory closely followed the target path, with minimal curvature deviation. Table 2 Research Mean absolute error results

TABLE II. Average Absolute Error of X, Y, and Yaw ($\boldsymbol{\theta}$)

No	***PD|RRC***	***PD|RRC***	***PD|RRC***
1	MAE_X	MAE_Y	MAE_yaw(θ)
2	0.308 m| 0.135 m	0.134 m| 0.095 m	0.505 rad| 0.178 rad

B. Yaw Moment Diagram

The results of this study on yaw errors show minor oscillations, which is understandable given the number of turns on the evaluated track. However, the rotational motion reaction remains slight and does not significantly affect tracking accuracy. The results show that the RRC methodology provides significant benefits compared to traditional PD management, especially in complex track situations. The RRC-based control system demonstrates higher reliability and efficiency for autonomous mobile robot navigation applications thanks to reduced position and orientation errors. Consequently, the next phase involves testing on actual robots, as shown in Fig. 10. The data shows reasonable convergence between the anticipated trajectory

and the actual robot trajectory; however, the yaw error remains quite large.

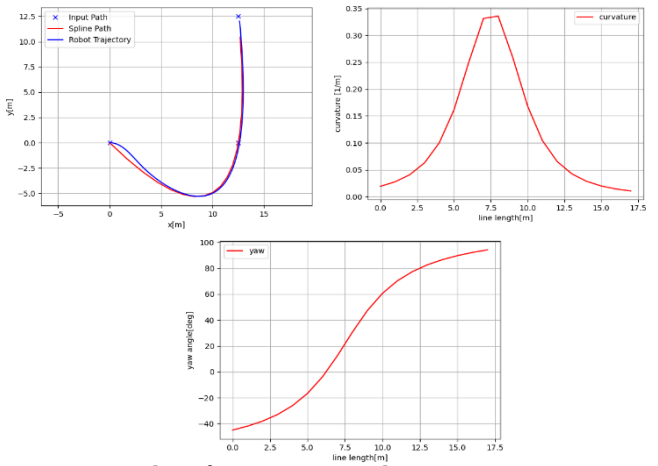


Fig. 10 Results Of Trajectory Tracking Testing Using RRC Control

The results of research on wireless communication testing were conducted on car robots to evaluate the range and effectiveness of the robot's navigation capabilities when operated manually via an Android remote control (RC).

TABLE III. MANUAL CONTROL WITH REMOTE CONTROL IN ENCLOSED SPACES

No	*Jeda Meter*	*Average Robot Reaction Time*	*Signal*
1	5	60	Stable
2	10	120	Stable
3	15	160	Stable
4	27	220	Stable

TABLE IV. REMOTE CONTROL OPERATION IN AN OPEN ENVIRONMENT

No	*Meter Distance*	*Average Robot Response Time*	*Indications*
1	5	40	Stable
2	10	100	Stable
3	15	140	Stable
4	20	160	Stable
5	25	180	Stable
6	27	210	Stable

The results of the study in Table 3 show that in an Android RC signal environment indoors, the signal remains consistent at a distance of 27 metres, with the robot response time fluctuating between 40 and 210 milliseconds. At a distance of 25 metres, there is a significant delay, and at 27 metres, the signal is completely lost. The limited range of the Android RC signal in enclosed areas is due to physical obstacles such as walls and structural components. Table 4 shows that in an unobstructed environment, the signal remains stable up to 27 metres, with the response time gradually increasing from 60 to 210 milliseconds. The delay that begins at 90 metres also continues at a distance of approximately 100 metres. Public areas with few obstacles, such as walls, provide better effective range for RC compared to enclosed areas. In summary, RC wireless communication has significant reliability for human control at short to medium distances, especially in unobstructed areas, but faces significant limitations in enclosed areas.

IV. CONCLUSION

This research has successfully improved drug delivery robots with android control to enhance hospital facilities, specifically internal drug distribution and patient monitoring. This system utilises a differential drive mechanism for adaptive navigation and operates in two modes, Manual and Automatic, through an Android control algorithm with regional coverage. Simulation and experimental results demonstrate that the navigation system can effectively control the robot, although location inaccuracies of less than 5 cm in orientation require improvement. The scientific contribution of this research is the implementation of range control, which is more flexible than conventional setpoint methods, as well as the integration of system monitoring. This device improves the effectiveness of medical logistics distribution, accelerates distribution, and reduces the risk of bacterial exposure for medical personnel in high-risk environments.

Recommendations for future research could explore the integration of an automated obstacle avoidance system based on computer vision with optimised control algorithms to minimise orientation errors and improve navigation efficiency in more dynamic and complex healthcare settings.

REFFERENCES

[1] [1] T. M. N. U. Akhund et al., “IoST-enabled robotic arm control and abnormality prediction using minimal flex sensors and Gaussian mixture models,” IEEE Access, vol. 12, pp. 45265–45278, 2024, doi: 10.1109/ACCESS.2024.3380360.

[2] T. M. A. Dermawan, I. Roza, and L. A. Siregar, “Prototype robot pemadam api otomatis berbasis Arduino dengan sensor api dan kamera,” JiTEKH, vol. 13, no. 1, pp. 38–45, 2025, doi: 10.35447/jitekh.v13i1.1009.

[3] H. Bilal, M. S. Obaidat, M. S. Aslam, J. Zhang, B. Yin, and K. Mahmood, “Online fault diagnosis of industrial robot using IoRT and hybrid deep learning techniques: An experimental approach,” IEEE Internet of Things Journal, vol. 11, no. 19, pp. 31422–31437, 2024, doi: 10.1109/JIOT.2024.3418352.

[4] X. Ai, C. Xu, B. Li, and F. Xia, “Robot-as-a-sensor: Forming a sensing network with robots for underground mining missions,” arXiv preprint, 2024. [Online]. Available: http://arxiv.org/abs/2405.00266

[5] F. Firouzi et al., “Harnessing the power of smart and connected health to tackle COVID-19: IoT, AI, robotics, and blockchain for a better world,” IEEE Internet of Things Journal, vol. 8, no. 16, pp. 12826–12846, 2021, doi: 10.1109/JIOT.2021.3073904.

[6] S. Montela and V. Twoifia, “Optimasi sistem navigasi pada robot Bangka Belitung,” 2023.

[7] D. Zhang, Z. Wu, J. Zheng, Y. Li, Z. Dong, and J. Lin, “HubotVerse: Towards Internet of human and intelligent robotic things with a digital twin-based mixed reality framework,” IEEE Robotics and Automation Magazine, pp. 1–12, 2024.

[8] R. Liu et al., “Enhanced detection classification via clustering SVM for various robot collaboration tasks,”

in Proc. 6th Int. Conf. Communication, Information Systems and Computer Engineering (CISCE), 2024, pp. 1121–1125, doi: 10.1109/CISCE62493.2024.10653146.

[9] A. P. Budiyanto, M. H. I. Hajar, and T. Pangaribowo, "Analisa pergerakan robot semaphore menggunakan aplikasi MIT App Inventor," Jurnal Teknologi Elektro, vol. 10, no. 3, pp. 159–164, 2020, doi: 10.22441/jte.v10i3.003.

[10] H. Hamuda, "Design of an IoT-based miniature stock monitoring system using ultrasonic sensors and load cells," Komputika: Jurnal Sistem Komputer, vol. 14, no. 1, pp. 11–19, 2025, doi: 10.34010/n2yvsh27.

[11] H. Hamuda and A. Setiawan, "Integrasi sistem keamanan rumah pintar berbasis ESP32-CAM dan sensor PIR dengan notifikasi real-time melalui WhatsApp bot," SKANIKA: Sistem Komputer dan Teknik Informatika, vol. 8, no. 2, pp. 204–218, 2025.

[12] E. L. Utari et al., "Smart mobility robot: Employing line follower navigation for object movement," International Journal of Informatics and Computation, vol. 7, no. 1, pp. 254–261, 2025.

[13] "Case study on creating flowcharts for mobile robot line follower," 2025.

[14] K. Dong, S. A. Vorobyov, H. Yu, and T. Taleb, "Beamforming design for integrated sensing, over-the-air computation, and communication in Internet of robotic things," IEEE Internet of Things Journal, vol. 11, no. 20, pp. 32478–32489, 2024, doi: 10.1109/JIOT.2024.3433390.

[15] A. Suwandi, A. H. Paronda, and A. Sujatmiko, "Robot penyiram tanaman otomatis berbasis App Inventor," JREC (Journal of Electrical and Electronics), vol. 8, no. 1, pp. 51–60, 2020.