

# IoT-Based UPS Device Electricity Usage Monitoring System with MQTT Protocol

Adit Oktopryadin <sup>1\*</sup>, Adi Purnama <sup>2\*</sup>

\* Fakultas Teknik, Universitas Widyatama, Bandung, Indonesia  
[adit.oktopryadin@widyatama.ac.id](mailto:adit.oktopryadin@widyatama.ac.id) <sup>1</sup>, [adi.purnama@widyatama.ac.id](mailto:adi.purnama@widyatama.ac.id) <sup>2</sup>

## Article Info

### Article history:

Received 2025-04-28

Revised 2025-06-02

Accepted 2025-07-03

### Keyword:

*Electric Power Monitoring,  
Internet of Things (IoT),  
Message Queuing Telemetry  
Transport (MQTT),  
Real-Time Monitoring,  
Uninterruptible Power Supply  
(UPS).*

## ABSTRACT

The continuity of network device operations heavily relies on stable power supply, especially in digital environments that demand uninterrupted connectivity. One commonly used solution to ensure power continuity is the Uninterruptible Power Supply (UPS). However, traditional UPS systems often lack real-time monitoring mechanisms, leaving users uninformed during the transition from main electricity to UPS power. To address this challenge, this study proposes the design of a UPS power consumption monitoring system based on the Internet of Things (IoT) using the Message Queuing Telemetry Transport (MQTT) communication protocol. The system integrates a PZEM-004T power sensor and ESP32 microcontroller to read electrical parameters such as voltage, current, and power in real-time, and displays the data through a digital dashboard built with Node-RED. The implementation results show that the system can automatically detect changes in power source status and record electrical parameters with an average error rate below 1%, both during normal grid operation and when switching to UPS power. This system is expected to serve as a practical and efficient solution for minimizing network downtime caused by power disruptions.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

## I. PENDAHULUAN

Dalam dunia digital saat ini, keberlangsungan operasional perangkat jaringan menjadi salah satu aspek yang sangat penting, terutama bagi perusahaan, institusi pendidikan, dan penyedia layanan [1]. Perangkat jaringan telah menjadi kebutuhan penting dan mendasar untuk komunikasi data dan konektivitas [2]. Konektivitas perangkat jaringan sangat bergantung pada pasokan daya listrik yang stabil dan berkelanjutan. Jika terjadi pemadaman listrik, komputer dan perangkat jaringan tidak akan berfungsi sehingga aktivitas pengguna akan terganggu. Meskipun pengelolaan listrik telah dirancang sedemikian rupa untuk meminimalkan terjadinya pemadaman, ancaman gangguan daya tetap mungkin terjadi [3]. Oleh karena itu, maka perangkat jaringan dilengkapi dengan pemasangan UPS (*Uninterruptible Power Supply*) sebagai solusi sementara ketika pemadaman listrik akibat gangguan atau *maintenance* untuk mem-backup suplai daya listrik [4], [5]. Perangkat jaringan yang tidak berfungsi akibat

pemadaman listrik utama atau baterai UPS yang habis dapat mengganggu aktivitas pengguna yang bekerja menggunakan komputer. Pemasangan UPS bertujuan untuk mencegah *downtime* akibat gangguan daya [6]. Namun, jika baterai UPS habis sebelum listrik utama kembali berfungsi, *downtime* perangkat jaringan akan lebih lama dan mengakibatkan gangguan serius pada konektivitas pengguna.

Salah satu kelemahan dalam penggunaan UPS adalah tidak adanya sistem pemantauan yang sistematis untuk mengetahui masa peralihan dari penggunaan listrik utama ke UPS. Perangkat jaringan akan terus menyala hingga baterai UPS tanpa pemberitahuan kepada pengguna. Akibatnya, pengguna baru menyadari pemadaman saat perangkat mereka kehilangan koneksi jaringan, yang memperpanjang waktu *downtime*. UPS juga memiliki keterbatasan, seperti kapasitas baterai yang terbatas dan risiko kerusakan akibat pemakaian yang tidak terpantau [7]. Oleh karena itu, sistem pemantauan daya pada UPS menjadi kebutuhan penting untuk memastikan efisiensi dan keandalan perangkat jaringan.

Teknologi *Internet of Things* (IoT) memungkinkan untuk dilakukannya pemantauan secara *real-time* [8], [9]. Dengan memanfaatkan sensor daya dan mikrokontroler, data tentang pemakaian listrik dapat dikumpulkan, dianalisis, dan ditampilkan melalui *platform* digital [10]. Sistem ini memungkinkan pengguna untuk memantau status UPS dari jarak jauh secara *real-time* dan memberikan notifikasi jika terjadi anomali daya, seperti penurunan kapasitas baterai yang kritis dan atau terjadi peralihan dari listrik utama ke UPS [11], sehingga tindakan preventif dapat dilakukan lebih cepat dan efisien. Data pemakaian listrik yang dihasilkan melalui sensor daya dan *microcontroller* ditransmisikan menggunakan protokol komunikasi IoT. Salah satu protokol yang umum digunakan adalah *Message Queuing Telemetry Transport* (MQTT), dirancang khusus untuk transfer data yang ringan dan efisien pada perangkat dengan sumber daya terbatas [12]. Protokol MQTT memiliki berbagai keunggulan, seperti penggunaan *bandwidth* yang rendah, pengiriman pesan yang andal, dan kompatibilitas dengan berbagai perangkat [13].

Dalam penelitian yang dilakukan oleh Y. Apriani, M. Rais Asadullah, dan M. Hurairoh, mereka mengembangkan sistem pemantauan UPS berbasis *Internet of Things* (IoT) menggunakan modul Sonoff POW R2 untuk memantau parameter listrik seperti tegangan, arus, dan daya secara *real-time*. Sistem ini dirancang agar pengguna dapat memantau status UPS dari jarak jauh melalui aplikasi eWeLink pada *smartphone*, dengan memanfaatkan konektivitas internet. Penelitian tersebut menunjukkan bahwa *monitoring* parameter UPS dapat dilakukan baik dari sumber PLN maupun baterai, dengan hasil yang menunjukkan kestabilan lebih baik pada penggunaan baterai dibandingkan PLN. Namun, penelitian ini belum membahas aspek komunikasi data yang lebih efisien atau keamanan transmisi data, karena sepenuhnya bergantung pada aplikasi pihak ketiga [4].

Dalam penelitian yang dilakukan oleh S. C. Prasath, N. Darwin, R. S. Ramkumar, S. Nithishkumar, dan P. L. Somasundharam, mengembangkan sistem *monitoring* baterai UPS berbasis IoT yang dirancang untuk mendukung sistem dengan ketersediaan tinggi seperti pusat data dan institusi keuangan. Sistem ini menggunakan berbagai sensor, termasuk sensor tegangan, sensor suhu, sensor ultrasonik, dan sensor kebakaran, yang terhubung dengan mikrokontroler ATMEGA328P dan NodeMCU sebagai *gateway* IoT. Sistem ini juga dilengkapi dengan *buzzer* untuk memberikan peringatan, LCD untuk menampilkan data lokal, dan mekanisme otomatis untuk memutuskan beban jika terdeteksi kondisi tidak normal seperti suhu tinggi atau kebakaran pada baterai. Namun, penelitian ini tidak secara spesifik membahas efisiensi komunikasi data antar perangkat dalam jaringan IoT yang digunakan. Sistem yang digunakan sepenuhnya bergantung pada *gateway* IoT umum tanpa mengevaluasi performa protokol komunikasi tertentu seperti MQTT atau HTTP, yang sangat penting untuk mengukur

efisiensi dan reliabilitas pengiriman data terutama pada jaringan dengan keterbatasan *bandwidth* [5].

Penelitian yang dilakukan oleh J. M. Ramadhan, R. Mardiaty, dan I. N. Haq telah mengembangkan sistem pemantauan berbasis IoT untuk pembangkit listrik tenaga surya (PLTS) yang menggunakan metode komunikasi *publisher-subscriber* dengan protokol MQTT. Sistem ini dirancang menggunakan ESP32 sebagai mikrokontroler dan *platform* Google Cloud sebagai *server* penyimpanan data, yang memungkinkan pemantauan kinerja PV *mini-grid* secara *real-time*. Data hasil pemantauan ditampilkan dalam basis data yang dikelola oleh komputer virtual yang bertindak sebagai *broker* MQTT. Hasil penelitian menunjukkan bahwa sistem ini memiliki kinerja yang mumpuni dengan tingkat akurasi yang tinggi, ditunjukkan oleh nilai kesalahan pengukuran yang sangat kecil. Selain itu, sistem ini dapat mengirimkan data dalam waktu respons minimum, menjadikan transmisi data berlangsung secara *real-time*. Namun, penelitian ini berfokus pada PLTS tanpa membahas penerapan serupa pada sistem UPS [14].

Penelitian ini berfokus pada pengembangan arsitektur IoT untuk pemantauan UPS pada perangkat jaringan secara *real-time* yang menggunakan protokol MQTT untuk komunikasi data secara efisien melalui jaringan dengan *bandwidth* terbatas. Berbeda dengan penelitian yang dilakukan oleh Y. Apriani, M. Rais Asadullah, dan M. Hurairoh yang berhasil memantau tegangan, arus, dan daya UPS secara *real-time* dari jarak jauh dan penelitian yang dilakukan oleh S. C. Prasath *et al.* mengusulkan sistem pemantauan baterai UPS berbasis IoT dengan berbagai sensor untuk mendeteksi anomali seperti suhu tinggi atau kebakaran. Namun kedua penelitian tersebut tidak membahas protokol komunikasi seperti MQTT dalam pengiriman data antar perangkat. Adapun penelitian yang dilakukan oleh J. M. Ramadhan *et al.* telah menggunakan metode komunikasi *publisher-subscriber* dengan protokol MQTT pada sistem pemantauan PLTS berbasis IoT. Namun, penelitiannya hanya berfokus pada penerapan untuk PLTS dan tidak mengimplementasikan pada sistem UPS.

Dalam penelitian ini, sistem pemantauan berbasis IoT dengan protokol MQTT akan dirancang untuk memantau konsumsi daya listrik UPS yang digunakan untuk perangkat jaringan. Sistem ini diharapkan dapat memberikan solusi efektif dalam pengelolaan daya, khususnya dalam lingkungan yang membutuhkan pemantauan kontinu, seperti server, router, atau perangkat jaringan lainnya sekaligus meminimalkan risiko *downtime* akibat kegagalan daya.

## II. METODE

Pada penelitian ini akan menggunakan metodologi *System Development Life Cycle* (SDLC) model *waterfall*. Model *waterfall* dalam pengembangan sistem digunakan sebagai pendekatan berurutan, di mana setiap tahap dilakukan secara bertahap seperti aliran air terjun. Proses ini mengikuti

serangkaian fase yang harus diselesaikan secara berurutan untuk menghasilkan sistem yang sesuai kebutuhan.

Dalam konteks ‘Sistem Pemantauan Pemakaian Daya Listrik pada UPS untuk Perangkat Jaringan berbasis IoT dengan Protokol MQTT’, model ini memastikan setiap fase mulai dari analisis kebutuhan, perancangan sistem, implementasi, hingga pengujian sistem. Setiap fase dilakukan secara terstruktur untuk menghasilkan sistem yang handal dan sesuai kebutuhan.

#### A. Analisis Kebutuhan

Pada tahap analisis kebutuhan, dilakukan identifikasi kebutuhan perangkat keras dan perangkat lunak yang diperlukan untuk membangun sistem pemantauan pemakaian daya listrik bagi UPS berbasis IoT. Analisis ini mencakup kebutuhan data dari sensor, seperti data tegangan, arus, daya, dan energi listrik yang akan dikumpulkan dari UPS. Selain itu, kebutuhan aktuator yang dapat mengidentifikasi kondisi sumber listrik yang memberikan suplai ke UPS. Sistem ini juga memerlukan mikrokontroler, seperti ESP32, yang mampu membaca data dari sensor dan mengirimkannya melalui protokol MQTT ke server broker. Selain itu, dibutuhkan aplikasi pemantauan berbasis *dashboard* untuk menampilkan data secara *real-time*. Tahap ini menghasilkan spesifikasi kebutuhan perangkat keras (sensor, mikrokontroler, server MQTT) dan perangkat lunak (program mikrokontroler, broker MQTT, dan *dashboard monitoring*).

Analisis kebutuhan non fungsional merupakan kebutuhan yang menunjang dalam pembuatan sistem agar dapat berjalan sebagaimana mestinya. Analisis kebutuhan non fungsional pada penelitian ini terdiri dari kebutuhan *hardware* dan *software*. Kebutuhan *hardware* yang digunakan untuk membuat sistem ini ditampilkan pada Tabel I. Sedangkan kebutuhan *software* digunakan untuk membuat sistem pemantauan listrik UPS mulai dari tegangan, arus, daya, dan status sumber listrik dari PLN (Perusahaan Listrik Negara) yang ditampilkan pada Tabel II.

TABEL I  
KEBUTUHAN *HARDWARE*

No	Nama <i>Hardware</i>	Keterangan
1	ESP32	Mikrokontroller untuk mengelola sensor
2	PZEM-004T	Modul pengukuran sensor parameter listrik
3	CT Split-Core	Merupakan bagian dari PZEM004T yang berfungsi untuk mengukur arus listrik
4	Modul Relay	Saklar yang digunakan untuk mendeteksi sumber listrik ‘Menyala’ atau ‘Mati’
5	Access Point	Perangkat keras jaringan untuk menghubungkan mikrokontroller dengan internet secara <i>wireless</i>

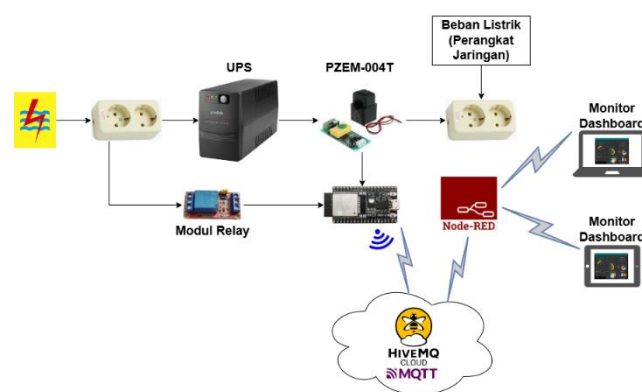
6	UPS Prolink PRO700SFC	Berfungsi untuk mem- <i>backup</i> dan menyimpan daya listrik pada saat listrik PLN padam
---	-----------------------	---

TABEL II  
KEBUTUHAN *SOFTWARE*

No	Nama <i>Hardware</i>	Keterangan
1	Arduino IDE	Aplikasi untuk menulis program
2	Node-RED	Alat pengembangan berbasis alur sebagai integrasi dan <i>dashboard</i> IoT
3	HiveMQ	Server MQTT Broker
4	Windows 11	Sistem Operasi laptop yang digunakan
5	Web Browser	Aplikasi untuk mengelola <i>logic</i> dan <i>dashboard</i> IoT dari Node-RED

#### B. Perancangan Sistem

Setelah kebutuhan teridentifikasi, langkah berikutnya adalah merancang arsitektur sistem pemantauan. Tahap perancangan bertujuan untuk memastikan setiap komponen sistem berfungsi sesuai kebutuhan. Pada tahap ini, desain dibuat untuk menggambarkan alur data mulai dari sensor hingga aplikasi *monitoring*.



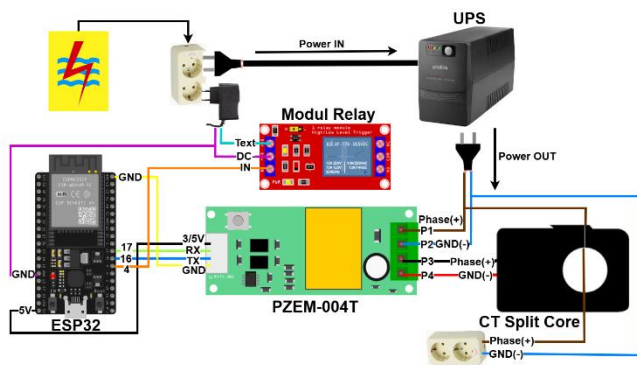
Gambar 1. *Quick Design*

Seperti yang ditunjukkan pada Gambar 1, sensor daya seperti PZEM-004T akan dihubungkan ke mikrokontroler ESP32 untuk membaca parameter daya listrik [15]. Mikrokontroler kemudian mengirimkan data ke broker MQTT melalui koneksi Wi-Fi. Broker MQTT HiveMQ Cloud akan mendistribusikan data ke aplikasi client MQTT, yang bertugas menampilkan informasi ke pengguna. Desain ini juga mencakup perancangan antarmuka pengguna yang berbentuk *dashboard* sehingga memungkinkan data ditampilkan dalam bentuk grafik atau tabel.

Sensor yang digunakan pada penelitian ini digunakan untuk mengukur tegangan, arus, dan daya listrik secara *real-time* dari perangkat jaringan seperti router, switch, dan access point sebagai beban listrik. ESP32 sebagai mikrokontroler dihubungkan dengan Wi-Fi untuk mengimplementasikan IoT yang terhubung dengan internet menggunakan MQTT dan ditampilkan melalui *dashboard*. Pada sistem ini, listrik dari

sumber PLN akan dideteksi menggunakan modul relay. UPS mendapatkan *input* dari sumber PLN dan *output* dihubungkan dengan sensor sebelum dihubungkan dengan stop kontak.

Setelah proses analisis dan perancangan sistem selesai, tahap selanjutnya adalah implementasi. Pada tahap ini, implementasi dilakukan untuk perakitan, skema diagram, dan Node-RED sebagai *logic flow* sistem yang dihubungkan dengan broker MQTT yang pada akhirnya akan menampilkan *traffic data* pada halaman *dashboard*.



Gambar 2. Skema Wiring Diagram

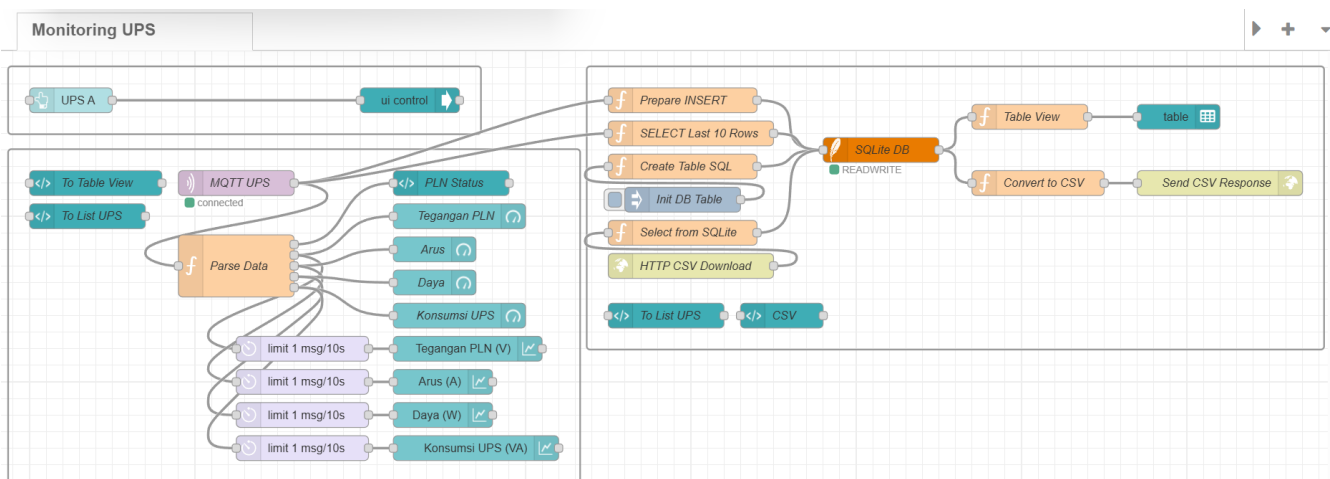
Berikut merupakan penjelasan skema diagram yang digunakan untuk koneksi antar *hardware*, sebagaimana ditunjukkan pada Gambar 2. Modul sensor PZEM-004T “GND” dihubungkan dengan “GND” pada ESP32. Modul sensor PZEM-004T “TX” dihubungkan dengan pin “17” pada ESP32. Modul sensor PZEM-004T “RX” dihubungkan dengan “16” pada ESP32. Modul sensor PZEM-004T “5V”

dihubungkan dengan “5V” pada ESP32. Modul sensor PZEM-004T “P1” dihubungkan dengan kabel phase (+) pada saklar. Modul sensor PZEM-004T “P2” dihubungkan dengan kabel GND (-) pada saklar. Modul sensor PZEM-004T “P3” dihubungkan dengan kabel phase (+) pada CT Split Core. Kabel phase (+) saklar yang sudah terhubung dengan PZEM-004T dihubungkan dengan stop kontak yang dilewati oleh CT Split Core, sedangkan kabel GND (-) saklar dihubungkan dengan stop kontak tanpa melewati CT Split Core.

Modul relay “IN” dihubungkan dengan pin “4” pada ESP32. Modul relay “DC-” dihubungkan dengan kabel GND (-) pada adapter AC 5V. ESP32 “GND” dihubungkan dengan kabel GND yang terhubung dengan modul relay “DC-”. Modul relay “DC+” dihubungkan dengan kabel phase (+) pada adapter AC 5V.

Adapter AC 5V pada modul relay dihubungkan dengan stop kontak yang dialiri listrik dari PLN. UPS mendapatkan listrik dari PLN sebagai power IN. Saklar PZEM-004T dihubungkan dengan UPS OUT. Stop kontak yang terhubung dengan PZEM-004T dan CT Split Core dihubungkan dengan perangkat jaringan sebagai beban listrik yang akan diukur dan dipantau.

Alur logika pengiriman data listrik dari perangkat ke *dashboard* diperlihatkan pada Gambar 3. Pengiriman parameter data listrik yang diolah dari ESP32 kemudian dikirim ke broker MQTT yang selanjutnya akan diolah dan ditampilkan pada *dashboard* dalam *tool* Node-RED. Sebelum data ditampilkan pada *dashboard*, dibuat terlebih dahulu daftar UPS yang akan dipantau.



Gambar 3. Tampilan Alur Logika Node-RED

Alur logika pengiriman data listrik dari perangkat ke *dashboard* diperlihatkan pada Gambar 3. Pengiriman parameter data listrik yang diolah dari ESP32 kemudian dikirim ke broker MQTT yang selanjutnya akan diolah dan

ditampilkan pada *dashboard* dalam *tool* Node-RED. Sebelum data ditampilkan pada *dashboard*, dibuat terlebih dahulu daftar UPS yang akan dipantau. Data yang diolah dan ditampilkan menggunakan Node-RED diantaranya mendeteksi listrik PLN, tegangan, arus, daya listrik, dan konsumsi UPS dalam bentuk *gauge* dan *chart*. Selain itu,

terdapat halaman *table view* sebagai penyimpanan data dengan *database* SQLite agar dapat dilihat dan diunduh dalam bentuk *file* csv.

### C. Implementasi

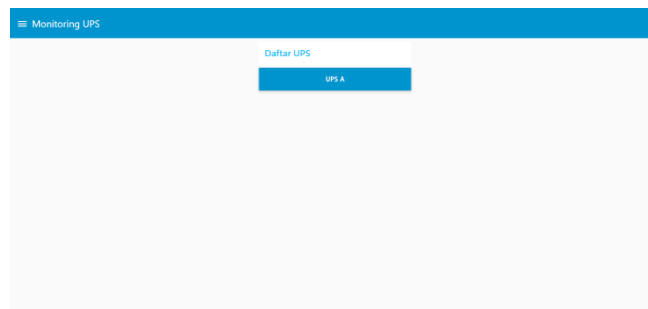
Tahap implementasi dimulai dengan pembangunan sistem berdasarkan desain yang telah dibuat. Pada perangkat keras, sensor daya dihubungkan dengan mikrokontroler ESP32. Mikrokontroler diprogram menggunakan Arduino IDE untuk membaca data dari sensor dan mengirimkannya ke server broker MQTT melalui koneksi *Wi-Fi*. Pada perangkat lunak, broker MQTT HiveMQ Cloud dikonfigurasi untuk mengelola komunikasi antara mikrokontroler dan aplikasi *monitoring*. Selanjutnya, aplikasi *monitoring* berbasis *dashboard* dikembangkan menggunakan Node-RED untuk menerima dan menampilkan data dari broker. Implementasi ini memastikan sistem siap untuk diuji.

Skema *wiring* diagram yang sudah digambarkan pada tahap perancangan kemudian dilakukan perakitan perangkat pada tahap implementasi. Perakitan dilakukan sesuai dengan *wiring* diagram yang sudah digambarkan agar sistem dapat berjalan sesuai dengan fungsinya. Berikut adalah hasil dari perakitan perangkat yang dilakukan.



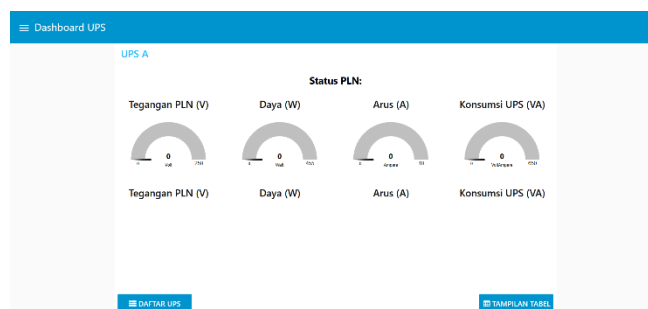
Gambar 4. Foto Perakitan Perangkat

Perakitan perangkat yang sudah dilakukan seperti pada Gambar 4, semua perangkat dihubungkan dan akan mendapatkan suplai listrik, sehingga perangkat tersebut siap digunakan untuk masuk ke tahap pengujian sistem. Sistem pemantauan UPS harus dapat dipantau secara mudah agar pengujian sistem berjalan dengan maksimal yang ditampilkan oleh *dashboard*. Berikut adalah tampilan halaman daftar UPS.



Gambar 5. Tampilan Halaman Monitoring UPS

Halaman monitoring UPS pada Gambar 5 digunakan untuk menampilkan daftar UPS. Pada penelitian ini dilakukan hanya menggunakan satu buah UPS untuk dipantau. Sehingga pada halaman monitoring UPS hanya terdapat satu UPS A. UPS A berbentuk *button* yang harus di-*click* oleh *user* untuk menuju halaman *dashboard* pemantauan UPS. Berikut adalah tampilan halaman *dashboard*.



Gambar 6. Tampilan Halaman Dashboard UPS

Halaman *dashboard* UPS pada Gambar 6 terdapat beberapa indikator sebagai parameter pemantauan penggunaan konsumsi listrik dan status kondisi PLN sebagai sumber listrik utama. Pengukuran penggunaan konsumsi listrik diimplementasikan dalam bentuk *gauge* dan *chart* sedangkan Status PLN akan menampilkannya dalam bentuk *text* yang memiliki *padding* yang berubah warna sesuai dengan kondisinya. Pada halaman ini juga terdapat *button* navigasi untuk kembali ke halaman monitoring UPS (Gambar 5). Terdapat juga *button* untuk menampilkan data dalam bentuk tabel yang ditampilkan sebagai berikut.

Waktu	PLN Status	Tegangan PLN (V)	Daya (W)	Arus (A)	Konsumsi UPS (VA)	MQTT Delay (ms)
16:58:52	X Mati	257.000	45.800	0.505	179.685	10071.000
16:58:52	X Mati	232.500	43.200	0.507	177.877	10072.000
16:58:52	X Mati	232.600	43.300	0.509	178.393	10071.000
16:58:52	X Mati	252.800	45.800	0.509	178.495	10071.000
16:57:52	X Mati	252.900	45.800	0.508	178.515	10072.000
16:57:42	X Mati	233.100	43.200	0.509	178.548	10071.000
16:57:32	X Mati	233.300	43.300	0.510	178.963	10076.000
16:57:21	X Mati	253.600	45.000	0.510	179.196	10072.000
16:57:11	X Mati	243.700	43.200	0.511	179.421	10071.000
16:57:01	X Mati	234.300	43.000	0.511	179.727	10072.000

Gambar 7. Halaman Tampilan Tabel



Pada halaman tampilan tabel yang ditunjukkan Gambar 7, terdapat penambahan kolom ‘Waktu’ dan kolom ‘MQTT Delay’. Kolom ‘Waktu’ digunakan untuk menunjukkan waktu pada saat data baru ditambahkan ke dalam tabel, sementara kolom ‘MQTT Delay’ digunakan untuk menunjukkan waktu yang dibutuhkan mulai dari proses *publish* data oleh ESP32 hingga data tersebut diterima dan ditampilkan pada *dashboard* dalam satuan milidetik (ms). Terdapat juga *button* ‘DAFTAR UPS’ untuk kembali ke halaman monitoring UPS (Gambar 5) dan terdapat *button* ‘CSV’ untuk mengunduh data tabel dalam bentuk CSV.

#### D. Pengujian Sistem

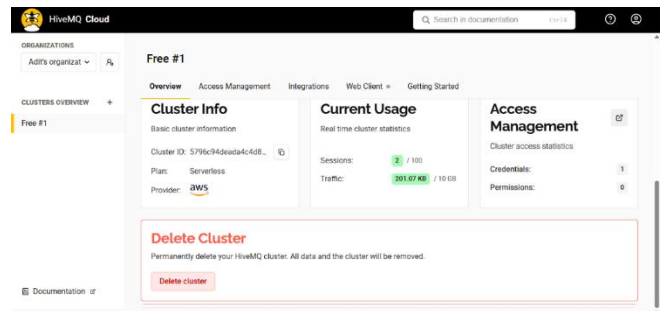
Pada tahap pengujian, sistem diuji untuk memastikan setiap komponen berfungsi dengan baik. Pengujian perangkat keras mencakup uji coba sensor daya untuk membaca parameter seperti tegangan, arus, dan daya listrik dengan akurat. Selain itu, konektivitas *Wi-Fi* pada mikrokontroler ESP32 diuji untuk memastikan data dapat dikirimkan ke *broker* MQTT. Pengujian perangkat lunak dilakukan untuk memastikan *broker* MQTT berfungsi sebagai perantara yang efektif dalam mendistribusikan data ke aplikasi *monitoring*. Aplikasi *monitoring* juga diuji untuk memastikan data ditampilkan secara *real-time* sesuai spesifikasi.

### III. HASIL DAN PEMBAHASAN

Pengujian sistem dilakukan untuk seberapa efektif sistem yang akan digunakan sebelum sistem tersebut diimplementasikan pada kondisi nyata dalam ekosistemnya. Alat pemantauan yang sudah dirakit akan ditempatkan di rak jaringan yang di dalamnya terdapat perangkat jaringan seperti router, switch, access point, dan sebagainya. Perangkat jaringan tersebut diukur untuk mengetahui parameter listrik seperti daya, arus, konsumsi UPS, dan tegangan listrik dari sumber PLN. Alat ini juga digunakan untuk mendeteksi status kondisi listrik PLN.

#### A. Pengujian MQTT

Pengujian pertama dilakukan pada MQTT sebagai protokol komunikasi perangkat IoT yang berfungsi untuk mentransformasikan data yang ditangkap oleh sensor dan diteruskan ke *broker* MQTT agar data tersebut dapat ditampilkan pada halaman *dashboard*. Berikut tampilan *broker* MQTT HiveMQ Cloud.

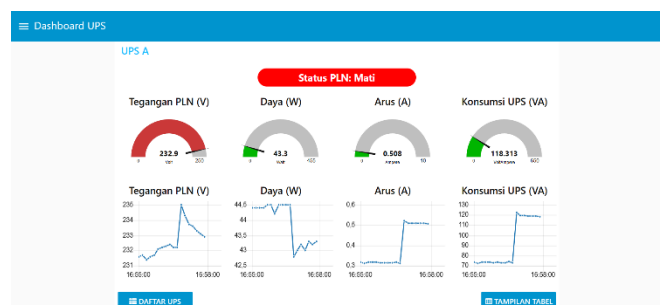


Gambar 8. Tampilan Broker HiveMQ Cloud

Pada tampilan *broker* MQTT HiveMQ yang ditunjukkan Gambar 8, *Current Usage* dapat menampilkan *traffic* data yang melalui *broker* MQTT HiveMQ, hal itu membuktikan bahwa penggunaan MQTT sudah berfungsi.

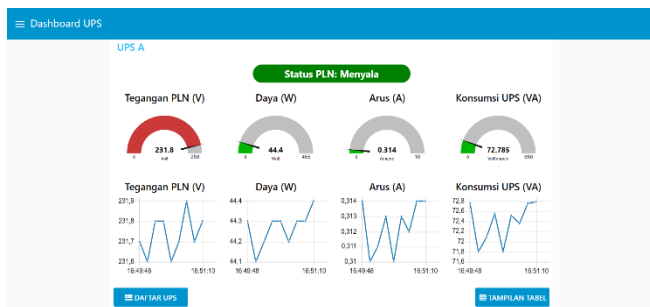
#### B. Pengujian Modul Relay dan PZEM-004T

Pengujian selanjutnya dilakukan pada Modul relay yang digunakan untuk mendeteksi status listrik PLN dan sensor PZEM-004T yang berfungsi untuk mengukur parameter listrik. Data yang dikumpulkan akan dikirimkan ke ESP32 dan selanjutnya akan ditampilkan di halaman *dashboard*. Berikut adalah tampilan halaman *dashboard* yang menunjukkan pengujian sistem.



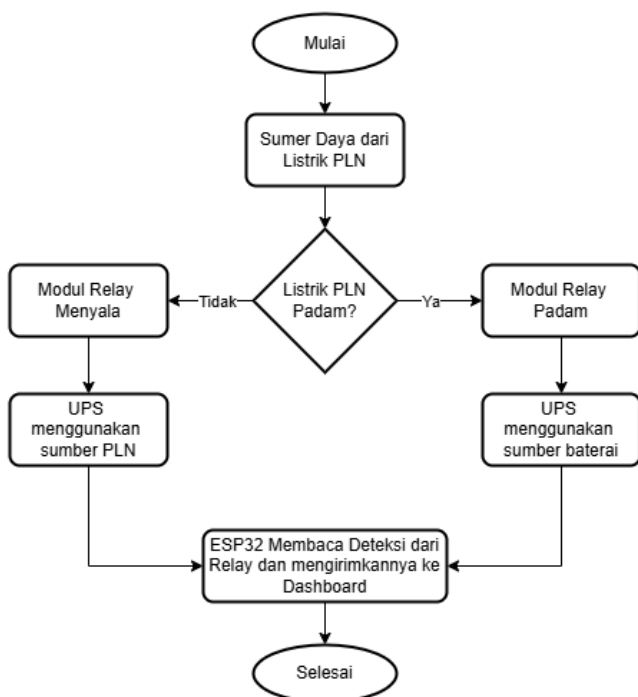
Gambar 9. Tampilan Status Dashboard PLN Mati

Pada saat listrik PLN sedang padam, maka status PLN pada *dashboard* akan menampilkan teks “Mati” dan *padding* berubah menjadi warna merah seperti yang ditampilkan pada Gambar 9. Sedangkan sensor PZEM-004T tetap berfungsi karena mendapatkan daya listrik dari UPS sehingga data parameter listrik dapat dipantau meskipun listrik PLN sedang padam. Pada saat listrik PLN berfungsi kembali, maka *dashboard* UPS ditampilkan sebagai berikut.



Gambar 10. Tampilan Status Dashboard PLN Menyala

Setelah listrik PLN berfungsi kembali, maka status PLN pada *dashboard* akan menampilkan teks “Menyala” dan *padding* berubah menjadi warna hijau seperti yang ditampilkan pada Gambar 10. dengan berubahnya teks dan warna *padding* yang sesuai dengan kondisinya, menunjukkan sistem pengujian dari *relay* berfungsi dengan baik. Pengukuran parameter listrik yang dilakukan oleh sensor PZEM-004T juga berfungsi baik pada saat listrik PLN menyala maupun mati.



Gambar 11. Flow Chart Pendeteksian Status Sumber Daya

Pendeteksian status sumber daya dilakukan oleh modul relay. Modul relay dihubungkan dengan adapter 5 volt untuk memberikan power kepada relay tersebut, kemudian adapter dihubungkan dengan sumber daya dari PLN. Seperti pada gambar 11. saat PLN padam, secara otomatis modul relay tersebut tidak akan mendapatkan power dan mengubah status listrik PLN dari status menyala ke status mati. Pada kondisi status listrik PLN mati, saat itu juga UPS berfungsi

menggunakan mode baterai dan halaman *dashboard* menampilkan status PLN sesuai kondisinya. Begitupun sebaliknya, pada saat listrik PLN menyala kembali, UPS berpindah dari sumber daya baterai menjadi sumber daya listrik PLN. UPS yang digunakan pada penelitian ini berjenis *non-smart* yaitu bertipe *line interactive* UPS.

Terdapat beberapa parameter listrik yang diukur pada penelitian ini, diantaranya yaitu tegangan AC (220V) digunakan untuk mengukur tegangan dari listrik PLN sebagai sumber listrik utama [16]. Listrik utama ini dihubungkan ke stop kontak yang telah dirakit dengan sensor PZEM-004T dan dihubungkan ke modul *relay* untuk mendeteksi kondisi listrik PLN. Daya nyata ( $P$ ) digunakan untuk mengukur arus pada setiap perangkat jaringan yang menjadi beban listrik dengan dihubungkan ke stop kontak yang telah dirakit dengan sensor PZEM-004T [17], [18]. Arus ( $I$ ) yang diukur pada penelitian ini berjenis AC yang merupakan parameter *default* yang sama dengan tegangan AC (220V). Meskipun perangkat jaringan menggunakan arus DC, namun parameter ini digunakan untuk menghitung parameter daya semu ( $S$ ) [19], [20]. Daya Semu ( $S$ ) digunakan sebagai peringkat daya dalam satuan *volt-ampere* yang menunjukkan beban maksimum pada UPS [21], [20]. Dengan parameter ini, beban listrik dapat diukur untuk mengetahui seberapa banyak kapasitas UPS yang sudah digunakan. Tegangan, daya, dan arus merupakan parameter yang dapat langsung diukur dengan sensor PZEM-004T [20]. Sedangkan parameter daya semu ( $S$ ) dihasilkan menggunakan rumus perhitungan pada Persamaan 1 berikut [22], [23].

$$S = V \times I \quad (1)$$

Pada Persamaan (1), diperlihatkan bahwa daya semu ( $S$ ) dihitung dengan mengalikan tegangan ( $V$ ) dan arus ( $I$ ). Satuan yang digunakan untuk daya semu adalah volt-ampere ( $VA$ ), sedangkan tegangan dinyatakan dalam volt ( $V$ ) dan arus dinyatakan dalam ampere ( $A$ ).

### C. Pengujian Akurasi PZEM-004T Terhadap Status PLN

Pengujian tahap akhir yaitu pengujian terhadap status kondisi listrik PLN yang diukur oleh modul relay dan pengujian terhadap parameter listrik yang diukur oleh sensor PZEM-004T dalam bentuk tabel. Beban listrik yang digunakan adalah Access Point U6-Pro yang memiliki konsumsi daya maksimal 13W [24], Switch USW-Lite-8-PoE memiliki konsumsi daya maksimal 8W tanpa PoE *Output* [25] dan Router RB4011iGS+5HacQ2HnD-IN memiliki konsumsi daya maksimal 23W tanpa perangkat yang terhubung [26]. Pada pengujian ini dilakukan perhitungan nilai galat rata-rata tegangan dari listrik PLN yang diukur menggunakan PZEM-004T dan alat pembanding menggunakan digital multimeter true RMS (*Root Mean Square*) Aneng M118A untuk menguji efektifitas sistem yang dihitung dari rumus Persamaan (2) berikut [27], [28].

$$\% \text{ Error} = \frac{|\text{eksak} - \text{perkiraan}|}{\text{eksak}} \times 100 \quad (2)$$

Persamaan (2) digunakan untuk menghitung nilai persentase kesalahan. Perhitungan ini dilakukan dengan menghitung selisih mutlak antara nilai eksak dan nilai perkiraan, kemudian membaginya dengan nilai eksak, dan hasilnya dikalikan dengan 100. Eksak merupakan nilai pembanding yang diukur menggunakan multimeter, sementara perkiraan adalah nilai yang diperoleh dari pengukuran menggunakan modul PZEM-004T.

Pada pengujian ini dilakukan 100 kali percobaan dengan waktu 10 detik setiap satu kali percobaan yang mencakup parameter tegangan dari sensor dan multimeter, arus, daya, konsumsi daya (VA), serta delay komunikasi MQTT. Data dikumpulkan selama proses pengujian sistem, di mana status PLN diujicoba dengan transisi dari status PLN 'Menyala' ke status PLN 'Mati' sebanyak 4 kali transisi secara bergantian yang menghasilkan total status PLN 'Menyala' sebanyak 52 data dan status PLN 'Mati' sebanyak 48 data. Pengujian ini dilakukan untuk mengetahui performa sistem saat transisi dan dalam kondisi stabil.

Berdasarkan Tabel III, terdapat perbedaan signifikan antara kondisi PLN menyala dan PLN mati pada beberapa parameter utama sistem. Rata-rata tegangan yang terukur oleh sensor PZEM-004T saat PLN menyala adalah 231,169 V, sedangkan saat PLN mati meningkat menjadi 234,898 V. Pengukuran tegangan menggunakan multimeter menunjukkan tren yang serupa, dengan nilai 232,231 V saat menyala dan 235,763 V saat mati. Perbedaan ini menunjukkan bahwa *output* UPS menghasilkan tegangan sedikit lebih tinggi dari jaringan PLN.

Selisih antara tegangan PZEM-004T dan multimeter digunakan untuk mengetahui tingkat akurasi sensor. Galat rata-rata saat PLN menyala adalah 0,464%, dan sedikit menurun menjadi 0,367% saat UPS aktif. Nilai ini

menunjukkan bahwa sensor PZEM-004T cukup akurat, dengan nilai galat pengukuran di bawah 0,5% terhadap alat ukur standar, bahkan saat terjadi perubahan sumber daya.

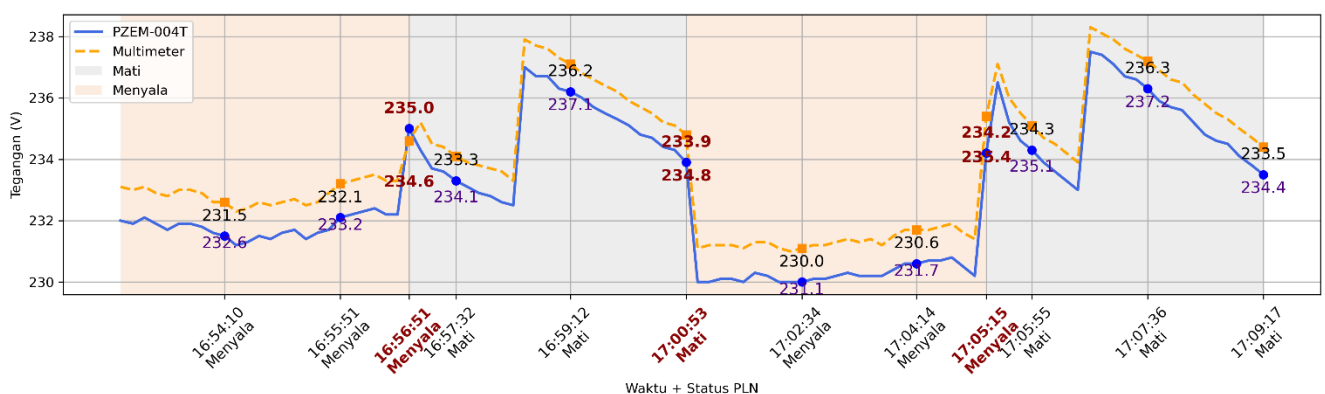
Dari sisi arus, terjadi peningkatan yang signifikan dari 0,326 A saat PLN menyala menjadi 0,487 A saat UPS menyuplai beban, meskipun daya (Watt) relatif stabil, yaitu 44,525 W saat menyala dan 43,552 W saat mati. Terlihat pula pada peningkatan signifikan konsumsi daya semu (VA), dari 73,6 VA menjadi 117,8 VA.

Sementara itu, *delay* pengiriman data melalui MQTT tetap sangat stabil di kedua kondisi, dengan rata-rata 10071 ms, menunjukkan bahwa komunikasi data dari ESP32 ke broker MQTT berjalan tanpa gangguan atau lonjakan waktu respons meskipun terjadi perpindahan sumber daya dari PLN ke UPS. Ini menunjukkan bahwa sistem dapat mempertahankan performa *monitoring real-time* secara konsisten.

Secara keseluruhan, data statistik ini menunjukkan bahwa sistem yang dirancang memiliki akurasi yang tinggi, komunikasi yang stabil, serta dapat mendeteksi dan merespons transisi daya secara efektif. Variasi antara kondisi PLN menyala dan mati dapat ditangkap dengan jelas oleh sistem, baik dari sisi tegangan, arus, maupun performa pengiriman data.

TABEL III  
STATISTIK PARAMETER SISTEM BERDASARKAN STATUS PLN

Parameter	Rata-Rata	
	PLN Menyala	PLN Mati
Tegangan PZEM (V)	231.169	234.898
Tegangan Multimeter (V)	232.231	235.763
Galat Tegangan (%)	0.464	0.367
Arus (A)	0.326	0.487
Daya (W)	44.525	43.552
Konsumsi UPS (VA)	75.362	114.316
MQTT Delay (ms)	10071.538	10071.688



Gambar 12. Grafik Tegangan PZEM dan Multimeter terhadap Waktu

Grafik tegangan terhadap waktu yang ditampilkan pada Gambar 12 menunjukkan perbandingan antara hasil pengukuran tegangan listrik menggunakan sensor PZEM-004T dan alat ukur multimeter. Sumbu horizontal (X)



menunjukkan waktu pengambilan data secara real-time, sementara sumbu vertikal (Y) menyatakan besarnya tegangan listrik dalam satuan volt (V). Dua garis yang ditampilkan mewakili masing-masing metode pengukuran yaitu garis biru untuk PZEM-004T dan garis oranye putus-putus untuk multimeter. Pada titik-titik tertentu, ditampilkan nilai tegangan untuk memperjelas perbandingan. Selain itu, grafik juga menampilkan area berwarna sebagai penanda status sumber listrik dari PLN, status PLN menyala dan PLN mati ditunjukkan dengan warna *background* yang berbeda sesuai dengan status PLN tersebut. Sampel yang ditampilkan terdiri dari 10 data yang diambil secara merata dari keseluruhan data, dengan interval waktu sekitar 1 menit 40 detik antar data. Selain itu, data tambahan ditampilkan secara khusus pada setiap terjadinya transisi status sumber daya PLN. Dari grafik ini dapat diamati bahwa hasil pembacaan PZEM-004T cenderung konsisten mengikuti pola dari multimeter, dengan perbedaan atau galat yang relatif kecil. Penanda dan label berwarna merah pada titik-titik tertentu menunjukkan terjadinya transisi status sumber daya, baik dari PLN ke UPS maupun sebaliknya.

Untuk memperjelas pengujian selama pengukuran dilakukan, disusun pula pada Tabel IV yang menunjukkan perubahan status daya berdasarkan waktu. Tabel ini

menampilkan data parameter utama seperti tegangan, arus, daya, dan waktu delay komunikasi MQTT yang ditampilkan dalam 10 data pertama untuk status PLN menyala dan 10 data terakhir untuk status PLN mati. Perbedaan nilai antara hasil pengukuran PZEM-004T dan multimeter yang ditampilkan pada Tabel IV berada dalam rentang galat yang kecil, umumnya kurang dari 1% yang sesuai dengan Tabel III. Pengujian ini menegaskan bahwa sensor memberikan hasil yang cukup akurat meskipun terjadi perubahan status sumber daya. Delay komunikasi MQTT yang relatif konstan di sekitar 10071–10076 ms selama kedua segmen waktu ini juga menunjukkan bahwa proses transmisi data berlangsung stabil tanpa adanya gangguan, baik dalam kondisi normal maupun saat terjadi switching ke UPS.

Berdasarkan hasil pengujian, sistem pemantauan yang dibangun menggunakan sensor PZEM-004T dan protokol komunikasi MQTT menunjukkan performa yang baik dalam mendeteksi dan merekam data parameter kelistrikan. Nilai galat yang dihasilkan tetap dalam rentang yang dapat diterima untuk keperluan pemantauan, dengan nilai tertinggi di bawah 1%. Hal ini membuktikan bahwa sensor PZEM-004T layak digunakan dalam sistem pemantauan daya untuk perangkat jaringan yang terhubung dengan UPS.

TABEL IV  
PENGAMATAN PARAMETER SISTEM BERDASARKAN STATUS PLN SECARA BERKALA

No	Waktu	Status PLN	Tegangan PLN (V)			Daya (W)	Arus (A)	Konsumsi UPS (VA)	MQTT Delay (ms)
			Multimeter	PZEM-004T	Galat (%)				
1	16.52.39	Menyala	233.1	232	0.472	44.3	0.311	72.152	10072
2	16.52.50	Menyala	233	231.9	0.472	44.1	0.312	72.3528	10071
3	16.53.00	Menyala	233.1	232.1	0.429	44.3	0.313	72.6473	10072
4	16.53.10	Menyala	232.9	231.9	0.429	44.2	0.311	72.1209	10071
5	16.53.20	Menyala	232.8	231.7	0.473	44.2	0.309	71.5953	10071
6	16.53.30	Menyala	233	231.9	0.472	44.3	0.313	72.58469	10072
7	16.53.40	Menyala	233	231.9	0.472	44.2	0.314	72.8166	10071
8	16.53.50	Menyala	232.9	231.8	0.472	44.3	0.314	72.7852	10071
9	16.54.00	Menyala	232.6	231.6	0.430	44.4	0.317	73.41721	10072
10	16.54.10	Menyala	232.6	231.5	0.473	44.3	0.315	72.9225	10076
...	...	...	...	...	...	...	...	...	...
91	17.07.46	Mati	236.9	235.9	0.422	43.9	0.457	107.8063	10071
92	17.07.56	Mati	236.6	235.7	0.380	43.8	0.454	107.0078	10072
93	17.08.06	Mati	236.5	235.6	0.381	43.6	0.453	106.7268	10071
94	17.08.16	Mati	236.1	235.2	0.381	43.9	0.454	106.7808	10071
95	17.08.26	Mati	235.8	234.8	0.424	44	0.454	106.5992	10072
96	17.08.36	Mati	235.5	234.6	0.382	43.9	0.453	106.2738	10071
97	17.08.46	Mati	235.3	234.5	0.340	43.8	0.453	106.2285	10072
98	17.08.56	Mati	235	234.1	0.383	43.8	0.452	105.8132	10071
99	17.09.07	Mati	234.7	233.8	0.383	43.9	0.451	105.4438	10071
100	17.09.17	Mati	234.4	233.5	0.384	44	0.449	104.8415	10072

Penggunaan protokol MQTT juga mendukung efisiensi pengiriman data secara *real-time* dengan latensi rendah, memudahkan integrasi dengan berbagai antarmuka

*dashboard monitoring*. Sistem mampu mendeteksi perubahan status PLN secara otomatis melalui modul relay dan merespons perubahan sumber daya dari PLN ke mode baterai UPS tanpa kegagalan pengiriman data. Sistem ini tidak mengganggu dan tidak diimplementasikan pada internal UPS itu sendiri, melainkan sistem yang terpisah dengan

sistem internal UPS dan berfungsi sebagai perpanjangan dari *output* UPS. Oleh karena itu, sistem ini dapat diimplementasikan pada skala yang lebih besar secara paralel dengan UPS komersial jenis lain yang mempunyai fitur komunikasi SNMP dan sebagainya. Penambahan node sebagai pengembangan jumlah UPS yang dipantau dapat dilakukan dengan protokol MQTT dengan beberapa penyesuaian seperti penambahan menu pada *dashboard*, penyesuaian *layout* dan broker MQTT yang lebih efektif dan efisien.

#### IV. KESIMPULAN

Kesimpulan pada penelitian ini menunjukkan bahwa sistem pemantauan pemakaian daya listrik bagi UPS untuk perangkat jaringan berjalan dengan baik. Penggunaan IoT pada sistem dengan Protokol MQTT, pemantauan parameter kelistrikan dapat dikumpulkan dan ditampilkan pada *dashboard* secara *real-time*. Sensor PZEM-004T terbukti dapat mengukur parameter kelistrikan berupa tegangan, arus, dan daya dengan tingkat akurasi yang baik, ditunjukkan oleh nilai galat rata-rata tegangan sebesar 0.464% saat status PLN menyala dan 0.367% saat PLN mati, berdasarkan perbandingan dengan multimeter digital. Hal ini menunjukkan bahwa sensor tersebut layak digunakan untuk sistem pemantauan dengan kebutuhan pengukuran harian. Sistem juga mampu mendeteksi perubahan status daya secara otomatis, baik saat PLN menyala maupun saat terjadi pemadaman dan daya disuplai oleh UPS. Perubahan tersebut berhasil direkam dengan baik oleh sistem dan ditampilkan melalui *dashboard* aplikasi *monitoring*. Selain itu, terdapat perubahan nilai rata-rata parameter kelistrikan saat sumber daya beralih dari PLN ke mode baterai UPS, di antaranya tegangan mengalami kenaikan yang diukur menggunakan PZEM-004T sebesar 1.62% dan multimeter digital sebesar 49.38%, peningkatan konsumsi daya oleh UPS sebesar 51.68%, perubahan mengalami kenaikan yang sedikit pada MQTT Delay sebesar 0.00149%. Hal ini menunjukkan bahwa sistem tidak hanya dapat membaca data secara *real-time*, tetapi juga mampu mengamati daya UPS saat terjadi perubahan sumber listrik.

Berdasarkan hasil pengujian dan pengalaman selama proses penelitian, terdapat beberapa saran yang dapat diberikan untuk pengembangan sistem ke depan. Penambahan fitur notifikasi seperti peringatan otomatis melalui *email*, atau *push notification* ke aplikasi *mobile* atau ke halaman *dashboard* ketika terjadi perubahan status daya seperti perpindahan dari PLN ke mode baterai UPS, agar pengguna dapat mengambil tindakan lebih cepat terhadap gangguan daya listrik. Integrasi penyimpanan data berbasis *cloud* seperti Firebase, Google Cloud, atau AWS IoT agar data hasil pengukuran dapat disimpan dalam jangka panjang dan memungkinkan analisis historis, pelaporan konsumsi daya, serta pengambilan keputusan berbasis data. Implementasi AI dan *Machine Learning* diperlukan untuk

mengamati dan analisis pola data untuk memprediksi waktu kapasitas baterai terhadap beban listrik yang terhubung.

#### DAFTAR PUSTAKA

- [1] Pita, "Serba-serbi Pentingnya Implementasi Manajemen Jaringan Bagi Bisnis," PT. Blue Power Technology. Accessed: Dec. 10, 2024. [Online]. Available: <https://www.bluepowertechnology.com/news-blog/serba-serbi-pentingnya-implementasi-manajemen-jaringan-bagi-bisnis/>
- [2] J. P. M and D. BMeenakshi Sundaram, "IoT Based Power Monitoring System," SSRG International Journal of Computer Science and Engineering, vol. 8, pp. 4–7, 2021, doi: 10.14445/23488387/IJCSE-V8I4P102.
- [3] Z. Sun *et al.*, "IoT Based Building Energy Monitoring and Controlling System Using LoRa Modulation and MQTT Protocol Low-Power Wide Area Network Construction Method Based on LoRa Technology for Agricultural Science and Technology Parks IoT Based Building Energy Monitoring and Controlling System Using LoRa Modulation and MQTT Protocol", doi: 10.1088/1757-899X/1096/1/012069.
- [4] Y. Apriani, M. Rais Asadullah, M. Hurairoh, U. Muhammadiyah Palembang, J. Jenderal Ahmad Yani, and U. Seberang Ulu, "Monitoring Uninterruptible Power Supply (UPS) Berbasis Internet Of Things (IoT)," vol. 9, no. 1, pp. 723–734, 2022.
- [5] S. C. Prasath, N. Darwin, R. S. Ramkumar, S. Nithishkumar, and P. L. Somasundharam, "IoT-Powered UPS Battery Monitoring: Ensuring High availability and reliability for Critical Systems," in *E3S Web of Conferences*, EDP Sciences, Jul. 2023. doi: 10.1051/e3sconf/202339904007.
- [6] J. Grace Jency Gnanammal, J. R. Khesore, P. V Anandam, R. R. Kamalakkannan, Y. R. Annie Beasent, and R. Royal, "IoT based smart UPS battery life monitoring system," in *International Conference on Computer Vision and Internet of Things 2023 (ICCV/IOT'23)*, 2023, pp. 324–329. doi: 10.1049/icp.2023.2896.
- [7] S. Kazadi, M. Thokozile, and K. A. Ogudo, "Design and Monitoring of a Voltage battery sensor of an Uninterruptible Power Supply (UPS) by means of an Arduino," in *2020 IEEE PES/IAS PowerAfrica*, 2020, pp. 1–5. doi: 10.1109/PowerAfrica49420.2020.9219950.
- [8] A. S. Salunkhe, Y. K. Kanse, and S. S. Patil, "Internet of Things based Smart Energy Meter with ESP 32 Real Time Data Monitoring," in *2022 International Conference on Electronics and Renewable Systems (ICEARS)*, 2022, pp. 446–451. doi: 10.1109/ICEARS53579.2022.9752144.
- [9] R. Andri Atmoko, H. Jovan Wardhana, I. Sapta Muktabar, M. Izzun Ni, and Z. Pramudia Alfarhisi, "Real-Time Energy Monitoring and Management in Educational Institutions: A Case Study of Universitas Brawijaya," *International Journal on Smart Material and Mechatronics*, vol. 11, no. 2, p. 2024, doi: 10.20342/IJSM.11.2.
- [10] A. S. Alexander, "Cloud-Based Mqtt Protocol For Power Monitoring And Socket Control System," 2023.
- [11] L. Shrinivasan and D. Samje, "UPS Power Measurement and Prediction Towards Video Surveillance Systems," in *2022 IEEE 4th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA)*, 2022, pp. 381–385. doi: 10.1109/ICCCMLA56841.2022.9989002.
- [12] G. P. Naik and A. U. Bapat, "A Brief Comparative Analysis on Application Layer Protocols of Internet of Things: MQTT, CoAP, AMQP and HTTP," *International Journal of Computer Science and Mobile Computing*, vol. 9, no. 9, pp. 135–141, Sep. 2020, doi: 10.47760/ijcsmc.2020.v09i09.014.
- [13] D. Silva, L. I. Carvalho, J. Soares, and R. C. Sofia, "A performance analysis of internet of things networking protocols: Evaluating MQTT, CoAP, OPC UA," *Applied Sciences (Switzerland)*, vol. 11, no. 11, Jun. 2021, doi: 10.3390/app11114879.
- [14] J. M. Ramadhan, R. Mardiati, and I. N. Haq, "IoT Monitoring System for Solar Power Plant Based on MQTT Publisher / Subscriber

- Protocol,” *Proceeding of 2021 7th International Conference on Wireless and Telematics, ICWT 2021*, 2021, doi: 10.1109/ICWT52862.2021.9678503.
- [15] K. A. Yasa *et al.*, “IoT-based Electrical Power Recording using ESP32 and PZEM-004T Microcontrollers,” 2023, doi: 10.32996/jcsts.
- [16] B. Demeianto *et al.*, “Rancang Bangun Panel Automatic Transfer Switch (ATS) Pada Pembangkit Listrik Tenaga Surya Sebagai Catu Daya Kincir Air Pada Tambak Perikanan Design And Build Of Automatic Transfer Switch (ATS) Panel On Solar Power Plant As A Main Power Supply For Paddle Wheel Aerator At Fishery Pond,” 2022.
- [17] A. Shodiq, S. Baqaruzi, and A. Muhtar, “Perancangan Sistem Monitoring dan Kontrol Daya Berbasis Internet Of Things,” *ELECTRON : Jurnal Ilmiah Teknik Elektro*, vol. 2, no. 1, pp. 18–26, May 2021, doi: 10.33019/electron.v2i1.2368.
- [18] N. Setiaji *et al.*, “Analisis Konsumsi Daya Dan Distribusi Tenaga Listrik.”
- [19] A. Rizal Tama and S. Winardi, “Monitoring Arus Listrik Dan Kontrol Circuit Breaker Untuk Arus Lebih Berbasis Internet Of Things (IoT).”
- [20] I. Surya *et al.*, “Sistem monitoring beban listrik dan perbaikan faktor daya menggunakan PZEM004T dan dashboard Adafruit berbasis IoT,” *JITEL (Jurnal Ilmiah Telekomunikasi, Elektronika, dan Listrik Tenaga)*, vol. 3, no. 3, pp. 235–246, Sep. 2023, doi: 10.35313/jitel.v3.i3.2023.235-246.
- [21] C. Loeffler and E. Spears, “Electrical: uninterruptible power supply system,” *Data Center Handbook: Plan, Design, Build, and Operations of a Smart Data Center*, pp. 483–520, 2021.
- [22] “VA Versus Watts | Eaton.” Accessed: Apr. 10, 2025. [Online]. Available: <https://www.eaton.com/us/en-us/products/backup-power-ups-surge-it-power-distribution/backup-power-ups/va-versus-watts--eaton.html>
- [23] D. Subhash Patil, M. Mujtahid Ansari, N. S. Mahajan, and D. S. Patil, “Review of kVAh (Kilo Volt Ampere Hour) Billing-Pros and Cons to Utility and Consumers,” 2020. [Online]. Available: <https://www.researchgate.net/publication/377575362>
- [24] “UniFi U6 Pro - Tech Specs.” Accessed: Apr. 12, 2025. [Online]. Available: <https://techspecs.ui.com/unifi/wifi/u6-pro>
- [25] “UniFi Lite 8 PoE - Tech Specs.” Accessed: Apr. 12, 2025. [Online]. Available: <https://techspecs.ui.com/unifi/switching/usb-lite-8-poe?subcategory=all-switching>
- [26] “MikroTik Routers and Wireless - Products: RB4011iGS+5HacQ2HnD-IN.” Accessed: Apr. 12, 2025. [Online]. Available: [https://mikrotik.com/product/rb4011igs\\_5hacq2hnd\\_in](https://mikrotik.com/product/rb4011igs_5hacq2hnd_in)
- [27] R. Gunawan, T. Andhika, . S., and F. Hibatulloh, “Monitoring System for Soil Moisture, Temperature, pH and Automatic Watering of Tomato Plants Based on Internet of Things,” *Telekontran : Jurnal Ilmiah Telekomunikasi, Kendali dan Elektronika Terapan*, vol. 7, no. 1, pp. 66–78, Apr. 2019, doi: 10.34010/telekontran.v7i1.1640.
- [28] F. Tanjung, S. T. Kismanti, M. Firdan Nurdin, and J. T. Mesin, “Analisis Konstruksi Rancang Bangun Mesin Uji Universal Menggunakan Sistem Komputasi”.