Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics e-ISSN: 2656-8624
Homepage: https://ijeeemi.org/; Vol. 8, No. 2, pp. 189-201, May 2026

RESEARCH PAPER

OPEN ACCES

Python-Based Backend Architecture Design for Commercial Medical IoT Device Integration: A Case Study of Omron HEM-7142T1

Tien Fabrianti Kusumasari[1], Yudhi Widyatama[2], Alaric Rasendriya Aniko[1], and Sinung Suakanto[1]

[1] Department of Information System, Telkom University, Bandung, Indonesia
[2] Finance & Asset Management Platform, IT Division, PT Telkom Indonesia (Persero) Tbk, Jakarta, Indonesia

Abstract

The current implementation of Remote Patient Monitoring (RPM) still faces crucial challenges related to the accuracy and integrity of medical data. Many healthcare IoT devices rely on generic sensors that require rigorous manual calibration and exhibit unstable error rates, failing to meet international clinical standards. This study aims to design and implement an integrated backend architecture that bridges certified commercial medical devices with digital health systems. The main contribution is a six-layer IoT architecture specifically designed to integrate the Omron HEM-7142T1 device to ensure data validity in remote blood pressure monitoring. Following the Design Science Research Methodology (DSRM), the system was developed using Python, the Bleak library for Bluetooth Low Energy (BLE) communication, and FastAPI to provide interoperable REST API services. Functional testing in Postman demonstrated that the system successfully extracts medical data, producing JSON output with an HTTP 200 OK status under single-access conditions. However, load testing using Apache JMeter with 10 virtual users revealed limitations in the hardware's point-to-point BLE protocol. The /scan endpoint showed stable performance with a 0% error rate and an average response time of 5.04 seconds. In contrast, endpoints /connect-and-read and endpoint /latest-bp-records recorded error rates of 100% and 90%, respectively, with an average response time of 23.29 seconds when accessed simultaneously, due to the Omron device's locking mechanism. This study concludes that while the six-layer architecture effectively ensures medical data integrity in single-access scenarios, it requires a database caching module in the Logic Tier to overcome parallel access constraints. The implementation provides a foundation for developing secure, standardized professional RPM systems for medical use.

Paper History
Received Feb. 10, 2026
Revised April 2, 2026
Accepted April 8, 2026
Published April 12, 2026

Keywords
IoT Architechture;
Omron Tensimeter;
API Testing;
Bluetooth Low Energy;
Design Science Research Methodology

Author Email
Tienkusumasari @telkomuniversity.ac.id
yudhi.widyatama@telkom.co.id
rasendriyaa @student.telkomuniversity.ac.id
sinung@telkomuniversity.ac.id

I. Introduction

The development of the Internet of Things (IoT) has made a significant contribution to the medical world [1]. One example is the Remote Patient Monitoring (RPM) system. The RPM system is effective for remote patient monitoring, especially for patients in remote locations or far from hospitals, critical conditions, and elderly patients [2]. Medical personnel can also detect and monitor patients' physiological parameters in real-time without having to meet face-to-face [3]. All physiological data required by doctors can be collected through the system and sent wirelessly to medical service centers. This allows doctors to evaluate patient conditions and make treatment decisions more quickly and accurately [4], [5]. Blood pressure measurement is one of the main parameters for diagnosing hypertension [6]. Previous research has shown that self-monitoring of blood pressure can help lower blood pressure levels in patients [7]. In addition, regular blood pressure monitoring is crucial for reducing the risk of chronic heart disease (CHD) and suppressing the prevalence of hypertension. Hypertension is one of the most common cardiovascular diseases and a leading cause of premature death worldwide. Therefore, the global target is to reduce the prevalence of hypertension by 25% by 2025 [8].

However, various studies have shown that healthcare IoT systems still face serious issues related to data accuracy. The majority of current RPM research still relies on generic sensors such as the MPX5050 or MPX5700 series, which require a rigorous calibration process. The use of the MPX5050GP analog blood pressure sensor processed through an Arduino Mega microcontroller, for example, is reported to have an accuracy rate ranging from 93% to 94% [9]. This accuracy fluctuation was also found in another study comparing the MPX5050GP sensor with an Omron digital tensiometer, with a maximum error rate of 5.06% for systolic and 6.45% for diastolic [10]. Then, an error was found in the average results of blood pressure ratio testing with the Arduino Uno-based MPX5050GP sensor. The error for the systolic

Corresponding author: Tien Fabrianti Kusumasari, tienkusumasari@telkomuniversity.ac.id, Department of Information System, Telkom University, Bandung, Indonesia, Jl. Telekomunikasi No.1, 40257, Bandung, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

ratio was 0.48, and for the diastolic ratio was 0.78 [11]. In fact, testing on ESP32 and NodeMCU-based systems shows that the difference in measurement results with the standard Omron HEM-8712 medical device can reach a significant difference with an average error of 7.09% in systolic and 6.88% in diastolic [12]. This high percentage of error, which in some cases in systolic/diastolic reaches 7.4% / 7.2% [13]. On the other hand, several studies have succeeded in reducing errors to around 2.49% [14] or achieving an accuracy of 96.83% [15]; these results still vary and have not reached the stability of certified medical devices.

Based on previous studies, generic sensors often fail to maintain consistent, accurate blood pressure measurements. Inaccurate blood pressure readings can lead to severe clinical consequences, such as misdiagnosis, inappropriate medication dosing, and failure to detect hypertensive crises. These inaccuracies ultimately elevate the risk of severe cardiovascular events like stroke or heart attack, highlighting the critical need for absolute data integrity in RPM systems. This encourages the need to integrate commercial medical IoT devices that have official marketing authorization, such as Omron (HEM-7142T1) [16], into a structured and interoperable system architecture. In addition, Omron has also gone through a testing process and obtained certification from leading international institutions, namely the American Association for the Advancement of Medical Instrumentation (AAMI) and the British Hypertension Society (BHS) [17]. These certifications guarantee that the device has passed rigorous clinical validation protocols. Meeting these specific standards is essential for ensuring that the data used for remote medical diagnostics is highly reliable.

Various studies have proposed IoT architecture models for smart healthcare systems. Research [18] proposed an IoT architecture based on four main layers: the perception layer, transport layer, processing layer, and application layer. This architecture emphasizes the integration of medical sensors and hospital systems, but still combines data processing functions and application services in a single, centralized layer, thus limiting backend flexibility and scalability. Meanwhile, research [19] proposed a more generic IoT architecture with five layers: the device layer, gateway layer, data management layer, application layer, and business layer. This model provides a broad conceptual overview. Neither study specifically discussed the architecture of IoT systems with commercial healthcare devices. This will be a research gap that will be filled in this study. Additionally, commercial healthcare devices differ from generic sensors in that they often use proprietary communication protocols, local locking mechanisms during data transfer, and closed-system data structures, requiring specialized parsing and integration layers.

To address these limitations, this study proposes a six-layer IoT architecture consisting of: a perception layer, a network layer, a service management layer, a data processing layer, an API service layer, and an application layer. The addition of separate API Service and Data Processing layers aims to improve modularity and data security when interacting with commercial healthcare devices via Bluetooth Low Energy (BLE). The primary objective of this study is to design a Python-based IoT architecture capable of securely and scalably extracting, processing, and distributing data from commercial medical devices.

The contributions of this study are summarized in four main points. The first point is the proposal of a six-layer IoT architecture optimized for standardized commercial medical device integration. The second point is the development of a backend using Python with the Bleak library for BLE communication and FastAPI for high-performance endpoint management. The third point is the development of a data parsing mechanism that converts device-exclusive data into a standard JSON Schema format ready for consumption by various platforms. The fourth point is the provision of a REST API service that supports full interoperability between commercial medical devices and the digital health ecosystem. Collectively, these four contributions provide a comprehensive blueprint for transitioning RPM systems from prototype-grade sensor networks to clinically reliable, interoperable digital health ecosystems.

The systematic structure of this article begins with Section I: Introduction, which presents the background and urgency of the problem. Section II: Materials and Methods provides a detailed explanation of the design for each architectural layer and the device specifications. Section III: Results presents the outcomes of the API functionality and latency testing. Section IV: Discussion analyzes the implications of the research findings in comparison with previous models, and the article concludes with Section V: Conclusion, which summarizes the findings and provides suggestions for future development.

II. Materials And Method

A. Remote Patient Monitoring

Remote Patient Monitoring (RPM) is an innovative healthcare system that allows for the regular tracking of patients' vital signs and health metrics outside of traditional clinical settings through wireless devices such as blood pressure cuffs, oximeters, and glucometers [20]. The implementation of RPM has been officially recommended by professional organizations such as the American College of Cardiology and the American Heart Association in the management of hypertension due to its potential to reduce disparities in healthcare access [21]. The main advantage of this method is its higher accuracy in predicting cardiovascular (CVD) morbidity and mortality compared to blood pressure measurements in medical offices, while also being an effective solution to overcome geographical limitations [22]. Furthermore, RPM has been shown to improve patients' quality of life through medication dose optimization, increased adherence, and early detection of health deterioration without relying on physical clinic visits, thereby significantly reducing

Corresponding author: Tien Fabrianti Kusumasari, tienkusumasari@telkomuniversity.ac.id, Department of Information System, Telkom University, Bandung, Indonesia, Jl. Telekomunikasi No.1, 40257, Bandung, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics e-ISSN: 2656-8624
Homepage: https://ijeeemi.org/; Vol. 8, No. 2, pp. 189-201, May 2026

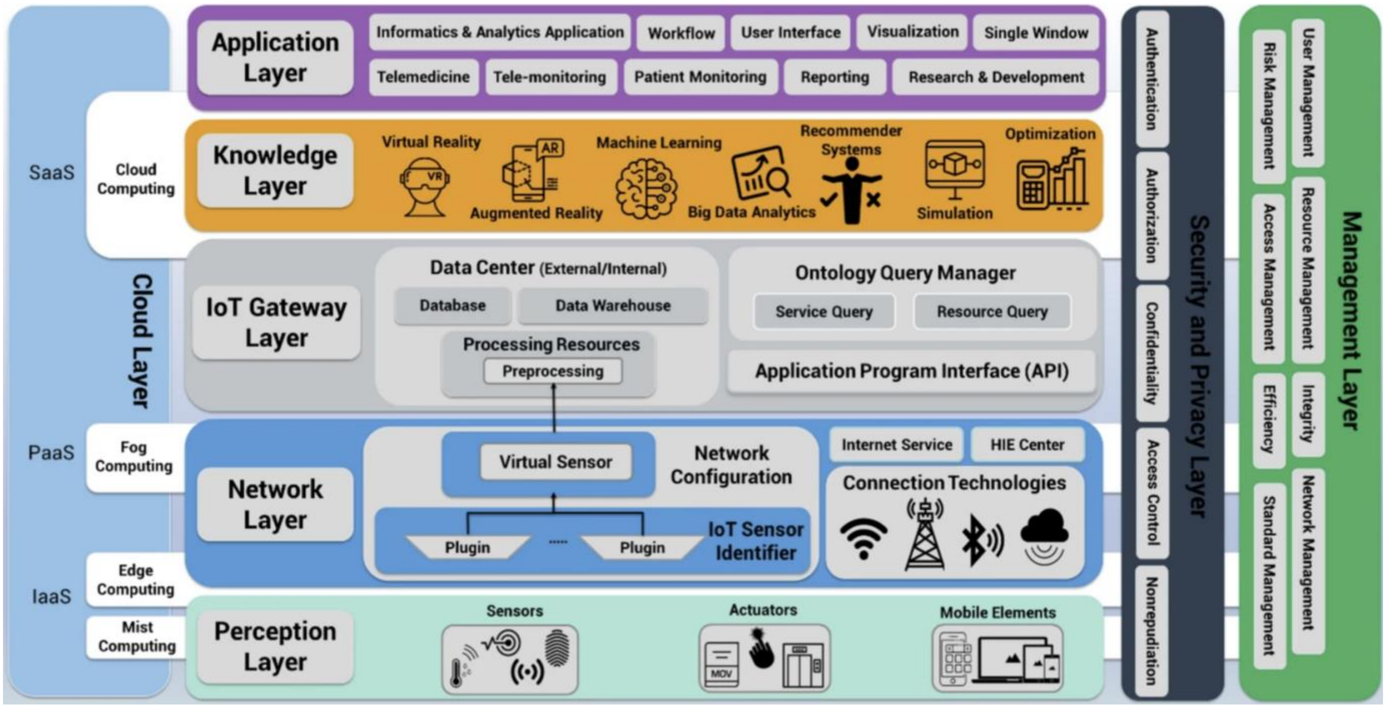


Fig. 2. Layered architecture consisting of five horizo´ntal layers and three vertical functional aspects [18]

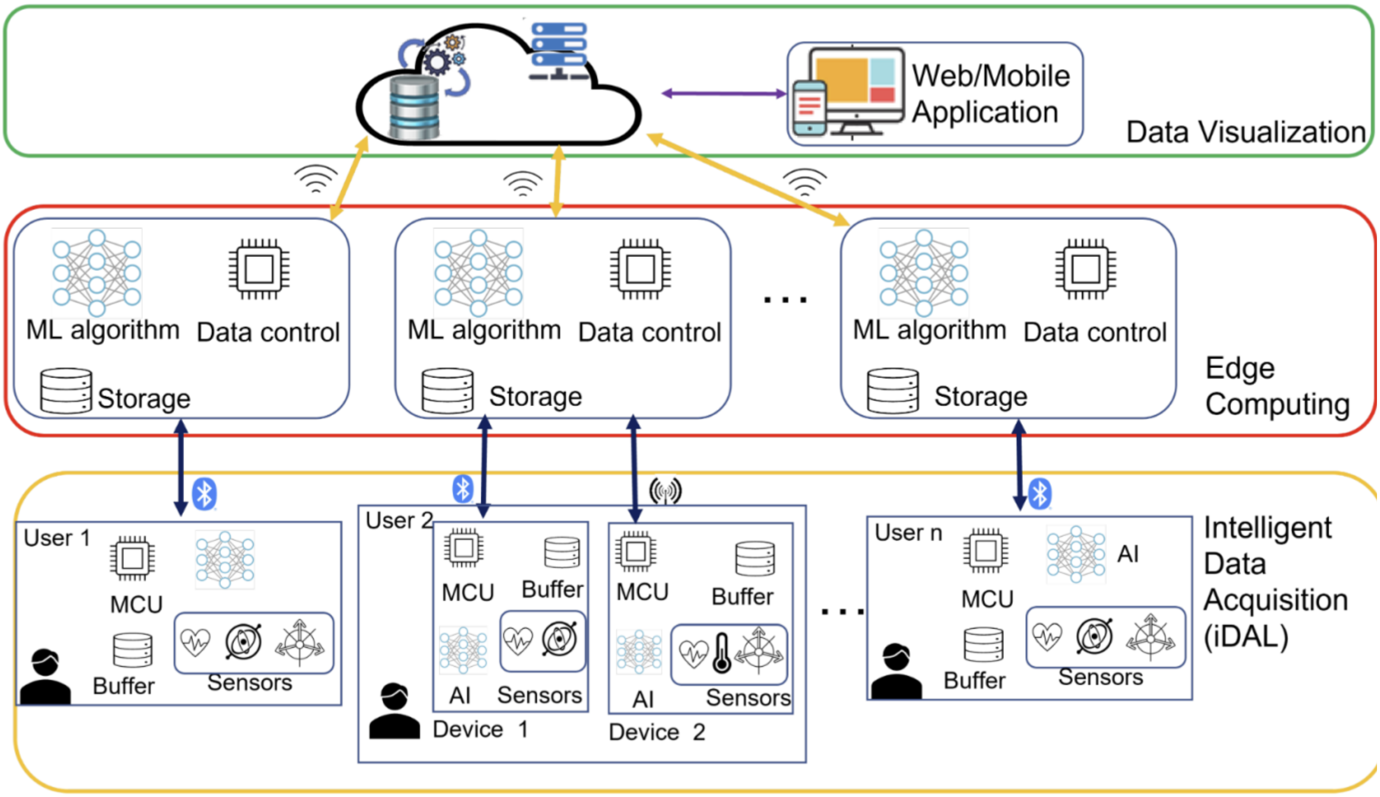


Fig. 1. A three-layer IoT architecture incorporating multi-sensing technology for healthcare usingartificial intelligence [27]

healthcare costs due to unnecessary hospitalizations or emergency department visits [23]. Despite the mentioned benefits, current RPM implementations face difficulties in ensuring ongoing data integrity, hardware interoperability constraints, and complex integration of commercial healthcare devices into backend systems.

B. IoT Implementation on RPM

Corresponding author: Tien Fabrianti Kusumasari, tienkusumasari@telkomuniversity.ac.id, Department of Information System, Telkom University, Bandung, Indonesia, Jl. Telekomunikasi No.1, 40257, Bandung, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

The implementation of the Internet of Things (IoT) in Remote Patient Monitoring (RPM) enables the integration of blood pressure sensors into wireless and cloud networks for automated acquisition, analysis, and early warning. This technology bridges sensors with microcontrollers to facilitate real-time monitoring by remote medical personnel. The sensor approach in RPM is dominated by oscillometric methods using cuffs and pressure sensors such as the MPX5050GP series, which boast stable accuracy through precise calibration [12]. Meanwhile, wearable innovations are beginning to utilize photoplethysmography (PPG) or piezoelectric sensors combined with machine learning for more continuous blood pressure estimation [24]. Microcontrollers such as the ESP32, ESP8266, and Arduino serve as the main processing units in this architecture. Some implementations integrate fuzzy logic for cardiovascular risk detection with systolic accuracy reaching 98.2% [25]. Apart from Wi-Fi, the use of LoRaWAN networks has also proven reliable for long-distance data transmission to databases such as Firebase with systolic accuracy of 97.5% and diastolic accuracy of 96.8% [26].

C. IoT Architecture

Fig. 1 shows the architecture proposed in the study [18], adopting a layered model consisting of five horizontal layers and three vertical functional aspects to support the system [18]. The main data flow begins with the Perception Layer, which captures patient vital signs through energy-efficient protocols, and continues to the Network Layer for wireless connectivity, supporting the exchange of health information across hospitals [18]. The next layer, the IoT Gateway, serves as an edge computing bridge to normalize data and significantly reduce transmission latency [18]. The system's intelligent core resides in the Knowledge Layer, which utilizes Big Data analytics and artificial intelligence for health risk prediction, while the Application Layer provides an interface for medical personnel for real-time data visualization [18]. To maintain operational stability, vertical aspects, including resource management, a cloud platform for large-scale storage, and a security layer with GDPR-standard encryption, are implemented across all horizontal layers [18]. This design follows the Design Science Research (DSR) methodology through three systematic phases: problem identification, design considerations, and an evaluation framework using the Architecture Tradeoff Analysis Method (ATAM) [18]. The use of ATAM allows for a comprehensive assessment of the system's qualitative attributes. Evaluation results demonstrate that the integration of edge computing and the HL7 FHIR standard API can accelerate clinical detection and effectively reduce readmission rates [18]. Furthermore, Fig. 2 shows an architecture designed by [27], enhanced by the use of advanced sensing technologies such as ultra-thin and flexible piezoelectric sensors that enable high-precision biosignal measurements [27]. Moving data processing to the sensor level through the Intelligent Data Acquisition Layer (iDAL) has been shown to provide drastic performance improvements, including 99.947% data reconstruction accuracy and efficient wireless data transmission [27]. Researchers [28] also developed an Ontology-Driven IoT System utilizing a three-layer architecture (Sensor, Fog,

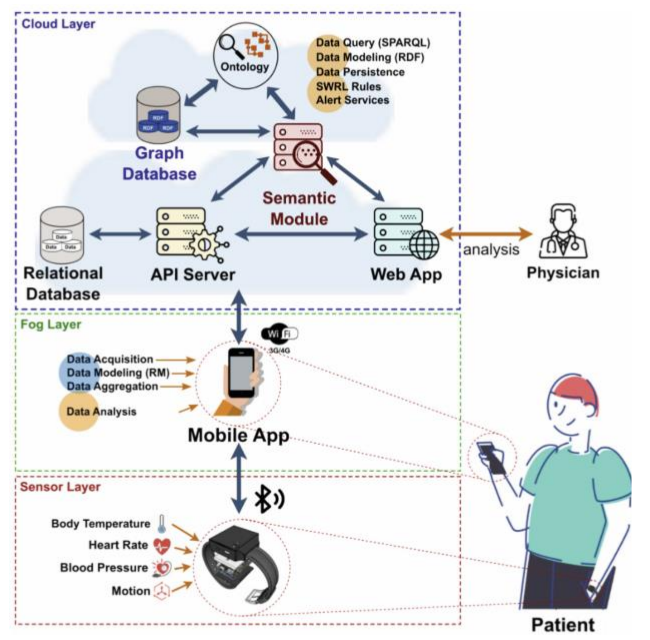


Fig. 3. Utilization of three-layer architecture in Ontology-Driven IoT System for Monitoring Hypertension [28]

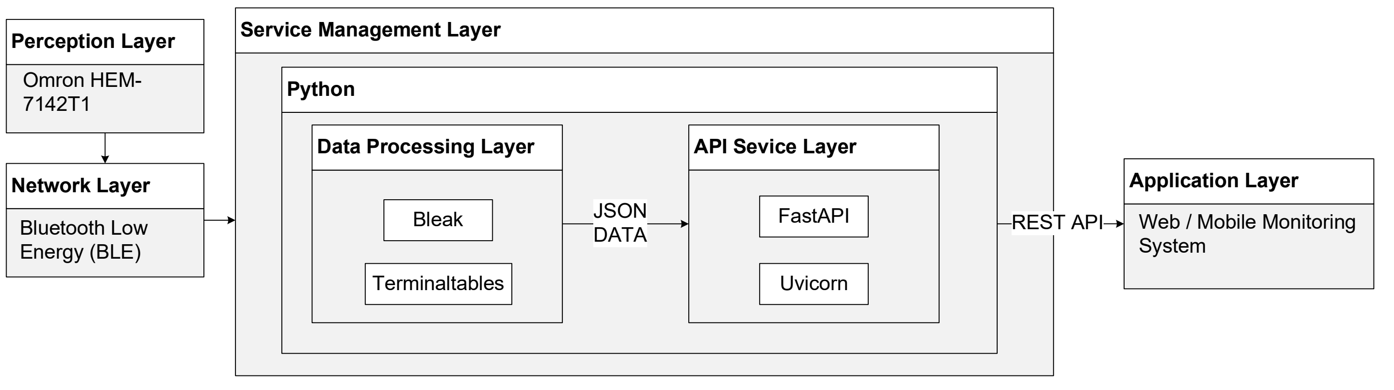


Fig. 4. Proposed Internet of Things architecture for blood pressure monitoring system

Corresponding author: Tien Fabrianti Kusumasari, tienkusumasari@telkomuniversity.ac.id, Department of Information System, Telkom University, Bandung, Indonesia, Jl. Telekomunikasi No.1, 40257, Bandung, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

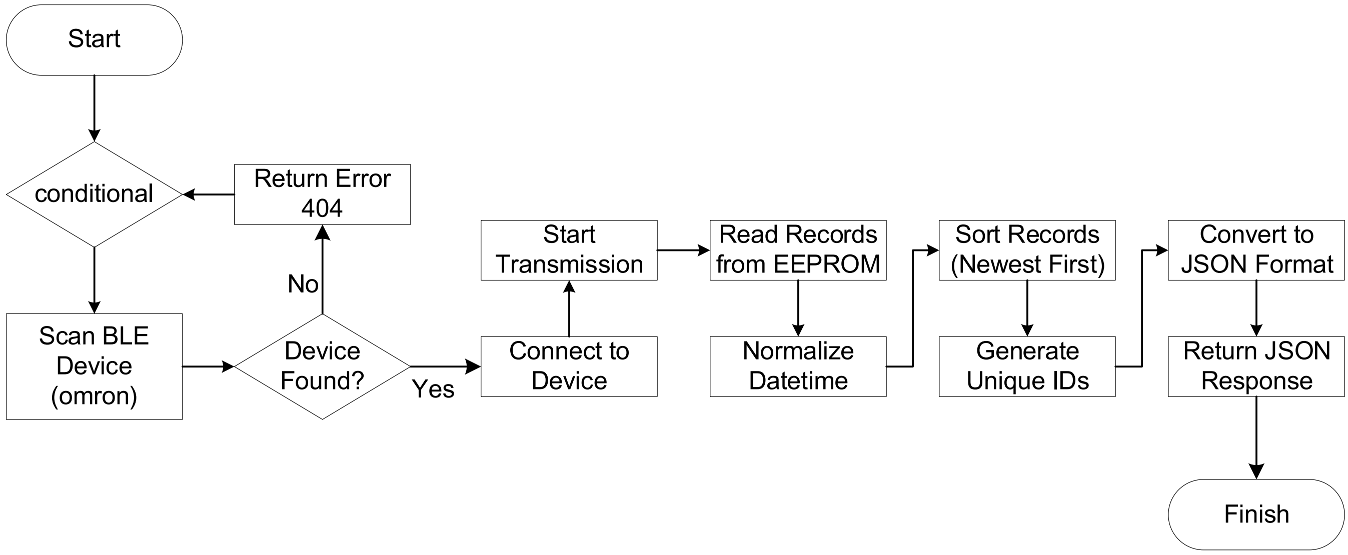


Fig. 5. Illustration of the system workflow steps depicted with a flowchart diagram

Table 1. Correlation between the application of design science research methodology and research

DSRM Phase	Implementation in This Study
Identify Problem & Motivate	Identifying issues with existing IoT system architecture, namely non-medical-grade blood pressure monitors. Emphasize the importance of accurate, real-time blood pressure monitoring to support hypertension diagnosis.
Define Objectives of a Solution	Proposes an IoT architecture design for blood pressure monitoring using the official Omron HEM-7142T1 medical device, connected via Bluetooth Low Energy (BLE), and processed by Python to produce a flexible and scalable API.
Design & Development	Designing a 6-layer IoT architecture by building a BLE–Python integration module, as well as developing an API for integration with digital health applications.
Demonstration	Testing the API using Postman and JMeter as client simulations to access the API, and ensuring that blood pressure data can be transferred and processed correctly.
Evaluation	Evaluate the system based on test results on Postman and JMeter, and compare performance with previous research.
Communication	Publish research results to journals with Sinta 2 index

and Cloud) as shown in Fig. 3. The system aims to monitor hypertensive patients in a structured manner, where clinical data is processed and archived for the analysis needs of healthcare professionals [28]. By combining the flexibility of a layered model and the efficiency of edge computing, the proposed architecture is able to provide a health monitoring solution that is modular, secure, and responsive to critical patient conditions. Previous architectures lacked a separation between the data processing and service layers, resulting in reduced scalability and tight coupling between system components.

D. Method

The Design Science Research Methodology (DSRM) [29], also known as the Peffers approach, consists of six main stages in conducting design science research. These stages are generally followed sequentially in problem-centered initiation research studies [30]. DSRM emphasizes the development of innovative solutions to address existing problems [31] and is now increasingly applied in engineering and other disciplines [32]. DSRM encompasses six phases: (1) identify the problem and motivate, (2) define the solution objectives, (3) design & development, (4) demonstration, (5) evaluation, and (6) communication [33]. Table 1 illustrates the relationship between the application of Design Science Research Methodology (DSRM) and the research conducted. In the Identify Problem & Motivate phase, this research focused on identifying problems in the IoT system architecture that still use non-medical blood pressure monitors, and emphasized the urgency of accurate, real-time blood pressure monitoring to support hypertension diagnosis.

Corresponding author: Tien Fabrianti Kusumasari, tienkusumasari@telkomuniversity.ac.id, Department of Information System, Telkom University, Bandung, Indonesia, Jl. Telekomunikasi No.1, 40257, Bandung, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

Next, the Define Objectives of a Solution phase was implemented by proposing an IoT architecture design based on the official Omron HEM-7142T1 medical device connected via Bluetooth Low Energy (BLE) and processed using Python to produce a flexible and scalable API. The Omron HEM-7142T1 was selected due to its widespread availability, BLE capability, and validated clinical accuracy. In the Design & Development phase, Python code was specifically developed that leverages the Bleak library to parse Omron's hexadecimal code into a standard JSON schema, enabling seamless API creation. The Demonstration phase was conducted by testing the API using Postman and JMeter as client simulations, ensuring that blood pressure data could be transferred and processed correctly. The Evaluation phase then assessed system performance based on the test results and compared them with previous research. Finally, the Communication phase was realized with the publication of the research results in a reputable journal with a Sinta 2 index.

III. Results

A. Proposed Architecture

This research produces an architecture designed to bridge data communication between a blood pressure measuring device, namely Omron (model HEM-7142T1), with a frontend application through an Application Programming Interface (API) service. This integration is carried out using the Bluetooth Low Energy (BLE) protocol, which enables wireless data transmission with low power consumption [34]. In addition to having medical standards, Omron devices are also used as instruments and validation references in various previous studies [10], [14], [25], [35], [36], so their selection strengthens the reliability and validity aspects of measurements in this study. This research is also in line with the concept of Smart and Secure Health Monitoring, especially in the data transmission and data collection phases, which emphasize the efficiency of patient data collection and transmission [37], [38].

Fig. 4 shows a proposed six-layer system architecture to support IoT-based health monitoring. In the Perception Layer, the Omron HEM-7121 medical device serves as the primary data source by measuring the patient's blood pressure. The measurement data is then sent via the Network Layer using the BLE communication protocol, ensuring efficiency and energy savings. Next, the data enters the Service Management Layer, built in the Python programming language. This layer is divided into two sublayers: the Data Processing Layer, which utilizes Bleak and TermiNatables to handle BLE connections and data processing, and the API Service Layer, which uses FastAPI and Uvicorn to provide API-based services. The separation of the Data Processing and API Services layers aims to increase modularity, enable secure data transformation, and support scalable system integration. The processed data is packaged in JSON format for easy integration. Finally, the data is sent to the Application Layer via a REST API. This layer displays the measurement results in the form of a web-based monitoring system or mobile application, allowing users and medical personnel to access real-time blood pressure information.

Fig. 5 illustrates the system workflow using a flowchart [39], [40]. The process begins with a scan for nearby BLE devices. If no device is found, the system returns a 404 error and re-scans. If found, the system connects to the device and initiates data transmission. Next, the system reads the blood pressure recordings from the EEPROM (Electrically Erasable Programmable Read-Only Memory), normalizes the time to match the system's time zone, and sorts the data from most recent to least recent. The system then generates a unique ID for each recording and converts it to JSON format. Finally, the measurement results are sent back as a JSON response.

B. API Endpoint Development

From the proposed architecture, a system was developed that has three main endpoints, namely /scan, /connect-and-read, and /latest-bp-records. The /scan endpoint functions to scan BLE devices around the system, in order to detect and identify Omron devices that will be used in the blood pressure data collection process. Meanwhile, the /connect-and-read endpoint functions to retrieve all measurement data stored in the Omron device memory, and the /latest-bp-records endpoint is used to obtain the latest blood pressure data measured by the user. These three endpoints were successfully implemented and tested using a functional testing approach to validate that communication between the server and client, as well as

Table 2. Data structure of Omron device measurement results in JSON format

Data	Description
Movement (mov)	Movement indicator during measurement
Irregular Heartbeat (ihb)	Indicator of an irregular heartbeat detected during measurement.
Systolic (sys)	Systolic blood pressure value (the highest pressure when the heart contracts).
Diastolic (dia)	Diastolic blood pressure value (the lowest pressure when the heart relaxes).
Heart Beat Rate (bpm)	Number of heart beats per minute
Datetime	Time and date the measurement was taken

Corresponding author: Tien Fabrianti Kusumasari, tienkusumasari@telkomuniversity.ac.id, Department of Information System, Telkom University, Bandung, Indonesia, Jl. Telekomunikasi No.1, 40257, Bandung, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

the request and response process, runs as expected [41]. After functional testing is carried out, load testing will be carried out. These tests are carried out using Postman, an API testing tool that has been widely used in previous studies due to its reliability in testing the performance and accuracy of API services [42], [43].

C. Functional Testing

Functional testing was conducted to ensure the /scan endpoint's ability to detect devices. Testing was conducted using the HTTP GET method via the Postman application, without requiring a request body. The test results showed that the server responded with an HTTP 200 OK code, indicating a successful BLE device scanning process. From the testing, a device named BLESmart_00000480FE4EAF4F7F7AC, namely the Omron HEM-7142T1, was detected. The device has a Received Signal Strength Index (RSSI) of -63 dBm, indicating a stable BLE connection, and can proceed to the data reading stage [44], [45].

Functional testing on the /latest-bp-records endpoint was performed to ensure the system's ability to retrieve the latest blood pressure measurement data from the Omron HEM-7142T1 device. The test results showed that the system returned an HTTP 200 OK response, indicating that communication between the Omron device and the backend was successful. Furthermore, functional testing on the /connect-and-read endpoint returned an HTTP 200 OK response code, confirming that the BLE connection, data handshake, and data parsing processes were successful.


```
{
  "mov": 1,
  "ihb": 0,
  "dia": 80,
  "sys": 110,
  "bpm": 86,
  "datetime": "04.10.2025 21:47:50"
}
```


Fig. 7. Successful data extraction produces standard structured JSON output for application use

The data received from the Omron device is structured in JSON format as Fig. 6, including vital parameters required for elderly health monitoring. This data structure is described in Table 2. This functional success demonstrates the validity of the proposed IoT architecture model in ensuring data integrity and interoperability with security-certified medical devices. Next, load testing was conducted to measure API performance under simultaneous access loads.

D. Load Testing

Load testing was performed using 10 Virtual Users (VUs) for 1 minute to evaluate the stability and efficiency of three main endpoints: /scan, /connect-and-read, and /latest-bp-records. As shown in Fig. 7, Postman recorded a total of 92 requests with an average response time of 6.24 seconds and an overall error rate of 59.78%. More specifically, the /scan endpoint demonstrated the most stable performance with 36 requests, no errors (0%), and an average response time of 5.04 seconds. This indicates that the BLE device scanning process is efficient because it only involves passive detection operations without direct interaction with the device's memory. In contrast, the

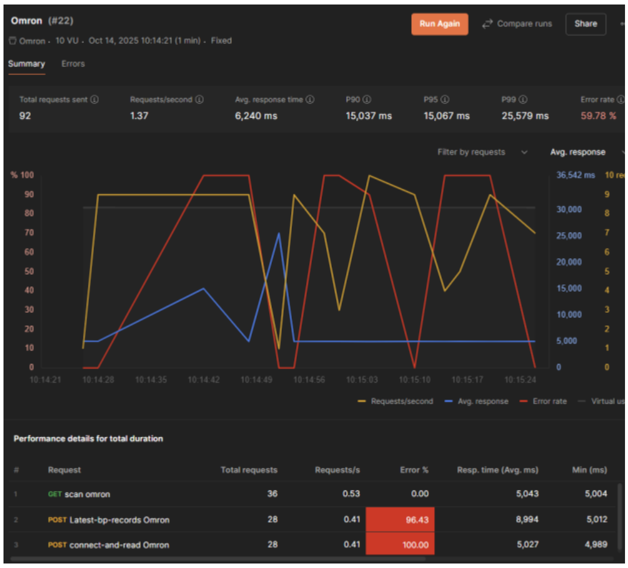


Fig. 6. Load testing results with Postman using 10 Virtual Users for 1 minute

Table 3. Load testing results with JMeter using 10 Virtual Users for 1 minute

Endpoint	Status Code	Response Message	Number of Samples	Success	Failed	Average Response Time (ms)	Time Span (ms)
/scan	200	OK	10	10	0	5030	5013 – 5067
/latest-bp-records	200 & 500	OK & Internal Server Error	10	1	9	24900	15044 – 41455
/connect-and-read	422	Unprocessable Entity	10	0	10	4–6	2 – 6

Corresponding author: Tien Fabrianti Kusumasari, tienkusumasari@telkomuniversity.ac.id, Department of Information System, Telkom University, Bandung, Indonesia, Jl. Telekomunikasi No.1, 40257, Bandung, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

/connect-and-read and /latest-bp-records endpoints had high failure rates, 100% and 96.43%, respectively, with average response times of 5.03 seconds and 8.99 seconds.

Load testing was also conducted using Apache JMeter with a configuration of 10 threads (virtual users), a ramp-up period of 1 second, and a loop count of 1, as shown in Table 3. JMeter is an open-source load-testing framework for measuring and analyzing the performance of various system services [46], [47]. The test results show that the /scan endpoint is able to respond to all requests with a successful status (HTTP 200 OK) consistently, with an average response time of around 5 seconds. This indicates that the BLE device scanning process runs stably, although it is still synchronous. In contrast, the /latest-bp-records and /connect-and-read endpoints experience a fairly high failure rate (errors). On /latest-bp-records, most requests experience delays with an average response time of 24.9 seconds, and some of them fail to be processed. The /connect-and-read endpoint even shows a complete failure (100% error), with an average execution time of 4-5ms (milliseconds), indicating that the BLE connection cannot be opened for parallel requests. This phenomenon is consistent with the test results in Postman, where the failure of simultaneous requests is caused by the limitations of point-to-point BLE communication on the Omron HEM-7142T1 device. The device can only handle one active connection at a time, so when multiple virtual users try to connect simultaneously, other requests are immediately rejected or fail because the device is still in an active session with another user.

IV. Discussion

A. Interpretation of Research Results

Functional testing results demonstrate that the proposed Python-based backend architecture reliably bridges data communication between the Omron HEM-7142T1 commercial medical device and a frontend application via a REST API in a single-user scenario. The success of the /scan, /connect-and-read, and /latest-bp-records endpoints in generating HTTP 200 OK responses indicates that the integration of Bluetooth Low Energy (BLE) with the FastAPI framework has been carried out in accordance with the proposed six-layer architecture design.

Specifically, the stability of the /scan endpoint demonstrates that the BLE device detection process is lightweight and does not require direct interaction with the device's memory. This confirms that the separation of functions in the Service Management Layer, between the scanning process and the data reading process, is an appropriate approach in the medical IoT backend architecture. In contrast, the high failure rate of the /connect-and-read and /latest-bp-records endpoints is due to the point-to-point nature of BLE communication, where the Omron device can only handle one active connection at a time. When multiple Virtual Users simultaneously attempt to access the same device, subsequent requests are rejected or fail because the device enters a busy or fail due to the Omron device's locking mechanism, which activates once an initial connection session is established. This phenomenon confirms that the current API architecture, which still runs synchronously and sequentially, does not yet support parallel multi-user access. Therefore, even if the backend system functions correctly under single-user conditions, system performance will degrade significantly under high-load scenarios due to communication limitations at the BLE hardware layer. Thus, these results confirm that the proposed architecture is successful in terms of system design and interoperability, but is still limited by physical limitations and communication protocols at the device layers (Perception and Network Layers). This is because the BLE transceiver on the Omron device does not support multiplexing of multiple incoming client requests simultaneously, which causes direct connection timeouts when the transceiver is busy.

B. Comparison with Previous Research

Compared to previous studies developing IoT-based blood pressure monitoring systems, most of these studies focused on sensor accuracy and measurement result validation using Omron devices as benchmarks. Studies such as [9], [48] reported systolic and diastolic accuracy rates ranging from 90% to 96% [15], but the system architecture used was still monolithic and device-oriented.

Unlike these approaches, this study does not use Omron as a benchmark device, but rather as the primary, medically certified data source. This approach shifts the research focus from sensor accuracy validation to designing a backend architecture that is interoperable, modular, and ready to integrate with various healthcare application systems.

Furthermore, previous studies generally relied on third-party IoT platforms such as Firebase, Cayenne, or ThingSpeak as the backend layer [13]. While this approach simplifies implementation, it limits architectural flexibility and control over medical data flow. In contrast, this study proposes a standalone backend based on Python and a REST API that enables integration across web and mobile platforms without reliance on proprietary services. Furthermore, a standalone backend provides full control over data processing, reduces reliance on third-party platforms, and ensures higher data integrity and privacy.

C. Research Limitations

Although the proposed architecture was successfully implemented and functionally tested, this study has several limitations. First, the system still relies on synchronous, direct BLE communication with the Omron device, thus not supporting parallel multi-user access. This causes API performance to degrade significantly in high-load scenarios, as demonstrated by load testing results using Postman and Apache JMeter.

Second, the system does not implement a centralized data storage mechanism (database), so each data read request still requires a direct connection to the device. This prevents the system from performing longitudinal

Corresponding author: Tien Fabrianti Kusumasari, tienkusumasari@telkomuniversity.ac.id, Department of Information System, Telkom University, Bandung, Indonesia, Jl. Telekomunikasi No.1, 40257, Bandung, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

health analytics, tracking historical hypertension trends, or integrating with a comprehensive Electronic Medical Record (EHR). This severely limits physicians' ability to make long-term, data-driven clinical decisions.

D. Research Implications

Theoretically, this research contributes to the development of healthcare IoT architectures by demonstrating that certified commercial medical devices can be directly integrated into an API-based backend without the need for additional sensors or recalibration. This broadens the perspective of healthcare IoT research, which has previously focused on hardware development. Practically, the proposed six-layer architecture can serve as a reference for developers of remote health monitoring systems, particularly in designing interoperable backends that are ready to be integrated with various frontend applications. Findings regarding the limitations of point-to-point BLE also provide important insights for developers in designing mitigation strategies, such as message queues that can serialize incoming requests, ensuring that devices only receive one command at a time. Furthermore, data caching can address the lack of centralized storage by temporarily storing the most recent readings, instantly serving subsequent user requests without waking the device.

V. Conclusion

This research has successfully addressed the need for a secure and accurate blood pressure monitoring system architecture by proposing a six-layer IoT architecture that integrates the commercial Omron HEM-7142T1 medical device via the Bluetooth Low Energy (BLE) protocol. The main objective of the research to design a Python-based backend system capable of bridging proprietary data from commercial medical devices into flexible API services has been fully achieved. Based on the test results, it was found that the basic functions on the three main endpoints (/scan, /connect-and-read, and /latest-bp-records) ran successfully with HTTP 200 OK status under single access conditions. However, load testing using Apache JMeter revealed limitations in the hardware point-to-point BLE protocol, where although the /scan endpoint remained stable with a 0% error rate, critical endpoints such as /connect-and-read experienced complete failure with a 100% error rate due to the Omron device's locking mechanism when accessed simultaneously. As a follow-up, future research will focus on integrating a database caching module with a message queue. Using this setup, incoming parallel requests will be queued sequentially to prevent BLE hardware lockups. Once successfully retrieved, the read will be cached, allowing subsequent immediate requests to retrieve this cached data without triggering excessive device connections.

References

[1] A. I. Paganelli *et al.*, “A conceptual IoT-based early-warning architecture for remote monitoring of COVID-19 patients in wards and at home,” *Internet of Things (Netherlands)*, vol. 18, 2022, doi: 10.1016/j.iot.2021.100399.

[2] T. Shaik *et al.*, “Remote patient monitoring using artificial intelligence: Current state, applications, and challenges,” 2023. doi: 10.1002/widm.1485.

[3] N. Sharma *et al.*, “A smart ontology-based IoT framework for remote patient monitoring,” *Biomed. Signal Process. Control*, vol. 68, 2021, doi: 10.1016/j.bspc.2021.102717.

[4] O. Cheikhrouhou, K. Mershad, F. Jamil, R. Mahmud, A. Koubaa, and S. R. Moosavi, “A lightweight blockchain and fog-enabled secure remote patient monitoring system,” *Internet of Things (Netherlands)*, vol. 22, 2023, doi: 10.1016/j.iot.2023.100691.

[5] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian, “Continuous Patient Monitoring with a Patient Centric Agent: A Block Architecture,” *IEEE Access*, vol. 6, 2018, doi: 10.1109/ACCESS.2018.2846779.

[6] A. N. Sari, P. C. N., M. R. Mak’ruf, H. G. Ariswati, Moch. P. A. T. P. T.P., and A. M. Maghfiroh, “Design of Ambulatory Blood Pressure Monitoring for IOT-Based Hypertension Patients,” *Jurnal Teknokes*, vol. 17, no. 1, pp. 37–42, 2024, doi: 10.35882/teknokes.v17i1.646.

[7] C. Li, S. Fan, and H. Li, “Study on the effectiveness of a community hypertension management model based on home smart blood pressure monitoring using IoT technology,” *Front. Public Health*, vol. 12, no. August, pp. 1–11, 2024, doi: 10.3389/fpubh.2024.1428310.

[8] G. Yenurkar, S. Mal, V. O. Nyangaresi, S. Kamble, L. Damahe, and N. Bankar, “Revolutionizing Chronic Heart Disease Management: The Role of IoT-Based Ambulatory Blood Pressure Monitoring System,” *Diagnostics*, vol. 14, no. 12, pp. 1–26, 2024, doi: 10.3390/diagnostics14121297.

[9] P. Tyas, R. Hidayati, and Suharyati, “Development of a Portable Digital System for Vital Sign Measurement using Multi-Sensor Integration,” *Electromedic Journal of Medical Electronic*, vol. 2, no. 2, pp. 86–95, 2025, doi: 10.1109/ELECTROMEDIC.v1.i1.1.

[10] H. Isyanto, A. S. Wahid, and W. Ibrahim, “Desain Alat Monitoring Real Time Suhu Tubuh, Detak Jantung dan Tekanan Darah secara Jarak Jauh melalui Smartphone berbasis Internet of Things Smart Healthcare,” *RESISTOR (Elektronika Kendali Telekomunikasi Tenaga Listrik Komputer)*, vol. 5, no. 1, 2022, doi: 10.24853/resistor.5.1.39-48.

[11] A. J. Puspitasari, E. Endarko, and I. Fatimah, “Blood Pressure Monitor Design Using MPX5050GP Pressure Sensor and Visual C# 2010 Express,” *Jurnal Fisika dan Aplikasinya*, vol.

Corresponding author: Tien Fabrianti Kusumasari, tienkusumasari@telkomuniversity.ac.id, Department of Information System, Telkom University, Bandung, Indonesia, Jl. Telekomunikasi No.1, 40257, Bandung, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

15, no. 3, 2019, doi: 10.12962/j24604682.v15i3.4929.

[12] I. G. M. N. Desnanjaya and I. M. A. Nugraha, "Real-time monitoring system for blood pressure monitoring based on internet of things," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 35, no. 1, p. 62, Jul. 2024, doi: 10.11591/ijeecs.v35.i1.pp62-69.

[13] A. Sukarno, A. Marwanto, and S. Alifah, "Vital Sign Monitoring in ICU Patients Based on MEWS (Modified Early Warning Score) with IOT (Internet of Things)," *Journal of Telematics and Informatics (JTI)*, vol. 7, no. 4, 2019.

[14] A. Wardhani, T. Anggraini, Y. Yultrisna, Z. Hendri, and R. Nandika, "Alat Monitoring Pengukuran Tekanan Darah Portable Dengan Output Suara Berbasis Iot," *SIGMA TEKNIKA*, vol. 6, no. 2, 2023, doi: 10.33373/sigmateknika.v6i2.5676.

[15] M. K. Baiduri, D. Perdana, and S. Sussi, "Blood Pressure Measuring Device for Hypertension Monitoring Based on Internet of Things with E-KTP Authentication," in *International Conference on Intelligent and Advanced Systems: Enhance the Present for a Sustainable Future, ICIAS 2021*, 2021. doi: 10.1109/ICIAS49414.2021.9642538.

[16] K. D. Farmalkes, "Izin Edar Omron Hem-7142t1," Direktorat Jenderal Farmasi dan Alat Kesehatan Kementerian Kesehatan Republik Indonesia. Accessed: Sep. 24, 2025. [Online]. Available: https://infoalkes.kemkes.go.id/#home/cari/frmCari/3/hem-7142/1758562569489

[17] W. Shi *et al.*, "Non-Invasive Blood Pressure Monitoring Using a Single-Channel PPG Sensor With Adaptive Kalman Algorithm and 4-LED Arrayed Structure," *IEEE Trans. Instrum. Meas.*, vol. 73, 2024, doi: 10.1109/TIM.2024.3480192.

[18] F. Yadegari and A. Asosheh, "A unified IoT architectural model for smart hospitals: enhancing interoperability, security, and efficiency through clinical information systems (CIS)," *J. Big Data*, vol. 12, no. 1, 2025, doi: 10.1186/s40537-025-01197-4.

[19] I. Zyrianoff *et al.*, "Architecting and deploying IoT smart applications: A performance–oriented approach," *Sensors (Switzerland)*, vol. 20, no. 1, 2020, doi: 10.3390/s20010084.

[20] W. Smith, B. M. Colbert, T. Namouz, D. Caven, J. A. Ewing, and A. W. Albano, "Remote Patient Monitoring Is Associated with Improved Outcomes in Hypertension: A Large, Retrospective, Cohort Analysis," *Healthcare (Switzerland)*, vol. 12, no. 16, 2024, doi: 10.3390/healthcare12161583.

[21] F. Chu, A. Stark, A. Telzak, and S. Rikin, "Patient Experience in a Remote Patient Monitoring Program for Hypertension: A Qualitative Study," *Am. J. Hypertens.*, vol. 37, no. 11, 2024, doi: 10.1093/ajh/hpae086.

[22] S. H. Park, J. H. Shin, J. Park, and W. S. Choi, "An updated meta-analysis of remote blood pressure monitoring in urban-dwelling patients with hypertension," 2021. doi: 10.3390/ijerph182010583.

[23] M. Makutonin, J. Dare, M. Heekin, A. Salancy, C. Hood, and L. W. Dominguez, "Remote Patient Monitoring for Hypertension: Feasibility and Outcomes of a Clinic-Based Pilot in a Minority Population," *J. Prim. Care Community Health*, vol. 14, 2023, doi: 10.1177/21501319231204586.

[24] D. Bernard, C. Msigwa, and J. Yun, "Toward IoT-Based Medical Edge Devices: PPG-Based Blood Pressure Estimation Application," *IEEE Internet Things J.*, vol. 10, no. 6, 2023, doi: 10.1109/JIOT.2022.3222477.

[25] R. Zulfiyani, A. W. Dani, and F. Sirait, "Rancang Bangun Alat Pengukur Tekanan Darah Untuk Deteksi Tingkat Risiko Cardiovascular Disease Dengan Metode Fuzzy Logic Mamdani Berbasis IoT," *Jurnal Teknologi Elektro*, vol. 16, no. 2, p. 66, Jul. 2025, doi: 10.22441/jte.2025.v16i2.001.

[26] M. Gusnam *et al.*, "Sistem Sistem Monitoring Telemedis untuk Pengukuran Tekanan Darah Berbasis Internet of Things (IoT) menggunakan MPX5700DP dan Jaringan LoRaWAN," *Journal of Applied Electrical Engineering*, vol. 8, no. 2, 2024, doi: 10.30871/jaee.v8i2.8853.

[27] A. Shumba, T. Montanaro, I. Sergi, L. Fachechi, M. De Vittorio, and L. Patrono, "Leveraging IoT-Aware Technologies and AI Techniques for Real-Time Critical Healthcare Applications," *Sensors*, vol. 22, no. 19, p. 7675, Oct. 2022, doi: 10.3390/s22197675.

[28] P. Lopes de Souza, W. Lopes de Souza, L. Ferreira Pires, J. L. R. Moreira, and R. R. Ciferri, "Ontology Engineering of an IoT System for Monitoring Hypertension," in *Lecture Notes in Business Information Processing*, 2024. doi: 10.1007/978-3-031-64755-0_3.

[29] A. R. Aniko, T. F. Kusumasari, S. Suakanto, and M. I. Fadilah, "User Experience Development in Elderly Heart Patient Monitoring System," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 14, no. 4, pp. 281–287, 2025, doi: 10.22146/jnteti.v14i4.18783.

[30] J. vom Brocke, A. Hevner, and A. Maedche, "Introduction to Design Science Research," 1st ed., no. September, J. vom Brocke, A. Hevner, and A. Maedche, Eds., Springer, 2020, pp. 1–19. doi: 10.1007/978-3-030-46781-4_1.

[31] B. Gledson, K. Rogage, A. Thompson, and H. Ponton, "Reporting on the Development of a Web-Based Prototype Dashboard for Construction

Corresponding author: Tien Fabrianti Kusumasari, tienkusumasari@telkomuniversity.ac.id, Department of Information System, Telkom University, Bandung, Indonesia, Jl. Telekomunikasi No.1, 40257, Bandung, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

Design Managers, Achieved through Design Science Research Methodology (DSRM)," *Buildings*, vol. 14, no. 2, p. 335, Jan. 2024, doi: 10.3390/buildings14020335.

[32] V. de Castro, M. L. Martín-Peña, E. M. Martínez, and M. Salgado, "Combining Action Research With Design Science as a Qualitative Research Methodology. An Application to Service (Operations) Management Research," *International Journal of Qualitative Methods* , vol. 24, pp. 1–15, 2025, doi: 10.1177/16094069241312018.

[33] T. Tuunanen, R. Winter, and J. vom Brocke, "Dealing With Complexity in Design Science Research: a Methodology Using Design Echelons1," *MIS Q.*, vol. 48, no. 2, pp. 427–458, 2024, doi: 10.25300/MISQ/2023/16700.

[34] I. Natgunanathan, N. Fernando, S. W. Loke, and C. Weerasuriya, "Bluetooth Low Energy Mesh: Applications, Considerations and Current State-of-the-Art," 2023. doi: 10.3390/s23041826.

[35] R. Librianty, "Faktor-Faktor Yang Mempengaruhi Hidroterapi (Rendam Kaki Air Hangat) Terhadap Tekanan Darah Pada Pasien Hipertensi," *Jurnal Kesehatan Bhakti Husada*, vol. 10, no. 2, p. 34, Dec. 2024, doi: 10.37848/jurnal.v10i2.218.

[36] A. B. Gultom, A. Batubara, and A. Afniwati, "Efek Relaksasi Otot Progresif Disertai Musik Klasik Trhadap Stress Dan Tekanan Darah Pasien Hipertensi Secara Cepat," *Jurnal Abdi Insani*, vol. 11, no. 2, pp. 1553–1560, 2024, doi: 10.29303/abdiinsani.v11i2.1595.

[37] N. Gavrilović and A. Mishra, "Software architecture of the internet of things (IoT) for smart city, healthcare and agriculture: analysis and improvement directions," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 1, 2021, doi: 10.1007/s12652-020-02197-3.

[38] N. Kamal and P. Ghosal, "Three tier architecture for IoT driven health monitoring system using raspberry Pi," in *Proceedings - 2018 IEEE 4th International Symposium on Smart Electronic Systems, iSES 2018*, 2018. doi: 10.1109/iSES.2018.00044.

[39] Z. U. Armaya'u, M. M. Gumel, and H. S. Tuge, "Comparing Flowchart and Swim Lane Activity Diagram for Aiding Transitioning to Object-Oriented Implementation," *American Journal of Education and Technology*, vol. 1, no. 2, 2022, doi: 10.54536/ajet.v1i2.612.

[40] I. Tarsini and R. Anggraeni, "Explore flowchart and pseudocode concepts in algorithms and programming," *Indonesian Journal of Multidisciplinary Science*, vol. 3, no. 5, 2024, doi: 10.55324/ijoms.v3i5.807.

[41] O. Baniaș, D. Florea, R. Gyalai, and D. I. Curiac, "Automated specification-based testing of REST APIs," *Sensors*, vol. 21, no. 16, 2021, doi: 10.3390/s21165375.

[42] A. Nugraheni and M. Maryam, "Penerapan Teknologi Quick Response Code Dan Application Programming Interface Pada Perancangan Aplikasi Perpustakaan (Studi Kasus : Smp Negeri 25 Surakarta)," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 7, no. 3, 2022, doi: 10.29100/jipi.v7i3.3096.

[43] S. A. B. C. Busro *et al.*, "Rancangan Pembuatan API Website Data Tanaman Obat Dan Langka Kabupaten Kediri," *Bulletin of Information Technology (BIT)*, vol. 3, no. 4, 2022, doi: 10.47065/bit.v3i4.373.

[44] F. Subhan *et al.*, "Experimental analysis of received signals strength in Bluetooth Low Energy (BLE) and its effect on distance and position estimation," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 2, 2022, doi: 10.1002/ett.3793.

[45] A. Strzoda, K. Grochla, and K. Połys, "Variability of BLE Advertisement Packets Received Signal Strength and Delivery Probability in the Presence of Interferences," in *DIVANet 2022 - Proceedings of the 12th ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, 2022. doi: 10.1145/3551662.3560933.

[46] Y. Abuzrieq, A. A. S. Ahmad, and M. B. Younes, "An experimental performance evaluation of cloud-API-based applications," *Future Internet*, vol. 13, no. 12, 2021, doi: 10.3390/fi13120314.

[47] A. Golmohammadi, M. Zhang, and A. Arcuri, "Testing RESTful APIs: A Survey," *ACM Transactions on Software Engineering and Methodology*, vol. 33, no. 1, 2023, doi: 10.1145/3617175.

[48] K. Shihab, D. Perdana, and S. Sussi, "Design and Implementation of IoT Based Blood Pressure Monitoring Tools," *International journal of simulation: systems, science & technology*, 2020, doi: 10.5013/ijssst.a.21.01.03.

AUTHOR BIOGRAPHY

Tien Fabrianti Kusumasari is a Senior Lecturer in the Department of Information Systems at Telkom University, Indonesia. She began her academic journey by earning a Bachelor's degree in Chemical Engineering from the Institut Teknologi Bandung (ITB) in 2001. Shifting her focus to information technology, she completed her Master's degree in Informatics in 2008 and subsequently obtained her

Corresponding author: Tien Fabrianti Kusumasari, tienkusumasari@telkomuniversity.ac.id, Department of Information System, Telkom University, Bandung, Indonesia, Jl. Telekomunikasi No.1, 40257, Bandung, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

Doctoral degree in Electrical Engineering and Informatics in 2016, both from the same institution. Currently, Dr. Kusumasari is an integral member of the Digital Enterprise System and Technology (DIGEST) research group. Her research expertise spans enterprise systems, digital transformation, and information management. With a strong multidisciplinary background, she is dedicated to advancing digital enterprise solutions and has contributed extensively to academic research and the development of intelligent information systems within the Indonesian higher education landscape.

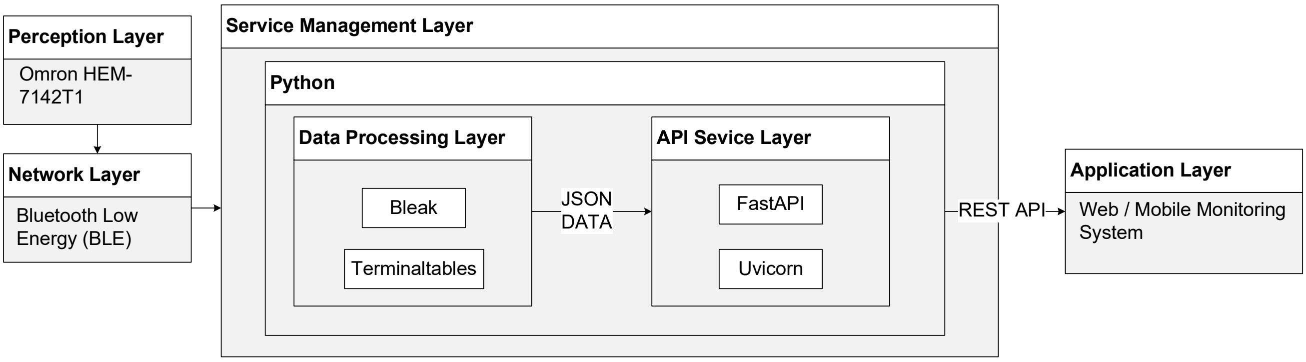
Yudhi Widyatama is a seasoned Information Technology practitioner at PT Telkom Indonesia (Persero) Tbk. He earned his Bachelor's degree in Electrical Engineering from the Institut Teknologi Bandung (ITB) in 2001. Committed to advancing his technical expertise, he continued his postgraduate studies at the same institution, receiving his Master's degree in Information Technology in 2006. Currently, he serves as the Finance & Asset Management Platform Manager within the Information Technology Division of PT Telkom Indonesia. In this strategic role, he oversees the development and integration of enterprise-scale digital platforms. His professional expertise encompasses IT strategic governance, enterprise architecture, and the implementation of robust information systems for corporate resource management. His contribution to this research leverages his extensive industrial experience in managing complex IT infrastructures and digital transformation initiatives within Indonesia's telecommunications sector.

Alaric Rasendriya Aniko is earned his Bachelor's degree (S.Kom.) in Information Systems from Telkom University in 2023. He is currently pursuing his Master's degree (M.Kom.) in Information Systems at the same institution. Alongside his postgraduate studies, Alaric actively serves as a teaching and laboratory assistant specializing in software development, bridging the gap between academic theory and industrial practice. He possesses advanced technical expertise in modern web and mobile frameworks, with a strong proficiency in Laravel, ReactJS, React Native, Python, and NodeJS (ExpressJS). His involvement in various development projects as an assistant has further refined his skills in building scalable software solutions. His research interests focus on software engineering, digital ecosystem development, and the implementation of robust backend architectures for IoT-integrated systems. Through his academic and practical contributions, he aims to advance the field of software development within the information systems domain.

Sinung Suakanto is a Senior Lecturer in the Department of Information System at Telkom University, Indonesia. He earned both his Bachelor's and Master's degrees in Electrical Engineering from the Institut Teknologi Bandung (ITB). He furthered his academic career by obtaining a Doctoral degree in Electrical Engineering and Informatics from the same institution in 2013. Dr. Suakanto is an active member of the Digital Enterprise System and Technology (DIGEST) research group. His research contributions are channeled through the Enterprise Intelligent System Development (EISD) Research Laboratory and the Center of Excellence for Integrated Management Systems and Solutions (CoE IMSS). His expertise lies in enterprise intelligent systems, digital transformation, and advanced software development. Throughout his career, he has focused on bridging academic research with industrial needs, specifically in developing scalable digital enterprise solutions and IoT-based architectures.

Corresponding author: Tien Fabrianti Kusumasari, tienkusumasari@telkomuniversity.ac.id, Department of Information System, Telkom University, Bandung, Indonesia, Jl. Telekomunikasi No.1, 40257, Bandung, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

Each author is required to provide a cover image as shown example below



Corresponding author: Triwiyanto, triwi@poltekkesdepkes-sby.ac.id, Department of Medical Electronics Technology, Poltekkes Kemenkes Surabaya, Jl. Pucang Jajar Timur No. 10, 60282, Surabaya, Indonesia (change this part according to yours).
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407