Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.15970
e-ISSN : 2541-2019
p-ISSN : 2541-044X

An API-Based Modular Architecture for Heterogeneous Multi-Device IoT Integration in Environmental Monitoring

Andi Marwan Elhanafi[1)*], Dedy Irwan[2)], Kissi Lola Armedia Br Siregar[3)]
[1,2,3)] Universitas Harapan Medan, Indonesia
[1)]andimarwanelhanafi@gmail.com, [2)]dedy_irwan.unhar@harapan.ac.id, [3)]kissiarmedia17@gmail.com

Submitted :March 5, 2026 | **Accepted** : May 11, 2026 | **Published** : July 5, 2026

Abstract: The Internet of Things (IoT) has increasingly played a significant role in the development of adaptive and real-time environmental monitoring systems. However, integrating multiple IoT devices remains challenging due to variations in data transmission intervals, communication protocols, and processing capabilities across devices. These differences often complicate system interoperability and data management within a unified monitoring platform. To address this issue, this study proposes an API-based modular architecture as a solution for integrating heterogeneous IoT devices in environmental monitoring systems. The proposed architecture separates core system functions into independent modules, including data acquisition, device management, and data visualization. The proposed architecture is evaluated through a multi-device environmental monitoring implementation configured with different logging intervals in order to assess communication performance and data consistency. The novelty of this study lies in its architectural approach to handling heterogeneous data transmission intervals in multi-device IoT environments using a modular API-based design. The experimental results indicate that the average communication latency is approximately 200ms, while the average daily data logging volume exceeds 3,500 entries per device. Furthermore, analysis of logging interval variations shows a time deviation of less than 3 seconds, which remains within the acceptable range for real-time environmental monitoring applications. The results demonstrate that the proposed architecture achieves success rate of over 97%, confirming the reliability of the proposed API-based modular architecture. Overall, the findings suggest that the modular API-driven architecture not only improves the flexibility and scalability of multi-device IoT integration but also maintains reliable data consistency and efficient communication performance.

Keywords: API; environmental monitoring; internet of things; modular architecture; multi-device

INTRODUCTION

Climate change, air pollution, and environmental degradation have become increasingly urgent global challenges that require effective technological solutions. Environmental monitoring plays a crucial role in providing accurate and real-time data to support informed decision-making, including disaster mitigation, air quality assessment, and natural resource management. In recent years, the Internet of Things (IoT) has emerged as a promising technological paradigm that enables automated collection, processing, and distribution of sensor data at relatively low cost. Through interconnected sensing devices, IoT systems can continuously monitor a wide range of environmental parameters such as temperature, humidity, air quality, pollutant concentration, and carbon dioxide (CO_2) levels. Recent developments in IoT-based environmental sensing systems indicate a growing shift toward real-time, low-power, and distributed monitoring architectures (Panduman et al., 2024; Polymeni et al., 2022). Modern IoT sensors are no longer limited to measuring basic environmental variables such as temperature and humidity but are also capable of monitoring more advanced parameters, including fine particulate matter (PM2.5 and PM10), hazardous gases (CO, NO_2, O_3), and environmental noise levels. In addition, several recent studies have integrated IoT sensing infrastructures with machine learning techniques to enable predictive analytics. For example, machine learning models can be used to identify air pollution patterns in urban areas or predict air

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

quality conditions using historical sensor data combined with meteorological information (Yan et al., 2023). These developments demonstrate that IoT-based environmental monitoring is evolving beyond simple data collection toward more intelligent systems capable of supporting data-driven decision making.

Despite these advancements, implementing IoT-based environmental monitoring systems still presents significant challenges, particularly when integrating multiple devices within a unified system. IoT devices are inherently heterogeneous in terms of hardware platforms, communication protocols, and data transmission intervals. Such heterogeneity often leads to interoperability issues, complex data management processes, and limited scalability as the system expands. Previous studies have shown that many traditional IoT monitoring systems rely on monolithic architectures (M. Alam et al., 2025; Islam et al., 2022), which makes them difficult to extend when new devices need to be added or existing components must be replaced. These limitations highlight the need for a modular system architecture that separates key functionalities such as data acquisition, communication, and visualization so that modifications in one component do not affect the entire system. In this context, Application Programming Interfaces (APIs) play an essential role in enabling modular IoT architectures. APIs provide standardized communication mechanisms that allow different modules and devices to interact efficiently while maintaining system flexibility and scalability.

To address the challenges associated with heterogeneous IoT device integration, this study proposes an API-based modular architecture for heterogeneous IoT integration, which is validated through the implementation of an environmental monitoring system. The proposed approach emphasizes modularity as a core design principle, allowing individual system components to operate independently while maintaining seamless communication through standardized interfaces. By adopting this architecture, the system can simplify the integration of multiple IoT devices with different configurations while maintaining flexibility for future expansion. The proposed architecture divides the system into several functional layers, including data acquisition, device communication, and data visualization. Each component operates as an independent module connected through an API. This modular separation enables new devices or services to be integrated without requiring substantial changes to the entire system. As a result, the architecture supports a more maintainable and scalable monitoring infrastructure, particularly in environments where the number of deployed IoT devices may increase over time.

To evaluate the effectiveness of the proposed architecture, the system integrates multiple IoT sensor devices built on the ESP32 platform. These devices are configured with different data logging intervals specifically 15 seconds and 30 seconds to simulate heterogeneous data transmission behavior commonly observed in real-world IoT deployments. By incorporating devices with varying logging frequencies, this study investigates how well the modular API-based architecture can manage asynchronous data flows while maintaining system stability and consistency. System performance is evaluated using several key indicators, including communication latency, daily data log volume, logging interval consistency, and the overall success rate of device integration. These metrics provide insights into both the technical performance and the operational reliability of the proposed system when handling data from multiple IoT devices simultaneously.

This study contributes to the field of IoT system architecture by proposing a modular environmental monitoring framework that separates system functionalities into independent layers connected through standardized APIs. The proposed architecture is validated through the implementation of a multi-device environmental monitoring platform capable of integrating heterogeneous IoT devices operating with different transmission intervals. In addition, this study provides a comprehensive evaluation of the proposed architecture by analyzing communication latency, logging interval consistency, data logging capacity, and transmission reliability under asynchronous multi-device communication scenarios. Through this approach, the study demonstrates how modular API-based architectural design can improve interoperability, scalability, and integration flexibility in heterogeneous IoT environments.

Unlike existing IoT implementations that primarily utilize common technologies such as REST APIs and modular system structures, the novelty of this study lies in the architectural integration of heterogeneous device behavior into the system design. This study does not introduce new communication technologies; instead, it addresses a practical but often overlooked problem in IoT systems, namely the asynchronous data transmission caused by different logging intervals across devices. The proposed architecture explicitly incorporates this heterogeneity into its modular API-based design, enabling consistent data handling without requiring uniform device configurations. Furthermore, this study provides real-world validation of the proposed architecture using multiple IoT devices with varying transmission intervals, which distinguishes it from many existing works that rely on simulation or homogeneous device setups. Therefore, the contribution of this study lies in the architectural design that bridges the gap between theoretical modularity and practical multi-device heterogeneity in IoT environments.

LITERATURE REVIEW

Recent research trends indicate an increasing adoption of API-based approaches in Internet of Things (IoT) systems to support interoperability and efficient communication among distributed devices. Common

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

implementations include RESTful APIs (Maurya, 2021) and Message Queue Telemetry Transport (MQTT) APIs (Alshammari, 2023), both of which enable lightweight data exchange and near real-time communication between sensors, gateways, and central servers. RESTful APIs are widely used due to their compatibility with web-based systems and ease of integration with various software platforms. Meanwhile, MQTT is specifically designed for IoT environments, offering low bandwidth consumption and efficient message delivery through a publish–subscribe communication model. The use of standardized APIs has been shown to simplify cross-platform integration, reduce system complexity, and minimize dependencies between different system components. Consequently, API-driven architectures are increasingly recognized as an important foundation for building open, modular, and extensible IoT ecosystems. Alongside the adoption of APIs, another emerging trend in IoT system design is the integration of edge computing and cloud–edge hybrid architectures (R. R. Singh et al., 2022). In this approach, part of the data processing is performed directly at the device or gateway level rather than relying solely on centralized cloud infrastructure. By processing data closer to the data source, edge computing can significantly reduce communication latency, decrease network bandwidth consumption, and improve the responsiveness of real-time monitoring systems. In environmental monitoring applications, this capability allows IoT devices deployed in the field to preprocess sensor data before transmitting it to the central server, enabling faster analysis and more efficient data management. Several studies have demonstrated that combining API-based architectures with edge computing frameworks can improve data exchange efficiency and reduce system latency in distributed IoT environments (Bukhsh et al., 2021). These developments highlight the importance of designing flexible architectures that can support both distributed processing and standardized communication.

Research related to IoT-based environmental monitoring has also grown rapidly in recent years. Various studies have developed systems for monitoring environmental parameters such as air quality, temperature, humidity, and pollutant concentration using IoT sensor networks. Many of these implementations focus on improving sensor accuracy, developing web-based monitoring dashboards, or enabling real-time data visualization. Similar research efforts have also been reported in Indonesia, where IoT technologies are increasingly applied for environmental monitoring purposes. However, many existing implementations are still designed as standalone systems that focus on a limited number of devices or a specific monitoring scenario (Hussein et al., 2024; Reddy & Kumar, 2026; Wahyuni Sabran et al., 2023). As a result, these systems often lack the flexibility required to support large-scale deployments involving multiple heterogeneous devices. Several studies, for example, focus primarily on monitoring air quality using a single type of IoT sensor device deployed in a specific area (Azhari et al., 2023; Martín-Baos et al., 2022; Wibawa & Putra, 2022). While such approaches are useful for demonstrating sensing capabilities, they often overlook the challenges associated with integrating multiple devices with different configurations and communication behaviors. Other research efforts concentrate on developing web-based dashboards for environmental monitoring data without addressing the underlying system architecture required for scalable device integration (T. Alam, 2021; Megantoro et al., 2021). In practice, environmental monitoring systems deployed at larger scales such as university campuses, industrial zones, or urban environments require reliable integration of multiple heterogeneous devices to ensure that collected data remain representative, consistent, and scalable.

Based on these research trends, an important research gap can be identified in the architectural design of IoT-based environmental monitoring systems. Many previous studies emphasize sensor deployment, data visualization, or specific monitoring applications, but comparatively little attention has been given to designing modular architectures that explicitly support heterogeneous multi-device integration through standardized APIs. In reality, modularity and interoperability are critical requirements for ensuring that IoT monitoring systems can evolve over time. Without a modular architecture, integrating new devices, modifying system components, or expanding the monitoring network may require significant changes to the entire system infrastructure. Therefore, there remains a need for a system architecture that explicitly addresses these challenges by providing a modular framework capable of integrating heterogeneous IoT devices while maintaining efficient communication and system scalability. Such an architecture should allow different system components such as data acquisition, device communication, and visualization to operate independently while remaining interconnected through standardized interfaces. Addressing this gap is essential for developing environmental monitoring systems that are not only functional but also adaptable to future technological developments and increasing system complexity.

Unlike existing studies that primarily focus on either API-based communication or application-specific IoT implementations, this study explicitly integrates architectural modularization with heterogeneous device behavior modeling. The proposed approach not only utilizes APIs as a communication mechanism but also structures the system into independent functional layers designed to handle asynchronous data transmission from multiple devices with varying logging intervals. In addition, this study provides an experimental evaluation that specifically examines the impact of heterogeneous transmission intervals on system performance, which is often overlooked in previous works that assume uniform device behavior. This distinction highlights that the novelty of this research lies in the combination of modular architectural design and practical validation under heterogeneous multi-device conditions.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.15970
e-ISSN : 2541-2019
p-ISSN : 2541-044X

METHOD

This study follows a structured research framework to design, implement, and evaluate a modular IoT–based environmental monitoring system capable of integrating multiple heterogeneous devices. The framework is intended to ensure that the development process proceeds systematically, starting from system requirement identification to performance evaluation and analysis of experimental results. By organizing the research workflow into several sequential stages, the proposed system can be developed and assessed in a controlled and reproducible manner.

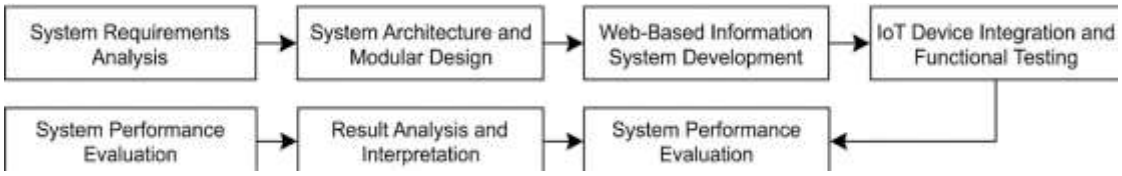

Fig. 1 Research Framework

The overall research process consists of six main stages: system requirements analysis, modular architecture design, web-based information system development, IoT device integration and functional testing, system evaluation, and analysis of experimental results. These stages collectively represent the workflow used to develop and validate the proposed modular monitoring system. The research framework is illustrated in Figure 1. The first stage involves system requirements analysis, where the operational needs of the environmental monitoring system are identified. This stage focuses on determining the types of environmental parameters to be monitored, the expected data transmission frequency, and the general system functionalities required to support real-time monitoring. The analysis also considers the challenges associated with integrating multiple IoT devices that may operate with different hardware configurations and data logging intervals.

The second stage focuses on system architecture design and modularization. In this phase, the overall architecture of the monitoring system is defined using a modular approach that separates the main functional components of the system. The architecture is designed to divide the system into several independent modules, including data acquisition, device communication, data processing, and visualization components. These modules interact through standardized APIs, enabling flexible communication between system components while maintaining scalability and maintainability. The third stage involves the development of the web-based information system, which serves as the central platform for managing and visualizing environmental data. This component is responsible for receiving sensor data from IoT devices through the API layer, storing the data in a database, and presenting the information through an interactive monitoring dashboard. The web-based system also provides basic device management functionalities and supports real-time data visualization. The fourth stage consists of IoT device integration and functional testing. In this phase, multiple IoT sensor devices are deployed and connected to the system using the proposed API-based communication mechanism. Each device transmits environmental data to the server at predefined logging intervals. Functional testing is then performed to ensure that the devices can successfully transmit data to the system and that the data can be properly processed and displayed by the monitoring platform. The fifth stage involves system evaluation and testing, where the performance of the implemented system is assessed using several evaluation metrics. Finally, the sixth stage focuses on the analysis of experimental results. In this stage, the collected evaluation data are analyzed to determine the reliability and efficiency of the proposed system architecture.

System Architecture

The system implementation in this study is primarily intended as a validation platform to evaluate the effectiveness of the proposed architecture under real-world conditions. The proposed environmental monitoring system is designed using a modular architecture that enables flexible integration of multiple IoT devices while maintaining system scalability and maintainability. The architecture separates the system into several functional components that operate independently but communicate through standardized APIs. This modular design approach allows each component to be developed, modified, or expanded without significantly affecting other parts of the system.

In general, the system architecture consists of four main layers: the IoT device layer, the communication layer, the data processing layer, and the application layer. These layers collectively support the acquisition, transmission, processing, storage, and visualization of environmental monitoring data. The IoT device layer represents the physical sensing devices deployed in the monitoring environment. Each device is responsible for collecting environmental parameters such as temperature, humidity, and air quality indicators through connected sensors. The devices periodically transmit the collected data to the central system using wireless network connectivity. Because the deployed IoT devices may operate with different configurations and data transmission intervals, the

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.15970
e-ISSN : 2541-2019
p-ISSN : 2541-044X

system must be capable of handling heterogeneous data flows generated by multiple devices simultaneously. The communication layer functions as the interface between IoT devices and the central monitoring system. In the proposed architecture, communication between devices and the server is facilitated through a REST-based API. This API layer acts as a standardized gateway that receives sensor data from IoT devices, validates the incoming data, and forwards it to the backend processing components. By using a standardized API interface, the system can support device interoperability and simplify the integration of new devices without requiring modifications to the core system components. The data processing layer is responsible for managing incoming data from IoT devices. This layer handles tasks such as data validation, storage, and basic preprocessing before the data are made available to the monitoring application. Sensor data transmitted through the API are stored in a centralized database, enabling efficient data management and retrieval for both real-time monitoring and historical analysis.

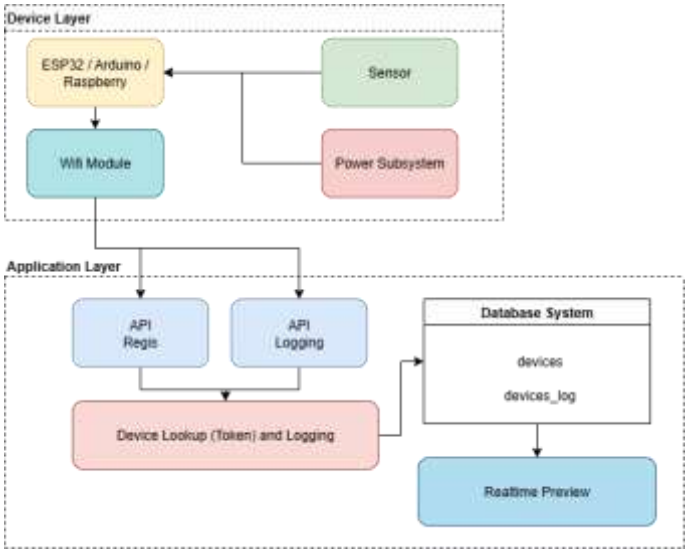

Fig. 2 System Architecture

Finally, the application layer provides the user interface for monitoring environmental conditions. This layer includes a web-based monitoring dashboard that visualizes sensor readings and historical data collected from multiple IoT devices. Through this interface, users can observe environmental parameters in near real-time, access historical data records, and monitor the operational status of connected IoT devices. The modular separation of these layers enables the monitoring system to support flexible expansion and long-term maintainability. For example, new types of sensors or additional IoT devices can be integrated into the system without requiring significant changes to the data processing or visualization modules. Similarly, improvements to the monitoring dashboard or data management components can be implemented independently of the device communication mechanism.

Beyond its implementation in environmental monitoring, the proposed architecture is designed to be domain-independent. The modular structure and API-based communication mechanism can be generalized to other IoT application domains that require integration of heterogeneous devices, such as smart agriculture, industrial monitoring, and smart city infrastructure. This abstraction highlights that the contribution of this study lies not only in a specific system implementation but in a reusable architectural design pattern for multi-device IoT integration.

IoT Device Configuration

The environmental sensing devices used in this study were developed using ESP32 microcontrollers due to their integrated Wi-Fi connectivity, low power consumption, and suitability for IoT-based applications (Hercog et al., 2023; Hussein et al., 2024; Pereira et al., 2023). Each device was equipped with environmental sensors capable of measuring several parameters relevant to environmental monitoring, including temperature, humidity, and air quality indicators. The IoT devices periodically collect environmental data and transmit the measurements to the central server through the API communication layer. To simulate heterogeneous device behavior commonly found in real-world IoT deployments, the devices were configured with different data logging intervals.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.15970
e-ISSN : 2541-2019
p-ISSN : 2541-044X

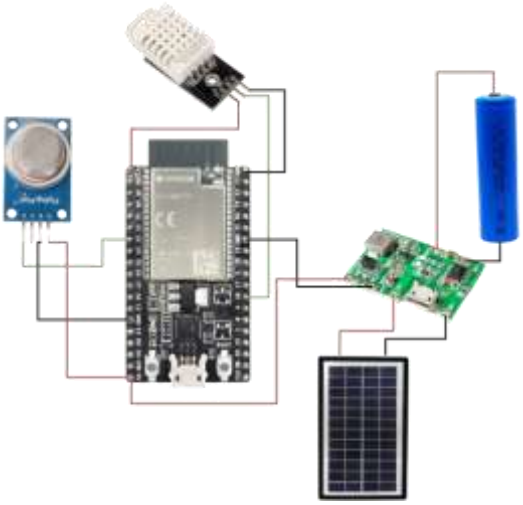

Fig. 3 IoT Device Schematic

In this study, two transmission configurations were implemented, where one group of devices transmitted data every 15 seconds, while another group transmitted data every 30 seconds. Each transmitted data record contains a timestamp and sensor readings generated by the device. The data are formatted using a lightweight JSON structure before being sent to the API endpoint. This format enables efficient communication between IoT devices and the server while maintaining compatibility with the modular architecture implemented in the system.

The schematic configuration of the IoT sensing device is illustrated in Figure 3. As shown in the diagram, the ESP32 microcontroller serves as the central processing unit that collects measurements from the connected sensors and manages wireless communication with the server. Sensor readings are processed locally before being transmitted through the Wi-Fi network to the API server. This configuration enables the device to operate as an autonomous sensing node capable of periodically sending environmental data to the monitoring system. By configuring devices with different logging intervals, the system can be evaluated under conditions that represent asynchronous data transmission from multiple devices. This configuration allows the proposed architecture to be tested in handling heterogeneous data streams while maintaining consistent data logging and reliable system performance.

Architectural Realization

The proposed environmental monitoring architecture was implemented using a modular architecture that integrates IoT sensing devices, an API-based communication layer, a centralized database, and a web-based monitoring interface. The following implementation details are presented to illustrate how the proposed architecture is realized in practice, rather than to emphasize system-specific engineering aspects. The implementation focuses on enabling reliable data exchange between multiple IoT devices and the server while maintaining system flexibility and scalability.

Fig. 4 API Communication

The communication between IoT devices and the server is handled through a REST-based API that serves as the primary interface for data transmission (Catovic et al., 2022; Sartaj et al., 2025). Each IoT device is required to register with the server before transmitting telemetry data. During this process, the device sends its unique identifier to the registration endpoint, after which the server verifies the device and issues an authentication token. This token is subsequently used to authorize all data transmission requests from the device. After successful registration, the device enters the telemetry transmission cycle. At each logging interval, the device collects sensor measurements and sends them to the server through the logging API endpoint using an HTTP POST request. The

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.15970
e-ISSN : 2541-2019
p-ISSN : 2541-044X

transmitted data are structured in JSON format, containing the device identifier, timestamp, and sensor readings. Upon receiving the request, the server performs authentication and data validation before storing the information in the database. The overall communication process between the IoT devices and the server is illustrated in Figure 4, which shows the API communication of device registration, token authentication, and periodic telemetry logging. On the server side, incoming sensor data are processed and stored in a relational database that maintains device information and historical sensor logs. The database structure is designed to support efficient data retrieval for both real-time monitoring and historical analysis.

Performance Evaluation Metrics

The performance of the proposed IoT monitoring system was evaluated using several quantitative metrics to assess the reliability of multi-device data integration. The evaluation focuses on communication latency, data logging capacity, logging interval consistency, and the integration success rate. Communication latency represents the delay between the time a sensor reading is transmitted by an IoT device and the time it is successfully received and recorded by the server (Sefati & Halunga, 2023; Shukla et al., 2023). The latency for each data transmission is calculated as the difference between the timestamp when the data are received by the server and the timestamp when the data are sent by the device, as shown in Equation (1), where latency is defined as the difference between the receive timestamp (T_receive) and the send timestamp (T_send), i.e., Latency = T_receive − T_send.

$$Latency = T\_receive - T\_send \quad (1)$$

The average latency value is then used to represent the overall communication performance of the system. The second metric is logging interval consistency, which evaluates the stability of the data transmission interval generated by IoT devices (Jiang et al., 2022). Since the devices operate with predefined logging intervals of 15 seconds and 30 seconds, the difference between consecutive timestamps is calculated to evaluate logging interval consistency, as shown in Equation (2), where the interval difference is computed as $\Delta t = t\_i - t\_\{i-1\}$, representing the difference between two consecutive timestamps.

$$\Delta t = t\_i - t\_\{i-1\} \quad (2)$$

$$Success\ Rate = (N\_success / N\_total) \times 100\% \quad (3)$$

Finally, the integration success rate measures the reliability of the system in handling data transmissions from multiple IoT devices (P. Singh et al., 2022). The success rate is calculated as the ratio between the number of successfully recorded transmissions and the total number of transmission attempts, as shown in Equation (3), where the success rate is defined as the ratio of successful transmissions to total transmission attempts, i.e., Success Rate = (N_success / N_total) × 100%.

RESULT

System Deployment

The proposed environmental monitoring system was deployed using several ESP32-based IoT sensing devices configured with different data transmission intervals. The deployment aimed to simulate heterogeneous IoT environments where devices may operate with varying logging frequencies. In this implementation, two logging configurations were used, namely 15 seconds and 30 seconds transmission intervals.

Fig. 5 IoT Device Deployment

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.15970
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Each device periodically collected environmental sensor readings and transmitted the data to the server through the REST-based API communication layer. The transmitted data were stored in the system database and subsequently visualized through the web-based monitoring dashboard. This setup enabled continuous monitoring of environmental parameters while simultaneously allowing the system to manage asynchronous data streams generated by multiple devices.

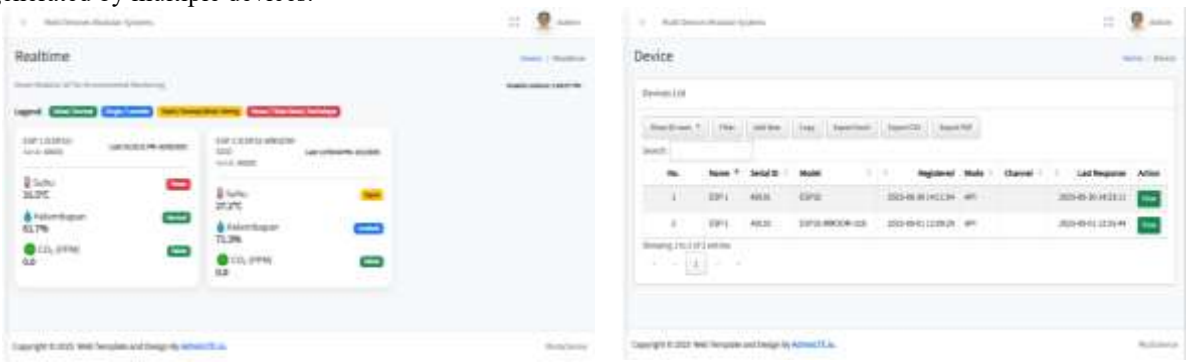

Fig. 6 Web Deployment

The monitoring interface provided real-time visualization of sensor readings as well as historical data access. Through the dashboard, users could observe the latest environmental conditions and verify whether data from each device were successfully received and recorded by the system. The overall system interface used for monitoring environmental data is shown in Figure 6. The dashboard displays real-time sensor values and graphical trends of environmental parameters collected from the connected IoT devices.

Communication Latency Analysis

Communication latency was analyzed to evaluate the responsiveness of the proposed IoT monitoring system in receiving sensor data from multiple devices. Latency represents the time difference between when the data are transmitted by the IoT device and when the server successfully records the data in the database. The measurement results show that the system maintains a relatively stable communication delay during the monitoring period. The observed latency values are generally within the range expected for wireless IoT communication over standard network infrastructure. Based on the collected data, the average communication latency is approximately 200ms, indicating that the system is capable of receiving sensor data from multiple devices with minimal delay.

Table 1
Latency Results

Latency Range (ms)	Frequency
100–150	280
150–180	385
180–210	770
210–240	1050
240–270	595
270–300	420

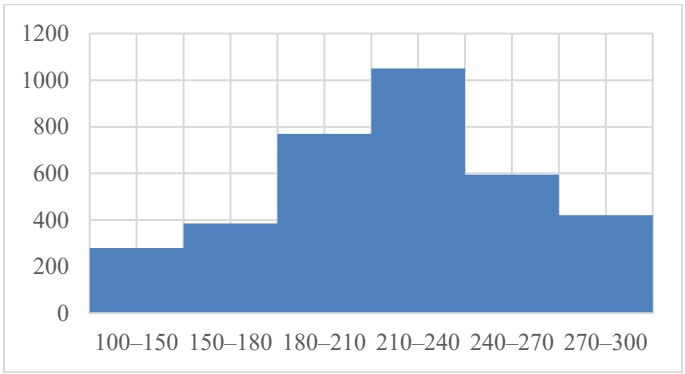

Fig. 7 Latency Distribution of 3,500 Data Logs

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.15970
e-ISSN : 2541-2019
p-ISSN : 2541-044X

The distribution of latency values is illustrated in Figure 7, which shows that most transmissions fall within a narrow latency range. Occasional variations in latency are primarily influenced by network conditions and server processing time. However, these variations remain within acceptable limits for environmental monitoring applications that require near real-time data updates. Overall, the results demonstrate that the proposed API-based architecture can support efficient communication between IoT devices and the server. The relatively low latency confirms that the system is suitable for applications that require timely environmental data acquisition and monitoring.

Logging Consistency and Success Rate Analysis

Logging interval consistency was analyzed to evaluate the stability of the data transmission cycle generated by the IoT devices. Since the sensing devices were configured with predefined logging intervals of 15 seconds and 30 seconds, the actual transmission intervals recorded by the server were examined to determine whether the devices maintained consistent data delivery. The analysis was performed by calculating the difference between consecutive timestamps recorded in the system database. The results show that the observed transmission intervals generally follow the configured logging schedules with only minor deviations. During the monitoring period, the majority of interval differences remain close to the expected transmission times.

Table 2 Sample Sensor Log Interval 15s

Serial_ID	Model	Last_Data	Temp (°C)	Humid (%)	ADC	PPM	TimeStamp
A0101	ESP32	8/30/2025 14:17:52	31.2	63.1	169	0	8/30/2025 14:18:07
A0101	ESP32	8/30/2025 14:18:07	31.3	62.8	160	0	8/30/2025 14:18:22
A0101	ESP32	8/30/2025 14:18:22	31.4	62.3	155	0	8/30/2025 14:18:37
A0101	ESP32	8/30/2025 14:18:37	31.4	61.8	175	0	8/30/2025 14:18:52
A0101	ESP32	8/30/2025 14:18:52	31.5	61.5	165	0	8/30/2025 14:19:07

Table 3 Sample Sensor Log Interval 30s

Serial_ID	Model	Last_Data	Temp (°C)	Humid (%)	ADC	PPM	TimeStamp
A0102	ESP32-WROOM-32U	9/1/2025 12:49:41	27.2	71.5	0	0	9/1/2025 12:50:11
A0102	ESP32-WROOM-32U	9/1/2025 12:50:11	27.2	71.4	0	0	9/1/2025 12:50:41
A0102	ESP32-WROOM-32U	9/1/2025 12:50:41	27.2	71.4	0	0	9/1/2025 12:51:11
A0102	ESP32-WROOM-32U	9/1/2025 12:51:11	27.2	71.3	0	0	9/1/2025 12:51:41
A0102	ESP32-WROOM-32U	9/1/2025 12:51:41	27.1	71.3	80	0	9/1/2025 12:52:11

Table 4 Logging interval deviation summary

Device ID	Configured Interval (s)	Avg Observed Interval (s)	Min Interval (s)	Max Interval (s)	Max Deviation (s)
A0101	15	15.2	14.7	16.8	1.8
A0102	30	30.4	29.3	32.7	2.7

The measurement results indicate that the maximum deviation of the logging interval is less than 3 seconds, which is considered acceptable for environmental monitoring applications that rely on periodic data acquisition.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.15970
e-ISSN : 2541-2019
p-ISSN : 2541-044X

These small variations are mainly influenced by network latency, processing delays on the server, and scheduling differences in device-side task execution.

In addition to evaluating the stability of logging intervals, the reliability of the multi-device data integration was also assessed through the transmission success rate. This metric measures the proportion of sensor data successfully received and stored by the server relative to the total number of transmission attempts generated by the IoT devices. During the observation period, the monitoring system demonstrated a consistently high level of communication reliability. The majority of sensor data transmitted by the IoT devices were successfully processed and recorded by the server through the API-based communication mechanism. Minor transmission failures were occasionally observed, primarily caused by temporary network instability or short communication interruptions.

Table 5 Multi-device transmission success rate over a 7-day monitoring period

Day	Transmission Attempts	Successful Records	Failed Records	Success Rate (%)
Day 1	8420	8205	215	97.4
Day 2	8560	8351	209	97.6
Day 3	8615	8394	221	97.4
Day 4	8490	8278	212	97.5
Day 5	8720	8502	218	97.5
Day 6	8650	8429	221	97.4
Day 7	8585	8370	215	97.5

The overall transmission performance observed during the seven-day monitoring period is summarized in Table 7. The results indicate that the system maintains an integration success rate above 97%, confirming that the proposed modular API architecture can reliably handle continuous data streams generated by multiple IoT devices operating with different logging intervals.

DISCUSSIONS

It is important to note that the objective of this evaluation is not to optimize system implementation, but to validate the effectiveness of the proposed architectural design in handling heterogeneous device integration. The experimental results demonstrate that the proposed modular API-based architecture is capable of supporting reliable multi-device integration for environmental monitoring applications. The communication latency analysis shows that the system maintains an average delay of approximately 200ms, which is sufficiently low for near real-time monitoring scenarios. Such latency levels indicate that the REST-based API communication mechanism does not introduce significant delays in the transmission and processing of sensor data.

Table 6. System Performance Summary

Metric	Result
Average latency	~200 ms
Logs per device per day	>3500
Interval deviation	<3 s
Integration success rate	>97%

The analysis of logging interval consistency further confirms the stability of the data acquisition process. Devices configured with 15-second and 30-second transmission intervals produced observed intervals that remained close to the configured values, with deviations generally below three seconds. These small variations are expected in practical IoT deployments and are mainly caused by network transmission delays and server-side processing time. Nevertheless, the deviations remain within acceptable limits for environmental monitoring systems that rely on periodic data collection. In terms of data logging capacity, the monitoring system successfully recorded thousands of sensor data entries per device per day without experiencing significant data loss. This result indicates that the implemented system architecture can handle continuous data streams generated by multiple devices operating simultaneously. The modular separation between device communication, data processing, and visualization components also contributes to improved system scalability. The integration reliability analysis further demonstrates the robustness of the proposed architecture. Throughout the seven-day observation period, the system maintained a data transmission success rate exceeding 97%. This high success rate confirms that the API-based integration mechanism is capable of reliably handling asynchronous data transmissions from heterogeneous IoT devices. The small percentage of failed transmissions is mainly attributed to temporary network

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

instability rather than limitations of the system architecture itself. These results validate that the proposed modular architecture, rather than the specific implementation, is capable of supporting reliable multi-device integration.

To further observe the scalability characteristics of the proposed architecture under asynchronous communication conditions, an additional lightweight concurrent transmission observation was conducted by increasing the transmission frequency of one IoT device to approximately two requests every two seconds while other devices continued operating using their normal logging intervals.

Table 7. Concurrent Transmission Observation

Device ID	Transmission Pattern	Approximate Interval	Observation Result
Device 1	Increased asynchronous transmission	~1–2 s	Stable data reception without packet ordering issues
Device 2	Normal periodic transmission	~6–7 s	No interruption observed during concurrent requests

The observation results indicate that the API-based modular architecture was able to maintain stable data reception and timestamp consistency without significant communication disruption or data ordering issues. The server successfully processed asynchronous requests generated from devices operating with different transmission frequencies, suggesting that the modular API communication layer can tolerate moderate increases in concurrent data transmission activity. Although the observation was conducted on a limited scale and was not intended as a full server stress test, the results provide an initial indication that the proposed architecture remains operationally stable under moderate asynchronous multi-device communication conditions.

Compared to previous IoT-based environmental monitoring systems, the proposed architecture demonstrates comparable or improved communication performance while offering greater flexibility in handling heterogeneous devices. For example, previous studies often assume uniform data transmission intervals or rely on monolithic system designs, which limits scalability. In contrast, the proposed modular API-based architecture maintains stable latency and high success rates even under asynchronous multi-device conditions, highlighting its advantage in real-world deployments. Despite the promising results, this study has several limitations. The system evaluation was conducted using a limited number of IoT devices and relatively simple environmental parameters. In large-scale deployments involving hundreds of devices or more complex sensing scenarios, additional challenges such as network congestion, data synchronization, and fault tolerance may arise. Furthermore, the current implementation relies on a centralized server architecture, which may introduce scalability constraints under high data loads. Future research should explore distributed or edge-based extensions of the proposed architecture to address these limitations.

CONCLUSION

The primary contribution of this study lies in the architectural design rather than the application implementation. The proposed system separates key functionalities into independent modules, including data acquisition, device communication, and data visualization, allowing the system to remain flexible and scalable when integrating heterogeneous IoT devices. Experimental results demonstrate that the system can reliably manage continuous data streams generated by multiple IoT sensor devices configured with different logging intervals. The observed communication latency remains around 200 ms, indicating that the API-based communication mechanism supports near real-time data transmission. In addition, the analysis of logging interval consistency shows that the devices maintain transmission intervals close to their configured values, with deviations generally below three seconds. The system also successfully records thousands of sensor data entries per device per day while maintaining a transmission success rate exceeding 97%. These results confirm that the proposed modular API architecture provides a reliable and efficient approach for integrating heterogeneous IoT devices in environmental monitoring systems. By enabling flexible device integration and stable communication performance, the architecture offers a practical foundation for developing scalable monitoring infrastructures. Future work may focus on expanding the system with additional sensor types, integrating edge computing capabilities for local data processing, and incorporating advanced data analytics techniques such as machine learning to support predictive environmental monitoring. This study demonstrates that innovation in IoT systems can be achieved not only through new technologies, but also through architectural design that addresses real-world integration challenges.

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

ACKNOWLEDGMENT

The author would like to express sincere gratitude to the Direktorat Penelitian dan Pengabdian kepada Masyarakat (DPPM) for providing the Beginner Lecturer Research Grant for the 2025 Fiscal Year under contract number: 019 / SK-M / VI / R.UnHar / 2025.

REFERENCES

Alam, M., Islam, M. M., Nayan, N. M., & Uddin, J. (2025). An IoT Based Real-Time Environmental Monitoring System for Developing Areas. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, *52*(1), 106–121. https://doi.org/10.37934/araset.52.1.106121

Alam, T. (2021). Cloud-based iot applications and their roles in smart cities. *Smart Cities*, *4*(3), 1196–1219. https://doi.org/10.3390/smartcities4030064

Alshammari, H. H. (2023). The internet of things healthcare monitoring system based on MQTT protocol. *Alexandria Engineering Journal*, *69*, 275–287. https://doi.org/10.1016/j.aej.2023.01.065

Azhari, Nasution, T. I., Sinaga, S. H., & Sudiati. (2023). Design of Monitoring System Temperature And Humidity Using DHT22 Sensor and NRF24L01 Based on Arduino. *Journal of Physics: Conference Series*, *2421*(1), 12018. https://doi.org/10.1088/1742-6596/2421/1/012018

Bukhsh, M., Abdullah, S., & Bajwa, I. S. (2021). A Decentralized Edge Computing Latency-Aware Task Management Method with High Availability for IoT Applications. *IEEE Access*, *9*, 138994–139008. https://doi.org/10.1109/ACCESS.2021.3116717

Catovic, A., Kadusic, E., Ruland, C., Zivic, N., & Hadzajlic, N. (2022). Air pollution prediction and warning system using IoT and machine learning. *International Conference on Electrical, Computer, Communications and Mechatronics Engineering, ICECCME 2022*, 1–4. https://doi.org/10.1109/ICECCME55909.2022.9987957

Hercog, D., Lerher, T., Truntič, M., & Težak, O. (2023). Design and Implementation of ESP32-Based IoT Devices. *Sensors*, *23*(15), 6739. https://doi.org/10.3390/s23156739

Hussein, N. M., Mohialden, Y. M., & Salman, S. A. (2024). Impact of IoT-Based Environmental Monitoring on Lab Safety and Sustainability. *Babylonian Journal of Internet of Things*, *2024*, 16–26. https://doi.org/10.58496/bjiot/2024/003

Islam, M. M., Kashem, M. A., & Uddin, J. (2022). An internet of things framework for real-time aquatic environment monitoring using an Arduino and sensors. *International Journal of Electrical and Computer Engineering*, *12*(1), 826–833. https://doi.org/10.11591/ijece.v12i1.pp826-833

Jiang, C., Fu, C., Zhao, Z., & Du, X. (2022). Effective Anomaly Detection in Smart Home by Integrating Event Time Intervals. *Procedia Computer Science*, *210*(C), 53–60. https://doi.org/10.1016/j.procs.2022.10.119

Martín-Baos, J. Á., Rodriguez-Benitez, L., García-Ródenas, R., & Liu, J. (2022). IoT based monitoring of air quality and traffic using regression analysis. *Applied Soft Computing*, *115*, 108282. https://doi.org/10.1016/j.asoc.2021.108282

Maurya, R. (2021). Application of Restful APIs in IOT: A Review. *International Journal for Research in Applied Science and Engineering Technology*, *9*(2), 145–151. https://doi.org/10.22214/ijraset.2021.33013

Megantoro, P., Pramudita, B. A., Vigneshwaran, P., Yurianta, A., & Winarno, H. A. (2021). Real-time monitoring system for weather and air pollutant measurement with html-based ui application. *Bulletin of Electrical Engineering and Informatics*, *10*(3), 1669–1677. https://doi.org/10.11591/eei.v10i3.3030

Panduman, Y. Y. F., Funabiki, N., Fajrianti, E. D., Fang, S., & Sukaridhoto, S. (2024). A Survey of AI Techniques in IoT Applications with Use Case Investigations in the Smart Environmental Monitoring and Analytics in Real-Time IoT Platform. *Information (Switzerland)*, *15*(3), 153. https://doi.org/10.3390/info15030153

Pereira, G. P., Chaari, M. Z., & Daroge, F. (2023). IoT-Enabled Smart Drip Irrigation System Using ESP32. *Internet of Things*, *4*(3), 221–243. https://doi.org/10.3390/iot4030012

Polymeni, S., Athanasakis, E., Spanos, G., Votis, K., & Tzovaras, D. (2022). IoT-based prediction models in the environmental context: A systematic Literature Review. *Internet of Things (Netherlands)*, *20*, 100612. https://doi.org/10.1016/j.iot.2022.100612

Reddy, C. D., & Kumar, D. K. (2026). Iot-Based Multi-Sensor Data Fusion for Precision Crop Yield Optimization Using Arduino,Esp32,and Webcam Integration. *Journal of Engineering and Technology for Industrial Applications*, *12*(57), 602–611. https://doi.org/10.5935/jetia.v12i57.2893

Sartaj, H., Ali, S., & Gjøby, J. M. (2025). REST API Testing in DevOps: A Study on an Evolving Healthcare IoT Application. *ACM Transactions on Software Engineering and Methodology*. https://doi.org/10.1145/3765744

Sefati, S. S., & Halunga, S. (2023). Ultra-reliability and low-latency communications on the internet of things based on 5G network: Literature review, classification, and future research view. *Transactions on Emerging Telecommunications Technologies*, *34*(6), e4770. https://doi.org/10.1002/ett.4770

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

Shukla, S., Hassan, M. F., Tran, D. C., Akbar, R., Paputungan, I. V., & Khan, M. K. (2023). Improving latency in Internet-of-Things and cloud computing for real-time data transmission: a systematic literature review (SLR). *Cluster Computing*, *26*(5), 2657–2680. https://doi.org/10.1007/s10586-021-03279-3

Singh, P., Saman Azari, M., Vitale, F., Flammini, F., Mazzocca, N., Caporuscio, M., & Thornadtsson, J. (2022). Using log analytics and process mining to enable self-healing in the Internet of Things. *Environment Systems and Decisions*, *42*(2), 234–250. https://doi.org/10.1007/s10669-022-09859-x

Singh, R. R., Banerjee, S., Manikandan, R., Kotecha, K., Indragandhi, V., & Vairavasundaram, S. (2022). Intelligent IoT Wind Emulation System Based on Real-Time Data Fetching Approach. *IEEE Access*, *10*, 78253–78267. https://doi.org/10.1109/ACCESS.2022.3193774

Wahyuni Sabran, F., Zalfiana Rusfian, E., Ilmu Administrasi dan Kebijakan Bisnis, M., & Ilmu Administrasi, F. (2023). Penggunaan Internet of Things pada eFishery untuk keberlanjutan Akuakultur di Indonesia. *Innovative: Journal Of Social Science Research*, *3*(2), 8142–8156. https://j-innovative.org/index.php/Innovative/article/view/1359

Wibawa, I. M. S., & Putra, I. K. (2022). Design of air temperature and humidity measurement based on Arduino ATmega 328P with DHT22 sensor. *International Journal of Physical Sciences and Engineering*, *6*(1), 9–17. https://doi.org/10.53730/ijpse.v6n1.3065

Yan, K., Zhou, X., & Yang, B. (2023). Editorial: AI and IoT applications of smart buildings and smart environment design, construction and maintenance. *Building and Environment*, *229*. https://doi.org/10.1016/j.buildenv.2022.109968

*name of corresponding author