Smart Plug Socket with ESP-NOW and MQTT for Efficient Power Monitoring and Control via Android

Anak Agung Ngurah Gede Dhananjaya[a1], I Made Agus Dwi Suarjaya[a2], I Putu Sugi Almantara[a3]

[a]Information Technology Departmen, Faculty of Engineering, Udayana University, Indonesia
e-mail: [1]dhananjayaad24@gmail.com , [2]agussuarjaya@it.unud.ac.id,
[3]sugik.almantara@gmail.com

Abstrak

Penggunaan listrik yang tidak terkontrol serta fluktuasi tegangan dapat menyebabkan kerusakan perangkat elektronik dan pemborosan energi. Penelitian ini merancang sistem plug socket yang mampu memantau dan mengendalikan konsumsi listrik secara efisien melalui aplikasi Android. Sistem terdiri dari dua perangkat, yaitu master dan slave, yang berkomunikasi menggunakan protokol ESP-NOW tanpa koneksi internet dan terhubung ke aplikasi Android melalui MQTT. Perangkat slave dilengkapi sensor PZEM-004T untuk memantau tegangan, arus, daya, dan energi. Sistem dirancang untuk secara otomatis memutus aliran listrik jika fluktuasi tegangan terjadi selama lebih dari 1 menit, dan menyambungkannya kembali saat tegangan normal. Aplikasi Android menyediakan fitur kontrol relay, penjadwalan, estimasi biaya, dan visualisasi data historis hingga 45 hari. Hasil pengujian menunjukkan bahwa sistem dapat berfungsi dengan baik serta membantu pengguna memantau dan mengendalikan listrik secara efisien dan aman.

Kata kunci: *Plug Socket, Monitoring Listrik, ESPNOW, MQTT, Android*

Abstract

Uncontrolled electricity usage and voltage fluctuations can damage electronic devices and lead to energy waste. This research designs a plug socket system capable of efficiently monitoring and controlling electricity consumption through an Android application. The system consists of two devices: a master and a slave, communicate using the ESP-NOW protocol without an internet connection, and connect to the Android app via MQTT. The slave device, equipped with a PZEM-004T sensor to monitor electrical parameters such as voltage, current, power, and energy. The system is designed to automatically cut off power if voltage fluctuations occur for more than one minute and to restore it when the voltage returns to normal. The Android application provides features such as relay control, scheduling, cost estimation, and historical data visualization for up to 45 days. Testing results shows the system functions well and assists users in monitoring and controlling electricity use more efficiently and safely.

Keywords: *Plug Socket, electricity monitoring, ESPNOW, MQTT, Android*

1. Introduction

Daily electricity usage is often accompanied by negligence in its management. Approximately 80% of energy waste is caused by human factors, such as leaving electronic devices in standby mode [1]. Moreover, excessive use of electronic devices without supervision increases the risk of accidents such as electrical short circuits [2]. Voltage fluctuations also have a detrimental effect on electronic devices, potentially shortening their lifespan and increasing maintenance costs. The Indonesian state electricity company (PLN) sets a voltage tolerance limit of +5% to -10% from the nominal voltage (PT. Perusahaan Listrik Negara (Persero), 1995).

Smart plugs offer remote control and monitoring of electricity usage through smartphone apps, providing convenience and potential energy savings. However, most rely heavily on Wi-Fi, which has limitations in range and connection stability. Many existing products also lack real-time monitoring for key electrical parameters. This reduces the user’s ability to respond quickly to changing power conditions.

To address these limitations, this research proposes a plug socket system consisting of two devices—master and slave—connected via ESP-NOW. The master device includes a GSM module for independent internet access and communicates with an Android app through an MQTT broker. The system supports real-time monitoring, control, and voltage protection. It offers automatic power cutoff during overload or voltage anomalies and can estimate electricity costs based on configurable PLN tariffs.

2. Research Method / Proposed Method

The Software Development Life Cycle (SDLC) is a methodology used in software creation and modification [3]. The Waterfall model in SDLC follows a sequential and linear flow through phases such as analysis, design, implementation, testing, and maintenance [4]. It serves as a structured framework guiding the entire software development process.

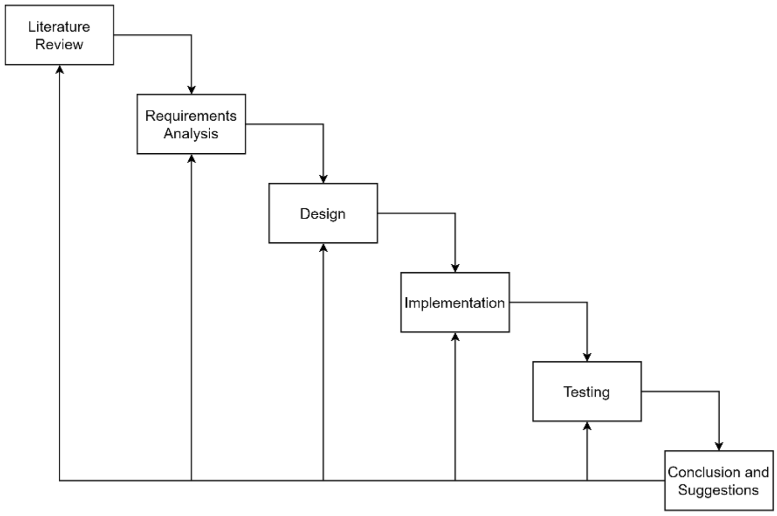

Figure 1. Software Development Life Cycle

Figure 1 shows the Software Development Life Cycle (SDLC) method adopted in this research, specifically the Waterfall model, to ensure a systematic and efficient software development process. The implementation of the Waterfall model in this study begins with a literature review phase to gather foundational and supporting theories from various sources such as books, journals, and online articles. The next phase involves system requirement analysis, identifying both hardware and software needs along with their functions. This is followed by the design phase, which includes system workflow, circuit schematics, database structure, 3D designs, and more. In the implementation phase, the system is developed based on the previously created designs. The testing phase ensures the system works correctly and assesses its performance. Finally, the development process is concluded with a summary and suggestions for future improvements, aligning with the structured approach of the Waterfall SDLC.

3. Literature Study

This section reviews relevant literature that forms the theoretical basis of the study. It covers research on power monitoring systems, ESP-NOW wireless communication, MQTT protocol, and sensor modules like PZEM-004T. The review guides the design of a plug socket system with real-time monitoring and control, and supports the selection of suitable technologies.

3.1. ESP-NOW Protocol

ESP-NOW is a communication protocol developed by Espressif that enables short-range data transmission between ESP32/ESP8266 devices using MAC addresses, based on the IEEE 802.11 standard [5]. It offers fast response, low power consumption by utilizing only one OSI layer, and stable performance even in unreliable Wi-Fi networks. The use of callbacks enhances signal range and reception, and supports both unicast and broadcast messaging for multilayer control. ESP-NOW allows fast, lightweight, energy-efficient communication without relying on an internet connection [5].

3.2. MQTT Protocol

MQTT (Message Queuing Telemetry Transport) is a lightweight publish/subscribe communication protocol designed for low-power devices [6]. With a small header size of only 2 bytes per message type, it operates efficiently in environments with limited bandwidth and power. MQTT ensures message delivery even when the connection is temporarily lost [7]. Its architecture involves three main components: publisher, subscriber, and broker, enabling real-time data exchange and device control—making it ideal for integrating IoT devices like plug sockets with Android applications over the internet.

3.3. Electric Sensor Modul PZEM-004T

The PZEM-004T, sensor module used to measure electrical parameters like voltage, current, power, and energy in AC systems. It operates based on current transformation principles, converting the current from a current transformer into a readable voltage signal. The module supports TTL serial communication, making it easy to interface with microcontrollers [8]. With a voltage range of 80VAC to 260VAC, a maximum load of 100A or 22,000W, and ±1% measurement accuracy, the PZEM-004T is more reliable and calibrated than alternatives like the ZMPT101B, which requires manual calibration, and the ACS712, which is prone to high noise.

3.4. Relay Modul

A relay is an electric operated switch component consisting of two main parts. two main parts such as an electromagnet (coil) and a contact mechanism. It operates based on electromagnetic principles, allowing a low-power signal to control higher voltage currents [9]. Relay contacts are categorized as normally closed (NC) or normally open (NO). When activated, the NC contact opens and the NO contact closes, enabling control over the electrical current flow.

3.5. ESP32 Microcontroller

The ESP32 is a System on Chip (SoC) microcontroller that supports Wi-Fi 802.11 b/g/n, Bluetooth 4.2, and includes a wide range of peripherals. It features a processor, memory, and access to GPIO (General Purpose Input Output) pins, making it a powerful alternative to Arduino with built-in Wi-Fi connectivity [10]. The board is available in 30- and 36-GPIO variants, with the 30-GPIO version preferred for its dual GND pins. It also includes a USB-to-UART interface, making it easy to program using development platforms like the Arduino IDE [11].

3.6. GSM Modul SIM800L

SIM800L is a dual-band GSM/GPRS module integrated into user applications with an SMT design. It supports standard interfaces and operates at 900/1800 MHz frequencies, enabling voice, SMS, and fax transmission [12]. The V2 version can operate directly at 5V without requiring a buck converter, and its TTL serial port allows easy connection to microcontrollers without using a MAX232. With compact size and low power consumption, SIM800L offers an efficient solution for cellular communication.

4. Result and Discussion

This section presents the results obtained from the development and testing of the plug socket system, along with a detailed analysis of the performance of each component. It begins with the physical implementation of the device circuit, followed by an overview of the Android application features that support system interaction. Sensor testing was conducted to assess the accuracy of electrical measurements, while communication performance was evaluated using the ESP-NOW and MQTT protocols. Finally, the system's functionality was tested in various scenarios to demonstrate the plug socket's response to real-world use, with a focus on accuracy, efficiency, and reliability.

4.1. Hardware Circuit

Implementation is carried out on the designed circuit. In this implementation stage, the circuit is tested before being inserted into the socket socket. The following is a diagram of the implemented circuit.

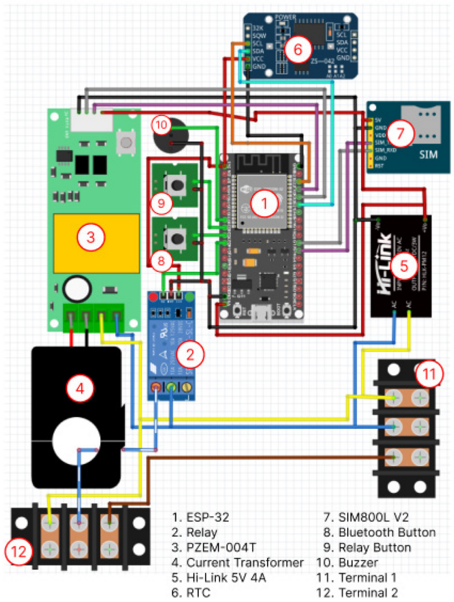

Figure 2. Plug Socket Master Circuit

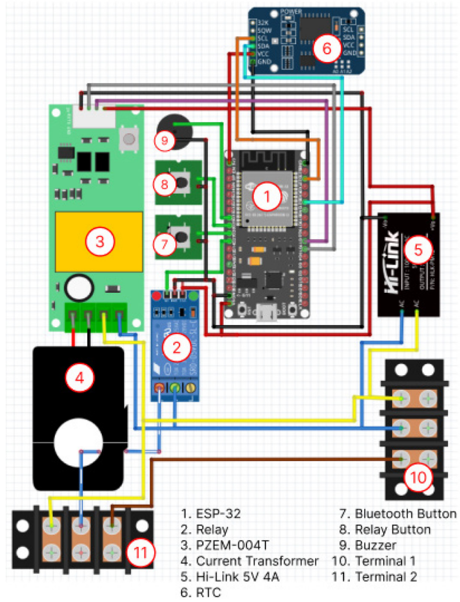

Figure 3. Plug Socket Slave Circuit

Figure 2 shows the implementation of the plug socket master circuit, where the ESP32 acts as the main microcontroller, controlling the relay and receiving real-time data from the PZEM-004T sensor. This sensor measures electrical parameters such as voltage, current, power, and energy, aided by a current transformer. The SIM800L V2 module enables cellular data communication using a SIM card with an active data plan and is powered by a Hi-Link 5V 4A module. An RTC (Real-Time Clock) is included to support scheduling features. The circuit also integrates terminal blocks for power input/output and manual buttons for relay and Bluetooth functions. Figure 3 depicts the slave circuit, which shares a similar design with the master, but excludes the SIM800L V2 module as it does not require direct cellular communication.

4.2. Application Feature

The Android application developed in this research serves as a user interface for monitoring and controlling the plug socket devices. It offers various features, including real-time display of electrical parameters such as voltage, current, power, and energy consumption. The application also allows users to configure power cut-off thresholds for overload protection. Additionally, it supports cost estimation based on configurable electricity tariffs from PLN, helping users manage their energy expenses. Scheduling features are provided to automate relay control based on specific times and days. Users also receive notifications in the event of anomalies, such as abnormal voltage or power usage, enhancing responsiveness and user awareness.

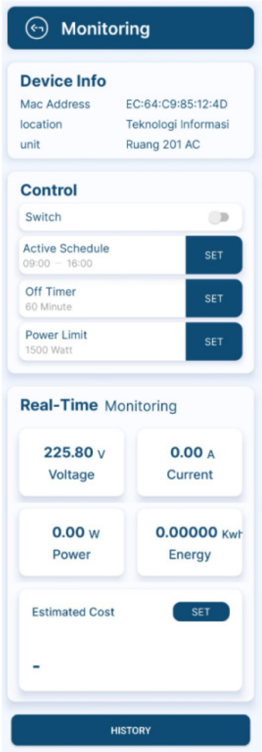

Figure 4. Application Monitoring User-Interface

Figure 4 shows the main display that provides an interface for monitoring and controlling the plug socket. Users can turn the switch on or off, set timers, and create schedules. It also shows real-time measurements of voltage, current, power, energy, and estimated cost. A history button at the bottom allows users to view past measurement records.

4.3. PZEM-004T Sensor Modul Test

To evaluate the performance of the PZEM-004T sensor along with the overall capability of the system components, a voltage accuracy test was conducted under various input conditions. This test aimed to identify the minimum voltage at which the sensor begins functioning, assess its accuracy across the operational range, and observe the behavior of the entire system including the power supply and relay under different voltage levels. By recording when the device could power on, maintain operation, or shut down, this test also provided insight into the voltage tolerance of the hardware components. The measured values were then compared to actual voltage inputs to determine the sensor's accuracy and the system's voltage resilience.

Table 1. PZEM-004T Modul Voltage Testing

Voltage (V)	Sensor Results (V)	Difference	Error
13	-	-	-
30	-	-	-
70	-	-	-
80	80.1	0.1	0.13%
120.1	120	0.1	0.08%
160	160.3	0.3	0.19%
190.2	190.8	0.6	0.32%
220.2	221.1	0.9	0.41%
240	241.2	1.2	0.50%
265	266.6	1.6	0.60%
Average Error			0.32%

Table 1 shows the result of voltage accuracy testing was conducted on the PZEM-004T sensor by supplying various AC voltage levels and observing its measurement response. At 13V, the power supply was no longer able to maintain output and shut down completely, while at 30V, it was just able to power on and remain active, although the sensor still failed to provide a reading. Even at 70V, the sensor could not yet detect or display any voltage, indicating its minimum operational threshold had not been reached. The sensor began reporting data at 80V with a very low error margin of 0.13%, although this was still below the power supply's recommended minimum.

From 120.1V to 240V, the device operated under normal conditions and the PZEM-004T produced accurate readings, with errors ranging from 0.08% to 0.50%. The highest voltage tested was 265V, at which point the error reached 0.60%. However, this value exceeded the maximum safe operating voltage specified by the datasheets for the power supply (264V), PZEM-004T sensor (260V), and the relay (250V), posing a potential long-term risk. Overall, the average error across all valid readings was recorded at 0.32%, indicating that the PZEM-004T sensor is sufficiently accurate for AC voltage monitoring within its specified range.

Table 2. PZEM-004T Modul Power Testing

Power (W)	Sensor Results (W)	Difference	Error
28.6	28.6	0.4	0.00%
40,7	41.1	0.4	0.98%
75,9	76.5	0.6	0.79%
188,3	188.1	0.2	0.11%
220.0	220.2	0.2	0.09%
534.6	523.3	11.3	2.11%
Average Error			**0.68%**

Table 2 shows the result of accuracy test of the PZEM-004T sensor in measuring active power (Watt) was conducted at a constant voltage of 220 V using purely resistive loads (PF = 1). Results showed small differences and low errors between actual and sensor readings, with an average error of 0.68%. The sensor provided accurate and stable measurements across a wide range of loads, from 28.6 W to 534.6 W.

4.4. Communication Performance via ESP-NOW and MQTT

This section discusses the performance of the communication system used in the plug socket, which combines ESP-NOW for device-to-device communication and MQTT for cloud-based data exchange. Both protocols play a crucial role in enabling real-time monitoring and control. Testing was conducted to evaluate the responsiveness, reliability, and delay under different conditions. The results provide insights into how each protocol supports the system's functionality in both local and internet-based scenarios.

Table 3. ESP-NOW Monitoring Delay 8m

Master Receive Time	Time Difference (ms)	Delay Over Ideal (ms)
14:38:13.536	-	-
14:38:16.602	3066	1066
14:38:19.704	3102	1102
14:38:22.760	3103	1056
14:38:25.863	3053	1103
14:38:28.916	3105	1053
14:38:31.968	3057	1105
14:38:35.025	3056	1057
14:38:38.081	3097	1056
14:38:41.178	3103	1097
Average ideal delay		1084.6

Table 4. ESP-NOW Monitoring Delay 6,9m

Master Receive Time	Time Difference (ms)	Delay Over Ideal (ms)
16:06:49.351	-	-
16:06:55.561	6210	4210
16:06:57.664	2103	103
16:07:01.774	4110	2110
16:07:14.256	12482	10482
16:07:22.529	8273	6273
16:07:28.739	6210	4210
16:07:30.836	2097	97
16:07:32.895	2059	59
16:07:37.049	4154	2154
Average ideal delay		6410.89

ESP-NOW monitoring data testing was conducted in two scenarios. Table 3 shows the result for the same room at an 8-meter distance, where the actual delay was reduced by 2 seconds based on the device's transmission interval. The observed delay ranged from 1103 ms to 2151 ms. In a different room, as shown in Table 4, which is separated by a 0.6-meter wall and a total distance of 6.90 meters, delays ranged from 59 ms to 10,482 ms. Data transmission failed when the devices were placed farther apart with the wall in between.

Table 5. ESP-NOW Control Delay 8m

Master Send Time	Slave Receive Time	Delay (ms)
15:55:28.536	15:55:31.659	3123
15:55:34.907	15:55:36.012	1105
15:55:41.332	15:55:42.402	1070
15:55:47.708	15:55:48.791	1083
15:55:51.993	15:55:53.090	1097
15:55:56.275	15:55:57.343	1068
15:56:02.504	15:56:05.607	3103
15:56:08.883	15:56:11.931	3048
15:56:15.112	15:56:18.246	3134
15:56:21.446	15:56:24.519	3073
Average delay		2190.4

Table 6. ESP-NOW Control Delay 6,85m

Master Receive Time	Slave Receive Time	Delay (ms)
16:26:43.736	16:26:49.420	5684
16:26:50.108	16:26:55.812	5704
16:26:56.521	16:27:02.280	5759
16:27:02.946	16:27:08.607	5661
16:27:09.344	16:27:12.903	3559
16:27:22.140	16:27:25.690	3550
16:27:28.770	16:27:32.326	3556
16:27:34.203	16:27:39.912	5709
16:27:41.825	16:27:46.980	5155
16:27:48.123	16:27:52.581	4458
Average delay		4879.5

ESP-NOW control command testing was conducted in two conditions. Table 5 shows the result for the same room with an 8-meter distance, where the slave received commands with delays ranging from 1068 ms to 3134 ms. However, at longer distances, most control transmissions failed. As shown in Table 6, in a different room separated by a 0.6-meter wall and a total distance of 6.85 meters, delays ranged from 3550 ms to 5759 ms. Communication failed when the devices were placed farther apart with the wall acting as a barrier.

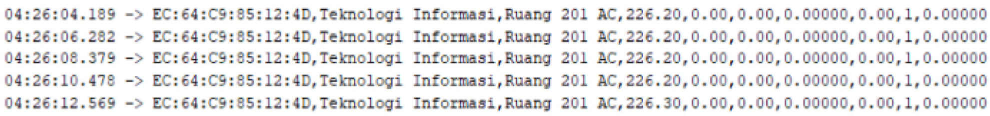

Figure 5. MQTT Monitoring Messages Sent from the Master Device

Figure 5 shows monitoring data using MQTT from the plug socket device, which includes several variables such as the MAC address, location, connected appliance, voltage, current, power, energy, and relay status. These parameters are combined into a single message and published using the MQTT communication protocol.

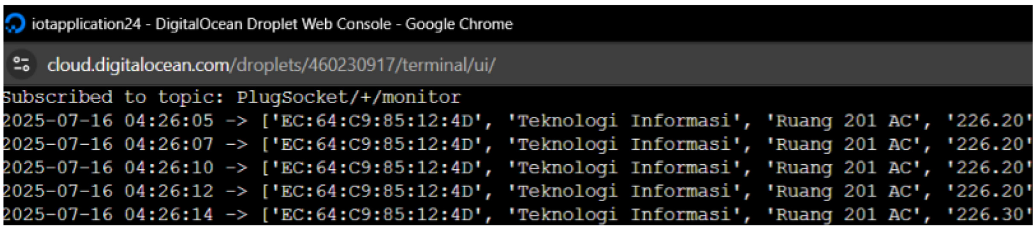

Figure 6. The MQTT Server Receiving the Monitoring Message

On the server side, as shown in Figure 6, a subscription is made to the topic PlugSocket/+/monitoring using a wildcard (+) to receive data from all plug socket devices regardless of their MAC address. This monitoring data is then stored in an InfluxDB database, allowing for historical review and analysis.

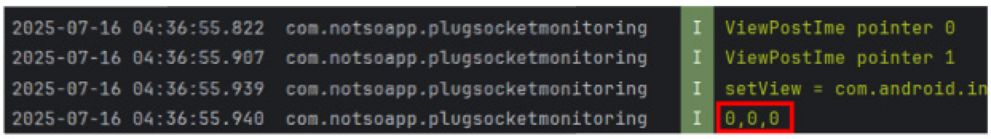

Figure 7. Android Application Sending Control Commands Via MQTT

As shown in Figure 7, the Android application sends control commands to the plug socket device via MQTT by publishing messages to specific topics. For example, the message 0,0,0 is published to the topic PlugSocket/EC:64:C9:85:12:4D/control to control the plug socket device associated with that MAC address.

Figure 8. Server Receiving the Control Message

Figure 8 shows the server, which is subscribed to the same control topic, receiving the command and verifies its delivery through the MQTT broker. This ensures that control messages from the Android application reach the server successfully.

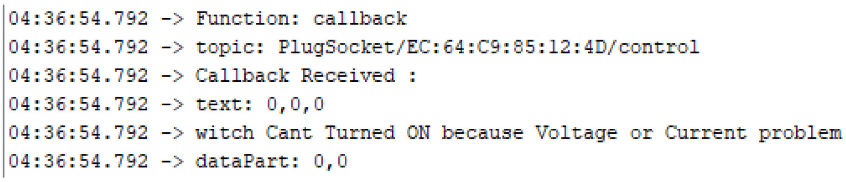

Figure 9. Master Device Receives Control Command Via MQTT

Likewise, Figure 9 shows that the plug socket device subscribes to its respective control topic. Upon receiving a command such as 0,0,0, the device interprets the message and executes the corresponding control action, completing the end-to-end communication process.

4.5. System Functionality Testing

This section presents the results of functional testing to evaluate the system's effectiveness in managing electrical devices through automatic monitoring and control features. Tests were conducted on various types of equipment to observe the performance of the plug socket system in real-world scenarios. Particular attention was paid to energy efficiency, user behavior, and the responsiveness of the automatic power cut-off feature. The findings highlight how the system contributes to energy savings in some cases, while in others, it serves as a preventative safety mechanism.

Table 7. Eficiency of Room Cooler Using Plug Socket Master

Day	Energy Usage (kWh) – Room Cooler		Description	Energy Saved (kWh)	Efficiency (%)
	Without Turning Off Plug Socket	If Plug Socket Turned Off			
Friday	19,1483	4.7392	Left On	14.4091	75.24%
Saturday	32,6117	0.0000	Left On	32.6117	100.00%
Sunday	29,9513	0.0000	Left On	29.9513	100.00%
Monday	7.3540	7.3470	Normal	0.0070	0.10%
Tuesday	9.0020	8.9930	Normal	0.0090	0.10%
Wednesday	0.01209	0.0000	Normal	0.01209	100.00%
Thursday	2.5310	2.5210	Normal	0.0100	0.40%
Friday	0.01077	0.0000	Normal	0.01077	100.00%
Saturday	0.01074	0.0000	Normal	0.01074	100.00%
Sunday	0.01070	0.0000	Normal	0.01070	100.00%

Table 7 shows the energy efficiency testing on air conditioners, indicating that a plug socket system with automatic control features can save up to 70.96% of energy. This highlights the significant energy waste caused by users' negligence in turning off devices when not in use. Without control, power consumption reached 108 kWh, while with automatic control enabled, power consumption dropped dramatically. This system encourages more practical and disciplined device management. Control features such as timers and scheduling have proven effective in reducing unnecessary power consumption. However, directly cutting power to active devices is not always safe, as some electronic devices require a proper power off to avoid data loss or corruption. Therefore, automatic switch features are best used in emergencies, unattended conditions, or when the device is not in use—to ensure energy savings without compromising device reliability.

5. Conclusion

The integration of ESP-NOW and MQTT enables efficient real-time monitoring and control with low and stable latency, while GPRS via SIM card serves as an alternative. The PZEM-004T sensor demonstrated high accuracy, with an average voltage error of 0.32% and power error of 0.68%, ensuring reliable protection against voltage fluctuations. The system also remained operational down to 30V, enhancing safety during voltage drops. All plug socket features—including switch, timer, power limit, scheduling, and cost estimation—functioned as intended. Testing on a room cooler showed that automatic control reduced energy consumption from 108.58 kWh to 31.53 kWh over 11 days, achieving 70.96% energy savings. These results confirm that the plug socket system improves energy efficiency, safety, and disciplined usage.

References

[1] Y. S. Akbar Gumilang, M. S. Hadi, and D. Lestari, “Stopkontak Pintar Berbasis Internet of things sebagai Solusi Manajemen Energi Listrik dengan Menggunakan Aplikasi Android,” *JASIEK (Jurnal Aplikasi Sains, Informasi, Elektronika dan Komputer)*, vol. 4, no. 2, pp. 55–66, Dec. 2022, doi: 10.26905/jasiek.v4i2.8332.

[2] D. Suryanto, “ANALISIS PERAWATAN AC (AIR CONDITIONER) UNIT SPLIT DUCT MENGGUNAKAN METODE FAILURE MODE AND EFFECT ANALYSIS FMEA DI HOTEL HARRIS YELLO.”

[3] M. Ridwan and I. Fitri, “Rancang Bangun Marketplace Berbasis Website menggunakan Metodologi Systems Development Life Cycle (SDLC) dengan Model Waterfall,” *Jurnal Teknologi Informasi dan Komunikasi)*, vol. 5, no. 2, p. 2021, 2021, doi: 10.35870/jti.

[4] D. Kurniawan, “PENDEKATAN SDLC MODEL WATERFALL DALAM PERANCANGAN APLIKASI PENDAFTARAN KURSUS,” 2023.

[5] J. A. Cujilema Paguay, G. A. Hidalgo Brito, D. L. Hernandez Rojas, and J. J. Cartuche Calva, “Secure home automation system based on ESP-NOW mesh network, MQTT and Home Assistant platform,” *IEEE Latin America Transactions*, vol. 21, no. 7, pp. 829–838, Jul. 2023, doi: 10.1109/TLA.2023.10244182.

[6] O. Pribadi, “Sistem Kendali Jarak Jauh Air Conditioner (AC) Berbasis IoT,” 2020. [Online]. Available: http://ejournal.stmik-time.ac.id

[7] H. A. Rochman, R. Primananda, and H. Nurwasito, “Sistem Kendali Berbasis Mikrokontroler Menggunakan Protokol MQTT pada Smarthome,” 2017. [Online]. Available: http://j-ptiik.ub.ac.id

[8] K. M. Monzer, Y. M. Tariq, and A. T. M. Farouq, “New design of socket modules for smart home applications,” *Instrumentation Mesure Metrologie*, vol. 18, no. 1, pp. 43–48, 2019, doi: 10.18280/i2m.180107.

[9] M. Saleh and M. Haryanti, “RANCANG BANGUN SISTEM KEAMANAN RUMAH MENGGUNAKAN RELAY,” 2017.

[10] M. Nizam, H. Yuana, and Z. Wulansari, “MIKROKONTROLER ESP 32 SEBAGAI ALAT MONITORING PINTU BERBASIS WEB,” 2022.

[11] A. Wagyana, J. Teknik Elektro, P. Negeri Jakarta, J. G. Siwabessy, and K. U. Depok, “Prototipe Modul Praktik untuk Pengembangan Aplikasi Internet of Things (IoT),” *Jurnal Ilmiah Setrum Article In Press*, vol. 8, no. 2, pp. 238–247, 2019.

[12] J. K. Brajamusti and C. K. Nurjanah, “RANCANG BANGUN SISTEM SATELIT BUATAN BERBASIS ESP32 DENGAN FITUR KOMUNIKASI MENGGUNAKAN MODUL GSM SIM800L,” *Jurnal Elektro dan Telekomunikasi Terapan*, vol. 10, no. 1, p. 17, Jul. 2023, doi: 10.25124/jett.v10i1.6123.

[13] M. Nasar and M. A. Kausar, “Suitability of influxdb database for iot applications,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 10, pp. 1850–1857, Aug. 2019, doi: 10.35940/ijitee.J9225.0881019.