

Desain dan Implementasi *Smart Energy Monitoring* Berbasis IoT Laboratorium Teknik Informatika POLNEP

Fitri Wibowo*¹, Suheri², Sarah Bibi³

^{1,2,3}Jurusan Elektro, Politeknik Negeri Pontianak, Pontianak

e-mail fitri.wibowo@polnep.ac.id, suheri@polnep.ac.id, sarah.bibi@polnep.ac.id

Abstrak

Laboratorium Teknik Informatika POLNEP saat ini belum memiliki sebuah sistem yang dapat digunakan untuk memantau penggunaan energi listrik. Hal tersebut menyebabkan penggunaan energi listrik cenderung tidak termonitor dengan baik. Dalam upaya penerapan smart laboratory di Laboratorium Teknik Informatika POLNEP, diperlukan sebuah sistem yang dapat mengindra data terkait penggunaan energi listrik. Tujuan dari penelitian ini adalah tersedianya sebuah sistem berbasis Internet of Things (IoT) yang dapat digunakan untuk memantau dan mengontrol penggunaan energi listrik di Laboratorium Teknik Informatika POLNEP. Selain itu sistem juga diharapkan dapat memberikan notifikasi melalui perangkat mobile kepada teknisi dan kepala laboratorium pada saat mendeteksi anomali pada nilai tegangan, arus, maupun perubahan status relay. Metodologi yang digunakan dalam penelitian ini adalah dengan menggunakan pendekatan pengembangan produk yang terdiri dari perangkat lunak (software) dan perangkat keras (hardware) dalam sebuah project. Secara garis besar tahapan-tahapan yang akan dilakukan adalah: (1) Mendefinisikan project goals; (2) Mendefinisikan project requirements berdasarkan project goals; (3) Mendesain arsitektur sistem; (4) Mendefinisikan alur User Experience (UX); (5) Mengidentifikasi bagian-bagian/tahapan-tahapan pengembangan; (6) Merakit hardware dan membuat kode program untuk setiap bagian/tahapan; (7) Mengintegrasikan setiap bagian/tahapan; serta (8) Uji coba dan troubleshooting.

Kata kunci : Smart Energy Monitoring, Smart Laboratory, IoT

Abstract

Pontianak State Polytechnic (POLNEP) Informatics Engineering Laboratory currently does not have a system that can be used to monitor the use of electrical energy. This causes the use of electrical energy tends not to be monitored properly. In an effort to implement a smart laboratory at the POLNEP Informatics Engineering Laboratory, a system is needed that can sense data related to the use of electrical energy. The purpose of this research is to provide an Internet of Things (IoT) based system that can be used to monitor and control the use of electrical energy in the POLNEP Informatics Engineering Laboratory. In addition, the system is also expected to be able to provide notifications via mobile devices to technicians and laboratory heads when detecting anomalies in voltage, current, or changes in relay status. The methodology used in this research is to use a product development approach consisting of software and hardware in a project. Broadly speaking, the steps to be carried out are: (1) Defining project goals; (2) Define project requirements based on project goals; (3) Designing system architecture; (4) Define User Experience (UX) flow; (5) Identify the parts/stages of development; (6) Assemble hardware and create program code for each section/stage; (7) Integrate each part/stage; and (8) Testing and troubleshooting.

Keywords : Smart Energy Monitoring, Smart Laboratory, IoT

1. PENDAHULUAN

Laboratorium Teknik Informatika POLNEP saat ini belum memiliki sebuah sistem yang dapat digunakan untuk memantau penggunaan energi listrik dari waktu ke waktu. Hal tersebut menyebabkan penggunaan energi listrik cenderung tidak termonitor dengan baik. Dalam upaya penerapan *smart laboratory* di Laboratorium Teknik Informatika POLNEP, diperlukan sebuah sistem yang dapat mengindra data terkait penggunaan energi listrik. Data tersebut selanjutnya akan diproses dan dianalisis menjadi informasi yang dapat dimanfaatkan untuk meningkatkan efisiensi penggunaan energi listrik di lingkungan laboratorium. Selain itu, data telemetri *realtime* terkait nilai tegangan dan arus juga dapat dimanfaatkan untuk memberikan peringatan dini ketika terjadi anomali seperti tegangan (*voltage*) yang tidak stabil maupun arus (*current*) yang di luar spesifikasi.

Tujuan dari penelitian ini adalah tersedianya sebuah sistem berbasis *Internet of Things* (IoT) yang dapat digunakan untuk memantau dan mengontrol penggunaan energi listrik di Laboratorium Teknik Informatika POLNEP. Selain itu sistem juga diharapkan dapat memberikan notifikasi melalui perangkat *mobile* kepada teknisi dan kepala laboratorium pada saat mendeteksi anomali pada nilai tegangan, arus, maupun perubahan status *relay*.

Urgensi dari penelitian ini adalah dibutuhkannya sebuah sistem dengan memanfaatkan teknologi terkini yang dapat menyajikan data tegangan, arus, dan daya yang digunakan pada perangkat elektronik di laboratorium Teknik Informatika POLNEP dalam upaya untuk meningkatkan efisiensi penggunaan energi listrik.

Terdapat beberapa penelitian serupa dalam beberapa tahun terakhir yang dapat dijadikan rujukan. Di antaranya penelitian yang dilakukan pada tahun 2021 oleh Pimple et al[1] yaitu *IOT Based Smart Energy Meter Using ESP32*. Penelitian tersebut bertujuan untuk menerapkan sebuah *smart energy meter* berbasis IoT untuk melacak konsumsi energi listrik secara otomatis pada perumahan. Energy meter yang dibangun dirancang untuk mengirimkan data konsumsi energi listrik kepada konsumen maupun perusahaan pemasok listrik. Sensor yang digunakan adalah HLW8012. Kemudian program akan menghitung total tagihan konsumsi energi listrik berdasarkan interval tertentu. *Microcontroller unit* yang digunakan adalah ESP32. Selain itu informasi tagihan juga dapat dilihat oleh konsumen melalui perangkat *smartphone* kapan pun. Sistem tersebut menghilangkan keterlibatan petugas perusahaan pemasok listrik dalam pemeriksaan KWh meter di rumah pelanggan dalam metode konvensional. Dalam penerapannya penelitian tersebut menggunakan platform *Virtuino*. Penelitian ini masih belum menerapkan sistem pemutus sambungan (ON/OFF) suplai listrik secara manual maupun otomatis.

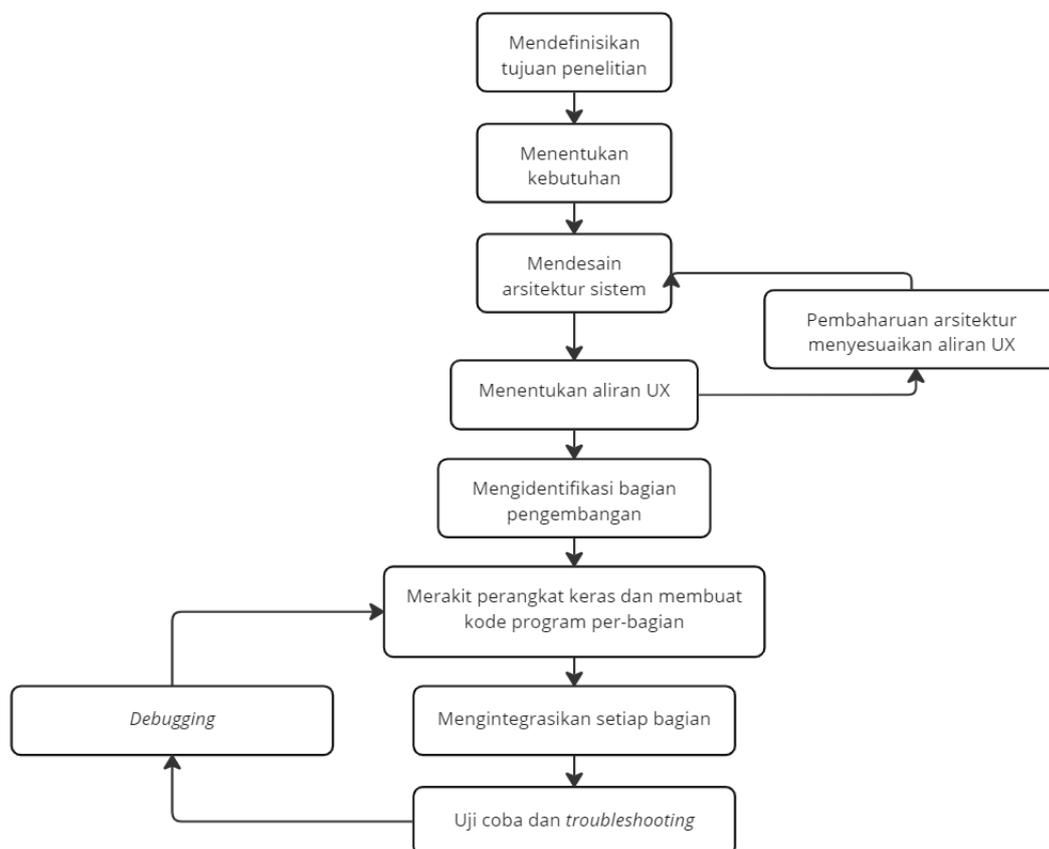
Penelitian selanjutnya dilakukan oleh Radu Gujarathi et al[2] pada tahun 2021, yang berjudul *IoT Based Smart Energy Meter*, juga menerapkan pengukuran penggunaan energi listrik dengan memanfaatkan teknologi IoT. Pada penelitian ini, digunakan *microcontroller* *Arduino* yang dihubungkan dengan KWh meter dan menggunakan modul GSM SIM800 untuk memberikan notifikasi kepada pengguna melalui perangkat seluler. Selain itu data penggunaan energi listrik juga dapat diakses oleh pengguna melalui halaman web yang dijalankan di atas web server pada ESP8266. Penelitian yang dibuat belum menerapkan sistem peringatan maupun notifikasi secara otomatis kepada pengguna dan tidak menggunakan platform IoT sehingga data telemetri tidak tersimpan. Selain itu penelitian tersebut masih membuka peluang untuk pengembangan lebih lanjut dalam hal integrasi dengan *website* yang menyediakan fitur pembayaran tagihan listrik.

Penelitian ketiga dilakukan oleh Souza et al[3] pada tahun 2020 dengan judul *Development of an Energy Meter based on IoT*. Pada penelitian tersebut, telah berhasil diterapkan sebuah pemantau penggunaan energi listrik dengan memanfaatkan ESP8266 sebagai *microcontroller unit*. Sensor arus dan tegangan dirakit secara manual dengan menggunakan sensor arus SCT013 100A dan sensor tegangan ZMPT101B. Data dari perangkat IoT dikirim ke platform IoT *Smart Automation using Internet of Things* (SAIOT) yang dikembangkan oleh Costa et al[4].

Berdasarkan perbandingan teknologi yang digunakan, metode dan tujuan dari penelitian sejenis yang ada, maka pada penelitian ini penulis mencoba untuk merancang dan menerapkan *smart energy meter* berbasis IoT di laboratorium Teknik Informatika Politeknik Negeri Pontianak.

2. METODE

Metodologi yang digunakan pada penelitian ini adalah dengan menggunakan metode pengembangan produk yang terdiri dari perangkat keras dan perangkat lunak pada pengembangan sebuah proyek berbasis IoT. Gambar 1 menunjukkan langkah-langkah metodologi yang digunakan pada pengembangan proyek berbasis IoT. Metode yang digunakan merupakan pengembangan dari metodologi pengembangan perangkat lunak di mana terdapat beberapa langkah tambahan khususnya pada bagian perancangan[5]. Secara umum, langkah-langkah yang dilakukan meliputi: (1) Mendefinisikan tujuan penelitian; (2) Menentukan kebutuhan berdasarkan tujuan yang telah ditetapkan; (3) Mendesain arsitektur sistem; (4) Menentukan aliran pengalaman pengguna (UX); (5) Mengidentifikasi bagian/fase pengembangan; (6) Merakit perangkat keras dan membuat kode program untuk setiap bagian/fase; (7) Mengintegrasikan setiap bagian/fase; dan (8) Melakukan uji coba dan *troubleshooting*. Selain langkah-langkah tersebut, terdapat aktivitas berulang lainnya yang menghubungkan langkah 4 dengan langkah 3 serta langkah 8 dan 6.

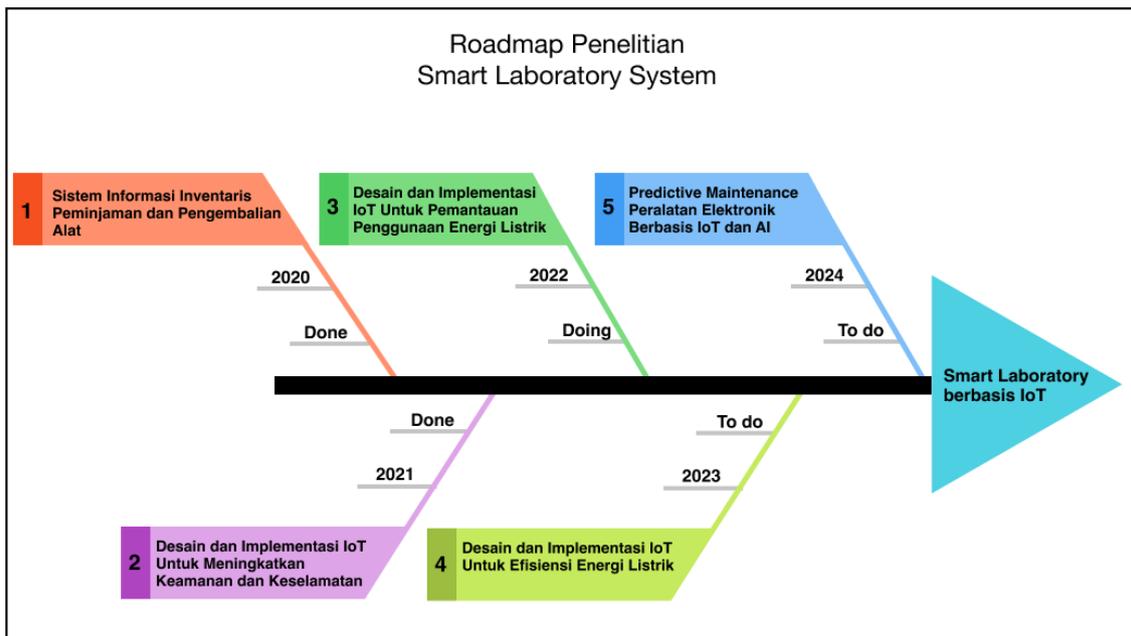


Gambar 1. Metodologi pengembangan proyek IOT

Aktivitas yang dilakukan di setiap tahapan pada metodologi yang digunakan adalah sebagai berikut:

2.1 Mendefinisikan tujuan penelitian

Tahapan ini merupakan tahapan pertama dalam metodologi penelitian yang digunakan, di mana tujuan utama dari penelitian didefinisikan. Tujuan dari penelitian ini adalah untuk membangun sebuah sistem berbasis *Internet of Things* (IoT) yang dapat digunakan untuk memantau dan mengontrol penggunaan energi listrik di Laboratorium Teknik Informatika POLNEP. Sesuai dengan *roadmap* penelitian pada Gambar 2, pada tahun 2022, penelitian akan berfokus pada pemanfaatan IoT untuk memantau penggunaan energi listrik di Laboratorium Teknik Informatika. Sistem juga akan diintegrasikan dengan aplikasi lain seperti Google Assistant dan Telegram. Google Assistant dimanfaatkan untuk mendapatkan maupun mengatur status relay pada perangkat IoT Sonoff POWR2 melalui *voice command* pada perangkat *smartphone*. Sedangkan Telegram akan dimanfaatkan sebagai aplikasi yang akan memberikan notifikasi kepada kepala laboratorium dan teknisi pada saat terjadi *event* tertentu seperti perubahan status *relay*, ketidakstabilan tegangan, dan arus yang melebihi spesifikasi.



Gambar 2. Metodologi pengembangan proyek IOT

2.2 Menentukan kebutuhan

Setelah mendefinisikan tujuan dari penelitian, langkah berikutnya adalah menentukan kebutuhan project pada penelitian. Pada tahapan kedua ini, kebutuhan terkait perangkat keras (*hardware*) maupun perangkat lunak (*software*) didefinisikan. Selain itu kebutuhan fungsional dan non-fungsional sistem juga ditentukan. Tabel 1 dan 2 menyajikan kebutuhan perangkat keras dan perangkat lunak sistem, sedangkan Tabel 3 menyajikan kebutuhan fungsional dan non-fungsional yang telah diidentifikasi.

Tabel 1. Kebutuhan perangkat keras

No.	Produk	Keterangan
1.	Sonoff POWR2	8 pcs (beban AC untuk 8 ruangan).
2.	Takida Smart Plug	8 pcs (beban peralatan elektronik per-ruangan)
4.	USB To Serial FTDI	1 pcs
5.	Ubiquity UniFi AP	1 unit
6.	Kabel NYM 2x1.5mm	1 roll
7.	Box casing	2 unit, ukuran 12x6x5 (cm)
8.	Cable duct 45x45	3 meter

9.	STB HG680P	1 unit
10.	Wemos D1 Mini	5 pcs
11.	CT sensor 100A	5 pcs
12.	ADC ADS1115	5 pcs
13.	LCD 16x02	5 pcs

Tabel 2. Kebutuhan perangkat keras

No.	Produk	Keterangan
1.	Home Assistant	Minimum versi 2022.5.5
2.	Armbian Linux Amlogic	Versi 2022
3.	Node-RED	Versi 2.2.2
4.	Tasmota	Versi 11.0.0
5.	Arduino IDE	Versi 1.8.19
6.	ESPHome	Versi 2022.8.1

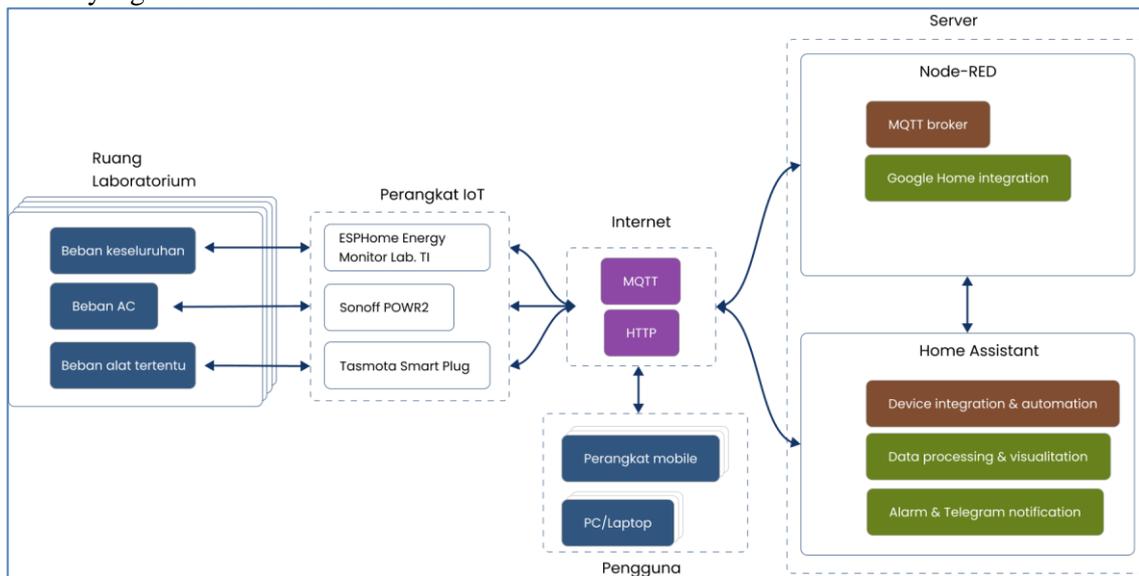
Tabel 3. Kebutuhan fungsional (FR) dan non-fungsional (NFR)

No.	Kode	Statement
1.	FR1	Sistem dapat menampilkan data tegangan, arus, dan daya dalam bentuk <i>dashboard</i> interaktif
2.	FR2	Sistem dapat menampilkan data penggunaan energi listrik per ruangan
3.	FR3	Sistem dapat menampilkan data penggunaan energi listrik total satu gedung
4.	FR4	Sistem dapat menampilkan data penggunaan energi listrik per beban perangkat elektronik tertentu
5.	FR5	Sistem mampu mengirimkan pesan melalui Telegram pada saat terjadi anomali nilai tegangan atau arus listrik, serta pada saat terjadi perubahan status <i>relay</i> (ON/OFF)
6.	FR6	Sistem dapat dikontrol melalui perintah suara <i>Google Assistant</i>
7.	FR7	Sistem dapat menampilkan laporan penggunaan energi listrik berdasarkan interval waktu tertentu (per hari, minggu, bulan, dan tahun)
8.	NFR1	Data <i>realtime</i> disajikan dalam bentuk grafik/ <i>chart</i> dengan interval waktu minimal 10 detik
9.	NFR2	Notifikasi melalui Telegram dapat terkirim dalam waktu kurang dari 1 menit
10.	NFR3	Perintah suara melalui <i>Google Assistant</i> dapat dijalankan oleh sistem dalam waktu kurang dari 10 detik

2.3 Mendesain arsitektur sistem

Tahapan selanjutnya adalah mendesain arsitektur sistem. Pada penelitian ini, pemasangan perangkat IoT akan dilakukan pada beberapa ruangan di laboratorium Teknik Informatika dengan mengukur tiga beban listrik yaitu; 1) lampu, 2) AC, dan 3) stop kontak per ruangan. Data telemetri tegangan, arus, dan status relay akan diambil menggunakan perangkat IoT Sonoff POWR2. Data-data tersebut kemudian dikirim menggunakan protokol MQTT ke MQTT broker yang berjalan di atas Node-RED. Selanjutnya data akan diteruskan ke perangkat lunak Home Assistant. Node-RED juga akan digunakan untuk mengintegrasikan sistem dengan Google Assistant dengan menggunakan pustaka `node-red-contrib-google-smarthome`[6]. Home Assistant digunakan untuk mengolah data dan menyajikan data dalam bentuk *dashboard* sehingga dapat diakses oleh

pengguna melalui web browser dengan protokol HTTPS. Home Assistant juga akan dikonfigurasi untuk memberikan notifikasi melalui Telegram. Gambar 3 menampilkan rancangan arsitektur sistem yang saat ini sudah diidentifikasi.



Gambar 3. Rancangan arsitektur sistem

2.4 Mendesain aliran *User Experience* (UX)

Tahapan berikutnya adalah pendefinisian alur *User Experience*. Pada tahapan ini dibuat alur penggunaan dari sisi pengguna. Pengguna yang dimaksud dalam sistem ini adalah kepala laboratorium dan teknisi laboratorium. Kepala laboratorium memiliki akses penuh pada dashboard aplikasi yang dapat diakses langsung melalui antarmuka pengguna Home Assistant. Home Assistant juga memiliki tampilan antarmuka pengguna dalam bentuk aplikasi *mobile* yang dapat diintegrasikan dengan sistem yang telah dibangun dan digunakan.

2.5 Mengidentifikasi bagian pengembangan

Tahapan selanjutnya adalah mengidentifikasi bagian-bagian pengembangan pada project yang akan dibuat. Pada tahapan ini project pada penelitian ini akan dibagi menjadi beberapa bagian yang lebih kecil. Bagian-bagian ini kemudian akan diselesaikan secara terpisah dan kemudian diintegrasikan secara keseluruhan.

2.6 Merakit perangkat keras dan membuat kode program

Langkah berikutnya adalah merakit perangkat keras, membuat kode program, serta melakukan konfigurasi pada *firmware/software*. Pada tahapan ini, bagian-bagian dari penelitian yang telah diidentifikasi selanjutnya akan dirakit dan dilakukan konfigurasi *firmware* pada setiap *microcontroller unit* (MCU) dari perangkat IoT. *Firmware* yang digunakan pada penelitian ini adalah Tasmota dan ESPHome. Kedua *firmware* tersebut menyediakan fitur untuk pengaturan dasar *firmware* seperti konfigurasi GPIO, WiFi, MQTT serta pengaturan *rule* pada perangkat IoT.

2.7 Mengintegrasikan setiap bagian

Tahapan berikutnya adalah integrasi bagian-bagian pengembangan. Pada tahapan ini setiap bagian-bagian pengembangan yang telah dirakit dan dibuat kode programnya akan diintegrasikan.

2.8 Uji coba dan *troubleshooting*

Langkah terakhir adalah uji coba dan *troubleshooting*. Langkah ini dilakukan untuk memastikan bagian-bagian yang telah diintegrasikan berjalan dengan baik dan tidak terjadi permasalahan. Metode pengujian dilakukan dengan menggunakan metode *black box testing*. Tabel 4 menyajikan

daftar pengujian kebutuhan fungsional dan non fungsional yang akan dievaluasi, prosedur pengukuran atau penilaian, serta kriteria penilaian.

Tabel 4. Daftar pengujian kebutuhan fungsional dan non-fungsional

No.	Kode Pengujian	Deskripsi Pengujian	Masukan	Hasil yang diharapkan	Kriteria Hasil
1.	A.FR1	Sistem dapat menampilkan data tegangan, arus, dan daya dalam bentuk <i>dashboard</i> interaktif.	Menekan menu <i>dashboard</i> pada aplikasi web.	Menuju halaman <i>dashboard</i> dan menampilkan data tegangan, arus, dan daya.	Sesuai/Tidak sesuai
2.	A.FR2	Sistem dapat menampilkan data penggunaan energi listrik per ruangan.	Menekan menu <i>dashboard</i> pada aplikasi web.	Menuju halaman <i>dashboard</i> dan menampilkan data penggunaan energi listrik per ruangan.	Sesuai/Tidak sesuai
3.	A.FR3	Sistem dapat menampilkan data penggunaan energi listrik total satu gedung.	Menekan menu <i>dashboard</i> pada aplikasi web.	Menuju halaman <i>dashboard</i> dan menampilkan data penggunaan energi total dalam satu gedung.	Sesuai/Tidak sesuai
4.	A.FR4	Sistem dapat menampilkan data penggunaan energi listrik per beban perangkat elektronik tertentu.	Menekan menu <i>dashboard</i> pada aplikasi web.	Menuju halaman <i>dashboard</i> dan menampilkan data penggunaan energi listrik per beban perangkat elektronik tertentu.	Sesuai/Tidak sesuai
5.	A.FR5	Sistem mampu mengirimkan pesan melalui Telegram pada saat terjadi anomali nilai tegangan atau arus listrik, serta pada saat terjadi perubahan status relay (ON/OFF).	Memasukkan data <i>dummy</i> arus atau tegangan dan atau mengubah status relay.	Sistem mengirim pesan melalui Telegram.	Sesuai/Tidak sesuai
6.	A.FR6	Sistem dapat dikontrol melalui perintah suara Google Assistant.	Memasukkan perintah suara melalui Google Assistant.	Sistem mengeksekusi perintah suara.	Sesuai/Tidak sesuai
7.	A.FR7	Sistem dapat menampilkan laporan	Menekan menu	Menuju halaman	Sesuai/Tidak sesuai

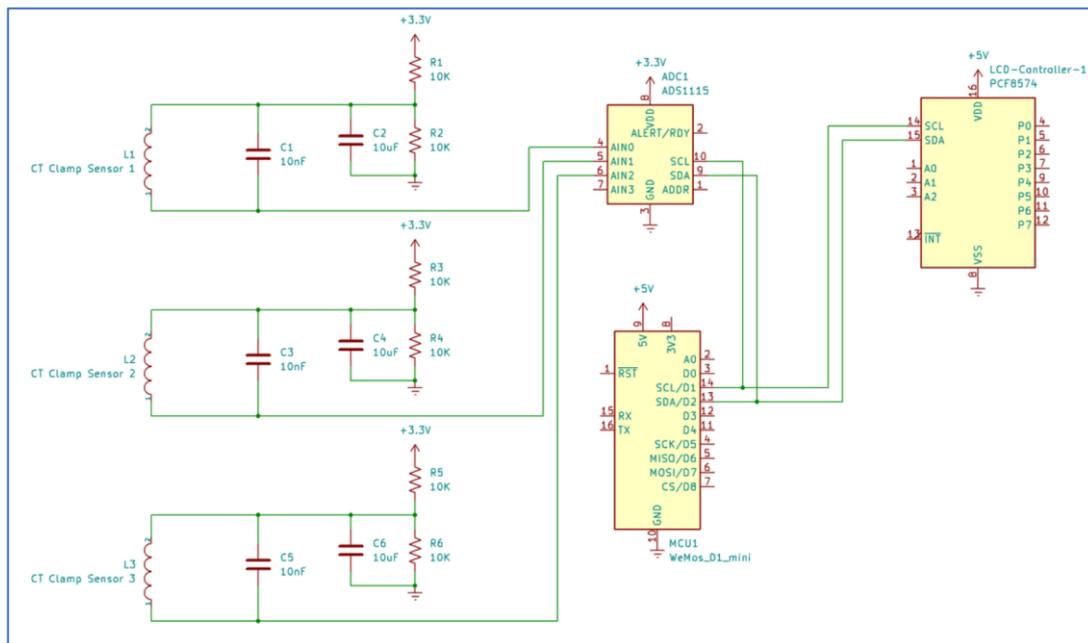
		penggunaan energi listrik berdasarkan interval waktu tertentu (per hari, minggu, bulan, dan tahun).	<i>dashboard history explorer</i> pada aplikasi web.	<i>dashboard history explorer</i> dan menampilkan laporan penggunaan energi listrik berdasarkan interval waktu tertentu (per hari, minggu, bulan, dan tahun).	
8.	A.NFR1	Data <i>realtime</i> disajikan dalam bentuk grafik/ <i>chart</i> dengan interval waktu minimal 10 detik.	Menekan menu <i>dashboard</i> pada aplikasi web.	Data <i>realtime</i> disajikan dalam bentuk grafik/ <i>chart</i> dengan interval waktu minimal 10 detik.	Sesuai/Tidak sesuai
9.	A.NFR2	Notifikasi melalui Telegram dapat terkirim dalam waktu kurang dari 1 menit.	Memasukkan data <i>dummy</i> arus atau tegangan dan atau mengubah status <i>relay</i> .	Notifikasi melalui Telegram dapat terkirim dalam waktu kurang dari 1 menit.	Sesuai/Tidak sesuai
10.	A.NFR3	Perintah suara melalui Google Assistant dapat dijalankan oleh sistem dalam waktu kurang dari 10 detik.	Memasukkan perintah suara melalui Google Assistant.	Perintah suara melalui Google Assistant dapat dijalankan oleh sistem dalam waktu kurang dari 10 detik.	Sesuai/Tidak sesuai

3. HASIL DAN PEMBAHASAN

3.1 Produk *hardware*

Hardware dibuat pada tahapan ke-6 yaitu pada tahapan merakit *hardware* dan konfigurasi *firmware*. Produk *hardware* pada penelitian ini berupa *kit* perangkat IoT untuk memantau penggunaan daya keseluruhan gedung Laboratorium Teknik Informatika Polnep yang disebut dengan *energy monitor* Lab. TI. Perangkat IoT kit tersebut dibuat dengan menggunakan *microcontroller* ESP8266 dengan *board* Wemos D1 mini. Sensor yang digunakan adalah *Current Transformer* (CT) *clamp sensor* dengan tipe YHDC SCT013-000. CT *clamp sensor* SCT013 adalah transformator sensor arus *non-invasif* yang mengukur intensitas arus yang melintasi konduktor tanpa perlu memotong atau memodifikasi konduktor itu sendiri. SCT013-000 dapat mengukur arus maksimal hingga 100A[7]. Hasil pembacaan arus pada CT *clamp sensor* selanjutnya diolah dengan *firmware* ESPHome dan ditampilkan pada LCD display dan dikirimkan ke perangkat lunak Home Assistant. Untuk *analog to digital converter* (ADC), digunakan *external ADC* yaitu ADS1115 agar hasil pembacaan lebih akurat dan memungkinkan untuk menggunakan hingga maksimal 4 input *analog* sekaligus. Hal tersebut berguna pada

kondisi pengukuran jalur 3 *phase*, di mana sensor CT *clamp* yang digunakan adalah 3 buah. Gambar 4 menampilkan rancangan *circuit diagram energy monitor* Lab. TI.



Gambar 4. *Circuit diagram energy monitor* Lab. TI

3.2 Produk *software*

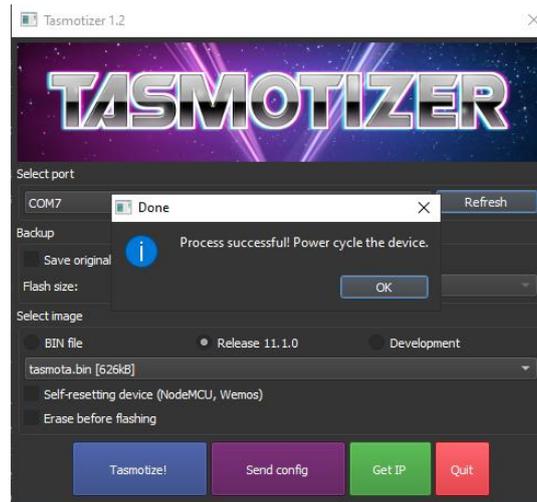
Selain pembuatan perangkat IoT, pada tahapan perakitan *hardware*, telah dilakukan pemasangan dan konfigurasi perangkat lunak yang digunakan pada penelitian. Berikut ini adalah daftar *firmware/software* yang digunakan dan telah dikonfigurasi sesuai dengan kebutuhan penelitian.

1. Firmware Tasmota.
2. Sistem operasi Armbian Linux Amlogic.
3. Node-RED.
4. Home Assistant.

3.2.1 Konfigurasi *firmware* Tasmota

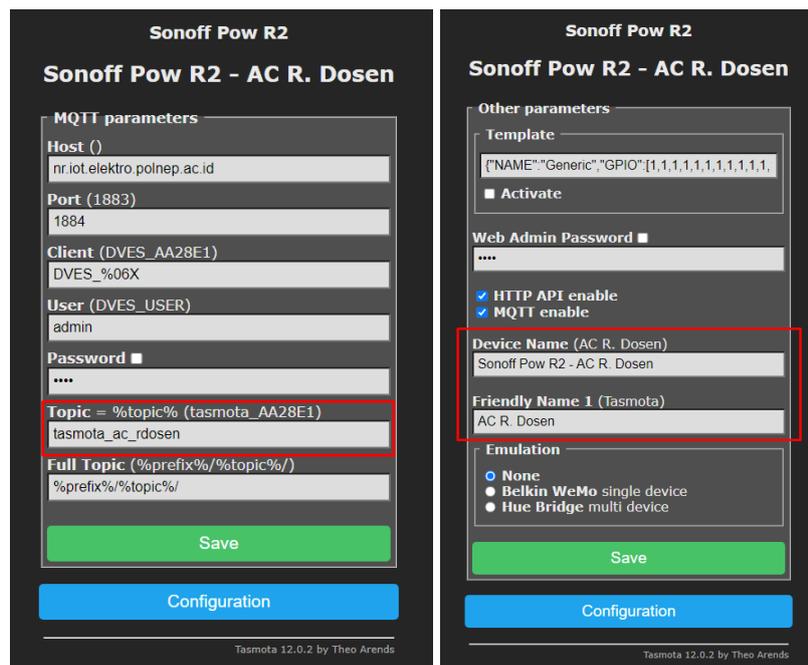
Firmware Tasmota digunakan untuk mempermudah penggunaan peralatan IoT. Dengan menggunakan *firmware*, kode program tidak perlu dibuat secara manual. *Firmware* ini mendukung berbagai jenis perangkat IoT[8]. Perangkat IoT yang dimanfaatkan dan telah dipasang *firmware* Tasmota di antaranya adalah Sonoff POWR2 dan Takida *Smart Plug*. Sonoff POWR2 digunakan sebagai sensor daya dan *circuit breaker* pada beban listrik AC pada setiap ruangan di gedung laboratorium. Sedangkan Takida *Smart Plug* digunakan sebagai sensor pembanding pada proses kalibrasi perangkat IoT *energy monitor* Lab. TI dan sebagai sensor tegangan pada jalur masuk rangkaian listrik di gedung laboratorium. Tegangan tersebut digunakan sebagai parameter input penghitungan daya listrik pada perangkat IoT *energy monitor* Lab. TI.

Pemasangan *firmware* Tasmota pada kedua perangkat IoT menggunakan metode yang berbeda. Pada perangkat IoT Sonoff POWR2, Tasmota dipasang dengan cara *flashing* menggunakan USB TTL. Dengan metode tersebut, perangkat IoT perlu dibongkar dan dipasang kabel *jumper* pada pin GPIO tertentu dan dihubungkan melalui USB TTL ke komputer. Adapun perangkat lunak yang digunakan pada proses *flashing* Sonoff POWR2 adalah aplikasi Tasmotizer.



Gambar 5. Proses flashing Sonoff POWR2 pada aplikasi Tasmotizer

Setelah proses *flashing*, *firmware* Tasmota dikonfigurasi melalui WebUI yang telah disediakan. Konfigurasi pada WebUI yang dilakukan meliputi konfigurasi pada menu WiFi, menu MQTT, dan menu “Other parameters”. Selain itu juga dilakukan kalibrasi tegangan yang dilakukan melalui Console pada WebUI. Gambar 13 menampilkan tangkapan layar konfigurasi pada menu “Other parameters” dan MQTT pada salah satu perangkat Sonoff POW R2.



Gambar 6. Konfigurasi menu MQTT dan Other parameters pada Tasmota

3.2.2 Konfigurasi Armbian Linux Amlogic

Langkah-langkah yang telah dilakukan agar sistem operasi Armbian Linux dapat berjalan dan digunakan meliputi:

1. Instalasi Armbian Linux.
2. Memindah instalasi Armbian Linux ke eMMC.
3. *Pengaturan hostname*.
4. *Pengaturan jaringan*.
5. *Pengaturan zona waktu*.

6. Pengaturan *remote login*.
7. Ekspansi ruang penyimpanan ke MicroSD.

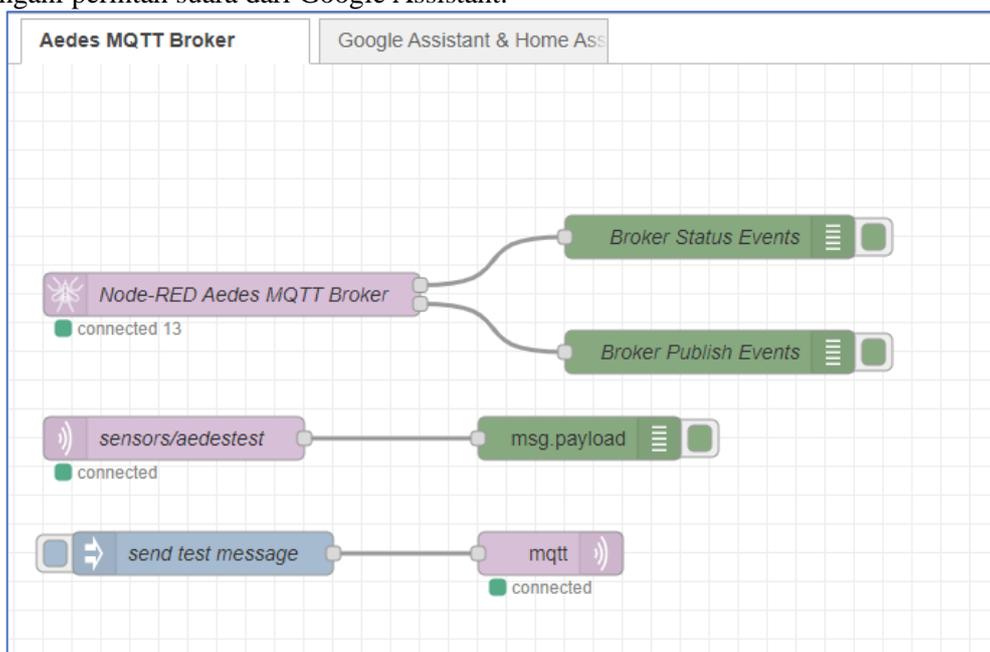
Instalasi Armbian dilakukan pada perangkat Set Top Box (STB) / TV box HG860-P. Instalasi dilakukan dalam beberapa tahapan. Sebelum konfigurasi jaringan dan konfigurasi *remote login* dilakukan, Armbian perlu dihubungkan terlebih dahulu ke layar monitor, *mouse* dan *keyboard*. Jika jaringan dan *remote login* sudah dikonfigurasi, maka sistem operasi dapat diakses melalui *remote login* dengan SSH.

3.2.3 Konfigurasi Node-RED

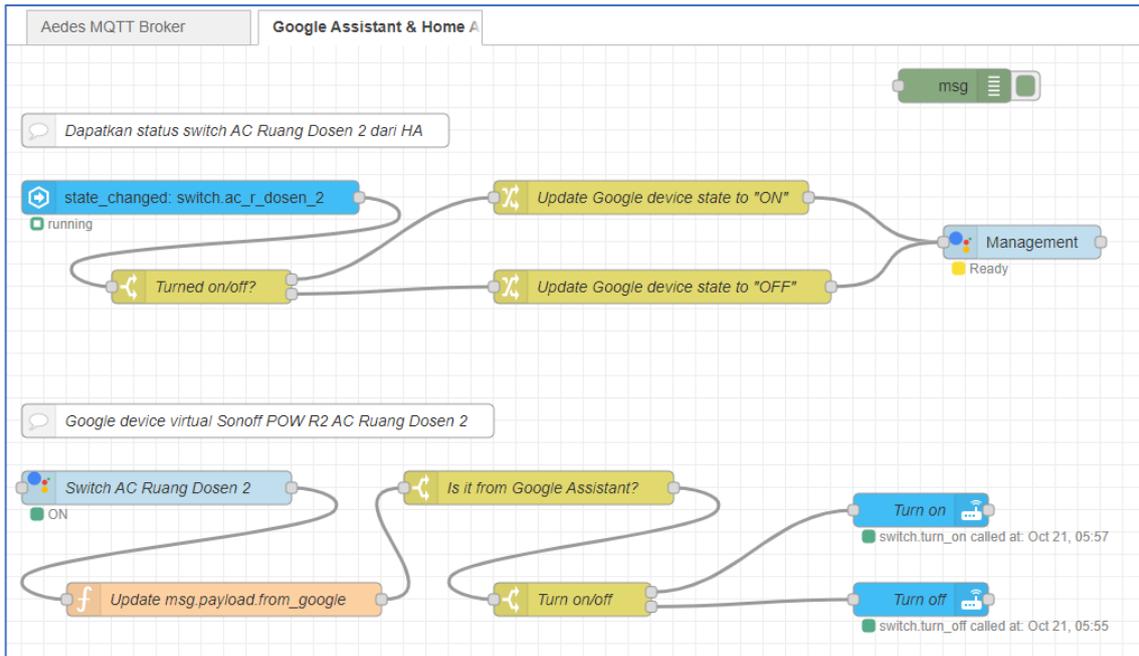
Node-RED telah dipasang pada *server* yang terpisah dari Home Assistant sebagai penyedia layanan MQTT dengan harapan jika terjadi permasalahan pada Home Assistant, perangkat IoT tetap dapat berfungsi atau mengirimkan data *telemetry* karena tetap dapat berkomunikasi dengan MQTT *broker*. Agar dapat diakses dari jaringan internet, telah dibuat domain khusus untuk Node-RED di alamat URL <https://nr.iot.elektro.polnep.ac.id>. Langkah-langkah yang dilakukan untuk instalasi dan konfigurasi Node-RED adalah sebagai berikut:

1. Memasang Node-RED melalui perintah Node Package Manager. Karena Node-RED adalah perangkat lunak yang dibangun dengan bahasa pemrograman Javascript maka Node-RED dapat dipasang dengan perintah `npm`[9].
2. Mengatur *reverse proxy* Node-RED dengan Apache.
3. Menambahkan autentikasi user ke Node-RED agar Node-RED hanya dapat diakses oleh user yang telah terdaftar[10].
4. Menambahkan service Node-RED ke sistem operasi agar dapat berjalan secara otomatis pada saat sistem operasi booting.

Agar dapat digunakan untuk mengintegrasikan perangkat IoT dengan Home Assistant dan Google Assistant, telah dipasang beberapa *palette* pada Node-RED, di antaranya: 1.) `node-red-contrib-aedes`, sebagai MQTT broker; 2.) `node-red-contrib-google-smarthome`, untuk integrasi Google Assistant; dan 3.) `node-red-contrib-home-assistant-websocket`, untuk integrasi Home Assistant. Berikut ini adalah *node* yang telah dikonfigurasi untuk menjalankan MQTT *broker* dan menangani perintah suara dari Google Assistant.



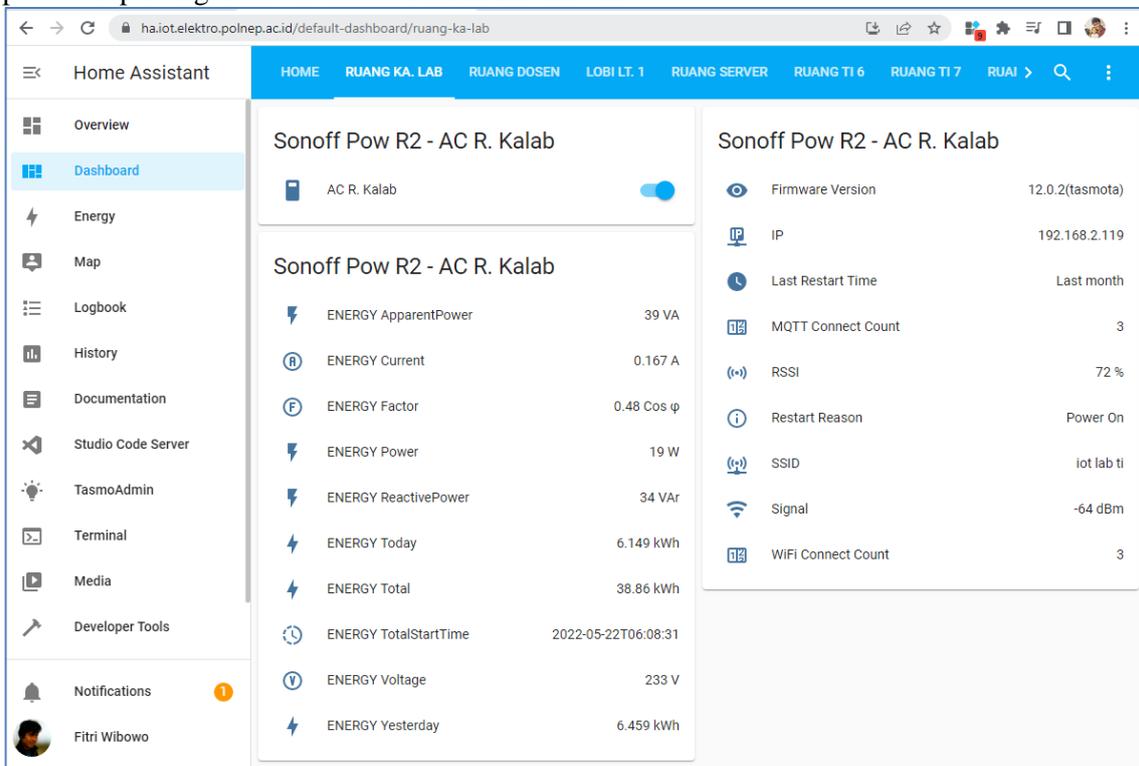
Gambar 7. Node Aedes MQTT broker pada Node-RED



Gambar 8. Node Google Assistant dan Home Assistant integration pada Node-RED

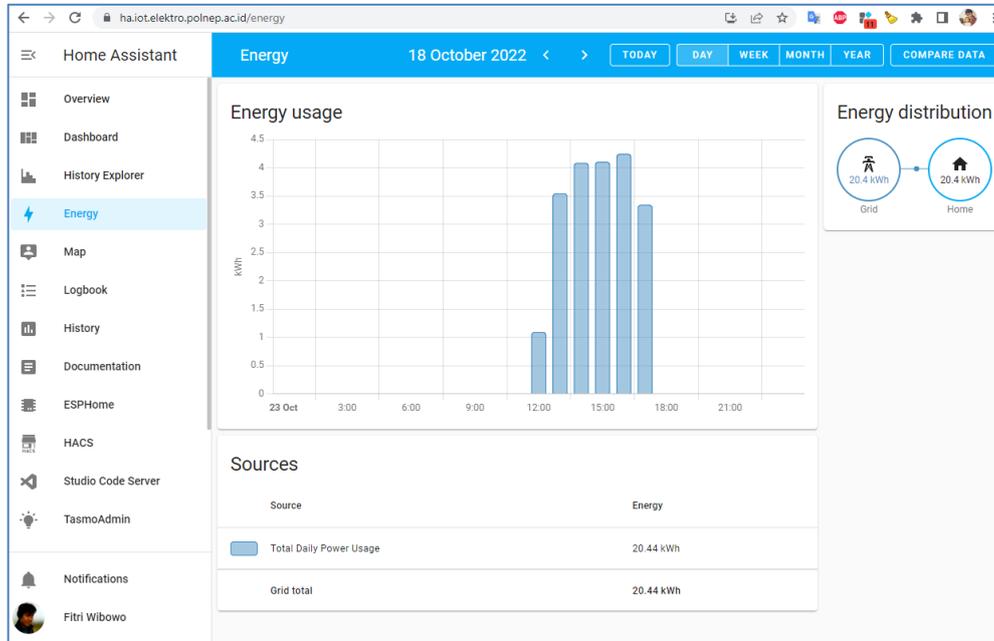
3.2.3 Konfigurasi Node-RED

Perangkat lunak Home Assistant telah berhasil dipasang dan dikonfigurasi untuk memantau dan mengontrol status dan penggunaan energi listrik pada beban listrik *Air Conditioner* (AC) dengan menggunakan Sonoff POW R2 di seluruh ruangan laboratorium Teknik Informatika. Berikut ini adalah tampilan *dashboard* untuk menampilkan statistik penggunaan energi listrik perangkat AC pada setiap ruangan laboratorium Teknik Informatika.



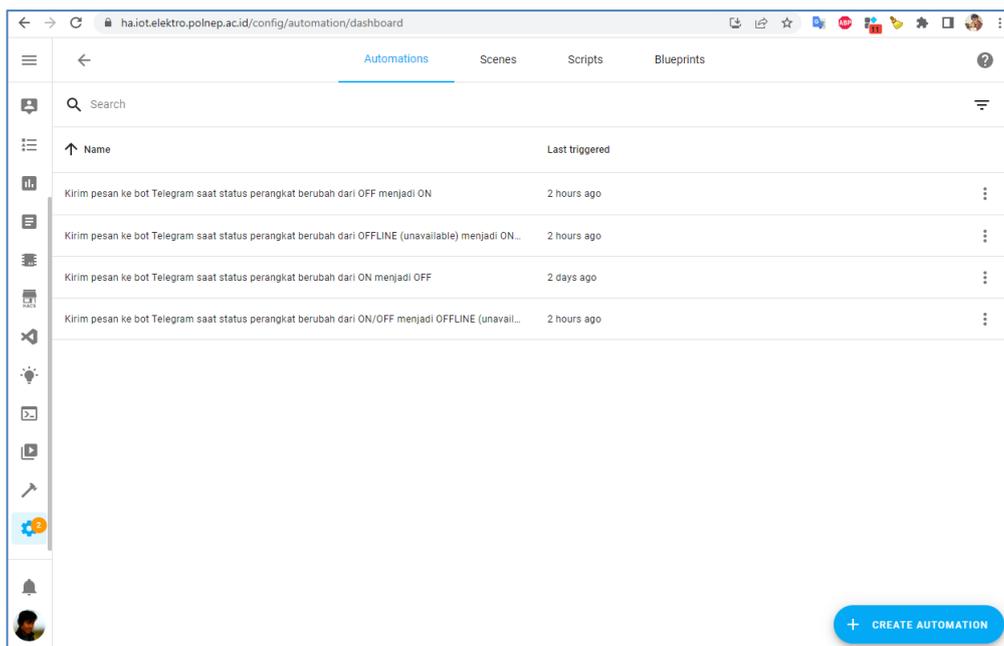
Gambar 9. Dashboard Ruang Kepala Laboratorium

Home Assistant juga telah dikonfigurasi untuk menampilkan penggunaan energi listrik satu gedung yang didapatkan dari perangkat IoT ESPHome energy monitor. Tampilan penggunaan energi listrik dapat diakses melalui menu Energy pada bagian navigasi sidebar. Tampilan penggunaan energi dapat diatur berdasarkan kurun waktu tertentu seperti per hari, minggu, bulan, dan tahun. Gambar berikut ini adalah tampilan grafis dari menu Energi pada Home Assistant.



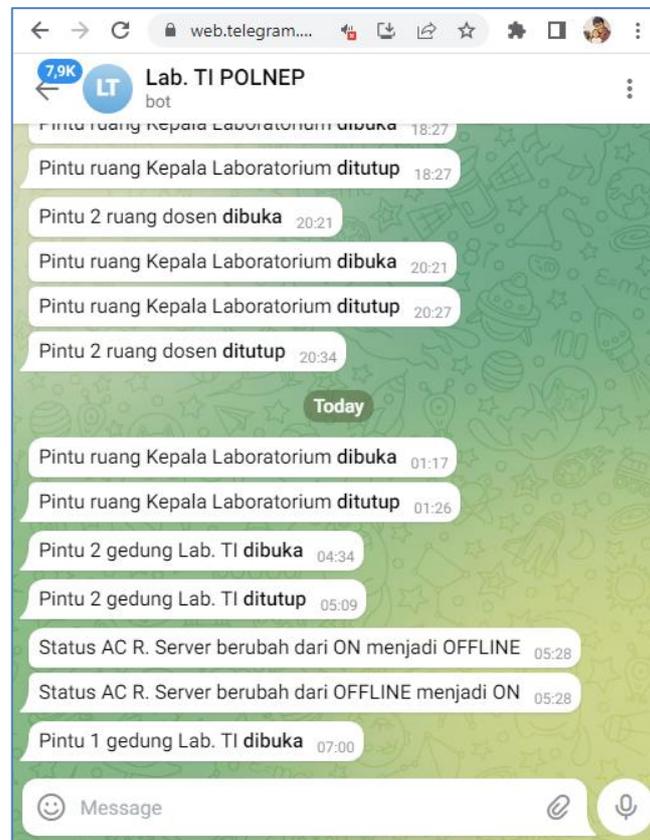
Gambar 10. Menu Energy pada Home Assistant

Selain konfigurasi *dashboard*, pada Home Assistant juga telah dilakukan konfigurasi menu *automation* yang digunakan untuk mengirimkan pesan melalui aplikasi Telegram pada saat terjadi perubahan status Sonoff POW R2 pada perangkat AC. Berikut ini adalah tampilan halaman konfigurasi *automation* pada Home Assistant.



Gambar 11. Halaman konfigurasi *automation* pada Home Assistant

Berikut ini adalah tampilan pesan notifikasi perubahan status relay pada perangkat IoT yang berhasil terkirim melalui aplikasi Telegram.



Gambar 12. Tampilan pesan hasil Home Assistant *automation* pada aplikasi Telegram

4. KESIMPULAN DAN SARAN

Adapun kesimpulan yang didapat pada penelitian ini adalah sebagai berikut:

1. *Firmware* Tasmota telah berhasil dipasang dan dikonfigurasi pada perangkat Sonoff POWR2. Perangkat tersebut telah berhasil digunakan sebagai perangkat IoT untuk memantau penggunaan energi listrik pada perangkat elektronik AC di setiap ruangan laboratorium Teknik Informatika serta status perangkat (ON/OFF) dapat dikendalikan melalui Home Assistant dan Google Assistant.
2. Node-RED telah berhasil dikonfigurasi dan digunakan sebagai MQTT broker yang menghubungkan perangkat IoT dengan Home Assistant. Node-RED juga telah berhasil digunakan untuk mengintegrasikan Google Home dengan Home Assistant.
3. Seluruh kebutuhan fungsional dan non-fungsional berhasil diterapkan dan dipenuhi meskipun belum semua ruangan belum tersedia perangkat IoT untuk memantau penggunaan energi listrik per perangkat misalnya lampu, komputer personal atau laptop.
4. Home Assistant telah berhasil dikonfigurasi dan digunakan untuk mengintegrasikan, menyajikan data, dan melakukan otomasi perangkat IoT berdasarkan kondisi tertentu. Contoh otomasi yang telah diterapkan adalah pengiriman notifikasi melalui aplikasi Telegram secara otomatis pada saat terjadi perubahan status *entity* saklar pada Sonoff POWR2 dan *smart wall switch*. Selain itu, notifikasi Telegram juga akan dikirim pada saat terjadi perubahan status *online* atau *offline* setiap perangkat IoT.

5. Perangkat IoT ESPHome *energy monitor* telah berhasil dibuat dan diterapkan untuk memantau penggunaan daya listrik di gedung laboratorium Teknik Informatika. Penggunaan daya yang didapat dari hasil pengukuran merupakan *apparent power* mengingat sebagian besar peralatan elektronik yang digunakan bersifat *non-resistive*. Meskipun demikian, tingkat akurasi yang didapatkan pada penggunaan *non-invasive CT clamp sensor* menunjukkan hasil yang cukup akurat dan masih ada peluang untuk diperbaiki lagi dengan pengambilan *sample* yang lebih banyak dan variasi beban peralatan elektronik saat kalibrasi. Selain itu, data penggunaan energi listrik tersebut tetap dapat digunakan bagi pemangku kepentingan dalam pembuatan kebijakan penggunaan energi listrik serta mengetahui kondisi perangkat elektronik berdasarkan tren penggunaan dayanya.

DAFTAR PUSTAKA

- [1] M. Pimple, S. Thopate, and A. Nikam, "IOT Based Smart Energy Meter Using ESP 32," SSRN Journal, 2021, doi: 10.2139/ssrn.3917892.
- [2] G. A. Gujarathi, S. R. Randive, C. S. Pawar, and Prof. M. Parmale, "IoT Based Smart Energy Meter," IJARSCT, pp. 461–465, Jul. 2021, doi: 10.48175/IJARSCT-1661.
- [3] W. W. V. Souza et al., "Development of an Energy Meter based on IoT," in 2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT, Roma, Italy, Jun. 2020, pp. 241–244. doi: 10.1109/MetroInd4.0IoT48571.2020.9138257.
- [4] J. Costa, D. Araujo, D. R. C. Silva, M. B. Nogueira, and M. C. Rodrigues, "Home Automation Architecture Based on IOT Technologies," in 2018 Workshop on Metrology for Industry 4.0 and IoT, Brescia, Apr. 2018, pp. 63–67. doi: 10.1109/METROI4.2018.8438863.
- [5] P. Desai, Python Programming for Arduino. Packt Publishing Ltd., 2015.
- [6] "NPM - node-red-contrib-google-smarthome library." <https://www.npmjs.com/package/node-red-contrib-google-smarthome> (accessed Apr. 07, 2022).
- [7] YDHC, "SCT013-100-100A-1V datasheet." YDHC.
- [8] "Tasmota Supported Devices Repository," 2022. <https://templates.blakadder.com/> (accessed Apr. 07, 2022).
- [9] "Running Node-RED locally," 2022. <https://nodered.org/docs/getting-started/local> (accessed May 02, 2022).
- [10] "Node-RED - Username/password based authentication," Editor & Admin API security, 2022. <https://nodered.org/docs/user-guide/runtime/securing-node-red#editor--admin-api-security> (accessed May 22, 2022).