Journal of Deep Learning, Computer Vision and Digital Image Processing
E-ISSN: 2986-8939 P-ISSN: 2986-8920
Vol. 4, No. 2, June 2026
DOI. https://doi.org/10.61255/decoding.v4i2.1318

An Intelligent IoT-Based Waste Bin System Utilizing Nearest Neighbor Algorithms for Optimized Waste Collection Routes

Muhammad Irfan Trinugroho
(Universitas Ahmad Dahlan, DI Yogyakarta, Indonesia)

Dinan Yulianto✉
(Universitas Ahmad Dahlan, DI Yogyakarta, Indonesia)

ARTICLE HISTORY
Received: April, 30 2026
Revised: May, 20 2026
Accepted: June, 18 2026

KEYWORDS
Digital twin;
Internet of things;
Nearest neighbor algorithm;
Route optimization;
Smart waste bin

ABSTRACT

Purpose – Despite advances in IoT-enabled waste monitoring, existing solutions generally fail to integrate real-time bin status information with adaptive route optimization, resulting in inefficient collection operations. This study aims to design and implement an integrated system that leverages real-time waste data to facilitate intelligent, data-driven route optimization for improved waste collection operations.

Methods –This study presents an ESP32-based smart waste system using reed switch event-driven control and deep-sleep mode for energy efficiency. Waste levels were estimated using the arithmetic mean fusion of four VL53L0X sensors. A cloud-based MQTT-over-TLS architecture enables secure real-time communication, whereas a priority-based nearest-neighbor routing algorithm is evaluated across 150 nodes.

Findings – The results demonstrate that the proposed system provides accurate waste-level estimation with a mean error of 1.98%, significantly reduces energy consumption by 90.9% through deep-sleep operation, and supports near-real-time communication with an average latency of 4.66 s. Moreover, the priority-based route optimization strategy decreased the travel distance by 42.7%, ensured the immediate servicing of all full-status bins, and maintained operational feasibility within a fleet capacity of 2,700 L.

Research implications – The evaluation results demonstrate the feasibility of integrating real-time monitoring and adaptive route optimization for smart waste management. Future research should extend the validation to large-scale real-world deployments and incorporate road network-based routing models to enhance operational realism and optimization accuracy.

Originality – This study proposes an integrated smart waste platform that combines energy-efficient event-driven sensing, dynamic priority-based nearest-neighbor routing, and hardware-assisted digital twin validation for scalable and cost-effective waste management evaluation.

Correspondence Author: ✉dinan.yulianto@tif.uad.ac.id

To cite this article: Trinugroho, M. I. & Yulianto, D. (2026). An Intelligent IoT-Based Waste Bin System Utilizing Nearest Neighbor Algorithms for Optimized Waste Collection Routes. *Journal of Deep Learning, Computer Vision and Digital Image Processing, 4*(2), 19-42. https://doi.org/10.61255/decoding.v4i2.1318

INTRODUCTION

Waste management is one of Indonesia's foremost infrastructure challenges, driven by rapid population growth and intensifying consumption patterns [1]. The effectiveness of waste management systems is shaped by a confluence of factors, including population density and spatial distribution, socioeconomic conditions, environmental characteristics, and community behavioral patterns [2]. According to data published by the Sistem Informasi Pengelolaan Sampah Nasional (SIPSN) in 2025, national waste generation reached approximately 34 million tons annually, most of which is still inadequately managed [3].

Analogous challenges are evident in Daerah Istimewa Yogyakarta (DIY). In 2024, the Department of Environment and Forestry (Dinas Lingkungan Hidup dan Kehutanan/DLHK) recorded a total daily waste generation of 1,923.67 tons in the DIY region [4]. Sleman Regency is the largest contributor, generating 601.79 tons of waste per day [4]. The SIPSN data for DIY further identified an annual waste accumulation of 219,653.64 tons in Sleman Regency, with waste reduction and waste handling rates of 13.88% and 10.17%, respectively, resulting in a total managed waste proportion of 24.04% [5]. These empirical figures indicate that approximately 75% of the generated waste remains inadequately managed, reflecting a substantial gap between waste generation volumes and existing management capacity. This condition may significantly increase the operational burden and accelerate waste accumulation.

Community-based waste processing facilities operating under the Reduce-Reuse-Recycle framework (TPS3R) serve a critical function in alleviating operational pressures, particularly in the upstream stages of waste collection and preliminary processing [6]. One actively operational TPS3R serving the Sleman Regency is TPS3R Sendangtirto, which is responsible for an extensive and geographically heterogeneous service area encompassing waste collection, transportation, and sorting operations.

Field interviews with operational personnel at TPS3R Sendangtirto indicated that waste collection activities are predominantly conducted using conventional practices that lack support from data-driven decision-making. As reported by the TPS3R Operational Officer (*personal communication*) [7], waste collection personnel must physically inspect each waste bin location to determine the level of waste accumulation, resulting in the absence of a systematic prioritization process based on the actual collection demand. This operational approach contributes to inefficient collection routes owing to the lack of planning based on real-time field conditions and may lead to delayed servicing of high-volume waste bins because resources are allocated without accurate information regarding bin fill levels. Furthermore, as reported by the TPS3R Operations Supervisor (*personal communication*) [7], the absence of real-time monitoring and data recording causes scheduling and operational decisions to rely heavily on individual experiences. Consequently, this practice increases operational inefficiencies, including unnecessary expenditures of time, labor and fuel resources.

A systematic review of the existing literature reveals a diverse range of research efforts aimed at improving the efficiency of waste collection and management systems via the integration of Internet of Things (IoT) technologies with routing optimization algorithms. Suryaningrat et al. [8] developed an IoT-based smart waste bin system utilizing Firebase for real-time volumetric fill-level monitoring. Although the system demonstrated improvements in monitoring efficiency, it did not incorporate waste-collection route optimization capabilities. Firmansyah et al. [9] proposed an IoT-based monitoring system augmented with additional environmental parameters, namely, humidity and ammonia gas concentration. While generating more comprehensive environmental datasets, the system remained limited to monitoring functions and did not incorporate mechanisms for managing the distribution of collection activities. Ismail et al. [10] developed an Arduino-based smart waste bin with a web interface for displaying waste volume data. Although effective at a local scale, the system lacks cloud integration and broader operational optimization support.

Verryando et al. [11] developed an IoT-based smart waste bin system that utilized ultrasonic sensors integrated with the Telegram application to enable real-time monitoring of bin conditions and automated notification delivery. Although the system enhanced monitoring efficiency and reduced manual user intervention, it remained focused solely on monitoring functionalities and did not support waste collection route optimization. Widodo et al. [12] developed an IoT-based smart waste bin system utilizing ultrasonic sensors and WeMos D1 Mini microcontrollers, capable of detecting waste fill levels and automatically transmitting notifications to field personnel. The system improved monitoring efficiency and minimized the requirement for manual field inspections; however, its functionality remained limited to notification delivery without incorporating route optimization algorithms or centralized management across multiple nodes.

Further examination of the literature identified the application of specific algorithms to establish optimal route configurations. Chaerul et al. [13] utilized the nearest-neighbor algorithm for waste collection route optimization, demonstrating measurable improvements in travel distance efficiency. Yanti et al. [14] similarly demonstrated that the Nearest Neighbor method effectively enhanced scheduling effectiveness while reducing travel distance, operational time, and associated costs. However, both studies relied on static datasets that could not accommodate real-time waste volume conditions.

Maturi et al. [15] developed an IoT-enabled smart waste bin platform for real-time waste monitoring and collection management within smart-city ecosystems. Alnanih et al. [16] incorporated software engineering principles into an IoT-based waste management framework to enhance its scalability and long-term maintainability. Barth et al. [17] proposed a digital twin–driven decision support approach for optimizing waste collection based on bin fill-level thresholds. These studies collectively highlight the growing importance of integrating real-time sensing technologies into waste collection optimization. However, existing solutions predominantly emphasize monitoring functionalities or employ static routing strategies, leaving a gap in adaptive and demand-driven collection optimization, which motivates the present research.

Based on the synthesis of prior research, it can be concluded that existing IoT-based monitoring systems are predominantly oriented toward data acquisition without supporting operational decision-making, while route optimization methods continue to depend on static data and have yet to leverage real-time data. Accordingly, a research gap emerges due to the lack of an integrated platform that combines real-time waste bin condition monitoring with dynamic collection route optimization. The integration of IoT technology with the nearest-neighbor algorithm offers a promising solution to address this gap, as it enables continuous data acquisition and real-time transmission from field-deployed devices. In the context of waste management, this technological integration facilitates ongoing fill-level monitoring and real-time data relay to a central system, thereby supporting more accurate operational decision-making and enhancing overall collection efficiency [18].

Wi-Fi was selected as the communication medium to support real-time data transmission because of its availability within the TPS3R Sendangtirto service area and its ability to provide reliable connectivity without additional infrastructure costs. Given the event-driven transmission scheme, Wi-Fi offers sufficient energy efficiency while providing a higher data throughput and flexible system configuration [19]. Although LPWAN technologies remain suitable for wider-area deployments, Wi-Fi is considered the most practical solution for the current operational context.

In response to the identified operational problem and research gap, this study presents the design and development of an IoT-based smart waste bin system that integrates real-time monitoring with route optimization through the implementation of the nearest-neighbor algorithm. The system was engineered using the ESP32 microcontroller, incorporating an event-driven mechanism and deep sleep mode for energy efficiency, alongside a multi-sensor approach to improve data accuracy. The system evaluation was conducted using a digital twin simulator to assess the performance scalability under large-scale operational scenarios. The proposed system is expected to serve as a viable solution for enhancing the operational efficiency of TPS3R Sendangtirto and advancing data-driven waste management practices.

This study introduces a system development framework that encompasses three distinct contributions. First, the integration of a reed-switch-based event-driven mechanism combined with a deep-sleep mode significantly reduced the energy consumption of the device while maintaining timely data transmission. Second, the implementation of the nearest-neighbor algorithm, enhanced by a dynamic prioritization mechanism based on fill-level status and fleet capacity constraints, enables real-time collection route optimization that responds to the actual field conditions. Third, a two-layer validation approach that combines physical hardware testing for sensor accuracy validation and hardware performance assessment with digital twin simulator-based testing for infrastructure scalability validation and algorithm performance evaluation under multi-node conditions, thereby ensuring a comprehensive system evaluation with optimized cost and implementation efficiency.

METHOD

This study employs the Waterfall model within the Software Development Life Cycle framework, encompassing the requirements analysis, system design, implementation, testing, and maintenance phases, where each phase is completed sequentially before proceeding to the next stage [20], [21]. The Waterfall model was selected because the system requirements—derived from a fixed fleet capacity (2,700 L), predefined fill-level classification thresholds, and stable findings obtained from structured field interviews at TPS3R Sendangtirto—were well understood and stable from the outset, and were therefore unlikely to change during development. These clearly defined and low-volatility requirements allow each development phase to be completed and validated before proceeding to the next, providing a structured and traceable development process suited to the scope of the system.

Requirements Analysis

Field observations and structured interviews with TPS3R Sendangtirto personnel [7] revealed that waste collection operations were carried out using a three-wheeled vehicle (Viar) with a 2,700-liter capacity. The collection process was conducted using conventional practices without real-time data support and without any location-based prioritization mechanism. Route determination is based on the experience of individual workers, which leads to increased travel distance, extended operational time, and higher fuel consumption. Accordingly, the proposed system must integrate real-time monitoring with a fill-level–driven decision-making mechanism that prioritizes full bins, followed by medium bins, thereby adopting a demand-driven waste collection strategy. The functional and non-functional requirements are summarized in Table 1.

Table 1. Functional and Non-Functional System Requirements

Code	Requirement	Description
FR-01	Fill Measurement	The fill level was measured using four VL53L0X sensors (error < 10%).
FR-02	Power Management	The device is activated from sleep mode via a reed switch.
FR-03	Data Transmission	JSON data are sent via the MQTT protocol.
FR-04	Route Optimization	Determine the route using the Nearest Neighbor; priority FULL > MEDIUM with fleet capacity.
FR-05	Telegram Monitoring	The monitoring and control were provided via the Telegram Bot.
NFR-01	Reliability	The data delivery success rate was ≥ 90%.
NFR-02	Latency	The end-to-end response time was ≤ 30 s.
NFR-03	Scalability	Supports new unit additions without firmware recompilation.

System Architecture Design

The system implements a four-layer IoT architecture [22], consisting of a sensing layer (ESP32 with WiFiManager, four VL53L0X sensors, and a reed switch), a network layer (EMQX Serverless MQTT broker and Firebase Hosting), a processing layer (Backend Worker deployed on Google Cloud Run, Cloud Firestore, and a digital twin simulator), and an application layer (Telegram Bot). The application layer provides information through automated push notifications triggered by Full or Low battery conditions, as well as via Telegram commands, including /AddBin, /Status, /CekAlert, and /OptimizeRoute. The resulting system architecture is shown in Figure 1.

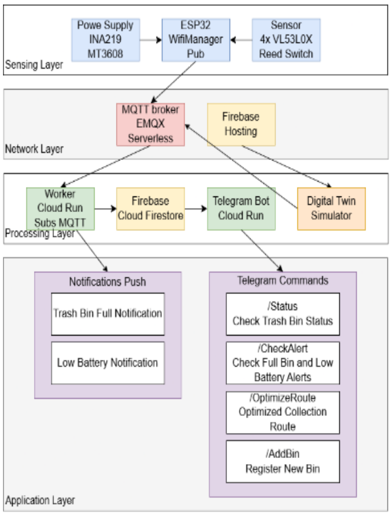

Figure 1. System Architecture

Hardware Integration Design

The control and sensing subsystem utilizes the ESP32 as the central processing unit [23], integrated with four VL53L0X sensors [24] interfaced through a TCA9548A I2C Multiplexer to prevent address conflicts [25]. The power subsystem utilizes an 18650 lithium-ion battery with a BMS [26], [27], real-time monitoring via an INA219 [28], and voltage stabilization through an MT3608 boost converter [29]. A reed switch is implemented to trigger the automatic deep-sleep mode when the bin lid is fully closed [30]. The hardware architecture and wiring diagrams are shown in Figures 2 and 3, respectively.

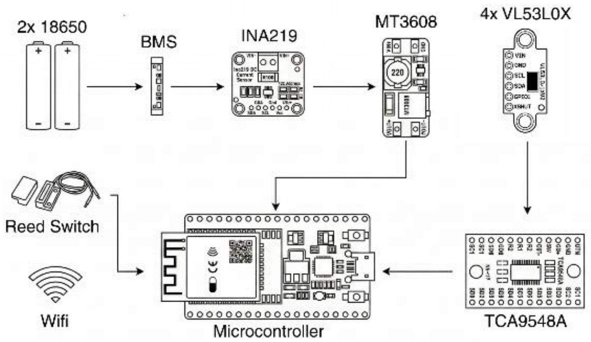

Figure 2. Hardware Architecture Diagram

The hardware integration design was further formalized by developing a wiring diagram that defines the physical pin-level interconnections between all system components. This diagram functions as an assembly reference to reduce the likelihood of short-circuit faults and data transmission errors, as shown in Figure 3.

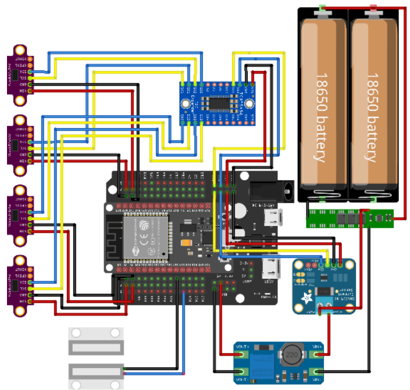

Figure 3. Hardware Wiring Diagram

Software Design

The software ecosystem comprises five integrated components: ESP32 Firmware, EMQX Serverless MQTT Broker, Backend Worker, Telegram Bot, and web-based IoT Simulator. The firmware is responsible for acquiring sensor measurements, performing arithmetic mean–based sensor fusion, classifying the bin fill-level status, and transmitting structured data in JSON format via MQTT secured with TLS. The sensor output of the fill-level classification maps to four status categories, as defined in Table 2.

Table 2. Waste Bin Fill Status Classification

Status	Fill Percentage
EMPTY	0%
NORMAL	1% - 49%
MEDIUM	50% - 79%
FULL	80% - 100%

The Backend Worker is responsible for validating incoming data, persisting records in Firestore, and executing route optimization algorithms. The IoT Simulator functions as a digital twin [31], facilitating scalable testing and evaluation without relying on physical hardware constraints.

Data Processing and Route Optimization

Sensor fusion was performed by computing the arithmetic mean of the four VL53L0X measurements to mitigate individual sensor noise [32], [33], expressed as

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

Where $\overline{\boldsymbol{x}}$ is the mean value, $\boldsymbol{x_i}$ is the i-th measurement, and $\boldsymbol{n}$ is the number of data points.

Arithmetic-mean fusion was selected instead of weighted averaging and Kalman filtering because of the homogeneous nature of the four VL53L0X sensors, which exhibit comparable noise characteristics and measure the same quasi-static parameter, namely the surface distance, during each wake-up interval. In such a sensing environment, the advantages of temporal state estimation are limited because the measured variable remains relatively stable within the observation period. Consequently, equal-weight averaging provides an effective approximation of the maximum likelihood estimator while maintaining low computational complexity and memory consumption. This property is particularly advantageous for energy-constrained and event-driven firmware architectures implemented on the ESP32 platform. Therefore, more sophisticated approaches, such as weighted fusion or recursive estimation methods, are considered potential enhancements for applications involving heterogeneous sensors or highly dynamic measurement conditions.

Route optimization applies the nearest-neighbor algorithm to solve the Traveling Salesman Problem [13], while inter-node distances are determined using the Haversine formula [34]:

$$d = 2R.arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos(\varphi1)\cos(\varphi2)\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right) \tag{2}$$

Where $\boldsymbol{d}$ is the distance between two points (km), $\boldsymbol{R}$ is the Earth's radius (≈6371 km), $\boldsymbol{\varphi1}$ and $\boldsymbol{\varphi2}$ are the geographic latitudes of the first and second points, respectively, and $\Delta\boldsymbol{\varphi}$ and $\Delta\boldsymbol{\lambda}$ represent the differences in latitude and longitude between the points, respectively.

The nearest-neighbor heuristic was selected because of its low computational cost. With a worst-case time complexity of $O(n^2)$, it remains highly efficient at the scale of a single TPS3R, which comprises tens to hundreds of nodes. For n = 150, the algorithm computes a complete route in less than one second on the Cloud Run backend, a duration that is negligible relative to the overall latency budget. However, at a city-wide scale involving several thousand nodes, its quadratic growth would justify the metaheuristic and spatial-indexing approaches that are identified as directions for future work.

Digital Twin Fidelity Validation

To assess the fidelity of the simulator, sensor measurements recorded from the physical device were replayed on a corresponding simulated node, and the resulting outputs were evaluated based on three externally observable aspects: (i) fill-level status classification (Empty, Normal, Medium, and Full), (ii) JSON payload structure, including key names and data types, and (iii) the calculated fill-level percentage. A total of N=10 paired physical–simulated trials were conducted, covering all status categories and boundary conditions near the 49–50% and 78–80% classification thresholds. The simulated node achieved complete consistency with the physical device in terms of status classification, correctly reproducing all classifications across the ten trials. In addition, the generated payloads exhibit identical key sets and numeric field types. The average deviation in the computed fill-level value was 0.24 percentage points (pp), with a maximum deviation of 0.65 pp. This discrepancy was attributable solely to the integer rounding applied during data storage and remained substantially lower than the measured sensing error of the physical device (1.98%). Moreover, the observed deviation did not affect the resulting fill-level classification in any of the trials.

These findings indicate that, under the conditions tested, the digital twin reproduces the externally observable behavior of the physical system with high fidelity, providing a sound basis for the scalability assessment presented in this study. However, it should be emphasized that these ten paired trials constitute initial fidelity evidence obtained under controlled conditions only. They do not validate long-term deployment behavior, performance under heterogeneous or degraded network conditions, the effects of outdoor environmental exposure such as temperature, humidity, dust, and sensor drift, or inter-device variability across a multi-node-field fleet. Therefore, the reported fidelity should be interpreted as a controlled-condition baseline, and comprehensive validation through extended multi-device field deployment remains an objective for future studies. The detailed validation results are listed in Table 3.

Table 3. Digital twin fidelity validation across N = 10 paired physical–simulated trials

Trial -N	Sensors (cm)	Device Fill/Status	Twin Fill/Status	Δ fill (pp)
1	41, 40, 47, 42	15.05% / Normal	15.00% / Normal	0.05
2	4, 5, 5, 4	90.90% / Full	91.00% / Full	0.10
3	50, 50, 50, 50	0.00% / Empty	0.00% / Empty	0.00
4	10, 10, 12, 11	78.20% / Medium	78.50% / Medium	0.30
5	35, 35, 39, 36	28.15% / Normal	27.50% / Normal	0.65
6	15, 14, 16, 14	70.70% / Medium	70.50% / Medium	0.20

7	21, 20, 25, 20	56.75% / Medium	57.00% / Medium	0.25
8	3, 4, 6, 3	92.00% / Full	92.00% / Full	0.00
9	38, 38, 42, 38	21.50% / Normal	22.00% / Normal	0.50
10	25, 25, 27, 25	49.30% / Normal	49.00% / Normal	0.30

A consolidated summary of the digital twin fidelity assessment across the four validation dimensions is presented in **Table 4**.

Table 4. Summary of digital twin fidelity across four validation dimensions

Fidelity dimension	Result across N = 10 paired trials
Fill-level status classification (Empty / Normal / Medium / Full)	100% agreement (10/10), including the empty edge case (0%) and near-threshold cases at 49–50% and 78–80%
JSON payload schema (keys & field types)	Identical key set {bin_id, sensors[4], avg, fill_level, battery, timestamp}
Computed fill-level value	Mean deviation Δ = 0.24 pp (max 0.65 pp), attributable to integer rounding only
Transmission success rate	100% (both)

Digital twin fidelity validation using N=10 paired physical–simulated trials. The simulated node reproduced the physical device's fill-level classification with 100% agreement across all status categories, including the near-threshold conditions. The average fill-level deviation was 0.24 pp (maximum 0.65 pp), attributable solely to integer rounding, supporting its use as a controlled-condition baseline for evaluating scalability.

System Workflow

The workflow of the system is illustrated in Figure 4. Upon activation by the reed switch during bin lid opening, the ESP32 initializes the Wi-Fi and MQTT connections, sequentially acquires sensor measurements through the TCA9548A multiplexer, processes the collected data according to Equation (1), determines the fill-level classification, and transmits the resulting JSON payload to the EMQX Serverless broker via TLS-secured MQTT communication. After successful transmission, the ESP32 returned to deep-sleep mode to minimize power consumption. Subsequently, the Backend Worker, acting as an MQTT subscriber, validates the incoming data, stores the processed records in Cloud Firestore, and automatically issues Telegram notifications when a full or low-battery condition is detected.

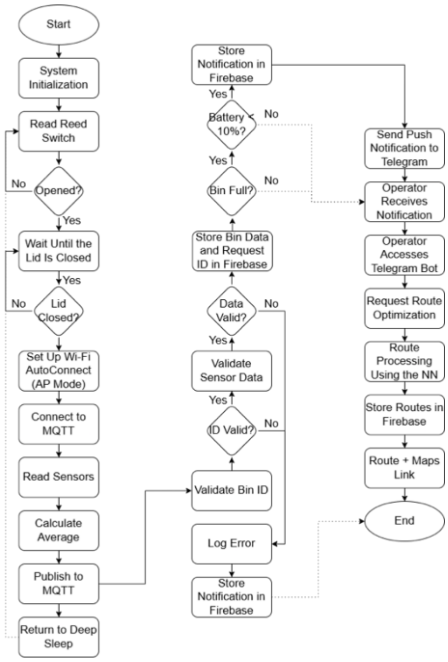

Figure 4. System Flowchart

Evaluation and Testing

The system performance was evaluated using a two-layer testing approach covering functional and non-functional requirements. The sensor accuracy was assessed by computing the mean absolute percentage error against the reference measurements at five height points, with an acceptance threshold of less than 10%. The power efficiency was evaluated by measuring the current consumption under three operating conditions. Transmission reliability was assessed as the success rate across 150 messages with a minimum threshold of 90%. Route optimization was evaluated through a comparative analysis of travel distance, critical node service ordering, and fleet capacity adherence between priority-based and baseline approaches. The end-to-end latency was characterized across 30 iterations with a 30-second acceptance threshold. Scalability was verified through the registration of new units without firmware recompilation.

Scope & Limitations

This study had several methodological limitations. Hardware validation was conducted on a single physical prototype in a controlled environment that may not fully represent outdoor deployment variability. Large-scale testing relies on digital twin simulations rather than full physical deployment. Route distance computation employs the Haversine geodesic distance rather than actual road network data, which may introduce discrepancies under irregular road infrastructure conditions.

RESULTS AND DISCUSSION

Results

Hardware Integration Implementation

The hardware implementation was performed by integrating all the system components into a single unified smart waste bin unit. The ESP32 DOIT DevKit V1 was employed as the central processing unit to concurrently manage the sensor communication and network connectivity. Four VL53L0X sensors were integrated via a TCA9548A I2C multiplexer to perform distance measurements between the bin lid and waste surface at four spatially distributed points, thereby yielding more representative volumetric data. An INA219 module was incorporated for real-time battery voltage monitoring, and a reed switch was positioned on the bin lid to serve as the system activation trigger. The complete hardware integration is shown in Figure 5.

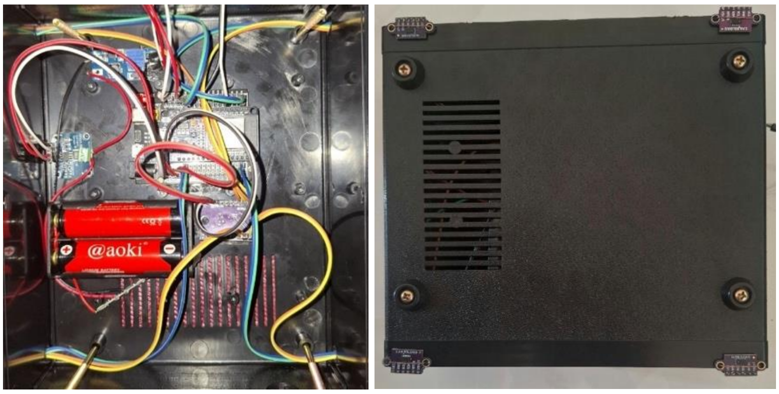

Figure 5. Hardware Integration

The system was implemented on a 50-liter waste bin with physical dimensions of 330 × 320 × 500 mm. All four sensors were installed on the inner side of the bin lid in a 2×2 grid arrangement, providing four spatially distributed measurement points across the entire cross-sectional area. This multipoint configuration was adopted to address the inherently nonuniform distribution and lateral accumulation of waste, where multisensor averaging provides a more representative estimation of the fill level than single-point sensing. In addition, each sensor was positioned at the center of its respective sensing region to reduce the effects of specular reflection from the bin walls, thereby ensuring improved measurement accuracy. The sensor implementation in the waste bin is illustrated in Figure 6.

Figure 6. Hardware Implementation on the Waste Bin

Software Implementation

The software comprises five functionally integrated modules. The ESP32 firmware was implemented using a reed-switch-based event-driven architecture, in which the device was activated only when the bin lid was opened, executed sensing and data transmission tasks, and subsequently returned to sleep mode to minimize energy consumption. The measurements from the four sensors were processed according to Equation (1) and categorized into four fill-level states: Empty, Normal, Medium, and Full. The processed data were transmitted in JSON format via the MQTT protocol with TLS encryption to the EMQX Serverless broker. The firmware Serial Monitor output is presented in Figure 7, and the EMQX dashboard interface is presented in Figure 8.

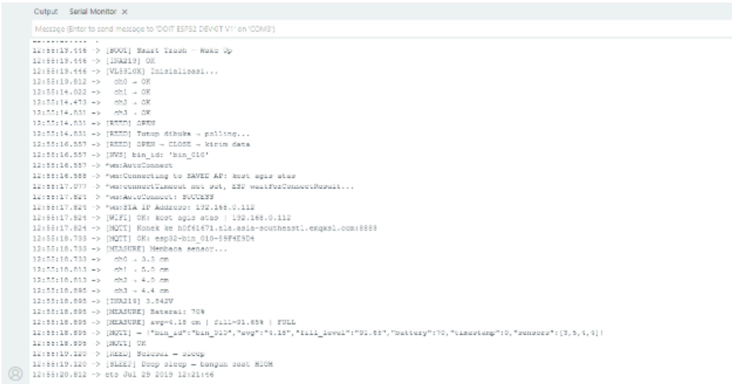

Figure 7. Firmware Serial Monitor Output

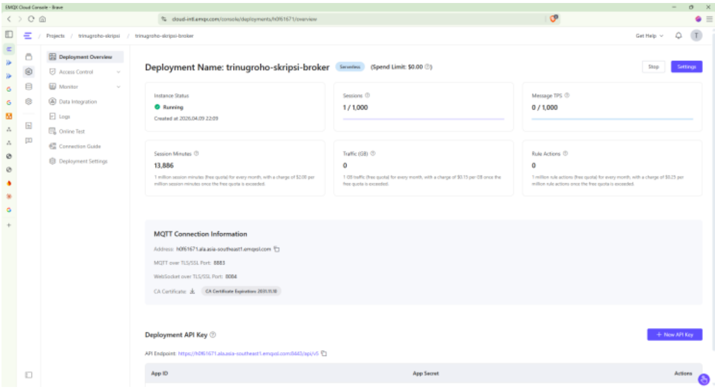

Figure 8. EMQX Dashboard Interface

The backend worker-sub service and Telegram Bot worker-bot service were deployed as two independent services on the Google Cloud Run platform in the asia-southeast2 region to minimize communication latency with IoT devices. The architectural separation of these two services was also intended to preserve operational independence, ensuring that updates or maintenance activities on one service do not affect the availability of the other service.

Both services were configured with a resource allocation of 512 MiB memory and 1,000 mCPU each. The backend service operates under a single-instance policy (minimum: 1, maximum: 1) with the CPU always-allocated mode to maintain persistent MQTT connection integrity and to prevent duplicate message processing. The Telegram Bot was configured with a horizontal scaling mechanism (minimum: 1, maximum: 3 instances) to dynamically accommodate fluctuating user request volumes. Both services implement TCP-based startup probes on port 8080 with a timeout of 300 s to facilitate the early detection of initialization failures. The configurations of both services are illustrated in Figure 9.

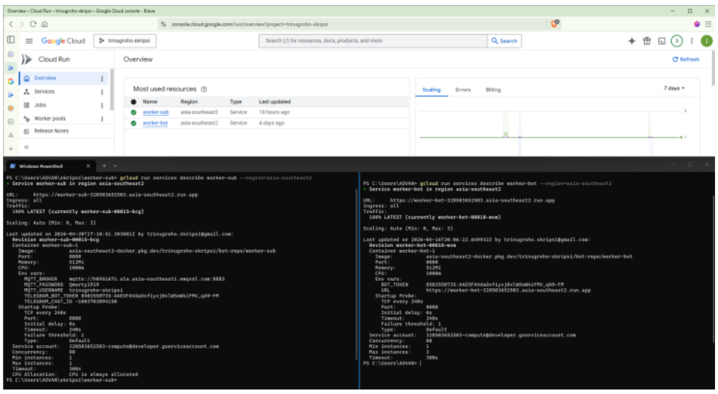

Figure 9. Backend Worker and Telegram Bot Configuration

The backend worker processes all incoming data received from the broker. Incoming data undergo a validation stage prior to persistence in Firestore, after which push notifications are automatically dispatched via Telegram upon detection of a full waste bin condition or a battery level falling below the predefined threshold. The Telegram Bot was developed as the primary operational interface for TPS3R Sendangtirto field personnel and provides four core features: Status Bin for monitoring current waste bin conditions, Check Alert for identifying critical status conditions, Route Optimization for determining collection sequences using the Nearest Neighbor algorithm integrated with Google Maps navigation links, and Add Bin for registering new units via an authentication mechanism. The push notification interface and Telegram Bot display are shown in Figure 10.

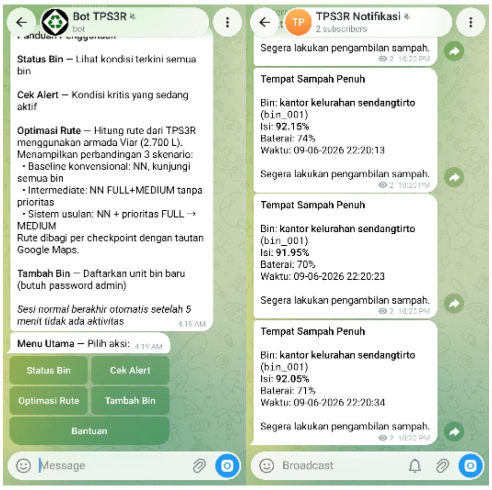

Figure 10. Push Notification and Telegram Bot Interface

The IoT Simulator was implemented as a web-based application connected to the broker via WebSocket Secure, enabling comprehensive end-to-end system validation without the need for extensive physical device deployment. The simulator supports two operational modes, namely Manual Input and Random Simulation, which facilitate controlled evaluation under a wide range of operational scenarios. In addition, the simulator serves as a lightweight digital twin that emulates the behavioral characteristics of physical IoT devices. This approach allows performance assessment, including route optimization experiments and large-scale deployment testing, without the limitations imposed by physical hardware, thereby enabling more cost- and time-efficient validation prior to the real-world implementation. The simulator interface is shown in Figure 11.

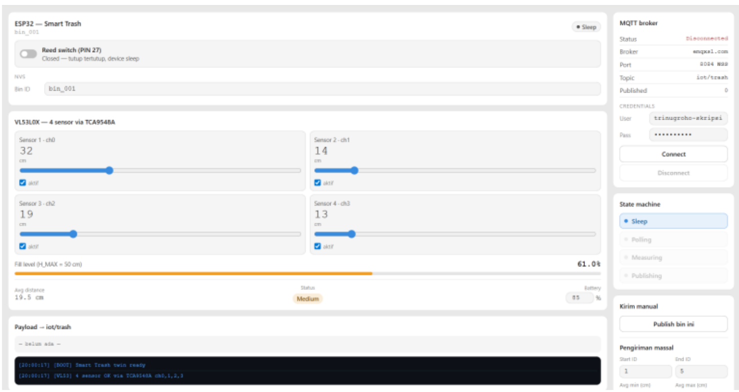

Figure 11. IoT Simulator Interface

Hardware and Software Testing

Sensor accuracy testing (FR-01) was conducted by comparing the mean readings from the four VL53L0X sensors against the reference values at five height measurement points. The test results are listed in Table 5.

Table 5. Per-point VL53L0X distance accuracy against ground truth

Reference (cm)	S1	S2	S3	S4	Mean (cm)	Error (cm)	Error (%)
10	10	10	10	9	9.75	0.25	2.5
20	21	22	20	20	20.75	0.75	3.75
30	30	32	31	30	30.75	0.75	2.5
40	40	39	40	40	39.75	0.25	0.63
50	49	50	50	52	50.25	0.25	0.5
Overall Mean (All Points)						0.45	1.98

As presented in Table 5, the mean sensor error across all five reference points was 1.98. The highest individual error of 3.75% was recorded at the 20 cm reference point, whereas the lowest error of 0.50% was observed at the 50 cm reference point. The sensor fusion approach, implemented through the arithmetic averaging of four spatially distributed measurement points, demonstrated measurable effectiveness in suppressing individual sensor deviation, consistent with the findings of [32] that the combination of homogeneous sensors consistently yields higher accuracy than a single-sensor configuration. The overall mean error of 1.98% falls substantially below the 10% tolerance threshold specified in FR-01, confirming that the system is suitable for deployment in fill-level detection under actual operational conditions.

Power consumption testing (FR-02) was conducted to verify the effectiveness of the deep-sleep mechanism based on a reed switch. The current consumption was measured using a digital multimeter under three operating conditions representative of the complete operational cycle of the device. The test results are listed in Table 6.

Table 6. Current Consumption Test Results

Operating Condition	Current (mA)
Deep Sleep Mode (lid closed)	± 20
Standby Mode (lid opened, waiting state)	± 100
Active Mode (connecting, measuring, and transmitting data)	± 220

The reduction in current consumption from approximately 220 mA in the active state to approximately 20 mA in deep-sleep mode represents an energy efficiency improvement of 90.9%. The observed deep-sleep current, which exceeded the theoretical specification of the ESP32 microcontroller (approximately 10 μA), was attributable to the residual power draw from external peripheral components that remained partially energized, including the VL53L0X sensors, MT3608 boost regulator circuit, and INA219 module. This behavior constitutes a well-recognized trade-off in battery-operated IoT systems that incorporate always-on peripheral architectures [26]. The implementation of the reed-switch-based event-driven mechanism allows the device to operate in the active state (approximately 220 mA) only for brief intervals within each collection cycle, thereby significantly minimizing the aggregate daily energy consumption and supporting sustained battery-based operation. The results of the deep sleep test are shown in Figure 12.

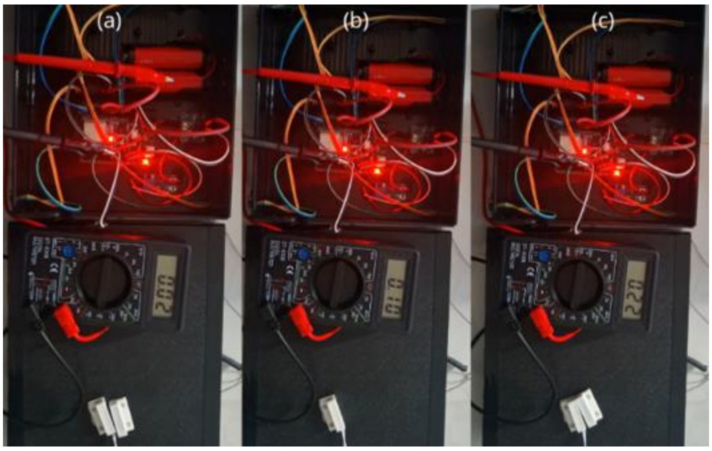

Figure 12. Deep Sleep Mechanism Test Results

To identify the origin of the residual current consumption during the deep-sleep operation, the nominal low-power current of each subsystem was obtained from the respective manufacturers' datasheets, as summarized in **Table 7**. The power consumption values reported for individual components were derived from manufacturer datasheets and therefore represent estimated rather than independently measured quantities. Direct experimental measurements were conducted only at the node level, where the overall current consumption during deep-sleep operation was approximately 20 mA.

Table 7. Deep sleep power budget estimated from datasheets, reconciled with the measured node level current of ≈20 mA

Component	State	Typical Current
ESP32 SoC [35]	Deep sleep (RTC timer)	≈ 10 μA
VL53L0X ×4 [24]	SW standby	≈ 6 μA each (≈ 24 μA total)
TCA9548A I^2C multiplexer [25]	Standby	< 0.1 mA
INA219 current monitor [36]	Active monitoring	≈ 1 mA
MT3608 boost converter [29]	No-load quiescent	≈ 1 mA
AMS1117-3.3 LDO, on-board [37]	Quiescent	≈ 5 mA (up to 11 mA)
On-board power-indicator LED	Continuously on	≈ 3–7 mA (estimated)

In the deep-sleep state, the ESP32 system-on-chip is the only component set to a fully suspended mode, exhibiting a current draw of approximately 10 μA while retaining the RTC timer functionality. Consequently, the remaining current consumption was primarily attributed to the power management and interface circuitry integrated into the development board, rather than the sensing subsystem itself.

Aggregating the current consumption values reported in the respective datasheets resulted in a baseline range of approximately 12–20 mA, which was consistent with the directly measured aggregate deep-sleep current of approximately 20 mA. This agreement indicates that the residual consumption, presented here as a power budget estimated from datasheets, is dominated by board-level power components, such as the AMS1117 linear regulator, MT3608 boost converter, and on-board power-indicator LED.

User-Perspective System Testing

System testing from the user perspective was conducted to verify that all end-to-end functional workflows operated in accordance with the defined specifications. Testing was performed exclusively through the Telegram Bot interface as the primary interaction medium, ensuring that the evaluation was conducted without exposing or requiring comprehension of the internal system implementation details.

Data transmission testing (FR-03) was performed by dispatching 150 messages originating from one physical device and 149 simulator units. Verification was conducted for successful data reception, data persistence in Firestore, and automatic push notification delivery via Telegram upon the detection of a full waste bin. The test results demonstrated that all messages were processed successfully, yielding a transmission success rate of 100%, which simultaneously satisfies the NFR-01 requirement mandating a minimum reliability threshold of 90%. Cloud Run log outputs are presented in Figure 13, and automatic push notification behavior is illustrated in Figure 14.

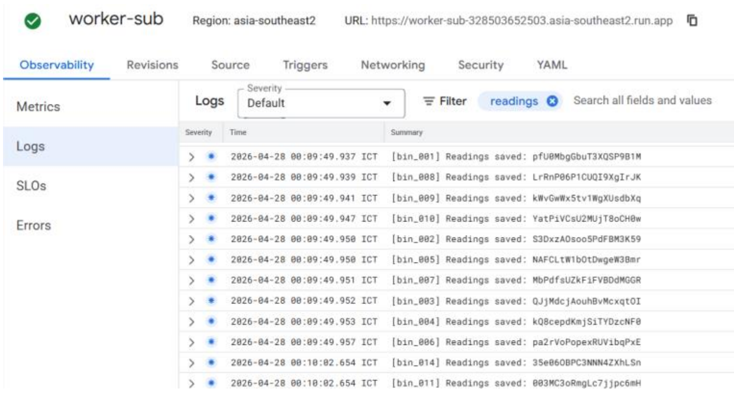

Figure 13. Cloud Run Log Output

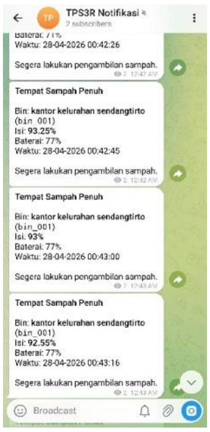

Figure 14. Automatic Telegram Push Notification

Route optimization testing (FR-04) was conducted to verify the system's capability to generate collection sequences that prioritize waste bins with full status while simultaneously optimizing route efficiency. The test involved 150 waste bin units, comprising one physical device and 149 simulator units, with coordinates randomly distributed across the TPS3R Sendangtirto service area. The system performance was evaluated through a comparative analysis of two approaches: a baseline method utilizing the nearest-neighbor algorithm without a priority mechanism, and the nearest-neighbor algorithm augmented with the dynamic priority mechanism. The route optimization test results are presented in Table 8.

Table 8. Route Optimization Test Results

Metric	Baseline (NN, all bins)	NN on FULL+MEDIUM (no priority)	Proposed (priority-enhanced)
Total Waste Bins	150	150	150

Bins Visited	150	113 (43 FULL + 70 MEDIUM)	67 (43 FULL + 24 MEDIUM)
FULL Bins Collected	43/43 (not prioritized)	43/43 (not prioritized)	43/43 (100%, prioritized)
MEDIUM Bins Collected	70/70 (not prioritized)	70/70 (not prioritized)	24 (remaining vehicle capacity)
Estimated Load Volume	± 5,047 L (exceeds vehicle capacity)	± 4,194 L (exceeds capacity)	± 2,689 L (within vehicle capacity)
Total Travel Distance	± 11.15 km	± 10.00 km	± 6.39 km
Number of Checkpoints	-	-	8
Distance Reduction	-	± 1.15 km (10.3%)	± 4.76 km (42.7%)

Table 8 compares the three routing strategies under an identical 150-node configuration, with only the routing policy varying. The conventional baseline, which visits all bins regardless of the fill level, produces the longest route (≈11.15 km) and a total load of ≈5,047 L, exceeding the 2,700 L vehicle capacity and thus is infeasible in a single trip. Condition-based filtering alone (servicing only FULL and MEDIUM bins without prioritization) yields a minor distance reduction to ≈10.00 km (10.3%), but still results in an excessive load of ≈4,194 L. In contrast, the proposed full-first, capacity-constrained approach services all 43 FULL bins and 24 selected MEDIUM bins (67 bins total, ≈2,689 L), reducing the route length to ≈6.39 km. This corresponds to reductions of 42.7% and 36.1% relative to the baseline and intermediate strategies, respectively. These results indicate that routing feasibility is achieved only through the joint application of prioritization and capacity-aware constraints, whereas condition filtering alone is insufficient.

These findings are consistent with those reported by [13] and [14], who demonstrated the effectiveness of the nearest-neighbor algorithm in reducing waste collection travel distances. This study extends the contributions of previous research by introducing a dynamic priority mechanism based on real-time data that was not addressed in earlier studies. The route optimization test interface via the Telegram Bot is shown in Figure 15.

Figure 15. Route Optimization Testing via Telegram Bot

Furthermore, the system generates a segmented route composed of a sequence of checkpoints, each enhanced with navigation hyperlinks, thereby improving the operational usability for field personnel during waste collection activities. These results indicate that the system not only satisfies the functional requirement FR-04 by prioritizing waste bins with a full status but also achieves notable improvements in route efficiency and fleet capacity utilization compared with the baseline approach. Functional testing of the Telegram Bot (FR-05) was conducted by executing all available features through the bot interface and assessing the response conformity across six predefined test scenarios. Telegram Bot functional test results are presented in Table 9.

Table 9. Functional test outcomes for the six Telegram Bot interaction scenarios

Feature	Scenario	Result
Status	Access the Bin Status menu	Displays the current status of all waste bins
Check Alert	Access the alert menu when at least one bin is FULL	Displays waste bins in critical condition
Optimize Route	Execute route optimization	Displays the optimized route summary and checkpoint links
Add Bin	Enter bin data with the correct password	Bin is successfully registered in Firestore without requiring firmware recompilation
Add Bin	Enter bin data with an incorrect password	Access is denied and the user is returned to the main menu
Push Notification	FULL bin and LOW battery conditions triggered by the physical device	Notification is successfully received in the Telegram channel

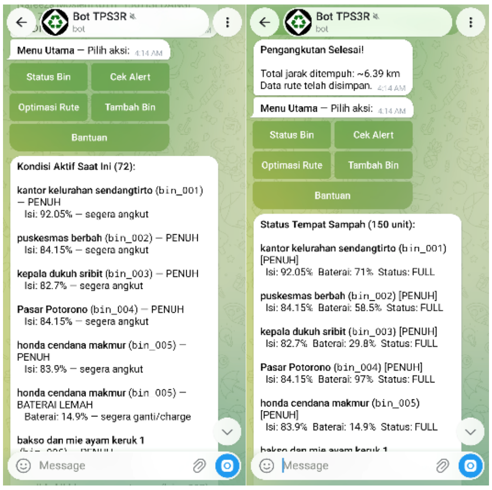

Figure 16. Telegram Bot Functionality Testing

End-to-end latency (NFR-02) was defined as the elapsed time from the lid-closure event, detected by the reed switch that wakes the ESP32 from deep sleep, to the delivery of the corresponding Telegram notification, thereby capturing the complete operational path (deep sleep wake-up, multi-sensor acquisition, Wi-Fi reconnection, MQTT-over-TLS publication, Cloud Run processing, Firestore persistence, and Telegram push). Measurements were taken over 30 consecutive steady-state cycles; a single first-cycle Wi-Fi provisioning event (captive-portal fallback, ~34 s) was excluded as a one-time configuration outlier that was not representative of the deployed operation. The results are shown in Figure 17.

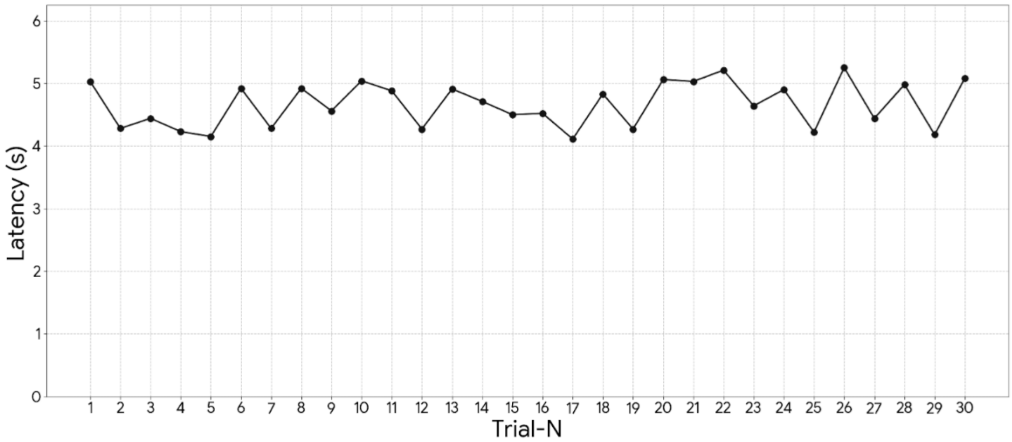

Figure 17. Latency Test Results

Across 30 consecutive steady-state test iterations, full end-to-end latency—measured from the lid-closure event (reed-switch activation) until delivery of the Telegram notification—ranged from 4.1 to 5.3 s, with a mean of 4.66 s (SD = 0.36 s) and a 95th-percentile latency of 5.2 s. This interval encompasses the complete operational pipeline: wake from deep sleep, sensor acquisition, Wi-Fi reconnection, MQTT over TLS publish, Cloud Run processing, Firestore persistence, and Telegram delivery. Reporting the 95th percentile characterizes the worst-case operational behavior, which is more relevant to reliability than the mean alone. Even at this percentile, the latency remains far below the 30 s threshold specified in NFR-02 (an 83% safety margin, rising to approximately 84% at the mean).

Scalability testing (NFR-03) was conducted using two mechanisms to verify that the registration of new waste-bin units did not necessitate firmware recompilation. The first mechanism involved registering a new waste bin using the Add Bin feature in the Telegram Bot by inputting the unit ID, location name, and geographic coordinates. The second mechanism involved configuring the bin ID on a physical device through the WiFiManager captive portal without any modification to the firmware source code. Both mechanisms successfully registered new waste bin units, with the resulting data processed by the system immediately upon registration, demonstrating that the system possesses adequate scalability for node expansion without the need for firmware recompilation. The scalability-testing interface is shown in Figure 18.

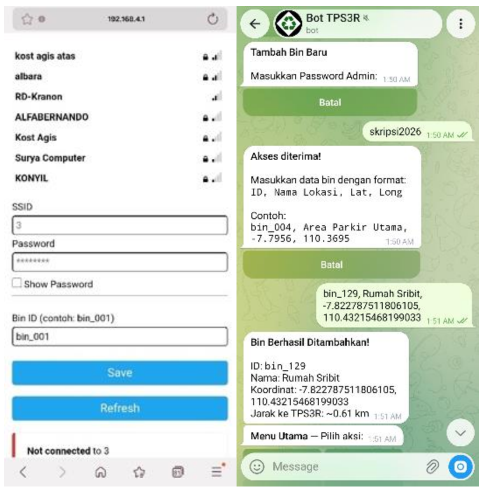

Figure 18. Scalability Testing for New Waste Bin Registration

A comprehensive summary of all functional and nonfunctional test results is presented in Table 10.

Table 10. Functional and Non-Functional Testing Results

Code	Requirement	Testing Method	Result
FR-01	Fill Measurement	Accuracy testing at five reference points using a measuring instrument	Mean error of 1.98%; satisfies the specified requirement of < 10%
FR-02	Power Management	Current consumption measurement using INA219 under three operating conditions	Power consumption reduced by approximately 90%, from ±220 mA (active mode) to ±20 mA (deep sleep mode)
FR-03	Data Transmission	Transmission of 150 messages (1 physical device and 149 simulated devices)	100% of messages were successfully stored and notifications were delivered
FR-04	Route Optimization	Evaluation using 150 waste bins (1 physical device and 149 simulated devices) with comparison against a baseline scenario	All FULL bins were collected; travel distance reduced by 42.7% compared to the baseline
FR-05	Telegram Monitoring	Functional testing of six Telegram Bot scenarios	All features responded according to the specified requirements
NFR-01	Reliability	Evaluation based on FR-03 results	Success rate of 100%, exceeding the minimum requirement of 90%
NFR-02	Latency	30 end-to-end latency measurements from physical device to notification delivery	Average latency of 4.66 s (range 4.1–5.3 s; 95th-percentile 5.2 s), satisfying the ≤ 30-second requirement
NFR-03	Scalability	Addition of new waste bins through Telegram Bot and WiFiManager	Successfully achieved without requiring firmware recompilation

Discussion

A review of prior studies indicates that the proposed platform is distinguished by its measured sensing accuracy (mean error of 1.98%), low average end-to-end latency (4.66 s), event-driven energy reduction of 90.9%, and, most importantly, a 42.7% reduction in travel distance achieved through a capacity-aware and priority-enhanced nearest-neighbor optimization. In contrast, existing systems primarily focus on SMS/GPS-based monitoring and notification frameworks [15], multi-platform management with emergency detection capabilities [16], or threshold-based collection scheduling [17], and generally do not report comparable quantitative metrics in terms of sensing accuracy, latency,andr measured route optimization. These differences highlight the integrative and empirically characterized nature of the proposed work, which combines low-cost sensing, secure cloud communication, and capacity-aware priority routing on a single validated platform. A summary of the related studies is presented in Table 11.

Table 11. Comparison with internationally published smart waste systems.

System	Sensing Accuracy	Communication	Latency	Routing / collection optimization
This Study	1.98% mean error (4× VL53L0X,	Wi-Fi / MQTT-over-TLS	4.66 s (SD = 0.36 s, 95th percentile = 5.2 s)	42.7% travel-distance reduction (capacity-

	arithmetic-mean fusion)			aware, priority-enhanced NN)
Maturi et al. [15]	HC-SR04 ultrasonic; accuracy NR	GSM/SMS (SIM900A) + GPS (NEO-6M)	Not Reported	SMS with GPS coordinates / Google Maps link; no quantified distance reduction
Alnanih et al. [16]	±3% margin (HC-SR04 ultrasonic fill-level)	Wi-Fi (ESP8266) + GSM; Google Firebase	Under 1 s app/cloud sync; fire alert under 5 s	Qualitative route optimization (no distance metric); 99.5% uptime in 50-bin, 6-month pilot
Barth et al. [17]	Fill-level sensor modules (98-bin Swiss field study); accuracy NR	Sensor modules to cloud (LoRaWAN-based PSS)	Not Reported	Digital-twin DSS optimizing cost vs service-quality trade-off (KPI Q≥90%); substantial cost savings without QoS loss — no single distance-reduction %

The experimental results demonstrate that integrating real-time IoT monitoring with priority-based route optimization substantially improves the operational efficiency of waste collection. The 42.7% travel distance reduction achieved by the priority-enhanced nearest-neighbor algorithm is attributable to two complementary mechanisms: prioritizing full-status bins eliminates unnecessary visits to low-priority locations, while the 2,700-liter fleet capacity constraint prevents route plans from exceeding vehicle capacity. Consequently, the optimized approach generated a feasible collection volume of approximately 2,689 liters over 6.39 km, whereas the baseline produced approximately 5,047 liters over 11.15 km.

The observed 42.7% reduction in travel distance corresponds to a savings of approximately 4.76 km per collection cycle, decreasing from 11.15 km in the baseline scenario to 6.39 km under the proposed optimized routing scheme. Although direct vehicle fuel consumption measurements were not collected in this study, an indicative estimation can be derived using the manufacturer-specified fuel efficiency of the Viar Karya 150 three-wheeler, which is approximately 45 km/L under no-load conditions and conservatively adjusted to approximately 30 km/L under realistic loaded, stop-and-go waste collection operations. Based on this conservative estimate, the reduction in travel distance corresponds to an approximate fuel saving of 0.16 L per collection cycle. Assuming one collection cycle per day and approximately 26 operational days per month, the cumulative annual fuel saving could reach approximately 50 L. Furthermore, considering a typical operating speed of 15–20 km/h during residential waste collection, the same distance reduction is equivalent to an estimated time saving of approximately 14–19 min per cycle. Additional operational benefits may include reduced vehicle wear and lower frequency of return trips caused by vehicle capacity constraints.

It should be noted that the reported 4.76 km distance reduction is scenario-dependent and may vary according to the spatial distribution of waste bins, the number of bins requiring collection, and the resulting route configuration. Accordingly, the associated fuel and travel time savings should be regarded as indicative estimates derived from published vehicle specifications rather than direct field measurements. Future studies should validate these findings using real-world operational data and comprehensive economic analyses.

The multi-sensor fusion approach achieved a mean measurement error of 1.98%, confirming that arithmetic averaging across the four VL53L0X sensors effectively suppressed individual sensor noise [32], [33]. The reed-switch-based event-driven mechanism reduced the current consumption by 90.9%, supporting long-term battery-powered operation while maintaining timely data transmission. The mean end-to-end latency of 4.66 s, while reflecting cumulative MQTT over TLS, Cloud Run, Firestore, and Telegram overhead, remained substantially below the 30-second operational threshold.

This study contributes three main advancements: an energy-efficient event-driven IoT architecture that reduces power consumption by 90.9%; a dynamic two-tier priority mechanism that enhances the Nearest Neighbor algorithm by incorporating real-time fill-level data and vehicle capacity constraints; and a two-layer validation framework combining physical experimentation and digital twin simulation for cost-effective large-scale evaluation.

Large-scale validation was conducted using digital twin simulations rather than full-scale physical deployment, which may not fully capture heterogeneous networks and environmental conditions. The route optimization model employs Haversine-based distance estimation rather than real road network routing, potentially introducing deviations in complex urban scenarios. Furthermore, the Wi-Fi-dependent communication architecture may limit applicability in areas with insufficient network coverage, and the current system operates reactively without predictive modeling of fill-level dynamics.

Future work should explore the integration of metaheuristic optimization techniques, such as Genetic Algorithms and Ant Colony Optimization, for large-scale deployments, incorporate real-time traffic-aware routing via external APIs to improve travel estimation accuracy, and employ machine learning-based predictive models to enable proactive waste collection strategies. Full-scale deployment in TPS3R Sendangtirto is recommended to further validate the long-term system performance under real operational conditions.

CONCLUSION

This study demonstrates an integrated IoT smart waste bin system that couples real-time fill-level monitoring with priority-enhanced nearest-neighbor route optimization. Physical and digital twin validation confirmed all requirements: 1.98% mean sensing error, 90.9% reduction in current consumption via event-driven deep sleep, and mean end-to-end latency of 4.66 s. At the system level, the Full > Medium priority mechanism reduced the travel distance by 42.7% over the non-prioritized baseline while servicing all critical bins within the 2,700 L fleet constraint. A 100% transmission success rate was observed within a single controlled 150-message session. Long-term operational reliability remains to be validated through extended field deployment, which, together with road network–based routing, constitutes the primary direction for future work.

ACKNOWLEDGMENT

The authors would like to express sincere gratitude to the field personnel of TPS3R Sendangtirto, Sleman Regency, for their willingness to participate in structured interviews and for providing valuable operational insights that were fundamental to the requirements analysis and system design phases of this study. The authors also extend their appreciation to the Regional Environmental and Forestry Agency of Yogyakarta City (Dinas Lingkungan Hidup dan Kehutanan/DLHK) for providing access to waste management data that supported the empirical grounding of this study. Finally, the authors wish to acknowledge Universitas Ahmad Dahlan for the institutional support and academic resources that facilitated the conduct and completion of this study.

AUTHOR CONTRIBUTION STATEMENT

DY: Conceptualization, Methodology, Supervision, Validation, Investigation, Writing—Review and Editing, Project Administration. MIT: Software, Hardware Development, Data Curation, Formal Analysis, Investigation, Visualization, Writing—Original Draft, System Implementation and Testing.

Both authors contributed to the interpretation of the results, manuscript revision, and approval of the final manuscript.

AI DISCLOSURE STATEMENT

The authors used Claude (Anthropic) during the preparation of this manuscript for English language translation, academic writing refinement, and structural formatting assistance. After using the tool, the authors thoroughly reviewed and edited the content as needed and took full responsibility for the content of the publication.

REFERENCES

[1] M. A. Maulana, A. Syukri, M. D. Nurfaisal, and I. Sari, "Optimalisasi Teknologi untuk Efisiensi dan Transparansi Dalam Pengawasan Pengelolaan Sampah di Kota Serang," *Kybernology J. Gov. Stud.*, vol. 5, no. 1, pp. 67–80, Aug. 2025, doi: 10.26618/kjgs.v5i1.17802.
[2] N. I. Chairunnisa *et al.*, "Faktor-Faktor yang Mempengaruhi Perilaku Masyarakat dalam Pengelolaan Sampah Rumah Tangga di Kecamatan Harau, Sumatera Barat," *J. Huk. Polit. DAN ILMU Sos.*, vol. 4, no. 4, pp. 01–13, Jan. 2026, doi: 10.55606/jhpis.v4i4.5722.
[3] "SIPSN - Sistem Informasi Pengelolaan Sampah Nasional." Accessed: Apr. 24, 2025. [Online]. Available: https://sipsn.menlhk.go.id/sipsn/
[4] Department of Environment and Forestry of the Special Region of Yogyakarta, "SIPSN KLHK Waste Generation Data of the Special Region of Yogyakarta for 2024," Department of Environment and Forestry of the Special Region of Yogyakarta, unpublished raw dataset (Microsoft Excel), provided to the authors, 2025.
[5] Department of Environment and Forestry of the Special Region of Yogyakarta, "SIPSN Achievement Data of the Special Region of Yogyakarta for 2024," unpublished raw dataset (Microsoft Excel), provided to the authors, 2025.
[6] G. G. H. Bhodas and M. R. Firdaus, "Efektivitas Pengelolaan Sampah Di Tempat Pengolahan Sampah 3R (TPS 3R) Di Kecamatan Tanta Kabupaten Tabalong," *JAPB*, vol. 8, no. 2, pp. 1499–1533, Oct. 2025, doi: 10.35722/japb.v8i2.1300.
[7] Field Operations Personnel, "Personal communication (structured interview)," Sendangtirto, Sleman, Yogyakarta, Apr. 22, 2025.
[8] A. Suryaningrat, D. Kurnianto, and F. T. Syifa, "Pemanfaatan Google Firebase Pada Sistem Tempat Sampah Pintar Berbasis Internet of Things," *Din. Rekayasa*, vol. 17, no. 1, Art. no. 1, 2020, doi: 10.20884/1.dr.2021.17.1.324.
[9] D. Firmansyah, A. Ullah, A. Faizal, and H. Zarory, "Perancangan Sistem Pemantauan Kondisi Tempat Sampah Kampus Berbasis Internet of Things (iot) (studi Kasus: Fakultas Sains Dan Teknologi Uin Suska Riau)," *Transm. J. Ilm. Tek. Elektro*, vol. 25, no. 4, pp. 165–171, Dec. 2023, doi: 10.14710/transmisi.25.4.165-171.
[10] M. Ismail, R. K. Abdullah, and S. Abdussamad, "Tempat Sampah Pintar Berbasis Internet of Things (IoT) Dengan Sistem Teknologi Informasi," *Jambura J. Electr. Electron. Eng.*, vol. 3, no. 1, Art. no. 1, Jan. 2021, doi: 10.37905/jjeee.v3i1.8099.
[11] K. A. Verryando, R. J. Iskandar, and A. Y. A. Putra, "Implementasi Sensor Ultrasonik Pada Tempat Sampah Pintar Berbasis IoT," *INTEKSIS*, vol. 11, no. 1, pp. 1–12, May 2024, doi: 10.66003/inteksis.v11i1.9921.
[12] Y. B. Widodo, T. Sutabri, and L. Faturahman, "Tempat Sampah Pintar Dengan Notifikasi Berbasis IOT," *J. Teknol. Inform. Dan Komput.*, vol. 5, no. 2, pp. 50–57, Oct. 2019, doi: 10.37012/jtik.v5i2.175.
[13] M. Chaerul, M. Puturuhu, and I. Artika, "Optimasi Rute Pengangkutan Sampah dengan Menggunakan Metode Nearest Neighbour (Studi Kasus: Kabupaten Manokwari, Papua Barat)," *J. Wil. Dan Lingkung.*, vol. 10, no. 1, pp. 55–68, Apr. 2022, doi: 10.14710/jwl.10.1.55-68.

[14] P. N. Yanti, R. Vikaliana, and I. N. Purnaya, "Implikasi Metode Nearest Neighbor Terhadap Efektivitas Penjadwalan Truk Pengangkutan Sampah Di Dinas Lingkungan Hidup Kota Mataram," *Pros. Semin. Nas. Manaj. Ind. Dan Rantai Pasok*, vol. 2, pp. 134–141, 2021.
[15] S. M. Maturi, S. R. Dhanikonda, and S. Giddaluru, "Smart Dustbins: Real-Time Monitoring and Optimization for Waste Management in Smart Cities through IoT Devices," *Eng. Technol. Appl. Sci. Res.*, vol. 15, no. 1, pp. 19246–19252, Feb. 2025, doi: 10.48084/etasr.8562.
[16] R. Alnanih, L. Elrefaei, and A. Al-Ahwal, "Advancing Sustainability Through an IoT-Driven Smart Waste Management System with Software Engineering Integration," *Sustainability*, vol. 17, no. 21, p. 9803, Jan. 2025, doi: 10.3390/su17219803.
[17] L. Barth, L. Schweiger, R. Benedech, and M. Ehrat, "From data to value in smart waste management: Optimizing solid waste collection with a digital twin-based decision support system," *Decis. Anal. J.*, vol. 9, p. 100347, Dec. 2023, doi: 10.1016/j.dajour.2023.100347.
[18] D. A. Fitriani, T. Khotimah, and A. Jazuli, "Implementasi Iot Untuk Pengelolaan Sampah Cerdas Berbasis Sensor Di Kawasan Perumahan," *JATI J. Mhs. Tek. Inform.*, vol. 9, no. 4, pp. 7320–7325, May 2025, doi: 10.36040/jati.v9i4.14483.
[19] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Express*, vol. 5, no. 1, pp. 1–7, Mar. 2019, doi: 10.1016/j.icte.2017.12.005.
[20] W. Royce, "Managing the Development of Large Software Systems (1970)," 2021, pp. 321–332. doi: 10.7551/mitpress/12274.003.0035.
[21] I. Sommerville, *Software engineering*, Tenth edition. in Always learning. Boston Columbus Indianapolis New York San Francisco Hoboken Amsterdam Cape Town Dubai London: Pearson, 2016.
[22] H. Mantik, "Revolusi Industri 4.0: Internet of Things, Implementasi Pada Berbagai Sektor Berbasis Teknologi Informasi (bagian 1)," *JSI J. Sist. Inf. Univ. Suryadarma*, vol. 9, no. 2, pp. 41–48, Jul. 2022, doi: 10.35968/jsi.v9i2.919.
[23] D. Ramadhan, A. Hakim, and D. R. Irawati, "Sistem Pemantauan Dan Keamanan Pada Toko Berbasis Mikrokontroler Esp32 Devkit V1, Esp32 Cam Ai Thinker, Sensor Am312 Dan Buzzer Berbasis Iot," *J. Inf. Syst. Inform. Comput.*, vol. 9, no. 1, pp. 191–199, Jun. 2025, doi: 10.52362/jisicom.v9i1.1926.
[24] J. Riyanto and A. Wasid, "Perancangan Alat Bantu Parkir Mobil Berbasis Esp32-Cam Dan Sensor Jarak Vl53l0x Menggunakan Vlc," *J. Inform. Dan Komputasi Media Bahasan Anal. Dan Apl.*, vol. 17, no. 1, pp. 1–5, Jun. 2023, doi: 10.56956/jiki.v17i1.173.
[25] "TCA9548A Low-Voltage 8-Channel I²C Switch With Reset," Texas Instruments, SCPS207D, 2012, rev. 2015. Accessed: Jun. 09, 2026. [Online]. Available: https://www.ti.com/lit/ds/symlink/tca9548a.pdf
[26] I. S. Sudibyo, B. F. T. K, and M. S. K. T. S. Utomo, "Analisis Manajemen Termal Cylindrical Battery Pack Li-Ion 18650 Secara Konveksi Paksa Dengan Variasi Temperatur Inlet dan Laju Aliran Udara Menggunakan Computional Fluid Dynamics (CFD)," *J. Tek. MESIN*, vol. 11, no. 1, Art. no. 1, Jan. 2023, doi: 10.36706/jrm.v24i2.554.
[27] I. W. Nugraha, S. Triwijaya, Y. Wiarco, and M. Rukmana, "Prototipe Battery Management System dengan mempertimbangkan State of Charge dan State of Health," *J. Perkeretaapi. Indones. Indones. Railw. J.*, vol. 7, no. 2, pp. 61–70, Oct. 2023, doi: 10.37367/jpi.v7i2.292.
[28] H. A. Kusuma, R. Ariandhi, S. Refly, and S. Nugraha, "Development Arduino Data Logger using INA219 Sensor for Battery Capacity Monitoring," *J. Tek. Elektro Dan Komputasi ELKOM*, vol. 5, no. 1, pp. 9–15, Mar. 2023, doi: 10.32528/elkom.v5i1.8352.
[29] "MT3608: High-Efficiency 1.2 MHz 2 A Step-Up Converter," Aerosemi Technology Co., Ltd., Datasheet, Rev. 1.0. Accessed: Jun. 09, 2026. [Online]. Available: https://www.olimex.com/Products/Breadboarding/BB-PWR-3608/resources/MT3608.pdf
[30] S. Dani, "Peringatan Pintu Belum Tertutup Menggunakan Sensor Magnetik (Reed Switch) dan ESP32," *Comput. J.*, vol. 4, no. 1, pp. 27–33, Feb. 2026, doi: 10.58477/cj.v4i1.356.
[31] A. Awouda, E. Traini, G. Bruno, and P. Chiabert, "IoT-Based Framework for Digital Twins in the Industry 5.0 Era," *Sensors*, vol. 24, no. 2, p. 594, Jan. 2024, doi: 10.3390/s24020594.

[32] S. Komarizadehasl, B. Mobaraki, H. Ma, J.-A. Lozano-Galant, and J. Turmo, "Low-Cost Sensors Accuracy Study and Enhancement Strategy," *Appl. Sci.*, vol. 12, no. 6, p. 3186, Jan. 2022, doi: 10.3390/app12063186.
[33] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Inf. Fusion*, vol. 14, no. 1, pp. 28–44, Jan. 2013, doi: 10.1016/j.inffus.2011.08.001.
[34] R. Palupi, D. A. Yulianna, and S. S. Winarsih, "Analisa Perbandingan Rumus Haversine Dan Rumus Euclidean Berbasis Sistem Informasi Geografis Menggunakan Metode Independent Sample t-Test," *JITU J. Inform. Technol. Commun.*, vol. 5, no. 1, pp. 40–47, Jul. 2021, doi: 10.36596/jitu.v5i1.494.
[35] "ESP32 Series Datasheet," Espressif Systems, Espressif Systems, 2023. Accessed: Jun. 09, 2026. [Online]. Available: https://documentation.espressif.com/esp32_datasheet_en.pdf
[36] "INA219 Zerø-Drift, Bidirectional Current/Power Monitor with I²C Interface," Texas Instruments, INA219 datasheet, Aug. 2008 [Revised Dec. 2011]. Accessed: Jun. 09, 2026. [Online]. Available: https://www.ti.com/lit/ds/symlink/ina219.pdf
[37] "AMS1117 1 A Low Dropout Voltage Regulator," Advanced Monolithic Systems, AMS1117 datasheet. Accessed: Jun. 09, 2026. [Online]. Available: http://www.advanced-monolithic.com/pdf/ds1117.pdf