Jurnal ELTIKOM :
Jurnal Teknik Elektro, Teknologi Informasi dan Komputer

Vol. 9, No. 1, June 2025, page. 1-10
ISSN 2598-3245 (Print), ISSN 2598-3288 (Online)
DOI: http://doi.org/10.31961/eltikom.v9i1.1363
Available online at http://eltikom.poliban.ac.id

INTEGRATION OF PM1200 AND IOT FOR ELECTRICAL ENERGY MONITORING WITH WEB-BASED MAP VISUALIZATION

Syafriyadi Nor, Zaiyan Ahyadi*
Departement of Electrical Engineering, Politeknik Negeri Banjarmasin, Banjarmasin, Indonesia
e-mail: syafriayadi.nor@poliban.ac.id, z_ahyadi@poliban.ac.id

Received: 8 November 2024 – Revised: 26 December 2024 – Accepted: 6 January 2025

ABSTRACT

This study aims to integrate the PM1200 device with Internet of Things (IoT) technology using the Modbus protocol to enable real-time monitoring of electrical energy. The current challenge lies in the limited flexibility of energy monitoring, which is typically restricted to local access and lacks map-based visualizations. To address this, the system integrates interactive maps to provide a clearer and more comprehensive view of energy distribution across different locations. This study seeks to offer an effective energy monitoring solution with data visualized through maps on an interactive web platform. The methodology includes reading data from the PM1200 device via the Modbus protocol, transmitting it to an IoT platform using the MQTT protocol, and displaying the data as maps on a web interface. The findings are expected to support effective energy monitoring and enhance energy management efficiency.

Keywords: *electrical energy, IoT, PM1200, web.*

I. INTRODUCTION

MONITORING electrical energy consumption is essential for effective energy management, especially as energy demand continues to grow and awareness of energy efficiency increases. Proper monitoring can help reduce energy costs, improve environmental sustainability, and enhance user comfort [1]. This situation highlights the need for efficient energy monitoring technologies. Various energy monitoring devices are available on the market, including the PM1200 from Schneider Electric. The PM1200 is a digital power meter capable of measuring voltage, current, power, and frequency with high accuracy. However, the PM1200 only provides local data and is not integrated with Internet of Things (IoT) technology or interactive map-based data visualization. This limitation makes it difficult for users to obtain real-time information, manage energy distribution across multiple locations, and analyze energy consumption trends comprehensively.

Web-based energy monitoring offers several advantages in addressing these challenges. A web platform can be accessed from any device with a web browser, such as computers, tablets, or smartphones, regardless of the operating system. This accessibility makes the monitoring system more flexible and user-friendly, allowing users to monitor energy usage from anywhere. Additionally, the availability of readily web development resources and tools for map-based visualization enables the creation of customizable interfaces tailored to specific user needs. Therefore, integrating energy monitoring systems with web-based platforms and interactive maps can provide a more effective and scalable solution for energy management.

IoT-based technology has made increasingly integrated energy management systems possible. Various studies have developed IoT-based energy monitoring systems. For example, the study by [2] monitors electrical power through a web interface but lacks map-based visualization for specific monitoring locations. Similarly, research in [3] utilizes a web interface for automatic monitoring and control of electrical loads using a Raspberry Pi 3 and PZEM-004T sensors, yet it does not include the capability to map energy data to physical locations interactively. Another study [4] employs the PM1200 device in an IoT-based monitoring system to observe air compressor conditions via a panel and web interface; however, its monitoring scope remains limited to specific devices and does not support broader location visualization or interactive maps. Some previous studies have also designed IoT-based systems to control and monitor electrical energy. Research [5] developed a tool to control four electrical devices and monitor electrical power via the Internet, although it still relied on HTTP protocols. Other studies [6] [7] used platforms such as Blynk and ThingSpeak to monitor electrical energy with sensors like LDR and PZEM-004T, employing ESP8266 and NodeMCU modules. However, these systems featured only text- or graph-based interfaces and lacked interactive map displays.

Some studies have also explored the use of Android applications as interfaces for energy monitoring. For example, Santos [9] developed Android applications integrated with IoT technologies to monitor water and electricity consumption, allowing homeowners and renters to access the data. Jokanan [10] used Firebase as a server and the Blynk platform to build an energy monitoring tool. Similarly, [11] designed a system to track real-time electricity usage using the Cayenne interface. These studies utilize Android applications to monitor electricity through IoT technologies.

While these approaches contribute to the development of IoT-based energy monitoring systems, they do not integrate map-based visualization for real-time monitoring. Research that combines real-time monitoring with interactive map-based visualization on web platforms remains limited, revealing a gap that warrants further investigation. This study aims to develop an IoT-based electrical energy monitoring system with real-time, map-based visualization on a web platform. The system enables users to interactively monitor energy data at specific locations, enhancing their understanding of usage patterns at each site. Consequently, this study contributes to the advancement of visual distributed energy monitoring and improves the overall efficiency of energy management.

II. RESEARCH METHOD

This section outlines the systematic steps taken to design, implement, and test a monitoring system for voltage, current, and frequency data using Internet of Things (IoT) technology. The system utilizes the NodeMCU ESP32 microcontroller, an RS485 module, and the Schneider PM1200 power meter. The MQTT protocol is selected for data transmission due to its low latency and efficient bandwidth usage, making it more suitable for real-time monitoring than traditional HTTP-based methods. The RS485 module ensures robust and noise-resistant communication between the PM1200 and the ESP32, enabling accurate data acquisition.

Traditional systems typically rely on HTTP-based communication, which often results in higher latency and greater bandwidth consumption due to frequent polling. In contrast, the MQTT protocol supports real-time, event-driven communication with significantly lower overhead. This enhances transmission speed and optimizes network resource usage—an essential requirement for IoT applications involving continuous data updates, such as energy monitoring. Compared to previous methods that used Arduino-based systems and HTTP protocols, the combination of MQTT and the NodeMCU ESP32 offers improved scalability and flexibility.

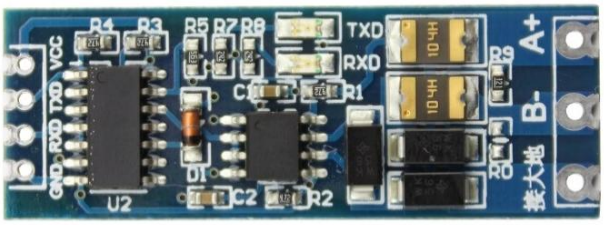

Figure 1. TTl to RS485 module

Figure 2. Schneider PM1200 Power Meter

A. Component Description

Abstracts should be explained at the beginning of the manuscript. The abstract section must clearly state the research's background, problems, objectives, results, and conclusions. The Introduction section must explicitly state the problem, update, and research objectives. The introduction must also be equipped with state-of-the-art research accompanied by the latest primary library sources.

1) NodeMCU ESP32

The ESP32 microcontroller serves as the core of the system, managing data acquisition and transmission. It features a Tensilica LX6 dual-core processor running at 240 MHz and supports Wi-Fi and Bluetooth communication, allowing seamless integration with IoT services [12].

2) TTL to RS485 module

The RS485 module functions as the communication interface between the ESP32 and the Schneider PM1200 power meter. It operates at both 3.3V and 5V and requires only two pins (RX and TX) for connectivity [13].

Figure 1 shows the TTL to RS485 module, which serves as the communication interface between the ESP32 and the PM1200. This module supports data transmission using the Modbus protocol, which is essential for acquiring electrical measurement data from the PM1200.

3) Schneider PM1200 power meter

The Schneider Electric PM1200 power meter is designed to accurately measure key electrical parameters. According to the data sheet [14], the PM1200 is capable of measuring voltage, current, and frequency, and supports Modbus communication via the RS-485 protocol. Figure 2 shows the Schneider Electric PM1200 power meter, which measures voltage, current, and frequency. The device transmits this data to the ESP32 microcontroller using the Modbus protocol for further processing.

B. Device Setup

1) Component connection:

The NodeMCU ESP32 connects to the RS485 module through UART2 communication pins—GPIO16 (RX2) and GPIO17 (TX2). The RS485 module is then linked to the PM1200 power meter using a twisted pair cable to support RS-485 communication. Figure 3 illustrates the schematic circuit connecting the NodeMCU ESP32 to the RS485 module, which interfaces with the PM1200 power meter. In this configuration:

- The ESP32 uses its UART2 communication pins (GPIO16 for RX2 and GPIO17 for TX2) to establish a bidirectional serial connection with the RS485 module. Using UART2 ensures that UART0 remains available for debugging.
- The RS485 module acts as an intermediary, enabling noise-resistant and reliable communication between the ESP32 and the PM1200, especially over longer cable distances.
- The PM1200 power meter provides real-time electrical data, including voltage, current, power, and frequency, via the Modbus RTU protocol.

This configuration is critical for acquiring electrical measurements directly from the PM1200 and transmitting the data to the ESP32 for processing.

2) Power meter installation:

The PM1200 power meter is installed in the electrical system connected to the panel located in the microprocessor laboratory. Voltage and current connections have been made according to the installation guidelines provided in the PM1200 data sheet.

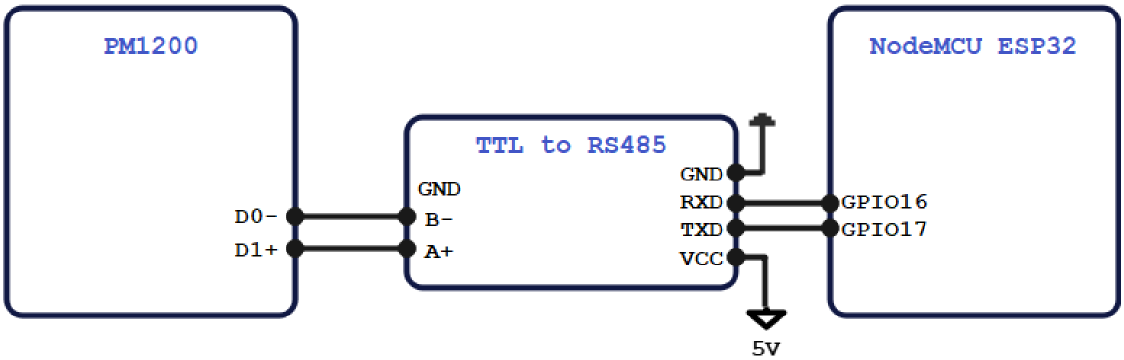

Figure 3. Schematic circuit

TABLE 1
PM1200 PARAMETER REGISTER ADDRESSES

Parameter	Description	Address	Data Type
W	Active power, total	3903	Float
VLN	Line-to-neutral voltage	3911	Float
A	Current average	3913	Float
F	Frequency, Hz	3915	Float

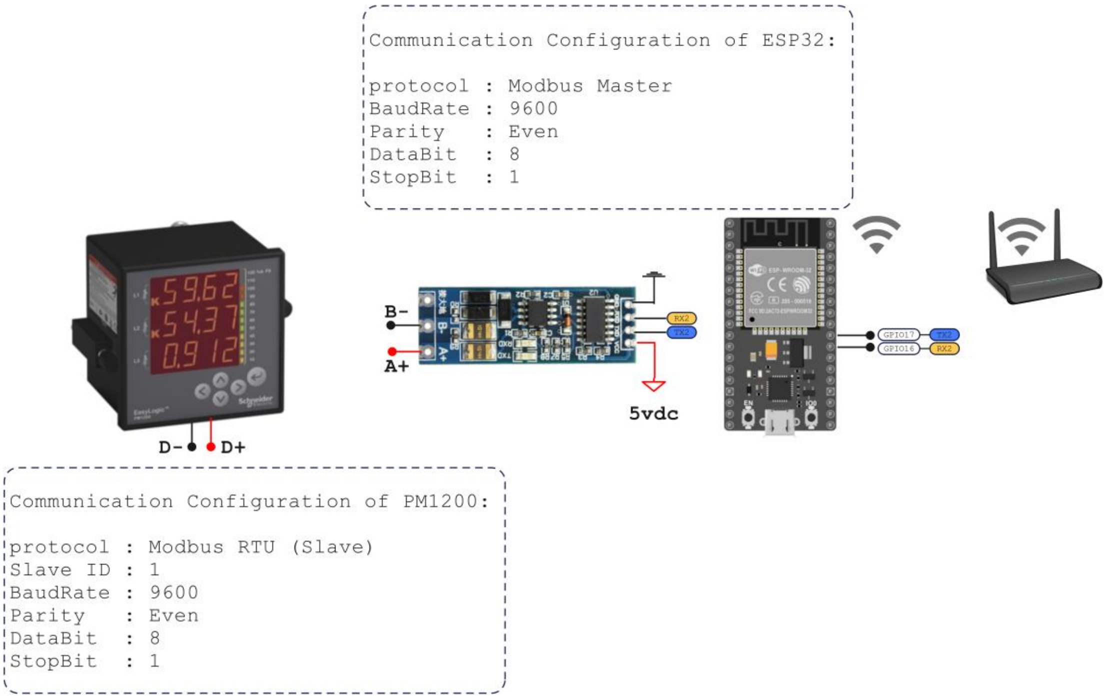

Figure 4. Inter-device communication protocols and settings

C. *Data collection and transmission via IoT*

1) Programming the ESP32:

The ESP32 is programmed using the Arduino IDE. In this implementation, communication with the RS485 module is established using the *ModbusMaster* library developed by Doc Walker [15] to read data from the PM1200 power meter. The successfully retrieved data is then stored in JSON (JavaScript Object Notation) format using the *ArduinoJson* library [16], and transmitted via the MQTT protocol using the *PubSubClient* library developed by Nick O'Leary [17]. These three libraries enable reliable and efficient integration of the ESP32, RS485 module, and MQTT protocol in this study.

2) Communication protocol:

The data collected from the PM1200 is encapsulated using the Modbus protocol and transmitted through the RS485 module. It is then forwarded to the server using the MQTT protocol.

Figure 4 shows the communication settings for each connected device, which are essential to ensure successful data acquisition from the PM1200 and transmission to the MQTT broker.

3) Register Address:

This study uses the *Read Holding Registers* function code (03) to access specific register addresses in the PM1200 operating under the Modbus protocol. Using this function, three standard measurement parameters—line-to-neutral voltage, current, and frequency—are retrieved. These parameters are defined in the register addresses listed in the PM1200 User Manual [18], as shown in Table 1. Table 1 presents the register addresses for PM1200 parameters, including line-to-neutral voltage (VLN), average current (A), active power (W), and frequency (F). Accessing these addresses allows the system to efficiently read the respective parameters and transmit the data to the server via the MQTT protocol.

D. *Data processing and storage*

1) IoT Server:

The data received by the IoT server is processed and stored in a MySQL database. The IoT platform includes a web application interface integrated with map visualization. This implementation uses the

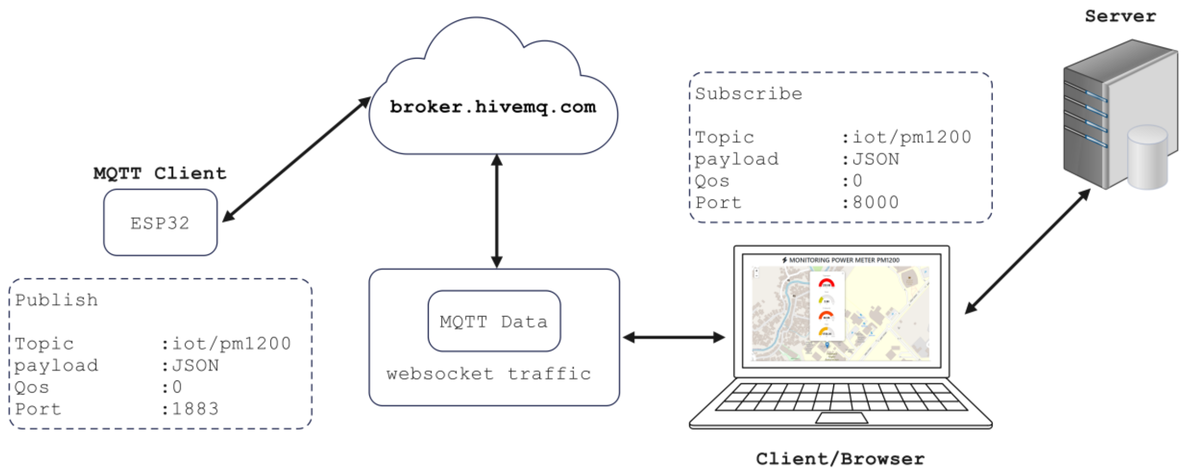

Figure 5. MQTT and WebSocket Architecture

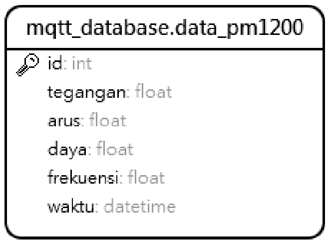

Figure 6. Entity-Relationship Diagram (ERD)

Paho JavaScript - MQTT Client library, which is designed for web browsers and utilizes WebSockets technology [19]. This library enables efficient and responsive communication with the MQTT broker, supporting real-time monitoring of electricity data through a WebSocket-based connection.

Figure 5 illustrates the data transmission process using MQTT and WebSocket. facilitates real-time, bidirectional communication between the client (web browser) and the server without requiring page refreshes or polling [20]. The MQTT broker embeds MQTT data within the WebSocket framework and transmits it to the web client. The client then extracts the MQTT packets from the WebSocket, processes them, and displays the data on gauges. Simultaneously, the data is stored in the database.

2) Data visualization in map form on the web:

A web interface was developed to visualize the measurement parameters on an interactive map. The monitoring point is located at Building K, Microprocessor Laboratory, Banjarmasin State Polytechnic, with geographic coordinates -3.295736490152364, 114.58224174357282, as shown on OpenStreetMap.

3) Data Storage in Database

The data collected from the power meter and the interactive monitoring system is stored centrally in a MySQL database. The database is named *mqtt_database*, and the table used is *data_pm1200*, designed to maintain data integrity and optimize accessibility. Each data entry is timestamped to support continuous monitoring and historical analysis.

Figure 6 presents the Entity-Relationship Diagram (ERD) of the database implemented in this system. The diagram outlines key entities, including:

- Data Table: stores real-time electrical data obtained from the PM1200 power meter. Its primary attributes include:
- Timestamp: records the exact time each data point is collected, which is essential for real-time analysis and historical tracking.
- Voltage, current, power, and frequency: represent the monitored electrical parameters, enabling detailed evaluation of energy usage patterns.

E. Power meter reading

Data acquisition begins by accessing the Schneider PM1200 power meter through a serial communication interface using the Modbus protocol. This protocol enables the ESP32 to send requests to specific register addresses on the power meter and receive responses containing the required data. The data collected includes line-to-neutral voltage, current, power, and frequency, based on the register table assigned to each measured parameter.

Since the Modbus protocol transmits floating-point data in 32-bit format, two 16-bit registers must be read for each parameter. When the PM1200 sends data to the ESP32, these two registers must be combined to form a complete 32-bit floating-point value. After successful retrieval, the ESP32 converts the data into readable units—volts for voltage, amps for current, watts for power, and hertz for frequency. For example, to read line-to-neutral voltage from register 3911, the system must read registers 3910 and 3911. These values are stored in a 16-bit array and then combined to form a complete float value.

This concatenation process uses bitwise operations to merge the two 16-bit registers into a single 32-bit floating-point number. The code snippet below illustrates how voltage data is read from registers 3910 and 3911:

```
// Get data Line to Netral
data[0] = node.getResponseBuffer(0); // register 3910 (indeks 0)
data[1] = node.getResponseBuffer(1); // register 3911 (indeks 1)
// Combines two registers into float form
uint32_t rawPhaseData = (data[1] << 16) | data[0];
float phase;
// Get data from rawPhaseData and copy it to phase
memcpy(&phase, &rawPhaseData, sizeof(float));
Serial.println(phase);
```

Other electrical parameters can be read in the same manner, based on the register table defined for each measurement.

F. Data Transmission to MQTT Broker

The data acquired from the PM1200 device via the ESP32 using the Modbus protocol is transmitted to the monitoring system using the MQTT protocol. This allows electrical energy data—such as voltage, current, power, and frequency—to be efficiently sent to the IoT infrastructure. Using the JSON format, the publisher sends data under a single topic, which is then forwarded to the MQTT broker. The subscriber receives the data, which is parsed and used for various purposes, including real-time monitoring. The use of MQTT facilitates seamless, real-time data integration and distribution.

Table 2 presents the average latency measurement, which is approximately 255.9 milliseconds—the time between the publisher sending a message and the subscriber receiving it in the MQTT protocol. This low latency enables responsive data monitoring. However, several factors may influence MQTT latency, including internet connection quality, the broker used, and message size.

G. Dashboard display on web application

A web application was developed to display real-time data from the PM1200 device via an interactive dashboard. The application uses map-based visualization with the Leaflet library [21] to show measurements of current, voltage, power, and frequency. Since web browsers cannot directly connect to MQTT, WebSocket serves as the intermediary, enabling real-time data flow from the MQTT broker to the web interface efficiently and effectively.

The map shown in Figure 7 is the actual output of the implemented system. It is generated using the OpenStreetMap platform and integrated into the web interface developed for this study. The dashboard employs WebSocket technology to dynamically display data received from the MQTT broker in real time. Electrical parameters—voltage, current, frequency, and power—are measured by the PM1200 power meter and read by the ESP32 microcontroller via Modbus RTU communication. The data is then transmitted to the server using the MQTT protocol.

On the client side, the system retrieves the data from the MQTT broker and displays it in real time through WebSocket communication. When a marker on the map is clicked, the system presents the data using gauges, allowing users to view detailed energy information—voltage, current, frequency, and power—for a specific location. In this study, the monitored location is the Microprocessor Laboratory.

This map-based interface offers several advantages for energy monitoring and management. By combining real-time data with geographic visualization via OpenStreetMap, users can spatially assess energy

TABLE 2
PM1200 PARAMETER REGISTER ADDRESSES

Payload	Latency Average
{ "voltage": 225.1648102, "current": 2.876006603, "power": 647.5754808984, "frequency": 50.07337189 }	255.9 milliseconds (ms)

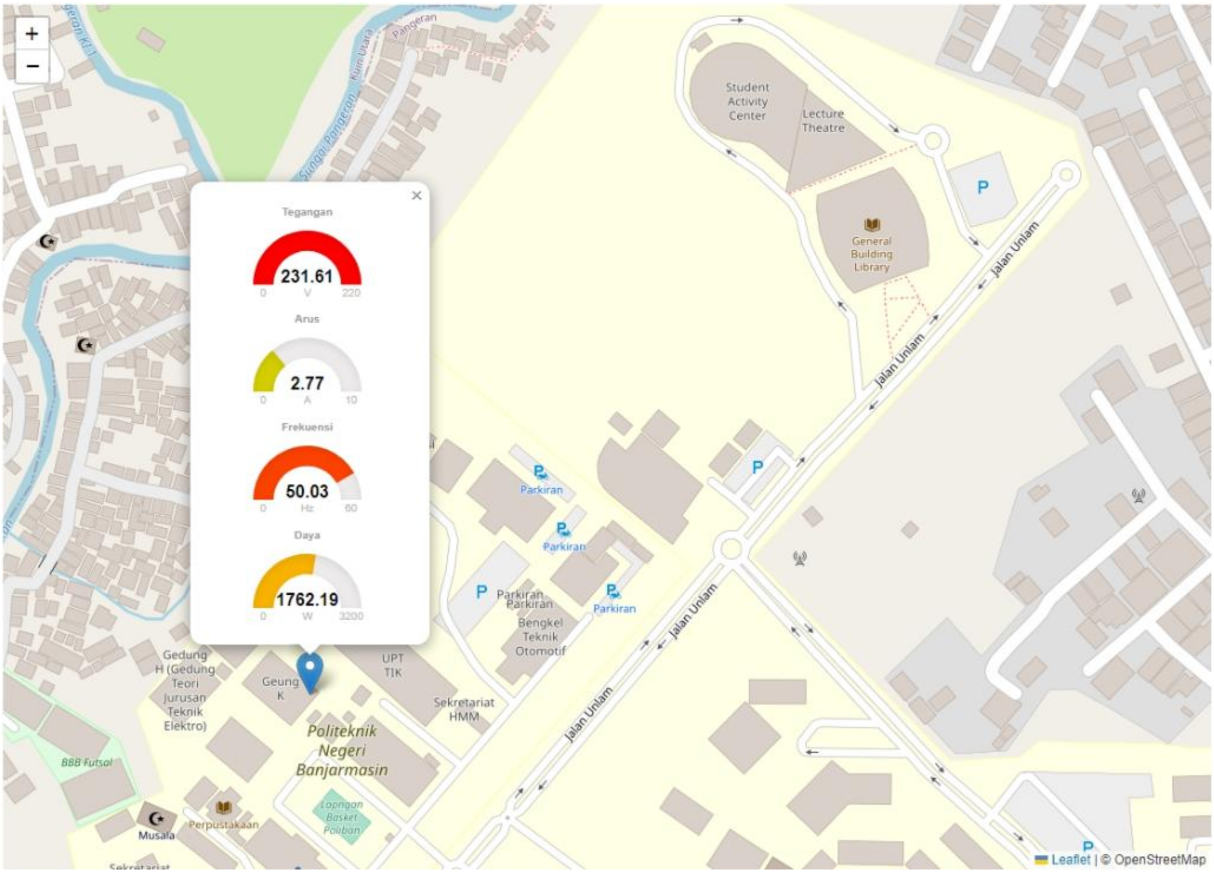

Figure 7: Dashboard view on the web

conditions and detect parameter changes effectively. The dashboard's responsive design ensures accessibility across various devices, enabling remote monitoring from desktops, tablets, or smartphones.

The system demonstrates the practical application of map-based visualization for real-time energy monitoring, allowing users to focus on specific sites and make informed decisions. Unlike static images, the map functions as an active component of the monitoring system and is designed to support future upgrades, including multi-location monitoring and additional data layers.

The methodology integrates key components to create a scalable and efficient energy monitoring system. The PM1200 power meter collects voltage, current, power, and frequency data, which is transmitted via the RS485 module to the ESP32 microcontroller for processing. The data is then sent to the server using the MQTT protocol, ensuring low-latency, real-time communication.

Once received by the IoT server, the data is stored in a MySQL database and displayed on an interactive map through the web interface. This system enables real-time monitoring while offering flexibility and scalability for future enhancements.

III. RESULT AND DISCUSSION

The results of the study are presented below and outlined in the following sections.

A. Data Analysis

An overview of the data characteristics for each variable is shown in Figure 8, including the central tendency (mean and median), dispersion (standard deviation), and percentiles for each measurement.

TABLE 3
STATISTICAL DESCRIPTION

	Voltage	Current	Power	Frequency
mean	230.705366	2.688822	1694.916402	50.005403
std	1.279127	0.278342	183.616369	0.051659
min	227.343000	1.601030	985.042000	49.874200
25%	229.556000	2.742472	1729.412500	49.969700
50%	231.430000	2.758705	1740.760000	50.013400
75%	231.824000	2.772825	1753.867500	50.046450
max	232.106000	3.460970	1907.560000	50.104400

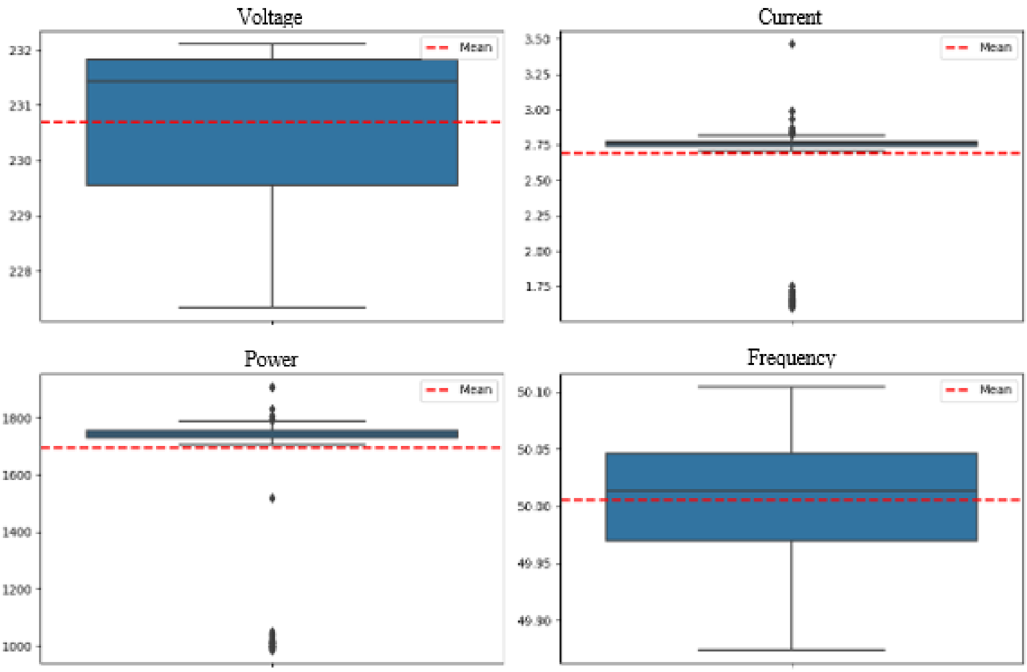

Figure 8. Data distribution and basic statistics for each variable

The statistical summary in Table 3 provides key insights into the characteristics of the collected measurements. The voltage has a mean value of approximately 230.71 V, with low variability indicated by a standard deviation of about 1.28 V. The current shows an average of around 2.69 A and a smaller standard deviation of about 0.28 A. The average power is approximately 1694.92 W, showing greater variability with a standard deviation of around 183.62 W. The frequency is highly stable, with a mean of about 50.01 Hz and a standard deviation of 0.05 Hz. These results suggest that voltage, current, and frequency are relatively consistent, while power readings exhibit more fluctuation.

This study demonstrates that integrating the PM1200 with IoT and web-based map visualization enables real-time, effective, and accurate monitoring of electrical energy. The processed and visualized data, displayed as maps, provides a clearer understanding of energy usage patterns at specific locations.

Compared to previous systems such as ThingSpeak, Blynk, and those using the HTTP protocol, the proposed system offers enhanced flexibility and customizability. While those platforms provide basic monitoring capabilities, they lack the adaptability offered by this system.

- Advantages of the Proposed System: The main advantages lie in its high degree of customizability and the ease with which the user interface can be modified or expanded to include additional data points. The system employs MQTT, a lightweight protocol optimized for real-time data transmission. Unlike HTTP, which relies on frequent polling and results in higher bandwidth usage and latency, MQTT supports bi-directional communication with significantly lower overhead. This makes it well-suited for IoT applications that demand low latency and efficient data handling. In addition, MQTT's compatibility with various brokers enhances scalability and supports long-term adaptability, offering an advantage over HTTP-based solutions.
- Shortcomings: The current implementation is limited to a single device (PM1200). Although the MQTT protocol can be scaled to support multiple devices, further testing is required to assess the system's performance and stability in more complex, large-scale environments.

The system updates real-time data on the map using MQTT, which enables low-latency, bi-directional communication. Data from the PM1200 power meter is transmitted to the web platform at regular intervals, typically every few seconds. This allows the map to display real-time energy consumption data—voltage, current, power, and frequency—dynamically and accurately.

Currently, the system visualizes data for only one location. However, it is designed for future scalability by incorporating additional data layers. In a multi-location setup, each site will have its own marker on the map, showing real-time energy data specific to that location.

B. Technical Challenges

Although the system has proven effective in monitoring energy data, several technical challenges remain in expanding its implementation:

- Compatibility with Different Energy Meters: This study focuses on the Schneider PM1200. Integrating other energy meter models may require adjustments to communication protocols or data formats. Ensuring compatibility with a range of devices is essential for scalability.
- Endianness Compatibility: In early tests using Arduino, the PM1200 data conformed to the standard little-endian format and functioned correctly. However, when implemented on the ESP32 microcontroller, the data appeared in big-endian format. This discrepancy required custom bitwise operations to accurately decode floating-point values such as voltage, current, and frequency.

C. Limitations and Future Research

This study was limited to a single PM1200 device at one location (Microprocessor Laboratory), which does not fully reflect the system's scalability or flexibility. Broader testing across multiple devices and locations is needed to evaluate performance in varied environments and assess factors such as network reliability and data concurrency. Future work will address the following areas.

- Scalability: The current system supports only one device and location. Future implementations will expand the map feature to monitor multiple sites, making the system more scalable and suitable for broader deployments of distributed energy systems.
- Historical Data Visualization: The system will integrate features for viewing historical energy data. Users will be able to select time ranges to analyze past consumption patterns, aiding in the identification of usage trends and supporting more informed energy management decisions.
- Machine Learning for Predictive Analysis: Machine learning algorithms will be explored to predict energy consumption patterns. By analyzing historical data, the system can forecast high- and low-demand periods, recommend energy usage adjustments, reduce costs, and promote off-peak consumption. This enhancement will improve system efficiency and support sustainable energy management.

IV. CONCLUSION

This study successfully demonstrates the integration of the PM1200 power meter with IoT technology for electrical energy monitoring. Parameter readings were conducted using two registers in float (little-endian) format on the ESP32 microcontroller. Data visualization through a map-based web platform enhances both readability and remote monitoring capabilities. The findings contribute to improved energy management efficiency and support the advancement of IoT-based energy monitoring systems. Future research is expected to expand monitoring coverage by adding new location points to the mapping system.

ACKNOWLEDGMENT

The authors would like to thank the Microprocessor Laboratory of Banjarmasin State Polytechnic for its support and resources during this research. Access to laboratory facilities and equipment significantly contributed to the successful implementation of this study.

REFERENCES

[1] I. B. G. Purwania, I. N. S. Kumara, and M. Sudarma, 'Application of IoT-Based System for Monitoring Energy Consumption', *International Journal of Engineering and Emerging Technology*, vol. 5, no. 2, pp. 81–93, 2020.

[2] G. Herandy and B. Suprianto, 'Monitoring Biaya dan Pengukuran Konsumsi Daya Listrik Berbasis Arduino Mega2560 Menggunakan Web'. *Jurnal Teknik Elektro*, vol. 8, no. 3, pp. 695 – 702. 2019.

[3] F. Y. Limpraptono, E. Nurcahyo, M. I. Ashari, E. Yandri, and Y. Jani, ‘Design of power monitoring and electrical control systems to support energy conservation’, *Proceedings of the Pakistan Academy of Sciences: Part A*, vol. 58, no. S, pp. 1–8, 2021.
[4] A. T. N. Mohammd Syafri, Hidayat, Dwi Sasmita Aji, Pambudi, ‘Elektriese: Jurnal Sains dan Teknologi Elektro’, *Sains dan Teknologi Elektro*, vol. 12, no. 02, pp. 126–140, 2022.
[5] S. Nor, ‘Penerapan Internet of Things (IoT) Sebagai Pengendali Peralatan Listrik Dan Pemantau Daya Listrik Berbasis Web’, *Jurnal EEICT (Electric, Electronic, Instrumentation, Control, Telecommunication)*, vol. 2, no. 2, 2019.
[6] V. Firmansyah, V. Nadhira, L. Silvi, and T. A. Dewi, ‘IoT Sistem Monitoring Meter Kwh Digital Menggunakan Sensor LDR dan Codeigniter Api Service’, *Jurnal Material dan Energi Indonesia,* vol. 9, no. 1, pp. 18-25, 2019.
[7] N. Nuryawan and M. Jannah, ‘Electrical Energy Monitoring and Control System In Boarding Rooms Based On The Internet of Things’, *International Journal of Engineering, Science and Information Technology*, vol. 2, no. 4, pp. 76–83, 2022.
[8] R. Risfendra, G. F. Ananda, and A. Stephanus, ‘Internet of Things on Electrical Energy Monitoring Using Multi-Electrical Parameter Sensors’, *MOTIVECTION: Journal of Mechanical, Electrical and Industrial Engineering*, vol. 3, no. 1, pp. 1–10, Jan. 2021.
[9] L. Q. Santos, J. C. D. Bautista, M. W. S. Estanque, C. J. B. Paloa, and A. B. A. Villaran, ‘Development of an IoT-Based Water and Power Monitoring System for Residential Building’, *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 2, pp. 744–751, 2021.
[10] J. W. Jokanan, A. Widodo, N. Kholis, and L. Rakhmawati, ‘Rancang Bangun Alat Monitoring Daya Listrik Berbasis IoT Menggunakan Firebase dan Aplikasi’, *Jurnal Teknik Elektro*, vol. 11, no. 1, pp. 47–55, 2022.
[11] M. Manfaluthy, A. Sunardi, S. Wilyanti, A. Jaenul, and E. Mulyana, ‘Design a Monitoring System of a Single-Phase Electricity Meter Based on The Internet of Things’, *Spektra: Jurnal Fisika dan Aplikasinya*, vol. 7, no. 3, pp. 113–126, 2022.
[12] Espressif, “ESP32 Series Datasheet,” in Espressif Systems, 2019, pp. 1–61. [Online]. Available: www.espressif.com
[13] H. S. Electronics, ‘UART TTL to RS485 Two-way Converter’, HAOYU STAR Electronics. Accessed: Aug. 23, 2023. [Online]. Available: https://www.hotmcu.com/uart-ttl-to-rs485-twoway-converter-module-p-267.html
[14] S. Electric, *PowerLogic PM1000 Series Power Meters*. Schneider Electric, 2010.
[15] D. Walker, ‘ModbusMaster’, 2016. [Online]. Available: https://github.com/4-20ma/ModbusMaster
[16] B. Blanchon, ‘ArduinoJson’, 2023. [Online]. Available: https://github.com/bblanchon/ArduinoJson
[17] N. O’Leary, ‘Arduino Client for MQTT’, 2020. [Online]. Available: https://github.com/knolleary/pubsubclient
[18] S. Electric, *Product Datasheet PM1200 Power Meter with Energy and Demand - with Communication*. 2023. [Online]. Available: https://www.etd.com.my/image/etd/image/data/pdf/schneider-METSEPM1200.pdf
[19] Eclipse, ‘Eclipse Paho JavaScript client’, 2018. [Online]. Available: https://github.com/eclipse-paho/paho.mqtt.javascript
[20] IoTEDU, ‘How to enable Mosquitto MQTT over WebSocket on Windows’. Accessed: Aug. 31, 2023. [Online]. Available: https://medium.com/@monishavasumacharla/how-to-enable-mosquitto-mqtt-over-websocket-on-windows-a909768f9c68
[21] Volodymyr Agafonkin, ‘Leaflet’. [Online]. Available: https://github.com/Leaflet/Leaflet