RANCANG BANGUN SMART RELAY BERBASIS IOT MENGGUNAKAN ESP32 UNTUK PENGENDALIAN KONSUMSI ENERGI LISTRIK RUMAH TANGGA

TUGAS AKHIR

Disusun Untuk Memenuhi Syarat Kelulusan Program Strata I
pada Sekolah Tinggi Manajemen Informatika Dan Komputer
(STMIK) Palangkaraya

OLEH

ALDYA SAPUTRA
C2155201085
PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
2025

RANCANG BANGUN SMART RELAY BERBASIS IOT MENGGUNAKAN ESP32 UNTUK PENGENDALIAN KONSUMSI ENERGI LISTRIK RUMAH TANGGA

TUGAS AKHIR

Disusun Untuk Memenuhi Syarat Kelulusan Program Strata I
pada Sekolah Tinggi Manajemen Informatika Dan Komputer
(STMIK) Palangkaraya

OLEH

ALDYA SAPUTRA
C2155201085
PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
2025

PERSETUJUAN

RANCANG BANGUN SMART RELAY BERBASIS IOT MENGGUNAKAN ESP32 UNTUK PENGENDALIAN KONSUMSI ENERGI LISTRIK RUMAH TANGGA

Tugas Akhir Ini Telah Disetujui dan Disahkan

Pembimbing I,	Pembimbing II,
Herkules, S.Kom., M.Cs.	Suparno, M.Kom.
NIK. 198510042010106	NIK. 196901041995105

Mengetahui
Ketua STMIK Palangkaraya,

Ir. Hj. Sri Maryamah, M.M.
NIK. 196902021995004

PENGESAHAN

RANCANG BANGUN SMART RELAY BERBASIS IOT MENGGUNAKAN ESP32 UNTUK PENGENDALIAN KONSUMSI ENERGI LISTRIK RUMAH TANGGA

Tugas Akhir ini Telah Diujikan, Dinilai, dan Disahkan
Oleh Tim Penguji pada Tanggal 14 Juni 2025

Tim Penguji Tugas Akhir :

1. Elia Zakharia, M.T. ..... 
   Ketua

2. Veny Cahya Hardita, M.Kom. ..... 
   Sekretaris

3. Amaya Andri Damaini, S.Kom., M.T. ..... 
   Anggota

4. Herkules, S.Kom., M.Cs. ..... 
   Anggota

5. Catharina Elmayantie, M.Pd. ..... 
   Anggota

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama Mahasiswa : **ALDYA SAPUTRA**
N I M : **C2155201085**

menyatakan bahwa Tugas Akhir dengan judul :

RANCANG BANGUN SMART RELAY BERBASIS IOT MENGGUNAKAN ESP32 UNTUK PENGENDALIAN KONSUMSI ENERGI LISTRIK RUMAH TANGGA

adalah hasil karya saya dan bukan merupakan duplikasi sebagian atau seluruhnya dari karya orang lain, kecuali bagian yang sumber informasi dicantumkan.

Pernyataan ini dibuat dengan sebenar-benarnya secara sadar dan bertanggungjawab dan saya bersedia menerima sanksi pembatalan Tugas Akhir apabila terbukti melakukan duplikasi terhadap Tugas Akhir atau karya ilmiah lain yang sudah ada.

Palangka Raya, 14 Juni 2025

Yang Membuat Pernyataan,



ALDYA SAPUTRA

MOTTO DAN PERSEMBAHAN

Keberhasilan adalah sebuah proses. Niat yang tulus, doa yang ikhlas, serta usaha yang maksimal adalah pondasi awal dari setiap keberhasilan. Doamu dan doa dari orang-orang di sekitarmu, ibarat bara api yang mematangkan setiap langkahmu. Allah SWT senantiasa menyertai orang-orang yang sabar dalam meraih keberhasilan. Kesabaran akan mengajarkanmu untuk memahami dan mensyukuri setiap makna di balik sebuah keberhasilan.

Puji syukur kehadirat Allah SWT atas rahmat dan ridho-Nya. Sholawat serta salam semoga tercurah kepada Nabi Muhammad SAW, yang menuntun umat ke jalan yang diridhoi. Tugas Akhir ini dipersembahkan kepada semua pihak yang telah memberikan dukungan dan semangat, sehingga terselesaikan dengan baik. Ucapan terima kasih
kepada :

1. ALLAH SWT,kerena hanya atas izin dan karunianya. Dan ridhonya nabi MUHAMMAD SAW, maka selesai lah Tugas Akhir ini.
2. Kepada kedua orang tua yang telah memberikan dukungan dan semangat serta doa yang tiada hentinya. Selesailah Tugas Akhir ini.
3. Bapak Abdul Hadi, S.T., M.Kom. selaku Ketua Prodi Teknik Informatika STMIK Palangkaraya.
4. Bapak Herkules, S.Kom., M.Cs. selaku pembimbing I dan Bapak Suparno, M.Kom., selaku pembimbing II yang selama ini telah tulus dan ikhlas meluangkan waktu untuk membimbing dalam pembuatan Tugas Akhir ini.
5. Ibu Resiana selaku pemilik barak rumah tangga.

INTISARI

Aldya Saputra, C2155201085, 2025. Rancang Bangun Smart Relay Berbasis IoT Menggunakan Esp32 Untuk Pengendalian Konsumsi Energi Listrik Rumah Tangga, Pembimbing I Herkules, S.Kom., M.Cs. Pembimbing II Suparno, M.Kom.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan smart *relay* berbasis *Internet of Things* (IoT) menggunakan Esp32 guna memantau dan mengendalikan konsumsi energi listrik rumah tangga secara efisien. Latar belakang dari penelitian ini adalah meningkatnya konsumsi listrik akibat bertambahnya perangkat elektronik rumah tangga serta penggunaan yang kurang terkontrol, yang berdampak pada pemborosan energi. Permasalahan yang diangkat yaitu bagaimana merancang sistem yang dapat mengontrol dan memantau penggunaan listrik secara jarak jauh melalui platform internet. Batasan penelitian meliputi penggunaan Esp32 sebagai mikrokontroler utama, aplikasi Blynk sebagai antarmuka kontrol berbasis Android, serta pengujian pada skala rumah tangga kecil.

Jenis penelitian yang digunakan adalah *Research and Development* (R&D) dengan pendekatan *prototype*, melalui tahapan analisis kebutuhan, desain cepat, pembangunan *prototype*, evaluasi pengguna, perbaikan prototype, serta implementasi produk. Teknik pengumpulan data yang digunakan meliputi observasi, dokumentasi, studi kepustakaan, dan wawancara dengan narasumber terkait. Seluruh teknik ini digunakan untuk memperoleh data kebutuhan sistem dan menyesuaikan rancangan alat sesuai dengan kondisi lingkungan rumah tangga.

Hasil penelitian diuji menggunakan *black box testing* untuk memastikan seluruh fungsi perangkat keras dan perangkat lunak berjalan sesuai spesifikasi. Seluruh fitur sistem seperti pengendalian *relay*, pembacaan sensor PZEM-004T, tampilan oled, dan koneksi blynk berfungsi dengan baik tanpa kendala. Pengujian konsumsi energi listrik selama 6 hari menunjukkan penghematan sebesar 1,87 kWh atau sekitar 29,68%, dari konsumsi awal 6,30 kWh menjadi 4,43 kWh setelah penerapan smart *relay*. Hasil ini membuktikan bahwa sistem smart *relay* berbasis IoT mampu mengontrol perangkat listrik secara efektif, terintegrasi, dan membantu mengurangi konsumsi energi listrik secara otomatis.

Kata kunci: *Internet of Things*, Esp32, Smart *Relay*, Energy Monitoring, Blynk.

ABSTRACT

Aldya Saputra, C2155201085, 2025. Design and Construction of IoT-Based Smart Relay Using Esp32 for Controlling Household Electricity Consumption, Advisor I Herkules, S.Kom., M.Cs. Advisor II Suparno, M.Kom.

This study aims to design and implement an Internet of Things (IoT)-based smart relay using Esp32 to monitor and control household electricity consumption efficiently. The background of this study is the increasing electricity consumption due to the increasing number of household electronic devices and uncontrolled use, which has an impact on energy waste. The problem raised is how to design a system that can control and monitor electricity usage remotely via an internet platform. Research limitations include the use of Esp32 as the main microcontroller, the Blynk application as an Android-based control interface, and testing on a small household scale.

The type of research used is Research and Development (R&D) with a prototype approach, through the stages of needs analysis, rapid design, prototype development, user evaluation, prototype improvement, and product implementation. Data collection techniques used include observation, documentation, literature study, and interviews with related sources. All of these techniques are used to obtain system requirements data and adjust the design of the device according to household environmental conditions.

The results of the study were tested using black box testing to ensure that all hardware and software functions run according to specifications. All system features such as relay control, PZEM-004T sensor reading, oled display, and blynk connection functioned properly without any problems. Testing of electrical energy consumption for 6 days showed savings of 1.87 kWh or around 29.68%, from initial consumption of 6.30 kWh to 4.43 kWh after the implementation of the smart relay. These results prove that the IoT-based smart relay system is able to control electrical devices effectively, integrated, and help reduce electrical energy consumption automatically.

Keywords: Internet of Things, Esp32, Smart Relay, Energy Monitoring, Blynk.

KATA PENGANTAR

Alhamdulillah penulis panjatkan kehadirat Tuhan Yang Maha Esa, karena dengan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini yang berjudul "RANCANG BANGUN SMART RELAY BERBASIS IOT MENGGUNAKAN ESP32 UNTUK PENGENDALIAN KONSUMSI ENERGI LISTRIK RUMAH TANGGA". Tugas Akhir ini disusun untuk memenuhi syarat dalam memperoleh gelar Sarjana Komputer di Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya.

Pada kesempatan ini penulis juga menyampaikan terimakasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bimbingan, motivasi, semangat, dalam menyelesaikan penulisan Tugas Akhir ini, yaitu kepada :

1. Resiana, selaku pemilik Barak Rumah Tangga yang memberikan izin selama penulis melakukan penelitian.
2. Herkules, S.Kom., M.Cs., selaku dosen Pembimbing I yang telah meluangkan waktunya untuk memberikan bimbingan dalam penyelesaian Tugas Akhir ini.
3. Suparno, M.Kom., selaku dosen Pembimbing II dalam penulisan Tugas Akhir ini, yang juga banyak memberikan saran dan masukan dalam penulisan Tugas Akhir ini.

Penulis mengharapkan masukan, saran dan kritik yang bersifat membangun untuk menyempurnakan Tugas Akhir ini. Semoga Tugas Akhir ini dapat bermanfaat bagi penulis dan pembaca terutama untuk untuk menambah ilmu pengetahuan.

Palangka Raya, 14 Juni 2025

Penulis

DAFTAR ISI

PERSETUJUAN ..... **ii**
PENGESAHAN ..... **iii**
PERNYATAAN KEASLIAN ..... **iv**
MOTTO DAN PERSEMBAHAN ..... **v**
INTISARI ..... **vi**
ABSTRACT ..... **vii**
KATA PENGANTAR ..... **viii**
DAFTAR ISI ..... **ix**
DAFTAR TABEL ..... **xi**
DAFTAR GAMBAR ..... **xii**
DAFTAR LAMPIRAN ..... **xiv**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang Masalah 1
1.2 Perumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan dan Manfaat 3
1.5 Sistematika Penulisan 4
BAB II TINJAUAN PUSTAKA **6**
2.1 Kajian Teori 6
2.1.1 Rancang Bangun 6
2.1.2 *Internet of Things* (IoT) 7
2.1.3 *Prototype* 8
2.1.4 Mikrokontroler Esp32 9
2.1.5 *Relay* 10
2.1.6 Adaptor 11
2.1.7 Step Down DC 12
2.1.8 Sensor Arus 13
2.1.9 OLED *(Organic Light Emitting Diode)* 15
2.1.10 Kabel Jumper 16
2.1.11 Blynk 17
2.2 Penelitian Yang Relevan 19

BAB III METODOLOGI PENELITIAN 24
3.1 Tinjauan Umum 24
3.2 Jenis penelitian 24
3.3 Desain Penelitian 25
3.4 Instrumen Penelitian 27
3.5 Analisis Kebutuhan 29
1.5.1 Kebutuhan Perangkat Keras 29
1.5.2 Kebutuhan Perangkat Lunak 30
1.5.3 Analisis Data 31
1.5.4 Analisis Proses 31
1.5.5 Analisis Kelemahan 32
3.6 Analisis Desain 32
3.6.1 Desain Proses 32
3.6.2 Desain Perangkat 34
3.6.3 Desain *Prototype* 40
3.7 Skenario Kendali Perangkat 41
BAB IV HASIL DAN PEMBAHASAN 43
4.1 Hasil 43
4.1.1 Implementasi 43
4.1.2 Implementasi Perangkat Keras *(Hardware)* 43
4.1.3 Implementasi Perangkat Lunak *(Software)* 44
4.1.4 Implementasi Alat 69
4.1.5 Pengujian 70
4.2 Pembahasan 78
4.2.1 Rangkaian 78
4.2.2 Pemrograman Perangkat Lunak 79
4.2.3 Alat 79
4.2.4 Analisis Konsumsi Energi 80
BAB V KESIMPULAN DAN SARAN 84
5.1 Kesimpulan 84
5.2 Saran 85

DAFTAR TABEL

Tabel 2.1. Penelitian Yang Relevan 19
Tabel 4.1. Widget Blynk 52
Tabel 4.2. *Black Box Testing* 71
Tabel 4.3. Tabel Perbandingan Konsumsi Energi Listrik 83

DAFTAR GAMBAR

Gambar 2.1. Mikrokontroler Esp32 10
Gambar 2.2. *Relay* dan Bagianya 11
Gambar 2.3. Adaptor 12
Gambar 2.4. Step Down LM25965 13
Gambar 2.5. Sensor Arus PZEM-004T 15
Gambar 2.6. OLED *(Organic Light Emitting Diode)* 16
Gambar 2.7. Kabel Jumper 17
Gambar 3.1. Diagram Blok 32
Gambar 3.2. Desain *Flowchart* 33
Gambar 3.3. Desain Perangkat Keras 34
Gambar 3.4. Desain *Interface* 36
Gambar 3.5. Desain *Interface* Perangkat 37
Gambar 3.6. Desain Antarmuka Pengendalian Perangkat 38
Gambar 3.7. Desain Antarmuka Notifikasi 39
Gambar 3.8. Desain *Prototype* 40
Gambar 4.1. Rangkaian Sistem Monitoring Energi Listrik 44
Gambar 4.2. Tampilan Board Manager 45
Gambar 4.3. Tampilan Instalasi Board Arduino dan Esp32 46
Gambar 4.4. Tampilan Pemilihan Board Esp32 47
Gambar 4.5. Istallasi Blynk 48
Gambar 4.6. Instal Blynk di Arduino IDE 48
Gambar 4.7. Tampilan Registrasi Blynk 49
Gambar 4.8. Login dan Pembuatan Proyek Blynk 50
Gambar 4.9. Widget Monitoring Energi di Blynk 51
Gambar 4.10. Tangkapan Layar Baris Deklarasi Library di Arduino IDE 54
Gambar 4.11. Tangkapan Layar Baris Pengaturan Waktu 55
Gambar 4.12. Tangkapan Layar Baris Konfigurasi Awal Layar Oled 55
Gambar 4.13. Tangkapan Layar Baris Konfigurasi Pin 56
Gambar 4.14. Tangkapan Layar Baris Pemantauan Energi Listrik 57
Gambar 4.15. Tangkapan Layar Baris atur *Timer*, *Relay*, dan Sensor 58
Gambar 4.16. Tangkapan Layar Baris Notifikasi Daya Threshold 58
Gambar 4.17. Tangkapan Layar Variabel Pembaca Sensor Energi 59
Gambar 4.18. Tangkapan Layar Baris Fungsi Setup() 60
Gambar 4.19. Tangkapan Layar Baris Tampil Logo di Oled 61
Gambar 4.20. Tangkapan Layar Baris Tampil Nama dan NIM di Oled 61
Gambar 4.21. Tangkapan Layar Baris Pesan Peringatan di Layar Oled 63
Gambar 4.22. Tangkapan Layar Baris Mengatur Status Relay 64
Gambar 4.23. Tangkapan Layar Baris Perbarui PZEM 65
Gambar 4.24. Tangkapan Layar Baris Matikan Semua *Relay* 66
Gambar 4.25. Tangkapan Layar Baris Perbarui Tampilan Layar Oled 67
Gambar 4.26. Tangkapan Layar Baris Fungsi BLYNK_WRITE() 68
Gambar 4.27. Tangkapan Layar Baris Kirim Pesan WhatsApp 69
Gambar 4.28. Implementasi Alat 70

Gambar 4.29. Konfigurasi Awal Wi-Fi..... 73
Gambar 4.30. Tampilan Wi-Fi Manager..... 73
Gambar 4.31. Tampilan Halaman Konfigurasi..... 74
Gambar 4.32. Antarmuka Aplikasi Blynk Saat *Relay* Aktif..... 75
Gambar 4.33. Tampilan Oled Saat *Relay* Aktif..... 76
Gambar 4.34. Fitur *Enable Timer*..... 77
Gambar 4.35. Fitur Threshold Power..... 77
Gambar 4.36. Token Sebelum Smart *Relay*..... 81
Gambar 4.37. Token Sesudah Smart *Relay*..... 82

DAFTAR LAMPIRAN

Lampiran 1. Surat Tugas Pembimbing Tugas Akhir
Lampiran 2. Lembar Konsultasi Bimbingan Tugas Akhir
Lampiran 3. Surat Izin Penelitian
Lampiran 4. Surat Keterangan Telah Melakukan Penelitian
Lampiran 5. Lembar Wawancara
Lampiran 6. Surat Tugas Penguji Tugas Akhir
Lampiran 7. Berita Acara Penilaian Tugas Akhir
Lampiran 8. Daftar Hadir Peserta Seminar Proposal TA
Lampiran 9. Lembar Pengujian *Black Box* TA

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Konsumsi energi listrik rumah tangga mengalami peningkatan seiring dengan berkembangnya teknologi serta bertambahnya jumlah perangkat elektronik yang digunakan. Kenaikan ini tidak hanya berdampak pada tingginya biaya tagihan listrik bagi pengguna, tetapi juga memberikan tekanan terhadap ketersediaan sumber daya energi yang terbatas. Di sisi lain, pola penggunaan listrik yang kurang efisien, seperti membiarkan perangkat menyala tanpa pengawasan, turut menyumbang pemborosan energi dan peningkatan emisi karbon yang berdampak negatif bagi lingkungan.

Teknologi Internet of Things (IoT) menawarkan solusi potensial untuk mengatasi tantangan tersebut melalui sistem pengelolaan energi listrik yang lebih cerdas, terintegrasi, dan dapat diakses secara real-time. Dengan memanfaatkan mikrokontroler Esp32, sistem berbasis IoT dapat dirancang untuk memantau serta mengontrol konsumsi listrik secara otomatis maupun manual dari jarak jauh menggunakan aplikasi berbasis internet (Shelemo, 2023). Salah satu inovasi yang dapat diterapkan adalah smart *relay*, yaitu perangkat yang memungkinkan pengguna untuk mengatur pengoperasian perangkat listrik rumah tangga melalui smartphone, sehingga dapat meminimalisir penggunaan listrik secara berlebihan.

Smart *relay* berbasis IoT memiliki kemampuan untuk mengotomatisasi pengaturan waktu operasi perangkat listrik, memberikan laporan konsumsi energi, serta mendeteksi dan memberikan notifikasi apabila terjadi penggunaan listrik yang tidak efisien (Shelemo, 2023). Selain itu, sistem ini juga dapat disesuaikan

dengan kebutuhan pengguna melalui pengaturan threshold daya dan jadwal operasional perangkat, sehingga memberikan kemudahan dalam mengelola konsumsi energi secara proaktif tanpa harus melakukan intervensi manual yang berulang (Ibrahim et al., 2022). Hal ini diharapkan dapat membantu rumah tangga dalam menekan biaya listrik sekaligus berkontribusi terhadap pengurangan dampak negatif penggunaan energi berlebih terhadap lingkungan.

Dalam konteks rumah tangga, penerapan smart relay berbasis IoT menggunakan Esp32 dapat menjadi solusi praktis untuk mendukung gaya hidup hemat energi dan ramah lingkungan. Teknologi ini mempermudah pengguna dalam melakukan pemantauan serta pengendalian konsumsi listrik secara efisien, baik melalui fitur notifikasi maupun pengaturan waktu otomatis yang terintegrasi dalam sistem.

Berdasarkan uraian tersebut, diperlukan penelitian untuk merancang dan mengimplementasikan sistem smart *relay* berbasis IoT yang efektif, efisien, serta andal untuk membantu rumah tangga dalam mengendalikan konsumsi listrik mereka secara cerdas. Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun prototipe smart relay berbasis IoT menggunakan Esp32, yang dilengkapi fitur notifikasi dan pengaturan waktu operasional perangkat, sebagai solusi inovatif dalam mendukung pengelolaan energi listrik rumah tangga secara terintegrasi dan berkelanjutan.

1.2 Perumusan Masalah

Bagaimana merancang dan mengimplementasikan smart *relay* berbasis IoT menggunakan Esp32 untuk memantau dan mengendalikan konsumsi energi listrik rumah tangga, sehingga dapat meningkatkan efisiensi energi dan membantu pengguna dalam mengurangi biaya listrik melalui pengelolaan yang terintegrasi dengan platform berbasis internet?

1.3 Batasan Masalah

Penelitian tugas akhir perlu diberikan batasan masalah, supaya penelitian sesuai dengan yang diharapkan. Beberapa batasan masalah dalam penelitian ini yaitu:

a. Sistem hanya dirancang untuk mengontrol perangkat listrik rumah tangga sederhana, seperti lampu atau perangkat elektronik kecil.
b. Sistem menggunakan Esp32 sebagai mikrokontroler utama untuk konektivitas IoT.
c. Aplikasi pengontrol hanya dirancang untuk platform Android.
d. Pengujian dilakukan dalam skala rumah tangga kecil.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Tujuan Penelitian ini bertujuan untuk merancang dan membangun sistem smart *relay* berbasis IoT dengan Esp32 yang dapat memantau dan mengendalikan konsumsi energi listrik di rumah secara efisien.

1.4.2 Manfaat

a. Bagi Peneliti

Menambah pengetahuan dan keterampilan dalam teknologi IoT serta menyediakan dasar untuk publikasi dan penelitian lebih lanjut.

b. Bagi Pengguna

Memberikan kemudahan dalam mengelola konsumsi energi listrik secara real-time, mengurangi pemborosan dengan pemantauan yang lebih efektif, serta membantu menekan biaya tagihan listrik.

c. Bagi STMIK Palangkaraya

Meningkatkan reputasi dalam pengembangan teknologi inovatif, mendorong kolaborasi industri, dan mendukung proyek penelitian yang aplikatif.

1.5 Sistematika Penulisan

BAB I PENDAHULUAN

Bab ini berisikan latar belakang masalah, rumusan masalah, Batasan masalah, tujuan dan manfaat, serta sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini berisikan tentang tinjauan pustaka yang diambil dari penelitian yang relevan beserta susunan kajian teori yang disesuaikan dengan tema Tugas Akhir.

BAB III METODOLOGI PENELITIAN

Bab ini berisikan tentang tahapan yang dilakukan peneliti dalam mengumpulkan informasi atau data yang dibutuhkan.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisikan tentang hasil dari penelitian dan pembahasan dari hasil yang didapat.

BAB V KESIMPULAN DAN SARAN

Bab ini berisikan tentang kesimpulan dan saran yang diberikan untuk pengembangan selanjutnya.

BAB II
TINJAUAN PUSTAKA

2.1 Kajian Teori

2.1.1 Rancang Bangun

Rancang bangun merupakan proses mengembangkan dan memperbaiki sistem atau aplikasi yang sudah ada ataupun belum ada dengan beberapa komponen yang digunakan dari hasil proses analisa sistem dimana mengharapkan dapat menghasilkan aplikasi yang berkualitas sesuai dengan kebutuhan yang diinginkan (Surahmat, 2023).

Rancang bangun merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh dan berfungsi. Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut atau memperbaiki sistem yang sudah ada (Gunawan, et al. 2021).

Rancang bangun adalah proses merancang dan mengembangkan sebuah sistem atau aplikasi, baik yang baru maupun yang sudah ada, dengan tujuan agar sistem tersebut dapat memenuhi kebutuhan dan harapan penggunanya. Proses ini dimulai dari hasil analisis sistem, kemudian diterjemahkan ke dalam bentuk desain atau kerangka kerja yang jelas. Pada dasarnya, rancang bangun adalah langkah konkret untuk mengubah konsep atau ide menjadi sebuah sistem yang nyata dan berfungsi dengan baik.

2.1.2 *Internet of Things* (IoT)

Ide awal *Internet of Things* pertama kali dimunculkan oleh Kevin Ashton pada tahun 1999 dimana benda-benda di sekitar kita dapat berkomunikasi antara satu sama lain melalui sebuah jaringan seperti internet. Berawal dari *Auto-ID Center*, teknologi yang berbasis pada *Radio Frequency Identification* (RFID) yang merupakan identifikasi kode produk elektronik yang bersifat unik ini kemudian berkembang menjadi teknologi bahwa pada setiap benda dapat memiliki alamat *Internet Protocol* (IP). Dengan semakin berkembangnya infrastruktur internet, maka kita menuju babak berikutnya, di mana bukan hanya *smartphone* atau komputer saja yang dapat terkoneksi dengan internet. Namun berbagai macam benda nyata akan terkoneksi dengan internet. Saat ini milyaran orang saling terhubung melalui jaringan internet, mereka saling bertukar pengetahuan, hiburan, dan lain-lain melintasi kota dan negara bahkan benua (Shelemo, 2023).

Dalam pengertian sederhana, IoT adalah ksonsep dasar yang menghubungkan perangkat satu dengan yang lainnya. Termasuk kulkas, televisi, mesin cuci, lampu, smartphone, mobil, dan masih banyak lagi. Selain peralatan sehari-hari, IoT juga bisa menghubungkan berbagai komponen mesin seperti mesin jet pesawat terbang, bor pertambangan minyak, dan lain-lain (Dwi, et al., 2023).

Seluruh aktivitas dalam IoT dapat dikendalikan oleh suatu perangkat elektronik yang merupakan pengertian sederhana bagaimana IoT bekerja. Pada dasarnya, jika kita memiliki peralatan yang mempunyai saklar *on* dan *off*,

peralatan tersebut mempunyai kesempatan yang besar untuk digunakan dalam ranah IoT (Dwi, et al., 2023).

IoT adalah sebuah konsep dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer. IoT telah berkembang dari konvergensi teknologi nirkabel, *micro-electromechanical systems* (MEMS), dan internet (Dwi, et al., 2023).

IoT adalah konsep di mana benda-benda di sekitar kita, seperti kulkas, lampu, atau mobil, bisa saling terhubung melalui internet tanpa perlu interaksi manusia langsung. Hal ini membuat hidup lebih praktis karena semuanya bisa dikendalikan secara otomatis.

2.1.3 *Prototype*

Prototype adalah suatu bentuk fisik pertama yang mewakili dari objek sebenarnya untuk digunakan sebagai media pembelajaran peserta didik dalam menerapkan pengetahuan atau konsep yang diperolehnya pada benda nyata. *Prototyping* disebut juga desain aplikasi cepat (*rapid application design* atau RAD) karena menyederhanakan dan mempercepat desain system (Handayani & Kurniawan, 2020).

Prototype dapat diterapkan pada pengembangan sistem kecil maupun besar dengan harapan agar proses pengembangan dapat berjalan dengan baik, tertata serta dapat selesai tepat waktu. Keterlibatan pengguna secara ketika *protype* terbentuk akan menguntungkan seluruh pihak yang terlibat, bagi pimpinan,

pengguna sendiri serta pengembang system (Meisak, Hendri and Agustini, 2022) (Meisak, et al., 2022).

Prototype adalah bentuk awal atau model dari suatu sistem yang digunakan untuk menguji konsep fungsi, dan desain sebelum implementasi penuh. *Prototype* memiliki peran penting dalam pengembangan sistem karena memungkinkan proses desain yang lebih cepat dan terstruktur, baik untuk proyek kecil maupun besar. Dengan pendekatan ini, pengguna dapat terlibat langsung dalam pengembangan, sehingga memastikan sistem yang dihasilkan lebih sesuai dengan kebutuhan.

2.1.4 Mikrokontroler Esp32

Esp32 adalah mikrokontroler yang dikenalkan oleh *Espressif System* merupakan penerus dari mikrokontroler Esp8266. Perbedaan yang menjadi keunggulan mikrokontroler Esp32 dibanding dengan mikrokontroler yang lain, mulai dari pin out nya yang lebih banyak, pin analog lebih banyak, memori yang lebih besar, terdapat bluetooth 4.0 *low energy* serta tersedia Wi-Fi yang memungkinkan untuk mengaplikasikan Internet of Things dengan mikokontroler Esp32 (Suriana, et al., 2022).

Esp32 ini dirancang untuk dapat digunakan pada aplikasi seluler, perangkat elektronik dan *Internet of Thing* (IoT). Mikrokontroler ini dapat digunakan pada daya yang rendah dan mengunakan *low-duty cycleyang* dapat meminimalkan energi yang dikeluarkan oleh chip. Esp32 adalah mikrokontroler yang terintegrasi sehingga dapat digunakan pada Wi-Fi dan Bluetooth IoT disertai oleh 20 komponen eksternal (Asmazori, 2021).

Esp32 adalah mikrokontroler yang sangat cocok untuk aplikasi IoT karena memiliki fitur unggulan seperti Wi-Fi, bluetooth low energi, memori yang besar, dan hemat daya. Dibandingkan dengan Esp8266, Esp32 lebih lengkap dan fleksibel, sehingga dapat digunakan untuk berbagai perangkat pintar dengan kebutuhan daya rendah dan konektivitas yang andal.



Gambar 2.1. Mikrokontroler Esp32
Sumber : (Yusri, et al., 2024)

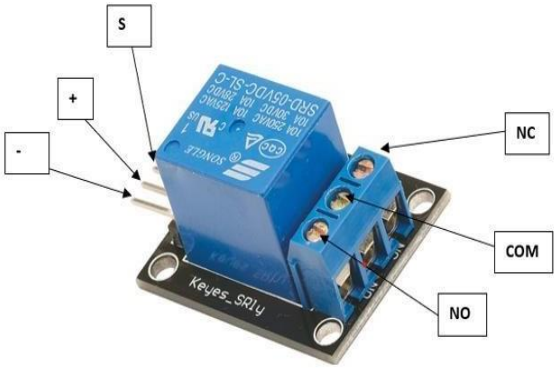
2.1.5 *Relay*

Relay merupakan saklar *(switch)* yang dioperasikan melalui listrik dan merupakan komponen Elektromechanical yang mempunyai dua bagian utama yaitu elektromagnet (koil) dan mekanikal (kontak saklar *switch*). Prinsip yang digunakan *relay* yaitu elektromagnetik yang digunakan untuk menggerakan kontak saklar sehingga dengan arus listrik yang kecil *(low power)* akan menghantarkan listrik yang bertegangan lebih tinggi (Budiyanto, et al., 2020).

Relay bekerja menggunakan prinsip elektromagnetik, di mana kumparan elektromagnetik pada *relay* akan menarik armature saat diberi tegangan. Akibatnya, kontak pada *relay* akan berubah posisi dan menghubungkan atau memutuskan aliran listrik pada perangkat yang dikontrol (Tri et al., 2023).

Relay adalah saklar yang dioperasikan dengan listrik menggunakan prinsip elektromagnetik. Saat arus listrik mengalir ke kumparan elektromagnetiknya, armature akan bergerak untuk mengubah posisi kontak saklar. Ini memungkinkan

*rela*y untuk mengendalikan aliran listrik yang lebih besar hanya dengan menggunakan arus listrik yang kecil. Teknologi ini sangat berguna untuk mengontrol perangkat elektronik dengan aman dan efisien.



Gambar 2.2. *Relay* dan Bagianya

Sumber : (Sutiari, 2024)

2.1.6 Adaptor

Adaptor adalah rangkaian listrik yang mampu menurunkan tegangan dan menyearahkan arus AC menjadi DC. Hal ini dikarenakanadaptor terdiri atas banyak dioda, yakni komponen yang mampu mengubah arus bolak balik menjadi arus searah. Adaptor bisa digunakan sebagai alternatif sumber tegangan DC selain baterai, akumulator, dan catu daya (Sutiari, 2024).

Adaptor adalah sebuah rangkaian elektronika yang dapat mengubah tegangan AC menjadi DC. Rangkaian ini adalah alternatif pengganti dari sumber tegangan DC, misalnya batu baterai dan accumulator. Keuntungan dari adaptor dibanding dengan batu baterai atau accumulator adalah sangat praktis berhubungan dengan ketersediaan tegangan karena adaptor dapat di ambil dari sumber tegangan AC yang ada di rumah, di mana pada jaman sekarang ini setiap rumah sudah menggunakan listrik. Selain itu, adaptor mempunyai jangka waktu

yang tidak terbatas asal ada tegangan AC, tegangan AC ini sudah merupakan kebutuhan primer dalam kehidupan manusia (Zahwa et al., 2022).

Adaptor adalah rangkaian elektronik yang mengubah tegangan AC menjadi DC, sehingga bisa digunakan sebagai alternatif sumber daya selain baterai atau akumulator. Kelebihan adaptor adalah praktis karena menggunakan listrik rumah yang selalu tersedia. Selama ada tegangan AC, adaptor bisa digunakan tanpa batas waktu, sehingga sangat berguna untuk memenuhi kebutuhan perangkat yang membutuhkan arus DC.



Gambar 2.3. Adaptor

Sumber : (Sutarti, et al., 2022)

2.1.7 Step Down DC

Modul step down yaitu komponen penurun tegangan yang berfungsi untuk menurunkan tegangan agar dapat digunakan sesuai dengan kebutuhan. Sistem yang akan dibuat ini menggunakan jenis modul DC LM2596. Modul step down DC LM2596 berguna untuk menurunkan tegangan dari 24VDC menjadi tegangan 5VDC yang berfungsi sebagai tegangan masukan dari arduino (Firdausi & Rohmah, 2023).

Modul step down LM25965 adalah modul yang memiliki IC LM2596 sebagai komponen utamanya. IC LM2596 adalah sirkuit terpadu / integrated circuit yang berfungsi sebagai Step-Down DC converter dengan current rating 3A. Terdapat beberapa varian dari IC seri ini yang dapat dikelompokkan dalam dua

kelompok yaitu versi *adjustable* yang tegangan keluarannya dapat diatur, dan versi *fixed voltage output* yang tegangan keluarannya sudah tetap (Giawa, et al., 2024).

Modul step down, khususnya LM2596, adalah perangkat yang digunakan untuk menurunkan tegangan DC agar sesuai dengan kebutuhan perangkat lain, seperti Arduino. Modul ini menggunakan IC LM2596 sebagai komponen utama, yang mampu menurunkan tegangan dengan arus hingga 3A. Ada dua jenis modul ini, yaitu yang bisa diatur tegangan keluarannya *(adjustable)* dan yang sudah memiliki tegangan tetap *(fixed).* Modul ini sangat praktis dan efisien untuk mengatur tegangan sesuai kebutuhan berbagai perangkat elektronik.



Gambar 2.4. Step Down LM25965

Sumber : (Firdausi & Rohmah, 2023)

2.1.8 Sensor Arus

PZEM-004T adalah sensor yang dapat digunakan untuk mengukur tegangan rms, arus rms dan daya aktif yang dapat dihubungkan melalui arduino ataupun platform opensource lainnya. Dimensi fisik dari papan PZEM-004T adalah 3,1 × 7,4 cm. Modul pzem-004t dibundel dengan kumparan trafo arus diameter 3mm yang dapat digunakan untuk mengukur arus maksimal sebesar 100A. Modul ini terutama digunakan untuk mengukur tegangan AC, arus, daya aktif, frekuensi,

faktor daya dan energi aktif, modul tanpa fungsi tampilan, data dibaca melalui interface TTL. Interface TTL dari modul ini adalah interface pasif, membutuhkan catu daya eksternal 5V, yang berarti ketika berkomunikasi, keempat port harus terhubung (5V, RX, TX, GND) jika tidak ia tidak dapat berkomunikasi (innovatorsguru, n.d.). PZEM-004T-10A: Rentang Pengukuran 10A (Built-in Shunt). PZEM-004T-100A: Rentang Pengukuran 100A *(External Transformer)* (Kiswantono & S, 2024).

PZEM-004T adalah sebuah modul elektronik yang berfungsi untuk mengukur tegangan, arus, daya, frekuensi, energy dan power factor. Dengan kelengkapan fungsi ini, maka modul PZEM-004T sangat ideal untuk digunakan sebagai projek maupun eksperimen alat pengukur daya pada sebuah jaringan listrik seperti rumah atau gedung. Modul PZEM-004T diproduksi oleh sebuah perusahaan bernama Peacefair, ada yang model 10 ampere dan 100 ampere (Ibrahim, et al., 2022).

PZEM-004T adalah modul elektronik buatan Peacefair yang dirancang untuk mengukur berbagai parameter kelistrikan seperti tegangan AC, arus, daya aktif, frekuensi, faktor daya, dan energi aktif. Modul ini tersedia dalam dua varian, yaitu PZEM-004T-10A dengan shunt internal untuk pengukuran arus hingga 10A dan PZEM-004T-100A dengan trafo arus eksternal untuk pengukuran hingga 100A. Dengan dimensi fisik 3,1 × 7,4 cm, modul ini dilengkapi interface TTL untuk membaca data, yang memerlukan koneksi 5V, RX, TX, dan GND agar dapat berfungsi. Modul ini sangat cocok untuk proyek atau eksperimen pengukuran daya pada jaringan listrik seperti rumah atau gedung, namun tidak

memiliki layar tampilan sehingga memerlukan perangkat tambahan untuk memantau data.



Gambar 2.5. Sensor Arus PZEM-004T

Sumber : (Kiswantono, et al., 2024)

2.1.9 OLED *(Organic Light Emitting Diode)*

Layar Oled *(Organic Light Emitting Diode)* adalah dioda pemancar cahaya organik yang terbuat dari bahan semikonduktor organik yang mengisolasi secara elektrik (Oktodinata & Purnomo, 2024).

Oled adalah suatu jenis LCD yang tidak memiliki polarisasi sehingga tampilan yang dihasilkan akan lebih halus dibanding jenis LCD biasa dan dapat menampilkan grafik hasil dari pengukuran secara jelas (Rahmad Timor & Kesuma, 2023).

OLED (Organic Light Emitting Diode) adalah teknologi layar yang menggunakan bahan semikonduktor organik untuk menghasilkan cahaya sendiri tanpa bantuan cahaya latar. Karena tidak menggunakan polarisasi seperti lcd biasa, tampilan oled lebih halus, tajam, dan jelas, sehingga sangat cocok untuk menampilkan hasil pengukuran dalam bentuk grafik secara akurat dan efisien.



Gambar 2.6. OLED *(Organic Light Emitting Diode)*

Sumber : (Oktodinata & Purnomo, 2024)

2.1.10 Kabel Jumper

Kabel jumper adalah kabel elektrik yang memiliki pin konektor di setiap ujungnya dan memungkinkanmu untuk menghubungkan dua komponen yang melibatkan Arduino tanpa memerlukan solder. Intinya kegunaan kabel jumper ini adalah sebagai konduktor listrik untuk menyambungkan rangkaian listrik.Biasanya kabel jumper digunakan pada breadboard atau alat prototyping lainnya agar lebih mudah untuk mengutak-atik rangkaian. Konektor yang ada pada ujung kabel terdiri atas dua jenis yaitu konektor jantan dan konektor betina (Sutarti, et al., 2022).

kabel jumper adalah kabel elektrik yang memiliki pin konektor di setiap ujungnya dan memungkinkanmu untuk menghubungkan dua komponen yang melibatkan arduino tanpa memerlukan solder (Tri et al., 2023).

kabel jumper adalah kabel elektrik yang sangat praktis, karena memiliki konektor di kedua ujungnya. kabel jumper memungkinkan kita untuk menghubungkan berbagai komponen, seperti Arduino, tanpa perlu menyolder. Kabel jumper sangat berguna dalam pengembangan rangkaian, terutama pada breadboard atau alat prototyping lainnya, karena memudahkan kita untuk menyusun dan mengubah rangkaian dengan cepat. Terdapat dua jenis konektor

pada kabel jumper, yaitu konektor jantan dan betina, yang bisa dipilih sesuai kebutuhan.



Gambar 2.7. Kabel Jumper

Sumber : (Sutarti, et al., 2022)

2.1.11 Blynk

Blynk adalah sebuah aplikasi open source yang memiliki API *(Application Programming Interface)* yang dapat digunakan untuk projek IoT (*Internet Of Things)* sehingga memungkinkan pengguna dapat menyimpan, menganalisis, menampilkan visual data serta dapat melakukan aksi atau tindakan atas program yang telah ditentukan. Blynk dapat dihubungkan dengan berbagai macam mikrokontroler seperti Arduino, Esp8266 dan NodeMCU dan lain-lain. Selain itu Blynk juga dapat digunakan pada platform IOS dan Android untuk mengontrol mikrokontroler tersebut selama keduanya terhubung ke internet. Pada aplikasi blynk kita juga data membuat tampilan kontroller dari widget yang telah disediakan di dalamnya. Sehingga dapat menyesuaikan dengan kebutuhan dari projek yang kita buat (Tri et al., 2023).

Blynk adalah platform aplikasi yang dapat diunduh secara gratis untuk iOS dan Android yangberfungsi mengontrol Arduino, Raspberry Pi dan sejenisnya

melalui Internet. Blynk dirancang untuk *Internet of Things* dengan tujuan dapat mengontrol hardware dari jarak jauh,dapat menampilkan data sensor, dapat menyimpan data, visual dan melakukan banyak hal canggih lainnya. Ada tiga komponen utama dalam platform yaitu Blynk App, Blynk Server, dan Blynk Library (Tri Sulistyorini, et al., 2022).

Blynk adalah aplikasi *open-source* untuk mengontrol perangkat IoT seperti Arduino, ESP8266, dan NodeMCU melalui ponsel Android atau iOS. Aplikasi ini memungkinkan pemantauan dan kontrol perangkat dari jarak jauh, menyimpan serta menganalisis data sensor melalui internet. Dengan fitur seperti widget yang dapat disesuaikan, serta komponen utama berupa Blynk App, Server, dan Library, Blynk mempermudah pengembangan proyek IoT secara praktis dan efisien.

2.2 Penelitian Yang Relevan

Seperti topic serupa atau memiliki mendekati persamaan Peneliti mengguakan referensi studi literature yang berasal dari peneliti sebelumnya untuk memudahkan peneliti. Berikut tebel dari beberapa peneliti yang relevan:

Tabel 2.1. Penelitian Yang Relevan

No	Penulis / Tahun	Topik Penelitian	Metode	Pembahasan	Hasil
1.	Afunia Bundha Lasera , Ibnu Hary Wahyudi, / 2020	Pengembangan Prototype Sistem Pengontrolan Daya Listrik berbasis IoT ESP32 pada Smart Home System	Pengembangan *prototype*	Pemasangan instalasi listrik tersebut dikaitkan dengan ketidaksesuaian beban penggunaan perangkat listrik, sehingga terjadi korsleting listrik yang memicu kebakaran. Kebakaran yang terjadi di rumah, salah satunya disebabkan adanya korsleting listrik	Alat ini dapat digunakan untuk menyalakan atau mematikan perangkat elektronik seperti lampu dan kipas angin secara jarak jauh menggunakan aplikasi blynk yang terkoneksi dengan internet melalui modul ESP32, sistem ini dapat menampilkan besarnya arus dan daya peralatan yang digunakan, sistem ini dapat menampilkan suhu alat dan lingkungan sekitar dan sistem ini mampu menampilkan prakiraan biaya rekening listrik yang harus dibayarkan tiap bulannya.

No	Penulis / Tahun	Topik Penelitian	Metode	Pembahasan	Hasil
2.	Krisna Joko Purjianto, Nurchim, Muhammad Nibras Faiq, / 2024	Prototipe Monitoring Energi Listrik Berbasis *Internet of Things* (IoT) guna Mewujudkan Rumah Pintar	SDLC *(System Development Life Cycle)* model *Waterfall*	kurangnya monitoring dan kontrol yang efisien. Banyak rumah tangga belum memiliki sistem yang mampu memantau penggunaan energi secara *real-time,* sehingga sering terjadi pemborosan energi. Selain itu, tidak adanya sistem otomatisasi yang efektif juga menjadi kendala dalam mengoptimalkan penggunaan energi listrik di rumah tangga	Prototipe rumah pintar berbasis IoT untuk monitoring energi listrik pada lampu berhasil diimplementasikan dengan menggunakan komponen ESP32, relay 5V, dan sensor PZEM-004T 100A. Data yang dikumpulkan oleh sensor dikirimkan ke aplikasi web melalui ESP32. Aplikasi web ini menampilkan informasi tegangan, arus, daya, dan konsumsi energi secara *real-time.*
3.	Suprianto, Herri Trisna Frianto, Ahmad Farid Bin Abidin, Mohd	Implementasi Sistem Otomatisasi Kelistrikan	Eksperimen	Penggunaan energi listrik yang tidak digunakan sesuai kebutuhan akan menyebabkan pemborosan energi listrik.	Setiap sensor pir yang berjumlah sembilan sensor yang diletakkan pada tiga ruangan akan mendeteksi orang didalam ruangan tersebut. *Delay* waktu yang dibutuhkan untuk merespon

No	Penulis / Tahun	Topik Penelitian	Metode	Pembahasan	Hasil
	Faizul bin Md Idros, Abdul Hadi Abdul Razak, / 2024	Untuk Penghematan Konsumsi Daya Listrik		mengimplementasikan sistem IoT untuk otomatisasi kelistrikan gedung dan monitoring besaran listrik pada suatu ruangan.	adanya keberadaan orang dalam ruangan dapat diatur melalui program. Pada saat sensor pir mendeteksi keberadaan orang maka seketika itu beban listrik akan aktif namun apabila sebaliknya yaitu pada beberapa waktu setelah itu orang meninggalkan ruangan maka peralatan system juga dapat diatur waktunya untuk menonaktifkan beban listrik.
4.	Rizki Nurul Hidayatullah, Nur Ariesanto Ramdhan, Abdul Khamid, / 2024	Rancang Bangun Instalasi *Smart Home* pada Sistem Proteksi Arus Lebih Menggunakan *Internet of Tings*	metode penelitian empiris eksperimental.	Gangguan arus lebih yaitu suatu permasalahan yang banyak terjadi di sistem rumah pintar.Tripnya rele masukan di sisitemrumah pintar yang diakibatkan oleh adanya kegagalan sistem proteksi atau gangguan yang menyebabkan adanya keterlambatan. Hal ini bisa	hardwareyang berfungsi untuk melakukan pengendalian atau memonitoring alat-alat elektro dalam rumah, selain itu juga berfungsi sebagi proteksi arus lebih, yang di monitoring atau di kendalikan dari jarak jauh melalui *internet of tings.*

No	Penulis / Tahun	Topik Penelitian	Metode	Pembahasan	Hasil
				menyebabkan terjadi trip di sisitem rumah pintar lainyayang tidak terjadi gangguan,tetapi mendapat suplai tegangan sertamasukannya tersebut.Sehingga untuk sistem proteksi di rumah pintar menjadi kurang maksimal serta dapat mempengaruhi keandalan di penyaluran sistem tenaga listrik.	
5.	Ifnu Akbar Rupianto, Rini Puji Astutik, Yoedo Ageng Surya, / 2023	Perancangan Aplikasi *Smart Home* Menggunakan Esp32 Berbasis Android	Pendekatan kuantitatif dengan metode studi	Pada teknologi analog, perangkat listrik seperti lampu atau kipas masih dikendalikan secara manual oleh pengguna melalui saklar yang terhubung dengan kabel. *Smart Home* merupakan	Pembacaan tegangan, arus, dan daya pada Aplikasi Android dapat membaca tegangan, arus, dan daya pada alat melalui komunikasi Bluetooth.

No	Penulis / Tahun	Topik Penelitian	Metode	Pembahasan	Hasil
			literatur	perangkat dengan sistem yang canggih yang memiliki teknologi untuk menngendalikan perangkat elektronik rumah tangga seperti lampu atau kipas melalui Smartphone Android dengan mikrokontroler sebagai basis pemrosesan dari sebuah perangkat *Smart Home.*	

BAB III
METODOLOGI PENELITIAN

3.1 Tinjauan Umum

Tugas akhir dengan judul Rancang Bangun Smart Relay Berbasis IoT Menggunakan Esp32 untuk Pengendalian Konsumsi Energi Listrik Rumah Tangga ini bertujuan untuk merancang dan membangun sebuah sistem kontrol berbasis *Internet of Things* (IoT) menggunakan mikrokontroler Esp32, yang dapat digunakan untuk memantau dan mengendalikan konsumsi energi listrik pada rumah tangga. Penelitian ini dilakukan di Kota Palangka Raya, tepatnya di Jl. Temanggung Tilung 22, Kelurahan Menteng, Kondisi lingkungan yang digunakan dalam penelitian ini adalah rumah tangga dengan sistem kelistrikan standar, di mana penerapan teknologi ini diharapkan mampu memberikan dampak positif dalam penghematan energi dan optimalisasi penggunaan listrik di tingkat rumah tangga.

3.2 Jenis penelitian

Jenis penelitian yang digunakan dalam penelitian ini adalah penelitian dan pengembangan, atau dalam bahasa Inggris disebut *Research and Development* (R&D). Penelitian *Research and Development* (R&D) sebagai acuan dan langkah-langkah penelitian untuk menghasilkan produk baru.

Dengan menggunakan metode penelitian R&D, peneliti akan lebih mudah mengidentifikasi tahapan-tahapan yang diperlukan dalam pengujian *prototype* yang akan dikembangkan. Pengujian ini akan dilakukan di lingkungan rumah

tangga sebagai media untuk mengontrol konsumsi energi listrik, dengan tujuan mengoptimalkan penggunaan energi dan mengurangi pemborosan listrik.

3.3 Desain Penelitian

Desain penelitian yang digunakan dalam penelitian ini oleh penulis adalah metode *prototype*, yang akan diterapkan di lingkungan rumah tangga. Metode *prototype* dipilih karena efektif untuk mengembangkan produk baru dan sistem. Dengan metode ini, penulis dapat menggambarkan dan menguji kinerja sistem serta melihat hasil yang diharapkan. Pendekatan ini memungkinkan penulis untuk mengamati langsung bagaimana sistem bekerja di dunia nyata, sehingga memudahkan perbaikan dan penyempurnaan produk yang dikembangkan.

Metode *prototype* mempunyai beberapa tahapan, tahapan pada metode *prototype* ini bertujuan agar membuat suatu projek yang dimana projek itu dapat berjalan sesuai yang diiginkan dan dilakukannya pemeliharaan sehingga dapat membuat pengguna nyaman.

a. Analisis atau pengumpulan kebutuhan *(Requerements)*

   Tahapan model *prototype* dimulai dari analisis kebutuhan. Dalam tahap ini kebutuhan sistem didefinisikan secara rinci. Dalam prosesnya maka penulis akan menganalisis kebutuhan alat yang akan digunakan pada penelitian. Alat yang digunakan pada penelitian ini adalah, Microkontroller Esp32, Sensor Arus PZEM-400T, Step Down LM2596, Adaptor, Oled, Kabel Jumper dan *Relay* 2 Chanel

   Tahap pengumpulan kebutuhan ini bertujuan untuk menentukan tujuan utama dari pembuatan *prototype* dan mengidentifikasi kebutuhan

dasar sistem yang akan dibuat. Peneliti mengumpulkan data melalui penelitian dan pengamatan langsung di lingkungan rumah tangga. Dalam proses ini, peneliti menganalisis masalah-masalah yang sering muncul akibat borosnya konsumsi listrik di rumah tangga. Hasil dari pengamatan ini digunakan sebagai dasar untuk merancang solusi yang tepat melalui sistem smart *relay* berbasis IoT.

b. Desain Cepat *(Quick Design)*

Pada tahapan ini penulis membuat rancangan atau pembuatan desain sederhana yang mengarah ke pembangunan *prototype* yang akan memberi gambaran tentang projek yang akan dibuat. *Prototype* dievaluasi oleh pengguna dan bagian analisis desain digunakan untuk menyesuaikan kebutuhan perangkat keras yang ingin dibangun. Tools yang digunakan penulis sebagai pembuatan perancangan perangkat keras adalah Fritzing.

c. Bangun *Prototype (Build Prototype)*

Memahami secara jelas dan detail apa yang perlu dilakukan. Tahapan ini merupakan tahapan secara nyata yang diterapkan dalam Merancang Smart Relay Untuk Pengendalian Konsumsi Energi Listrik Rumah Tangga dengan dibangunnya rangkaian *prototype* yang sudah ditentukan oleh penulis. Pada tahap ini penulis melakukan perancangan setelah melalui tahap 1 dan tahap 2 sehingga *prototype* dapat dibangun.

d. Evaluasi Pengguna *(User Evaluation)*

Pada tahapan ini penulis akan melakukan percobaan alat pada listrik lingkungan rumah tangga untuk pengendalian konsumsi energi listriknya,

penulis akan melakukan evaluasi terhadap *prototype* yang telah dibuat apabila terdapat eror tau kerusakan.

e. Memperbaiki *Prototype (Refining Prototype)*

Apabila *prototype* mengalami eror atau kerusakan maka penulis akan melakukan perbaikan terhadap *prototype* tersebut. Setelah melakukan perbaikan maka prosesnya akan kembali ketahap 4, dan apabila sudah tidak ada eror dan kerusakan pada alat maka proses dapat dilanjutkan ketahap berikutnya.

f. Implementasi dan Pemeliharaan *(Implement Product and Maintain)*

Pada tahap akhir ini maka *prototype* sudah sempurna dibuat berdasarkan hasil akhir dari evaluasi pengguna dan melakukan pemeliharaaan secara teratur agar sistem berjalan dengan lancar tanpa kendala.

3.4 Instrumen Penelitian

Teknik pengumpulan data menggunakan informasi yang relevan dan objek penelitian secara langsung. Ada berbagai karakteristik dan pendekatan yang berbeda berdasarkan penelitian, seperti penyesuaian konteks. Cara yang mendukung untuk mendapatkan data penelitian adalah sebagai berikut:

a. Observasi

Proses pengamatan secara langsung dari lingkungan, peristiwa, atau memahami hubungan sebab-akibat pada sistem monitoring dan pengontrol arus listrik. Observasi dilakukan untuk mengamati alat dan tempat agar

tujuan penelitian tercapai sesuai keinginan peneliti, yang berlokasi di Kota Palangka Raya, Jl. Temanggung Tilung 22, Kelurahan Menteng.

b. Studi Dokumentasi

Salah satu metode pengumpulan data yang melibatkan analisis dokumen dan pengumpulan informasi dari berbagai sumber seperti dokumen elektronik, laporan, foto, dan dokumen tertulis yang dapat mendukung proses penulisan.

c. Kepustakaan

Pada metode ini peneliti melakukan pengumpulan data maupuan mempelajari dari jurnal dan beberapa proposal skripsi terdahulu yang terkait atau relevan dengan penelitian yang sedang dilakukan oleh peneliti. Jurnal yang telah dikumpulkan oleh peneliti adalah sebagai acuan dari penelitian sehingga penelitian ini sesuai dengan yang diharapkan.

d. Wawancara

Metode wawancara dilakukan untuk memperoleh informasi langsung dari narasumber terkait objek penelitian. Wawancara dilakukan dengan Ibu Resiana selaku pemilik rumah atau barak yang menjadi lokasi pemasangan alat, yang beralamat di Kota Palangka Raya, Jl. Temanggung Tilung 22, Kelurahan Menteng. Wawancara ini bertujuan untuk mengetahui kebutuhan serta kebiasaan penggunaan alat listrik di lokasi tersebut, guna menyesuaikan rancangan sistem agar lebih efektif dan sesuai kebutuhan pengguna.

3.5 Analisis Kebutuhan

Analisis kebutuhan sistem dilakukan untuk menentukan spesifikasi perangkat keras *(hardware)* dan perangkat lunak *(software)* yang diperlukan dalam merancang dan membangun smart *relay* berbasis IoT untuk mengendalikan konsumsi energi listrik di rumah tangga.

3.5.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras *(hardware)* yang diperlukan untuk merancang *smart relay* dalam pengendalian konsumsi energi listrik rumah tangga adalah sebagai berikut:

a. Mikrokontroler Esp32

   Mikrokontroler Esp32 digunakan sebagai otak sistem yang memiliki kemampuan pemrosesan data dan koneksi jaringan nirkabel (Wi-Fi). Mikrokontroler ini akan mengelola komunikasi antara perangkat keras dan perangkat lunak, serta memproses data dari sensor arus.

b. *Relay*

   Relay digunakan sebagai saklar elektronik untuk mengontrol aliran listrik ke perangkat rumah tangga. *Relay* dapat diaktifkan atau dinonaktifkan secara otomatis berdasarkan perintah dari mikrokontroler.

c. Adaptor

   Adaptor digunakan untuk menyediakan sumber daya listrik ke sistem. Adaptor memastikan bahwa semua komponen perangkat keras mendapatkan suplai daya yang sesuai.

d. Step Down DC

Komponen ini digunakan untuk menurunkan tegangan DC agar sesuai dengan kebutuhan daya dari mikrokontroler Esp32 dan komponen lain dalam sistem.

e. Sensor Arus

Sensor arus berfungsi untuk mengukur konsumsi energi listrik rumah tangga secara real-time. Data dari sensor ini akan dikirimkan ke mikrokontroler untuk dianalisis dan diproses lebih lanjut.

f. Oled berfungsi untuk menampilkan informasi tegangan arus listrik secara lokal dan menampilkannya pada layar, Oled menerima data dari mikrokontroler Esp32.

3.5.2 Kebutuhan Perangkat Lunak

Adapun kebutuhan perangkat lunak *(Software)* agar *prototype* bekerja dengan baik adalah sebagai berikut:

a. Sistem Operasi Microsoft Windows 10

Sistem operasi ini digunakan sebagai platform utama untuk pengembangan perangkat lunak, seperti instalasi Arduino IDE dan aplikasi terkait lainnya.

b. Blynk

Blynk merupakan aplikasi berbasis IoT yang digunakan untuk memonitor dan mengontrol perangkat melalui *smartphone*. Aplikasi ini

memungkinkan pengguna untuk mengelola *relay* dan memantau konsumsi energi listrik secara jarak jauh.

c. Arduino IDE

Arduino IDE digunakan untuk menulis, mengunggah, dan menguji program pada mikrokontroler Esp32.

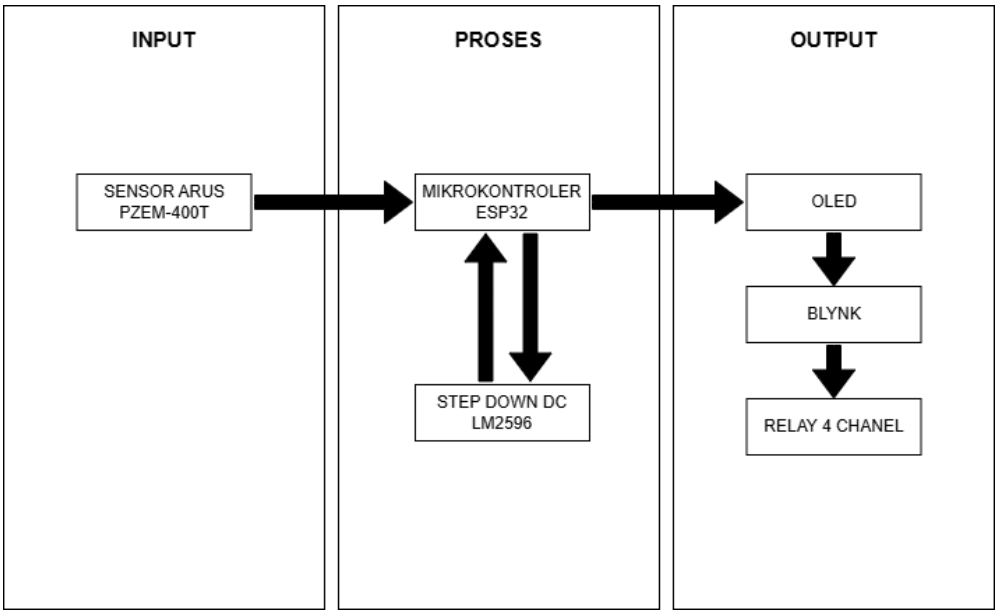
3.5.3 Analisis Data

Penulis menggunakan teknik analisis data statistik untuk membandingkan tingkat keakuratan dan kestabilan pengukuran arus listrik yang diterapkan dalam sistem pengendalian konsumsi listrik. Sistem ini melibatkan penggunaan data logger, sensor arus, serta tampilan nilai arus yang terhubung dengan aplikasi Blynk pada *smartphone*. Pengujian dilakukan dengan tujuan mengevaluasi kinerja keseluruhan sistem agar dapat beroperasi secara optimal selama proses pengamatan.

3.5.4 Analisis Proses

Tahapan Analisis Proses kinerja sistem Smart *Relay* dimulai dengan sensor arus yang mengukur konsumsi listrik perangkat rumah tangga dan mengirimkan data ke mikrokontroler Esp32. Mikrokontroler ini memproses data arus dan menampilkan informasi konsumsi energi pada layar oled secara real-time. Pengguna dapat memantau konsumsi energi dan, berdasarkan informasi tersebut, mengontrol perangkat secara manual melalui aplikasi Blynk yang terhubung dengan sistem. Melalui blynk, pengguna dapat menghidupkan atau mematikan perangkat dengan mengirimkan perintah ke Esp32, yang kemudian mengaktifkan atau memutuskan aliran listrik ke perangkat melalui *relay*. Step down digunakan

untuk menurunkan tegangan agar mikrokontroler dan komponen lainnya beroperasi dengan aman. Sesuai pada Gambar diagram blok 3.1. Dibawah ini.



Gambar 3.1. Diagram Blok

3.5.5 Analisis Kelemahan

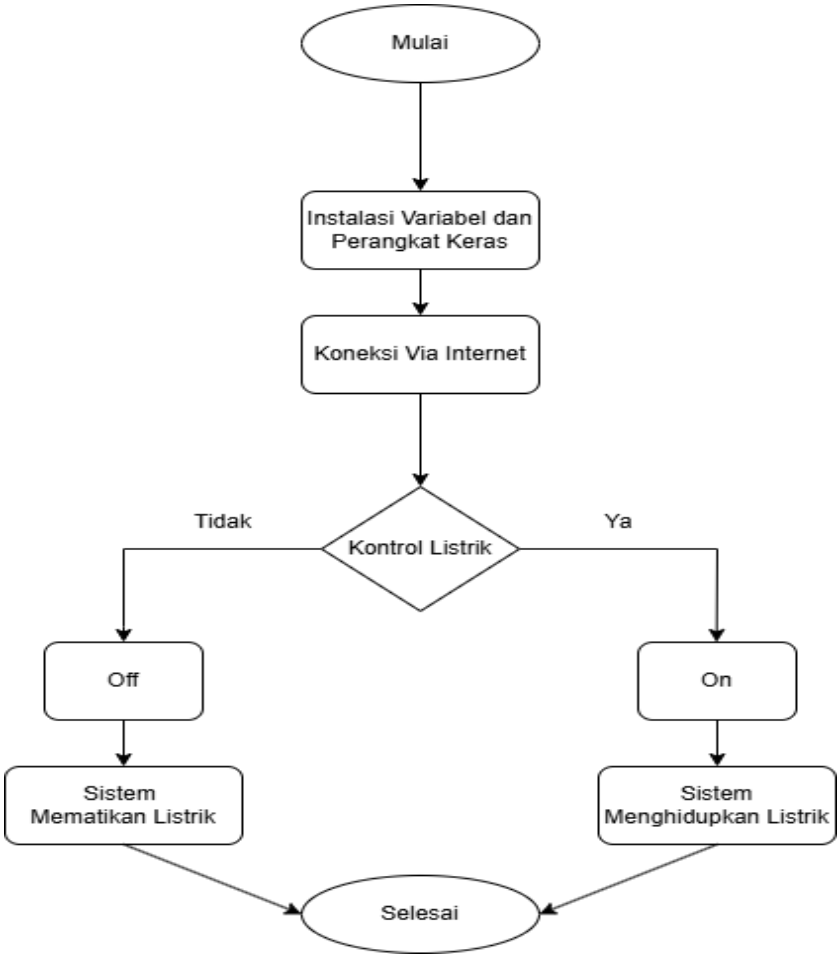
Analisis Kelemahan menurut peneliti yaitu, terletak pada ketergantungannya pada koneksi internet. Jika jaringan internet terganggu, seperti sinyal yang lemah atau terputus, sistem ini tidak dapat berfungsi dengan baik. Hal ini membuat pengendalian perangkat listrik menjadi terhambat dan kurang andal, terutama di area dengan jaringan yang tidak stabil.

3.6 Analisis Desain

3.6.1 Desain Proses

Pada analisis desain ini proses dimulai dengan instalasi variabel dan perangkat keras, diikuti dengan koneksi ke internet melalui Esp32 untuk memungkinkan komunikasi dengan platform IoT. Selanjutnya, sistem melakukan

kontrol listrik dengan memutus *(Off)* atau menyambungkan *(On)* aliran listrik melalui *relay* berdasarkan perintah pengguna. Jika sistem mematikan atau menghidupkan listrik, proses selesai dan kembali siap untuk memantau serta menerima perintah selanjutnya. Rancangan ini memungkinkan pengguna mengontrol dan memantau konsumsi listrik secara efisien dan *real-time* melalui koneksi internet. *Flowchart* pada Gambar 3.2. Menggambarkan proses kerja sistem *smart relay* berbasis IoT menggunakan Esp32 untuk mengendalikan konsumsi energi listrik rumah tangga.

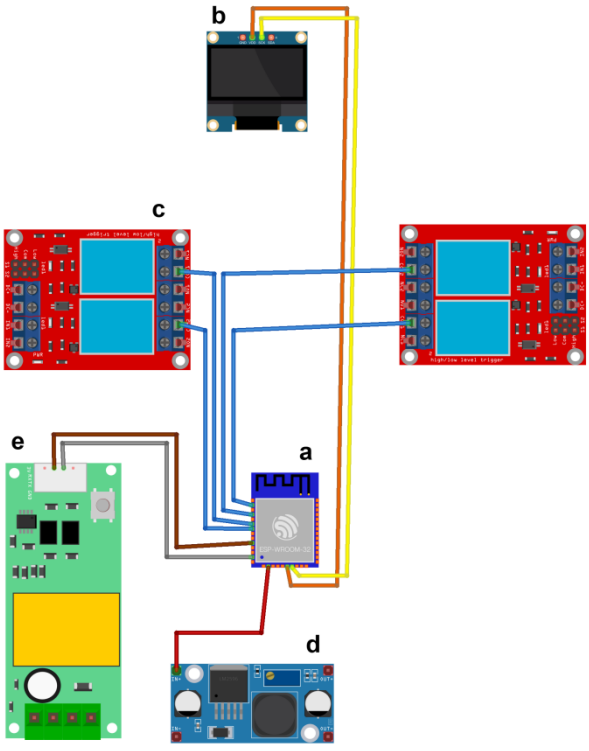


Gambar 3.2. Desain *Flowchart*

3.6.2 Desain Perangkat

1. Perangkat Keras

Pada desain perangkat keras ini menjelaskan struktur perangkat keras yang digunakan penulis pada saat penelitian, mikrokontroler Esp32 berukuran kecil namun fleksibel, perangkat keras tersebut disusun dan dihubungkan sehingga penulis mendapat gambaran desain Perangkat keras tersebut terhubung dan akan mejadi suatu *prototype* yang dapat menjadi salah satu sistem monitoring kualitas air. Berikut bagian perangkat keras yang digunakan penulis, sesuai pada gambar 3.3. Dibawah ini.



Gambar 3.3. Desain Perangkat Keras

Pada rancangan perangkat keras pada diatas terdapat beberapa alat yang saling terhubung satu sama lain, sebagai berikut:

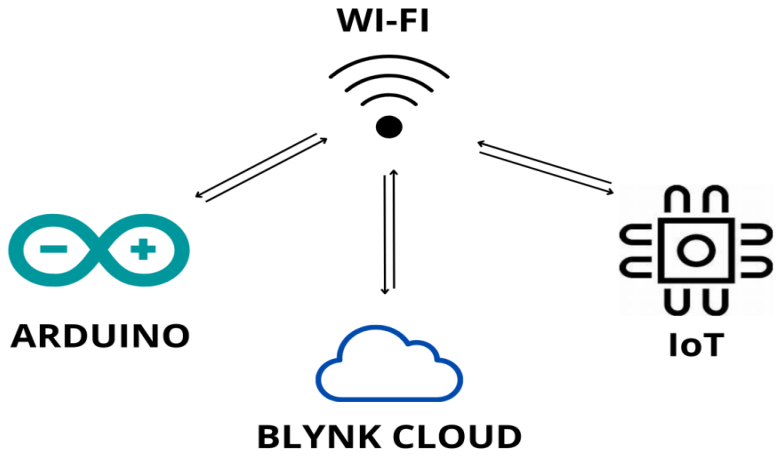
a. Microkontroller Esp32
b. Oled 12 C1
c. *Relay* 2 Chanel dua buah
d. Step Down DC LM2596
e. Sensor Arus PZEM-004T

2. Perangkat Lunak

Dalam desain perangkat lunak ini, penulis menjelaskan tentang penggunaan sistem pengendalian konsumsi energi listrik berbasis IoT di lingkungan rumah tangga.

a. Desain *interface*

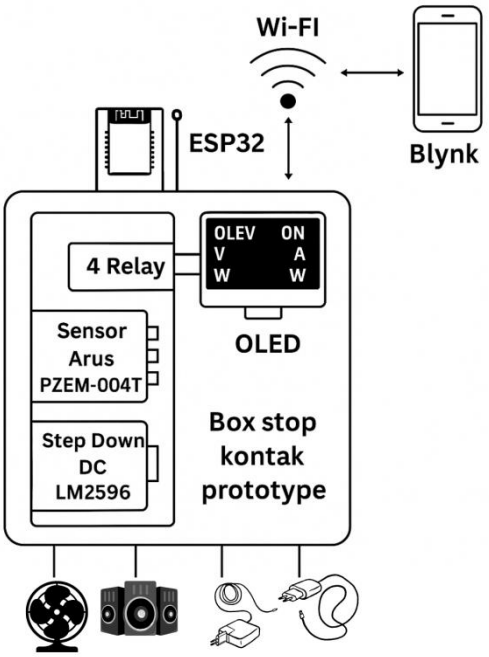
Pada desain *interface* menjelaskan alur kerja sistem IoT yang terdiri dari Arduino, modul Wi-Fi, Blynk Cloud, dan perangkat IoT. Arduino bertindak sebagai pengendali utama yang membaca data dari sensor atau mengontrol aktuator, dengan modul Wi-Fi sebagai penghubung untuk mengirim dan menerima data melalui internet. Data tersebut dikirim ke Blynk Cloud, platform berbasis cloud yang memungkinkan pengguna memantau dan mengontrol perangkat secara *real-time* melalui aplikasi blynk. Pengguna dapat memberikan perintah melalui aplikasi yang diteruskan ke Arduino melalui Wi-Fi, sehingga perangkat dapat beroperasi sesuai kebutuhan. Berikut desain *interface* disesuaikan pada Gambar 3.4 di bawah ini.



Gambar 3.4. Desain *Interface*

b. Desain *interface* perangkat

Desain interface perangkat keras pada sistem ini bekerja dengan mengintegrasikan beberapa komponen utama yang disusun dalam satu box stop kontak *prototype*. Mikrokontroler Esp32 berfungsi sebagai pusat kendali yang menerima dan mengirimkan data melalui jaringan Wi-Fi ke aplikasi Blynk pada *smartphone*. Esp32 terhubung dengan sensor arus PZEM-004T untuk membaca parameter kelistrikan dari perangkat rumah tangga yang dikendalikan. Data ini ditampilkan secara lokal melalui layar oled, yang juga menampilkan status dari *relay* 1 hingga *relay* 4. Keempat *relay* ini terhubung ke perangkat listrik seperti kipas angin, speaker GMC, charger laptop, dan charger HP, yang dapat dikontrol langsung melalui aplikasi Blynk. Berikut desain *interface* perangkat disesuaikan pada Gambar 3.5 di bawah ini.

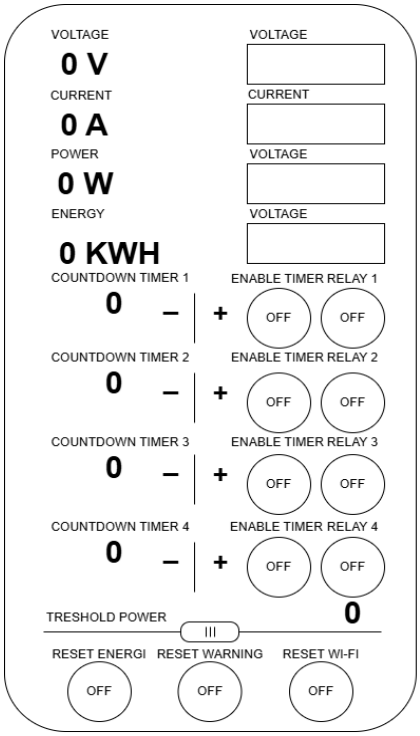


Gambar 3.5. Desain *Interface* Perangkat

c. Desain antarmuka pengendalian perangkat

Desain antarmuka pengendalian perangkat digunakan untuk memantau dan mengendalikan konsumsi energi listrik rumah tangga dengan menampilkan berbagai parameter listrik, termasuk Voltage (tegangan listrik dalam volt/V) untuk menunjukkan besaran tegangan yang digunakan, Current (arus listrik dalam ampere/A) untuk mengukur jumlah arus yang mengalir, Power (daya listrik dalam watt/W) sebagai indikasi besarnya daya yang dikonsumsi oleh perangkat yang terhubung, dan Energi (energi dalam kilowatt-hour/KWH) untuk mencatat total energi yang telah digunakan dari waktu ke waktu, serta dilengkapi dengan fitur Countdown Timer yang memungkinkan pengguna mengatur waktu hitung mundur untuk menyalakan atau mematikan perangkat secara otomatis sesuai dengan kebutuhan, *Enable Timer Relay* 1 sampai 4 yang terdiri dari tombol

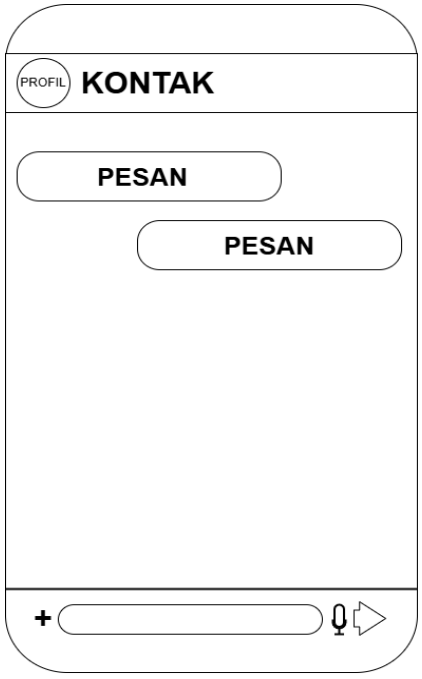
ON/OFF untuk mengaktifkan atau menonaktifkan pengendalian timer pada masing-masing *relay*, *Relay* 1 sampai 4 sebagai saklar elektronik yang dikendalikan secara manual untuk mengontrol aliran listrik ke perangkat yang terhubung, serta Threshold Power berupa pengaturan batas daya maksimum yang diperbolehkan agar sistem dapat melakukan tindakan pencegahan jika konsumsi listrik melebihi batas yang ditentukan, dengan tambahan tiga tombol reset, yaitu Reset Energi untuk menghapus data konsumsi energi yang telah tercatat, Reset Warning untuk menghapus notifikasi peringatan sistem jika terjadi kesalahan, dan Reset Wi-Fi untuk mengatur ulang koneksi jaringan jika terjadi gangguan komunikasi, disesuaikan berdasarkan Gambar 3.6. Di bawah ini.



Gambar 3.6. Desain Antarmuka Pengendalian Perangkat

d. Desain antarmuka notifikasi

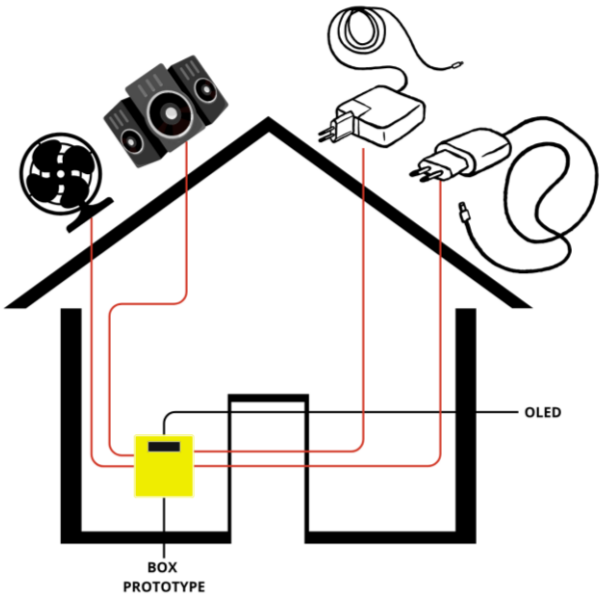
Desain antarmuka notifikasi merupakan bagian dari sistem pemantauan konsumsi daya yang berfungsi untuk memberikan peringatan apabila konsumsi listrik melampaui ambang batas yang telah ditetapkan dalam Threshold Power. Notifikasi ini dapat dikirim secara otomatis melalui aplikasi pesan WhatsApp. sehingga memungkinkan pengguna memperoleh informasi secara *real-time* dan segera mengambil tindakan *preventif*, seperti menonaktifkan perangkat atau menyesuaikan pengaturan daya, guna menghindari risiko kelebihan beban listrik dan meningkatkan efisiensi penggunaan energy disesuaikan berdasarkan Gambar 3.7. Di bawah ini.



Gambar 3.7. Desain Antarmuka Notifikasi

3.6.3 Desain *Prototype*

Pada desain *prototype* ini harus di sesuaikan dengan denah rumah yang akan digunakan, pada penelitian ini penulis menggunakan microkontroller Esp32 sebagai perangkat utama untuk mengumpulkan data dari sensor arus, Tujuan dibuatnya desain *prototype* ini agar mudah dipahami. Pemasangan atau peletakan *prototype* berada di dalam box untuk melindungi komponen, meningkatkan keamanan, merapikan kabel, mempermudah pemeliharaan, dan membuat perangkat lebih rapi. Seperti yang dijelaskan pada Gambar 3.8. Dibawah ini.



Gambar 3.8. Desain *Prototype*

Dari gambar desain *prototype* 3.8. Di atas atas berikut adalah perangkat-perangkat yang dikontrol dalam *prototype*:

1. Perangkat Yang Terhubung
    a. Kipas Angin Miyako
        - Jumlah perangkat: 1 unit

- Daya maksimum: 45 watt

b. Speaker Bluetooth GMC 888K

- Jumlah perangkat: 1 unit
- Daya maksimum: 60 watt

c. Charger Laptop *Acer*

- Jumlah perangkat: 1 unit
- Daya maksimum: 45 watt

d. Charger Hp Realme

- Jumlah perangkat: 1 unit
- Daya maksimum: 45 watt

2. Analisis Kebutuhan Daya

- Total daya maksimum seluruh perangkat: 45 W+60 W+45 W+45 W= 195 Watt.

3.7 Skenario Kendali Perangkat

Skenario uji coba pengendalian listrik rumah tangga dilakukan dengan menghubungkan empat perangkat listrik, yaitu kipas angin, speaker GMC, mejikom, dan setrika ke masing-masing *relay* (R1–R4) yang dikendalikan oleh Esp32 dalam box *prototype*. Pengujian dimulai dengan mengaktifkan satu per satu tombol kendali di aplikasi Blynk, untuk menyalakan dan mematikan setiap perangkat secara jarak jauh melalui koneksi Wi-Fi. Selama proses pengendalian, sensor arus PZEM-004T membaca nilai tegangan, arus, daya, dan energi yang digunakan oleh masing-masing beban, dan hasil pembacaan ditampilkan secara

real-time di layar oled serta pada aplikasi. Uji coba dianggap berhasil jika setiap perangkat merespons perintah *ON/OFF* dari aplikasi dengan cepat, dan nilai parameter kelistrikan yang muncul sesuai dengan karakteristik daya perangkat. Selain itu, dilakukan juga pengujian saat beban listrik tinggi untuk memastikan sistem dapat memberikan notifikasi peringatan jika konsumsi daya melebihi ambang batas (threshold power), sebagai bentuk proteksi dini terhadap risiko kelebihan beban listrik.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

4.1.1 Implementasi

4.1.2 Implementasi Perangkat Keras *(Hardware)*

Pada tahap ini, penulis mengimplementasikan hasil desain ke dalam perangkat keras berupa *prototype* yang telah dirancang, kemudian melakukan pengujian untuk mengevaluasi apakah sistem telah berfungsi sesuai dengan yang diharapkan dan mampu memenuhi tujuan perancangan.

Implementasi sistem monitoring konsumsi energi listrik rumah tangga ini menggunakan mikrokontroler Esp32 sebagai pusat kendali utama dan aplikasi Blynk sebagai media komunikasi dengan perangkat *smartphone*. Sistem ini memanfaatkan sensor arus, modul step down dc, dan *relay* untuk memantau konsumsi energi listrik, di mana data digital yang dikirimkan oleh sensor diproses oleh mikrokontroler dan ditampilkan pada layar oled. Selain itu, sistem dilengkapi fitur notifikasi otomatis yang memberikan peringatan saat konsumsi daya melebihi ambang batas yang ditentukan (Threshold Power), sehingga memungkinkan pengguna memperoleh informasi secara *real-time* dan mengambil tindakan *preventif* guna mencegah kelebihan beban listrik.

Rangkaian sistem monitoring konsumsi energi listrik terdiri dari beberapa komponen yang dirancang secara terintegrasi, antara lain mikrokontroler Esp32, sensor arus, modul step down dc, *relay*, dan oled, yang dihubungkan

menggunakan kabel jumper. Dalam implementasinya, seluruh perangkat keras tersebut disatukan dalam sebuah box stop kontak dan dihubungkan dengan aliran listrik, sehingga memungkinkan sistem untuk mengontrol konsumsi energi listrik melalui komponen-komponen yang saling terintegrasi dalam *prototype*.

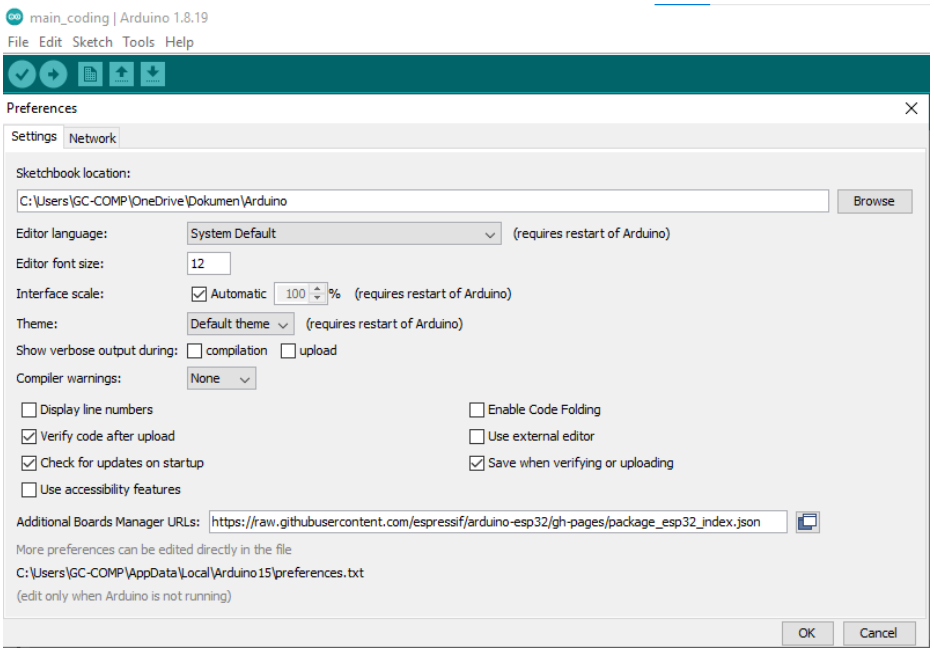


Gambar 4.1. Rangkaian Sistem Monitoring Energi Listrik

4.1.3 Implementasi Perangkat Lunak *(Software)*

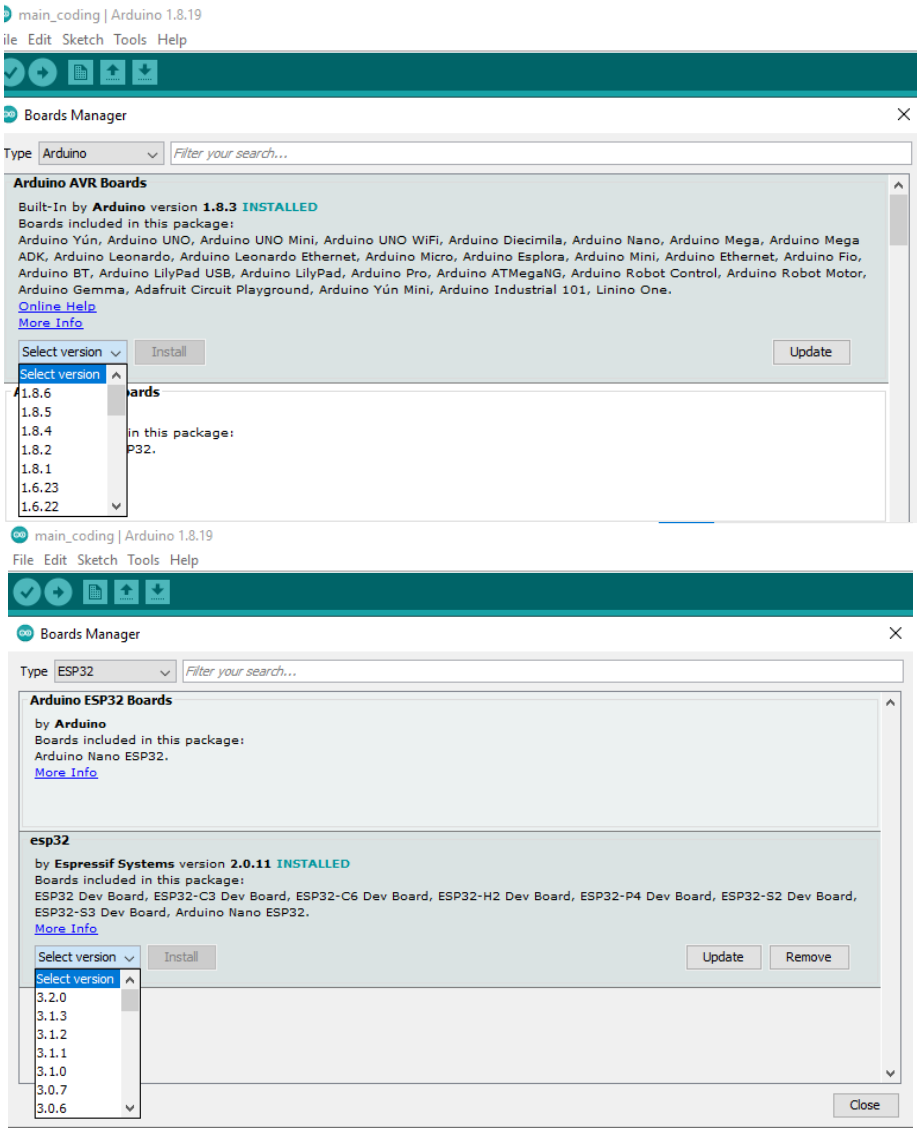
a. Konfigurasi Aplikasi Arduino IDE

Langkah pertama dalam instalasi mikrokontroler Esp32 adalah membuka aplikasi Arduino IDE, kemudian masuk ke menu Preferences melalui jalur File > Preferences. Pada bagian bawah jendela Preferences, terdapat kolom Additional Board Manager URLs tempat pengguna menambahkan URL berikut: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json berikut ini merupakan tampilan gambar menu Preferences pada aplikasi Arduino IDE sebagai referensi dalam proses instalasi.



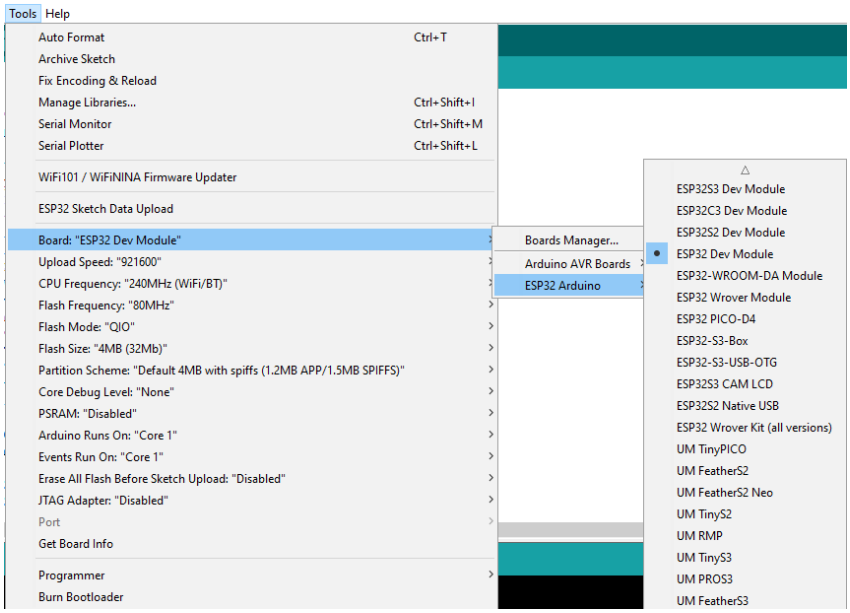
Gambar 4.2. Tampilan Board Manager

Setelah menambahkan URL pada kolom *Additional Board Manager URLs*, langkah selanjutnya adalah membuka *Board Manager* melalui menu Tools > Board > Board Manager. Pada kolom pencarian, ketik "Arduino", lalu setelah muncul Arduino AVR Boards, pilih versi 1.8.3 dan lakukan instalasi. Setelah itu, ketik "ESP32" pada kolom pencarian, kemudian pilih versi 2.0.11 dari paket board ESP32 dan lanjutkan dengan proses instalasi dilihat pada gambar 4.3. berikut:



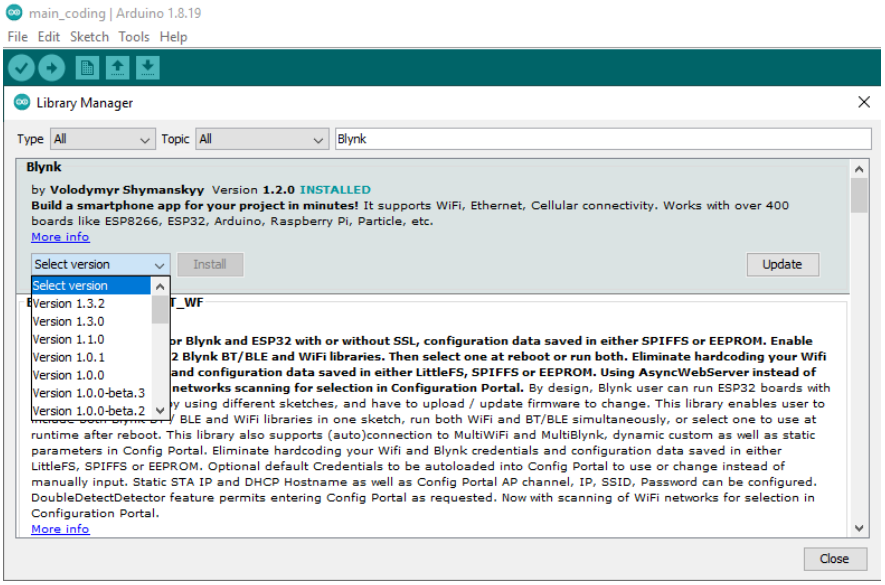
Gambar 4.3. Tampilan Instalasi Board Arduino dan Esp32

Langkah kedua adalah melakukan pemilihan jenis board mikrokontroler ESP32 yang akan digunakan. Pemilihan board dilakukan melalui menu Tools > Board > ESP32 Arduino > ESP32 Dev Module, seperti yang dapat dilihat pada gambar 4.4. berikut:



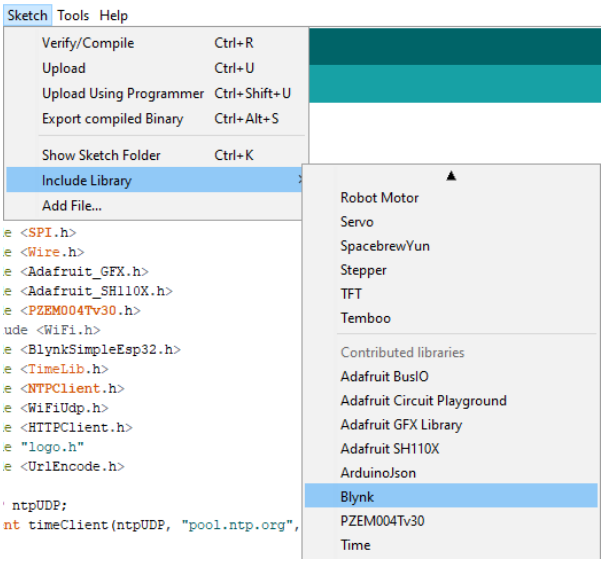
Gambar 4.4. Tampilan Pemilihan Board Esp32

Langkah ketiga adalah menginstal Blynk di Arduino IDE agar mikrokontroler ESP32 dapat berkomunikasi dengan aplikasi Blynk. Untuk memprogram ESP32, digunakan Blynk Arduino Sketch. Blynk dapat diunduh melalui menu Sketch > Include Library > Manage Libraries, kemudian cari Blynk pada kolom pencarian di bagian atas dan klik Install. Pastikan untuk menginstal Blynk versi 1.2.0 atau versi terbaru yang tersedia. Berikut ini adalah tampilan menu Blynk pada aplikasi Arduino IDE, sebagaimana dapat dilihat pada gambar 4.5. berikut:



Gambar 4.5. Istallasi Blynk

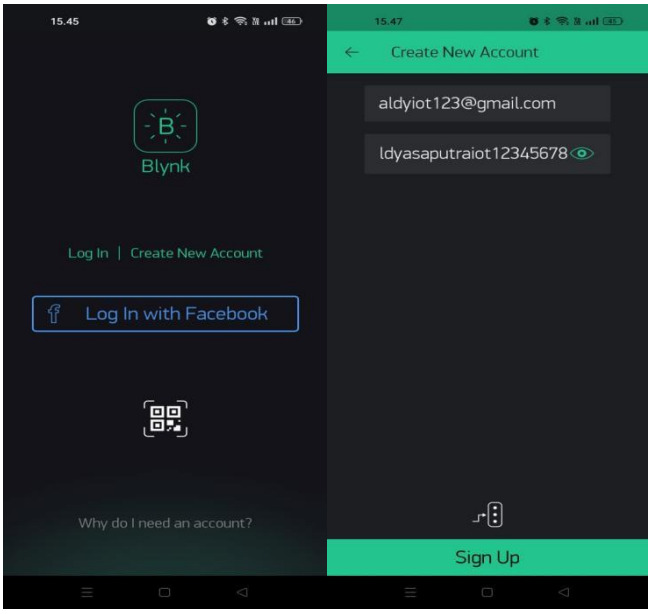
Apabila Blynk sudah diinstal, kembali ke menu Sketch > Include Library, maka Blynk akan sudah tersedia pada daftar pustaka yang dapat dipilih. Berikut adalah tampilan gambar untuk menambahkan Blynk pada aplikasi Arduino IDE, sebagaimana dapat dilihat pada gambar 4.6. berikut:



Gambar 4.6. Instal Blynk di Arduino IDE

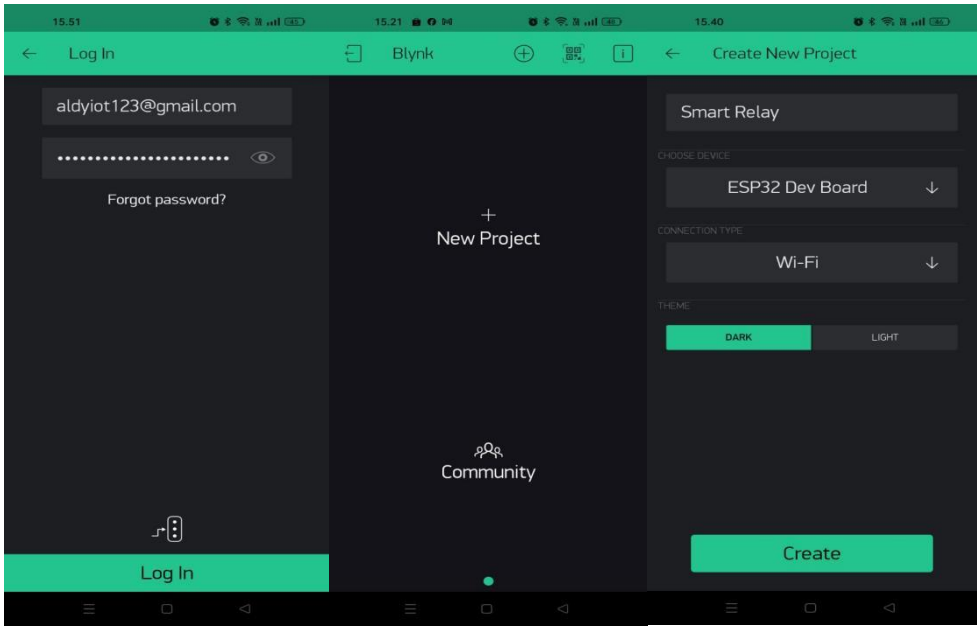
b. Konfigurasi Aplikasi Blynk

Setelah selesai melakukan installasi aplikasi Blynk, langkah berikutnya adalah mendaftar atau membuat akun dengan melakukan registrasi menggunakan akun Gmail. Setelah berhasil membuat akun dan melakukan verifikasi email, langkah selanjutnya adalah login pada aplikasi Blynk.



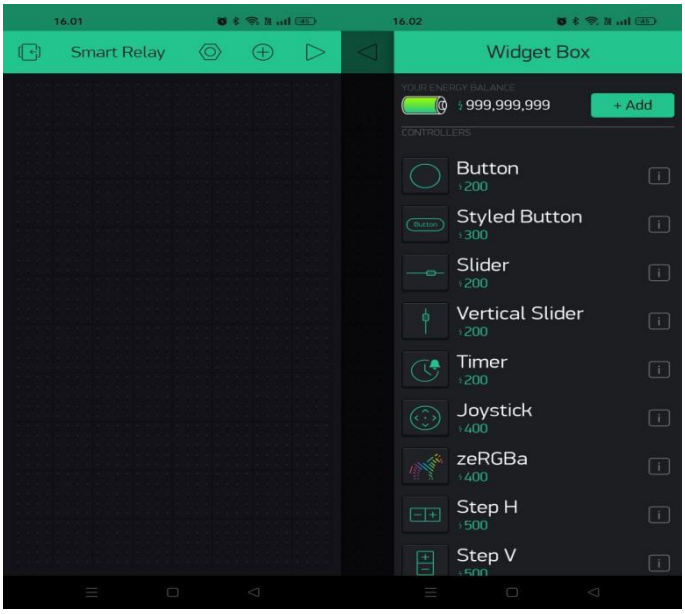
Gambar 4.7. Tampilan Registrasi Blynk

Pada tampilan gambar aplikasi Blynk di atas, langkah pertama yang dilakukan adalah memilih menu Log In. Setelah berhasil masuk, pilih opsi New Project, kemudian isi Project Name dengan *Smart Relay*. Selanjutnya, pada bagian Choose Device, pilih ESP32 Dev Board, dan untuk Connection Type, pilih Wi-Fi. Terakhir, klik tombol Create untuk membuat proyek baru.



Gambar 4.8. Login dan Pembuatan Proyek Blynk

Pada tampilan Quickstart Template pada aplikasi Blynk, ditambahkan berbagai widget seperti Button, Slider, Step H, Value Display, Labeled Value, dan Level H untuk memonitor konsumsi energi listrik. Tujuan penggunaan widget ini adalah untuk menyediakan antarmuka yang interaktif dan memudahkan pengguna dalam memantau serta mengontrol konsumsi energi listrik di rumah tangga secara efisien melalui aplikasi Blynk, sehingga pengguna dapat dengan mudah mengatur dan melihat data penggunaan energi secara langsung.



Gambar 4.9. Widget Monitoring Energi di Blynk

Langkah ketiga adalah melakukan pengaturan pada widget sesuai dengan program, khususnya penyesuaian terhadap penggunaan virtual pin yang telah ditentukan. Setiap widget dihubungkan dengan virtual pin masing-masing untuk memastikan data dapat diterima dan dikirimkan dengan benar melalui aplikasi Blynk. Berikut merupakan tampilan antarmuka pengguna (user interface) yang telah dibuat.

Pada tampilan antarmuka yang telah dibuat, setiap widget memiliki fungsi masing-masing untuk membantu proses monitoring dan pengendalian konsumsi energi listrik. Untuk memudahkan pemahaman, berikut ini adalah penjelasan dari fungsi setiap widget yang digunakan dalam system dapat dilihat pada tabel dibawah ini:

Tabel 4.1. Widget Blynk

No.	Nama Widget	Fungsi/Deskripsi	Pin Virtual
1	Button	Kontrol *Relay* 1	V4
2	Button	Kontrol *Relay* 2	V8
3	Button	Kontrol *Relay* 3	V16
4	Button	Kontrol *Relay* 4	V20
5	Button	Reset Energi	V22
6	Button	Reset Sistem	V5
7	Button	Reset WiFi	V21
8	Slider	Pengaturan Threshold Power	V1
9	Step H	Countdown Timer 1	V2
10	Step H	Countdown Timer 2	V6
11	Step H	Countdown Timer 3	V14
12	Step H	Countdown Timer 4	V18
13	Value Display	Menampilkan Counter Timer 1	V2
14	Value Display	Menampilkan Counter Timer 2	V6
15	Value Display	Menampilkan Counter Timer 3	V14
16	Value Display	Menampilkan Counter Timer 4	V18
17	Labeled Value	Menampilkan Tegangan (Voltage)	V10
18	Labeled Value	Menampilkan Arus (Current)	V11
19	Labeled Value	Menampilkan Daya (Power)	V12

No.	Nama Widget	Fungsi/Deskripsi	Pin Virtual
20	Labeled Value	Menampilkan Energi (Energy)	V13
21	Level H	Visualisasi Tegangan (Voltage)	V10
22	Level H	Visualisasi Arus (Current)	V11
23	Level H	Visualisasi Daya (Power)	V12
24	Level H	Visualisasi Energi (Energy)	V13

c. Implementasi Program Arduino IDE

Setelah melakukan konfigurasi pada aplikasi Arduino IDE dan aplikasi Blynk, langkah selanjutnya adalah mengimplementasikan program yang telah disiapkan. Program tersebut diprogramkan menggunakan software Arduino IDE dengan tujuan untuk menemukan kesalahan atau kekurangan, sehingga dapat diperoleh hasil yang sesuai dengan yang diharapkan.

Pada proses ini, penulis akan membahas source code program pada Arduino IDE yang digunakan untuk membangun prototipe sistem monitoring konsumsi listrik pada rumah tangga. Adapun proses pemrogramannya adalah sebagai berikut:

Baris pertama pada source code ini berisi deklarasi library yang digunakan dalam program. Library FS.h dan SPIFFS.h digunakan untuk mengelola sistem file pada ESP32. WiFiManager.h dan file custom wifimanagerr.h digunakan untuk konfigurasi koneksi WiFi secara dinamis. ArduinoJson.h digunakan untuk parsing dan pembuatan data JSON. #define BLYNK_PRINT Serial mendefinisikan agar

proses debug Blynk dicetak melalui Serial Monitor. Library SPI.h, Wire.h, Adafruit_GFX.h, dan Adafruit_SH110X.h dipakai untuk komunikasi SPI/I2C serta pengelolaan tampilan pada layar oled berbasis SH110X. PZEM004Tv30.h digunakan untuk membaca data dari sensor energi listrik PZEM-004T v3.0. BlynkSimpleEsp32.h menghubungkan Esp32 dengan platform IoT Blynk. TimeLib.h, NTPClient.h, dan WiFiUdp.h dipakai untuk pengaturan waktu menggunakan server NTP. HTTPClient.h digunakan untuk melakukan request HTTP, logo.h untuk menampilkan gambar pada oled, dan UrlEncode.h untuk pengkodean URL dalam komunikasi data. Semua library ini menjadi dasar utama dalam membangun sistem untuk memantau konsumsi listrik.



```
#include <FS.h>
#include <WiFiManager.h>
#include <ArduinoJson.h>
#include <SPIFFS.h>
#include "wifimanagerr.h"
#define BLYNK_PRINT Serial
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SH110X.h>
#include <PZEM004Tv30.h>
//#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <TimeLib.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <HTTPClient.h>
#include "logo.h"
#include <UrlEncode.h>
```

Gambar 4.10. Tangkapan Layar Baris Deklarasi Library di Arduino IDE

Bagian ini berisi deklarasi objek WiFiUDP dan NTPClient yang digunakan untuk pengaturan waktu secara real-time menggunakan protokol NTP (Network Time Protocol). Objek WiFiUDP ntpUDP dibuat untuk membuka koneksi UDP

yang diperlukan dalam komunikasi dengan server waktu. Kemudian, NTPClient timeClient diinisialisasi menggunakan server "pool.ntp.org" dengan pengaturan offset waktu sebesar 25.200 detik (setara UTC+7 untuk zona waktu Indonesia Barat) dan memperbarui data waktu setiap 60 detik. Pengaturan ini sangat penting agar perangkat Esp32 dapat mengetahui waktu saat ini secara akurat, yang nantinya berguna untuk fungsi-fungsi seperti pencatatan data energi berdasarkan waktu atau pengendalian *relay* berbasis jadwal.

```
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org", 25200, 60000); // Sesuaikan UTC+7
```

Gambar 4.11. Tangkapan Layar Baris Pengaturan Waktu

Bagian ini digunakan untuk konfigurasi awal layar oled yang akan digunakan sebagai media tampilan data monitoring. #define SCREEN_WIDTH 128 dan #define SCREEN_HEIGHT 64 mendefinisikan ukuran layar OLED yaitu 128 piksel lebar dan 64 piksel tinggi. #define OLED_RESET -1 menunjukkan bahwa tidak ada pin reset khusus yang digunakan untuk oled. Kemudian, objek Adafruit_SH1106G display dibuat menggunakan library Adafruit dengan menghubungkan parameter ukuran layar dan komunikasi melalui protokol I2C (Wire). Inisialisasi ini bertujuan agar ESP32 dapat mengirimkan data tampilan seperti tegangan, arus, daya, dan energi ke layar oled secara real time.

```
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
Adafruit_SH1106G display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
```

Gambar 4.12. Tangkapan Layar Baris Konfigurasi Awal Layar Oled

Bagian ini berisi deklarasi dan konfigurasi pin yang digunakan untuk mengontrol *relay*. #define RELAY1, RELAY2, RELAY3, dan RELAY4 masing-masing mengatur pin output ESP32 yang terhubung ke modul *relay*, yaitu pada pin GPIO 32, 33, 25, dan 26. Kemudian dibuat array relayPins[4] untuk menyimpan daftar pin *relay* yang digunakan, mempermudah pengaturan *relay* dalam program. Selain itu, relayEnablePins[4] mendefinisikan Virtual Pin dari aplikasi Blynk (V3, V7, V15, V19) yang berfungsi untuk mengaktifkan atau menonaktifkan relay melalui aplikasi. Sedangkan relayCounterPins[4] digunakan untuk menghubungkan Virtual Pin lain (V2, V6, V14, V18) yang nantinya berfungsi mencatat jumlah aktifasi atau status dari masing-masing *relay*. Konfigurasi ini sangat penting untuk mengontrol perangkat listrik dengan lebih fleksibel.

```
#define RELAY1 32
#define RELAY2 33
#define RELAY3 25
#define RELAY4 26
const int relayPins[4] = {32, 25, 26, 33};
const int relayEnablePins[4] = {V3, V7, V15, V19}; // Virtual pin untuk enable relay
const int relayCounterPins[4] = {V2,V6,V14,V18}; // Virtual pin untuk enable relay
```

Gambar 4.13. Tangkapan Layar Baris Konfigurasi Pin

Bagian ini mengatur konfigurasi untuk sensor PZEM-004T v3.0 yang digunakan untuk memantau konsumsi energi listrik. PZEM004Tv30 pzem(Serial2, 16, 17); mendeklarasikan objek pzem yang menghubungkan sensor PZEM ke ESP32 melalui komunikasi serial dengan pin RX dan TX pada pin GPIO 16 dan 17. Variabel resetenergy1 diinisialisasi dengan nilai 0, kemungkinan digunakan untuk menyimpan status atau reset nilai energi yang terbaca dari sensor. Kode yang dikomentari di bawahnya adalah konfigurasi untuk koneksi

WiFi dan Blynk, yang sebelumnya sudah tidak digunakan atau digantikan dengan sistem konfigurasi dinamis menggunakan WiFiManager. Selain itu, terdapat konfigurasi API WhatsApp yang belum diaktifkan, yang bisa digunakan untuk mengirim pemberitahuan ke nomor WhatsApp ketika kondisi tertentu tercapai, misalnya jika daya melebihi batas yang ditentukan.

```
int resetenergyl =0;
//char ssid[] = "GAZEL";
//char pass[] = "1234qwer";
//char auth[] = "-iUOry8LuuTwLugrtZ_K6l_UW0cvlcFF";
//
//// Variabel global untuk API Key dan nomor WhatsApp
//const char* WA_API_KEY = "your_whatsapp_api_key";  // Ganti dengan API Key WhatsApp
//const char* WA_PHONE_NUMBER = "your_phone_number"; // Ganti dengan nomor WhatsApp tujuan
//String notifmessage = "Daya melebihi batas, semua relay dimatikan!";
```

Gambar 4.14. Tangkapan Layar Baris Pemantauan Energi Listrik

Bagian ini mendeklarasikan beberapa variabel yang digunakan untuk pengaturan timer, status relay, serta pembacaan data sensor. BlynkTimer timer adalah objek yang digunakan untuk mengatur interval pengiriman data ke aplikasi Blynk. #define LOGO_WIDTH 128 dan #define LOGO_HEIGHT 64 mendefinisikan ukuran logo yang akan ditampilkan pada layar oled. Variabel relayTimer[4], relayCounter[4], dan lastrelayCounter[4] digunakan untuk mengatur waktu aktif relay, menghitung jumlah operasi relay, dan mencatat status sebelumnya dari masing-masing relay. relayEnabled[4] adalah array boolean yang menyimpan status apakah relay aktif atau tidak, sedangkan relayState[4] menyimpan status *ON/OFF* dari masing-masing relay. Variabel lastVoltage, lastCurrent, lastPower, dan lastEnergy digunakan untuk menyimpan nilai pembacaan terakhir dari sensor energi, seperti tegangan, arus, daya, dan energi, yang nantinya akan ditampilkan atau digunakan dalam pengendalian *relay*.

```
BlynkTimer timer;
#define LOGO_WIDTH   128
#define LOGO_HEIGHT  64
int relayTimer[4] = {0, 0, 0, 0};
int relayCounter[4] = {0, 0, 0, 0};
int lastrelayCounter[4] = {0, 0, 0, 0};
bool relayEnabled[4] = {false, false, false, false};
bool relayState[4] = {false, false, false, false};
float lastVoltage = 0;
float lastCurrent = 0;
float lastPower = 0;
float lastEnergy = 0;
```

Gambar 4.15. Tangkapan Layar Baris atur *Timer*, *Relay*, dan Sensor

Bagian ini berisi variabel yang digunakan untuk memantau dan mengontrol pengiriman notifikasi terkait daya. float powerThreshold = 0; mendeklarasikan variabel powerThreshold yang digunakan untuk menyimpan nilai ambang batas daya yang ditetapkan oleh pengguna melalui aplikasi Blynk. Jika daya yang terdeteksi melebihi nilai ini, sistem dapat melakukan tindakan tertentu, seperti menonaktifkan relay atau mengirimkan notifikasi. bool powerAlertSent = false; adalah flag yang digunakan untuk memastikan bahwa notifikasi terkait ambang batas daya hanya dikirimkan sekali, sehingga mencegah pengiriman notifikasi berulang jika daya terus-menerus melebihi batas yang telah ditentukan.

```
float powerThreshold = 0;  // Threshold daya dari Blynk
bool powerAlertSent = false; // Flag untuk mencegah pengiriman notifikasi berulang
```

Gambar 4.16. Tangkapan Layar Baris Notifikasi Daya Threshold

Bagian ini mendeklarasikan variabel untuk menyimpan nilai-nilai yang dibaca dari sensor energi, yaitu voltage (tegangan), current (arus), power (daya), dan energy (energi). Semua variabel ini diinisialisasi dengan nilai 0, yang akan diperbarui dengan pembacaan terbaru dari sensor PZEM-004T. Fungsi sendWhatsAppMessage(String message); adalah deklarasi dari fungsi yang

digunakan untuk mengirim pesan notifikasi melalui WhatsApp. Fungsi ini akan menerima parameter message, yang berupa string pesan yang akan dikirimkan, kemungkinan besar digunakan untuk memberi tahu pengguna mengenai status atau peringatan terkait konsumsi energi listrik.

```
float voltage = 0;
float current = 0;
float power = 0;
float energy = 0;
void sendWhatsAppMessage(String message);
```

Gambar 4.17. Tangkapan Layar Variabel Pembaca Sensor Energi

Bagian ini adalah fungsi setup() yang dijalankan sekali saat perangkat pertama kali dinyalakan. Pertama, Serial.begin(9600); digunakan untuk memulai komunikasi serial dengan baud rate 9600, yang memungkinkan untuk proses debugging melalui Serial Monitor. Wire.begin(); memulai komunikasi I2C untuk perangkat yang menggunakan protokol tersebut, seperti layar OLED. Kemudian, kode if (!display.begin(0x3C, true)) mengecek apakah layar OLED terhubung dengan benar. Jika tidak, program akan mencetak pesan kesalahan dan berhenti berjalan.

Fungsi showLogo(); digunakan untuk menampilkan logo pada layar oled selama 3 detik, diikuti dengan showID(); untuk menampilkan ID atau informasi lain pada layar. Setelah itu, int_wifimanager(); memulai proses pengaturan koneksi WiFi secara dinamis menggunakan WiFiManager, sehingga pengguna dapat mengonfigurasi koneksi WiFi tanpa harus mengedit kode.

Fungsi Blynk.begin() digunakan untuk menghubungkan ESP32 dengan platform Blynk menggunakan kredensial WiFi yang didapatkan dan server yang

ditentukan, dalam hal ini "iot.serangkota.go.id". Fungsi Blynk.syncVirtual() kemudian menyinkronkan status Virtual Pins dari aplikasi Blynk dengan ESP32, agar data dari aplikasi Blynk dapat diperbarui dengan data yang ada di perangkat.

Pada bagian selanjutnya, dilakukan inisialisasi pin untuk *relay* dengan pinMode() dan digitalWrite(), yang mengatur pin GPIO sebagai output dan memastikan *relay* dalam keadaan OFF (HIGH) saat pertama kali dinyalakan. Timer kemudian diatur menggunakan timer.setInterval(), yang akan menjalankan berbagai fungsi secara berkala: checkRelays untuk memantau status *relay*, updatePZEM untuk memperbarui data dari sensor energi, dan updateOLED untuk memperbarui tampilan pada layar oled.

```
void setup() {
    Serial.begin(9600);
    Wire.begin();
    if (!display.begin(0x3C, true)) {
        Serial.println(F("SH1106 allocation failed"));
        for (;;);
    }
    showLogo();
    delay(3000);
    display.setTextColor(SH110X_WHITE);
    showID();
    delay(3000);
    int_wifimanager();
    //Blynk.begin(auth, ssid, pass);
    Blynk.begin(auth, WiFi.SSID().c_str(), WiFi.psk().c_str(), "iot.serangkota.go.id", 8080); // U may use other free server, this is an example only.
    Blynk.syncVirtual(V1,V2, V3, V4,V5, V6, V7, V8,V9, V14, V15, V16,V17, V18, V19, V20);

    for (int i = 0; i < 4; i++) {
        pinMode(relayPins[i], OUTPUT);
        digitalWrite(relayPins[i], HIGH);
    }

    timer.setInterval(1000L, checkRelays);
    timer.setInterval(500L, updatePZEM);
    timer.setInterval(1000L, updateOLED);
    //timer.setInterval(1000L, updateBlynkCountdown);


}
```

Gambar 4.18. Tangkapan Layar Baris Fungsi Setup()

Fungsi showLogo(): Fungsi ini bertanggung jawab untuk menampilkan logo pada layar oled. Pertama, layar oled dibersihkan dengan display.clearDisplay(), kemudian logo digambar di tengah layar menggunakan display.drawBitmap().

Posisi logo dihitung agar berada di tengah layar dengan mempertimbangkan ukuran logo dan ukuran layar. display.display() digunakan untuk memperbarui tampilan layar setelah logo digambar.

```
// Fungsi untuk menampilkan logo di OLED
void showLogo() {
  display.clearDisplay();
  display.drawBitmap((display.width() - LOGO_WIDTH) / 2, (display.height() - LOGO_HEIGHT) / 2, logo_bmp, LOGO_WIDTH, LOGO_HEIGHT, SH110X_WHITE);
  display.display();
}
```

Gambar 4.19. Tangkapan Layar Baris Tampil Logo di Oled

Fungsi showID(): Fungsi ini digunakan untuk menampilkan informasi terkait proyek pada layar oled, yaitu nama proyek, nama penulis, dan NIM. Fungsi ini dimulai dengan membersihkan layar oled terlebih dahulu dengan display.clearDisplay(). Kemudian, fungsi ini mengatur ukuran teks menggunakan display.setTextSize(1) dan menetapkan posisi kursor dengan display.setCursor(x, y). Setiap informasi (judul proyek, nama, dan NIM) ditampilkan pada posisi yang sudah ditentukan dengan display.print(), dan akhirnya, display.display() digunakan untuk memperbarui tampilan layar.

```
void showID()
{
    display.clearDisplay();
    display.setTextSize(1);
    display.setCursor(26, 15);
    display.print("SMART RELAY");
    display.setCursor(21, 30);
    display.print("ALDYA SAPUTRA");
    display.setCursor(30, 45);
    display.print("C2155201085");
    display.display();
}
```

Gambar 4.20. Tangkapan Layar Baris Tampil Nama dan NIM di Oled

Fungsi warning(): Fungsi ini digunakan untuk menampilkan pesan peringatan pada layar oled, khususnya ketika daya yang terdeteksi melebihi batas yang telah ditentukan. Fungsi ini dimulai dengan membersihkan layar oled menggunakan display.clearDisplay(). Kemudian, teks "WARNING!" ditampilkan dengan ukuran font yang lebih besar (setTextSize(2)) di posisi tengah layar. Fungsi display.getTextBounds() digunakan untuk mendapatkan ukuran dan posisi teks, yang memastikan teks "WARNING!" ditempatkan dengan tepat di tengah layar.

Setelah itu, fungsi ini menampilkan pesan peringatan lebih lanjut ("Daya melebihi batas!" dan "Semua relay dimatikan!") dengan ukuran font lebih kecil (setTextSize(1)), yang juga ditampilkan di posisi tengah layar menggunakan display.getTextBounds() untuk perhitungan posisi teks secara otomatis. Semua teks ditampilkan secara berurutan, dan akhirnya, tampilan layar diperbarui dengan display.display().

```
void warning()
{
    display.clearDisplay();

    // Tampilkan "WARNING!" besar dan di tengah
    display.setTextSize(2); // Ukuran lebih besar
    int16_t x, y;
    uint16_t w, h;
    display.getTextBounds("WARNING!", 0, 0, &x, &y, &w, &h);
    display.setCursor((SCREEN_WIDTH - w) / 2, 5);
    display.println("WARNING!");

    // Tampilkan pesan peringatan lebih rapi di tengah
    display.setTextSize(1);
    display.getTextBounds("Daya melebihi batas!", 0, 0, &x, &y, &w, &h);
    display.setCursor((SCREEN_WIDTH - w) / 2, 25);
    display.println("Daya melebihi batas!");

    display.getTextBounds("Semua relay dimatikan!", 0, 0, &x, &y, &w, &h);
    display.setCursor((SCREEN_WIDTH - w) / 2, 40);
    display.println("Semua relay dimatikan");

    display.display();
}
```

Gambar 4.21. Tangkapan Layar Baris Pesan Peringatan di Layar Oled

Fungsi checkRelays() digunakan untuk memantau dan mengatur status *relay* secara berkala. Fungsi ini dimulai dengan mereset penghitung jumlah *relay* yang aktif dan kemudian memeriksa status setiap *relay*, mengidentifikasi jika ada perubahan status (*relay* yang sebelumnya mati menjadi hidup). Jika *relay* diaktifkan dan memiliki waktu hitung mundur (relayCounter[i] > 0), maka status relay akan diperbarui menjadi *"ON*", menghitung mundur waktu, dan menampilkan pembaruan status melalui aplikasi Blynk. Setelah waktu habis, *relay* dimatikan dan status tombol di Blynk diperbarui. Selain itu, *relay* akan diatur melalui pin yang sesuai, dengan menggunakan logika untuk menghidupkan atau mematikan *relay.* Fungsi ini juga mencetak status semua *relay* ke Serial Monitor

untuk memudahkan debugging, dan menambahkan penundaan jika lebih dari satu *relay* aktif agar perubahan status dapat terlihat jelas.

```
void checkRelays() {
      static int counterRelayAktif = 0; // Menyimpan jumlah relay yang baru aktif
    counterRelayAktif = 0; // Reset setiap pemanggilan checkRelays()
    for (int i = 0; i < 4; i++) {
        bool previousState = (digitalRead(relayPins[i]) == LOW); // LOW berarti relay sebelumnya ON

        if (!previousState && relayState[i]) {
            // Jika relay sebelumnya OFF (HIGH) lalu berubah menjadi ON (LOW)
            counterRelayAktif++;
        }
    }
    for (int i = 0; i < 4; i++) {
        if (relayEnabled[i] && relayCounter[i] > 0) {
            relayState[i] = true;
            relayCounter[i]--;
            Blynk.virtualWrite(relayCounterPins[i], relayCounter[i]); // Update tombol di Blynk ke OFF
            if (relayCounter[i] == 0) {
              relayState[i] = false;
              relayEnabled[i] = false;
              relayCounter[i] = lastrelayCounter[i];
              Blynk.virtualWrite(relayEnablePins[i], 0); // Update tombol di Blynk ke OFF
              Blynk.virtualWrite(relayCounterPins[i], lastrelayCounter[i]); // Update tombol di Blynk ke OFF
              }
        }


         digitalWrite(relayPins[i], relayState[i] ? LOW : HIGH);
         if(counterRelayAktif >1){delay(500);}
    }
    Serial.printf("[Relays] R1:%s R2:%s R3:%s R4:%s\n",
                  relayState[0] ? "ON" : "OFF",
                  relayState[1] ? "ON" : "OFF",
                  relayState[2] ? "ON" : "OFF",
                  relayState[3] ? "ON" : "OFF");
}
```

Gambar 4.22. Tangkapan Layar Baris Mengatur Status Relay

Fungsi updatePZEM() bertugas untuk memperbarui data dari sensor PZEM-004T yang digunakan untuk memantau konsumsi listrik. Pertama, jika variabel resetenergy1 bernilai 0, fungsi akan mengambil data baru seperti tegangan, arus, daya, dan energi dari sensor PZEM. Data yang diambil akan dicek validitasnya, dan jika valid, nilai-nilai tersebut akan diperbarui. Jika resetenergy1 tidak sama dengan 0, maka energi pada sensor akan direset, dan status reset akan dicetak ke Serial Monitor. Selanjutnya, fungsi ini memeriksa apakah daya yang terdeteksi

melebihi nilai ambang batas (powerThreshold). Jika daya melebihi batas dan notifikasi belum dikirim, maka semua relay akan dimatikan, pesan pemberitahuan akan dikirimkan melalui WhatsApp, dan status notifikasi akan diubah menjadi terkirim untuk mencegah pengiriman notifikasi berulang.

```
        void updatePZEM() {
            if(resetenergy1 ==0)
            {
            float newVoltage = pzem.voltage();
            float newCurrent = pzem.current();
            float newPower = pzem.power();
            float newEnergy = pzem.energy();

            // Cek apakah data valid, jika valid baru update nilai
            if (!isnan(newVoltage)) voltage = newVoltage;
            if (!isnan(newCurrent)) current = newCurrent;
            if (!isnan(newPower)) power = newPower;
            if (!isnan(newEnergy)) energy = newEnergy;
            }
            else {

               if(pzem.resetEnergy())
              {Serial.println("energy berhasil direset");resetenergy1=0;}
              else {Serial.println("gagal energy direset");}

            }
        // Deteksi jika power melebihi threshold
    if (power > powerThreshold) {
        if (!powerAlertSent) {
            matikanSemuaRelay();
            //Blynk.logEvent("power_alert", "Daya melebihi batas, semua relay dimatikan!");
            sendWhatsAppMessage(notifmessage);
            powerAlertSent = true; // Pastikan hanya mengirim notifikasi sekali
        }
    }
}
```

Gambar 4.23. Tangkapan Layar Baris Perbarui PZEM

Fungsi matikanSemuaRelay() berfungsi untuk mematikan semua relay yang terhubung dalam sistem. Fungsi ini melakukan iterasi melalui array relay dan mengubah status setiap relay menjadi mati dengan mengatur relayState[i] dan relayEnabled[i] menjadi false. Setelah itu, status tombol di aplikasi Blynk diperbarui untuk mencerminkan status *relay* yang mati, baik untuk tombol enable

maupun penghitung *relay*. *Relay* juga dimatikan dengan menulis nilai HIGH pada pin yang sesuai. Selain itu, fungsi ini juga mengirimkan peringatan ke aplikasi Blynk dengan mematikan beberapa virtual pin (V4, V8, V16, V20), serta menampilkan pesan peringatan di layar oled menggunakan fungsi warning(). Pesan peringatan juga dicetak ke Serial Monitor untuk memberikan informasi bahwa daya melebihi batas dan semua relay telah dimatikan.

```
void matikanSemuaRelay() {
    for (int i = 0; i < 4; i++) {
        relayState[i] = false;
        relayEnabled[i] = false;
        Blynk.virtualWrite(relayEnablePins[i], 0); // Matikan di Blynk
        Blynk.virtualWrite(relayCounterPins[i], lastrelayCounter[i]); // Update tombol di Blynk ke OFF
        digitalWrite(relayPins[i], HIGH);
    }
    Blynk.virtualWrite(V4, 0); // Matikan di Blynk
    Blynk.virtualWrite(V8, 0); // Matikan di Blynk
    Blynk.virtualWrite(V16, 0); // Matikan di Blynk
    Blynk.virtualWrite(V20, 0); // Matikan di Blynk
    warning();

    Serial.println("WARNING: Power limit exceeded! All relays turned OFF.");
}
```

Gambar 4.24. Tangkapan Layar Baris Matikan Semua *Relay*

Fungsi updateOLED() bertugas untuk memperbarui tampilan pada layar oled dan mengirimkan data terkait konsumsi listrik ke aplikasi Blynk. Di dalam fungsi ini, nilai-nilai seperti tegangan (voltage), arus (current), daya (power), dan energy (energy) ditampilkan pada aplikasi Blynk menggunakan Blynk.virtualWrite(), yang mengirimkan data ke pin virtual yang sesuai. Selain itu, data ini juga ditampilkan secara serial di monitor untuk memudahkan pemantauan. Di layar oled, fungsi ini menampilkan informasi seperti nama sistem, nilai-nilai tegangan, arus, daya, dan energi dengan format yang teratur agar mudah dibaca. Status relay juga ditampilkan pada dua baris di layar, yang menunjukkan apakah masing-masing relay dalam keadaan "*ON*" atau *"OFF"*. Fungsi ini hanya

berjalan jika tidak ada peringatan daya yang dikirimkan, yang memastikan bahwa tampilan oled tetap memperbarui informasi sepanjang waktu.

```
        void updateOLED() {
            Blynk.virtualWrite(V10, voltage);
            Blynk.virtualWrite(V11, current);
            Blynk.virtualWrite(V12, power);
            Blynk.virtualWrite(V13, energy);

            Serial.print("Voltage: "); Serial.println(voltage);
            Serial.print("Current: "); Serial.println(current);
            Serial.print("Power: "); Serial.println(power);
            Serial.print("Energy: "); Serial.println(energy);

            if( !powerAlertSent){
                display.clearDisplay();
                // Tampilkan judul di tengah
                display.setTextSize(1);
                display.setCursor((SCREEN_WIDTH - 6 * strlen("SmartRelay")) / 2, 0);
                display.print("SmartRelay");
                // Format nilai agar sejajar
                display.setCursor(5, 10);
                display.printf("Voltage : %.2f V", voltage);
                display.setCursor(5, 18);
                display.printf("Current : %.2f A", current);
                display.setCursor(5, 28);
                display.printf("Power   : %.2f W", power);
                display.setCursor(5, 38);
                display.printf("Energy  : %.2f kWh", energy);


            // Tampilkan status relay dalam dua baris
            display.setCursor(10, 46);  // Posisi kiri atas untuk R1 dan R2
            display.printf("R1: %s    R2: %s", relayState[0] ? "ON " : "OFF", relayState[1] ? "ON " : "OFF");

            display.setCursor(10, 56);  // Posisi di bawahnya untuk R3 dan R4
            display.printf("R3: %s    R4: %s", relayState[2] ? "ON " : "OFF", relayState[3] ? "ON " : "OFF");

            display.display();
        }
    }
```

Gambar 4.25. Tangkapan Layar Baris Perbarui Tampilan Layar Oled

Bagian kode ini berisi beberapa fungsi BLYNK_WRITE() digunakan untuk menerima dan memproses input dari aplikasi Blynk, yang mengontrol berbagai parameter dalam sistem. Fungsi pertama mengatur ambang batas daya (powerThreshold) melalui pin virtual V1, sementara pin V5 digunakan untuk mereset status peringatan daya. Fungsi lainnya mengelola counter dan status relay

(relayCounter dan relayState) untuk masing-masing relay (V2-V8, V14-V20), mengubah nilai berdasarkan input dari aplikasi Blynk. Fungsi BLYNK_WRITE(V21) dan BLYNK_WRITE(V22) digunakan untuk mereset pengaturan WiFi dan energi, masing-masing. Secara keseluruhan, kode ini memungkinkan kontrol dinamis terhadap relay, pembaruan status daya, dan pengelolaan konfigurasi melalui aplikasi Blynk, memberikan fleksibilitas dalam pengoperasian sistem secara real-time.



```
BLYNK_WRITE(V1) { powerThreshold = param.asFloat(); Serial.print("Threshold daya diperbarui: "); Serial.println(powerThreshold);}
BLYNK_WRITE(V5) { if( param.asInt()==1){powerAlertSent =false;} }
BLYNK_WRITE(V2) { relayCounter[0] = param.asInt(); }
BLYNK_WRITE(V3) { relayEnabled[0] = param.asInt(); relayState[0] = relayEnabled[0];if(relayEnabled[0]==1){lastrelayCounter[0]= relayCounter[0] ;}if(relayEnabled[0]==0){
BLYNK_WRITE(V4) { if (!relayEnabled[0]) relayState[0] = param.asInt(); }

BLYNK_WRITE(V6) { relayCounter[1] = param.asInt(); }
BLYNK_WRITE(V7) { relayEnabled[1] = param.asInt(); relayState[1] = relayEnabled[1]; if(relayEnabled[1]==1){lastrelayCounter[1]= relayCounter[1] ;}if(relayEnabled[1]==0)
BLYNK_WRITE(V8) { if (!relayEnabled[1]) relayState[1] = param.asInt(); }

BLYNK_WRITE(V14) { relayCounter[2] = param.asInt(); }
BLYNK_WRITE(V15) { relayEnabled[2] = param.asInt(); relayState[2] = relayEnabled[2]; if(relayEnabled[2]==1){lastrelayCounter[2]= relayCounter[2] ;}if(relayEnabled[2]==0
BLYNK_WRITE(V16) { if (!relayEnabled[2]) relayState[2] = param.asInt(); }

BLYNK_WRITE(V18) { relayCounter[3] = param.asInt(); }
BLYNK_WRITE(V19) { relayEnabled[3] = param.asInt(); relayState[3] = relayEnabled[3];if(relayEnabled[3]==1){lastrelayCounter[3]= relayCounter[3] ;}if(relayEnabled[3]==0)
BLYNK_WRITE(V20) { if (!relayEnabled[3]) relayState[3] = param.asInt(); }


Blynk.virtualWrite(relayCounterPins[0], lastrelayCounter[0]);}}


{Blynk.virtualWrite(relayCounterPins[1], lastrelayCounter[1]);}}


)){Blynk.virtualWrite(relayCounterPins[1], lastrelayCounter[1]);} }


{Blynk.virtualWrite(relayCounterPins[1], lastrelayCounter[1]);} }
```

Gambar 4.26. Tangkapan Layar Baris Fungsi BLYNK_WRITE()

Fungsi sendWhatsAppMessage ini digunakan untuk mengirim pesan WhatsApp melalui API CallMeBot. Fungsi ini pertama-tama memastikan bahwa ESP32 terhubung ke WiFi sebelum melanjutkan untuk mengirimkan pesan. URL yang dibentuk mencakup nomor telepon penerima, API Key yang telah diberikan oleh CallMeBot, serta pesan yang ingin dikirim. URL tersebut kemudian

dikodekan menggunakan urlEncode untuk memastikan pesan dapat diterima dengan benar oleh server. Dengan menggunakan HTTP POST, fungsi ini mengirim permintaan ke server CallMeBot untuk mengirim pesan WhatsApp. Setelah itu, kode memeriksa respons dari server untuk memastikan pengiriman berhasil, jika berhasil akan menampilkan pesan "WhatsApp notification sent successfully," atau jika gagal akan mencetak kode respons error dari server.

```
void sendWhatsAppMessage(String message) {
    String apiKey = String(WA_API_KEY);  // Ganti dengan API Key dari CallMeBot
    String phoneNumber = String(WA_PHONE_NUMBER);  // Ganti dengan nomor tujuan (format internasional tanpa "+")

    if (WiFi.status() == WL_CONNECTED) {
  // Data to send with HTTP POST
  String url = "https://api.callmebot.com/whatsapp.php?phone=" + phoneNumber + "&apikey=" + apiKey + "&text=" + urlEncode(message);
  HTTPClient http;
  http.begin(url);

  // Specify content-type header
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");

  // Send HTTP POST request
  int httpResponseCode = http.POST(url);

    if (httpResponseCode == 200) {
        Serial.println("WhatsApp notification sent successfully.");
    } else {
        Serial.print("Error sending WhatsApp notification: ");
        Serial.println(httpResponseCode);
    }

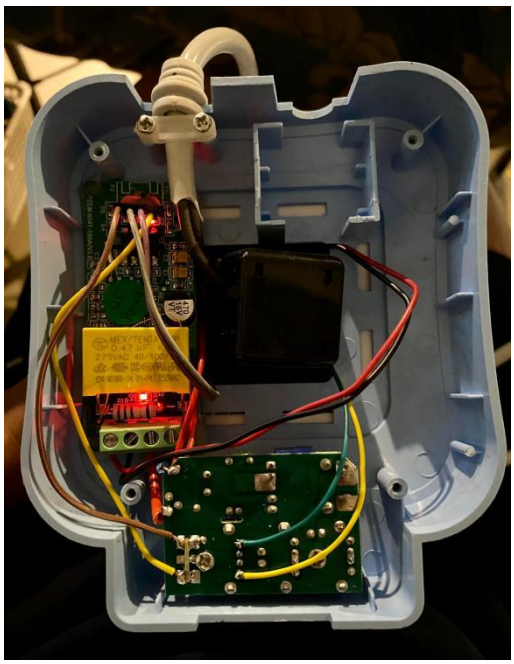
    http.end();
    }
}
```

Gambar 4.27. Tangkapan Layar Baris Kirim Pesan WhatsApp

4.1.4 Implementasi Alat

Hasil implementasi alat yang dibuat menunjukkan bahwa sistem ini mampu memantau konsumsi listrik secara langsung dan mengontrol penggunaan listrik secara otomatis melalui smart *relay*. Alat ini bekerja dengan akurat dalam mencatat jumlah energi yang digunakan dan dapat memberikan notifikasi jika pemakaian listrik melebihi batas yang sudah ditentukan. Dengan fitur ini, pengguna bisa lebih sadar terhadap konsumsi listrik mereka. Selain itu, alat ini

juga membantu mengurangi penggunaan listrik yang tidak perlu, sehingga bisa meningkatkan efisiensi dan menghemat energi dalam kehidupan sehari-hari.



Gambar 4.28. Implementasi Alat

4.1.5 Pengujian

a. Pengujian *Black Box*

Dalam melakukan pengujian alat, metode yang digunakan adalah *Black Box Testing*. Metode ini berfokus pada pengujian fungsi dan keluaran program tanpa memperhatikan struktur internal kode. Tujuan dari pengujian ini adalah untuk menemukan kesalahan atau ketidaksesuaian pada sistem. Berdasarkan hasil pengujian yang telah diuji oleh **Abdul Hadi, S.T., M.Kom.**, program yang dirancang dapat berfungsi dan berjalan sesuai dengan yang diharapkan dapat dilihat pada tabel berikut.

Tabel 4.2. *Black Box Testing*

No	Pengujian	Keterangan	Hasil
1	Mengoneksikan Board Mikrokontroler ESP32 ke Laptop	Board ESP32 dapat terkoneksi dengan baik ke laptop dan terdeteksi di perangkat.	Berhasil
2	Uji Coba Upload Kode Program ke Mikrokontroler ESP32	Proses upload program berjalan lancar tanpa ada error.	Berhasil
3	Uji Coba Menghubungkan Mikrokontroler ESP32 ke Jaringan Wi-Fi	Mikrokontroler berhasil terhubung ke jaringan Wi-Fi tanpa masalah.	Berhasil
4	Uji Coba *Relay*	Semua *relay* dapat berfungsi dengan baik sesuai perintah yang diberikan.	Berhasil
5	Uji Coba Sensor Arus	Sensor arus dapat mendeteksi dan membaca arus listrik dengan akurat.	Berhasil
6	Uji Coba Modul Step Down DC	Modul step down DC bekerja dengan baik dalam menurunkan dan menstabilkan tegangan.	Berhasil

b. Hasil pengujian

1. Mengoneksikan Board Mikrokontroler Esp32 Ke Laptop

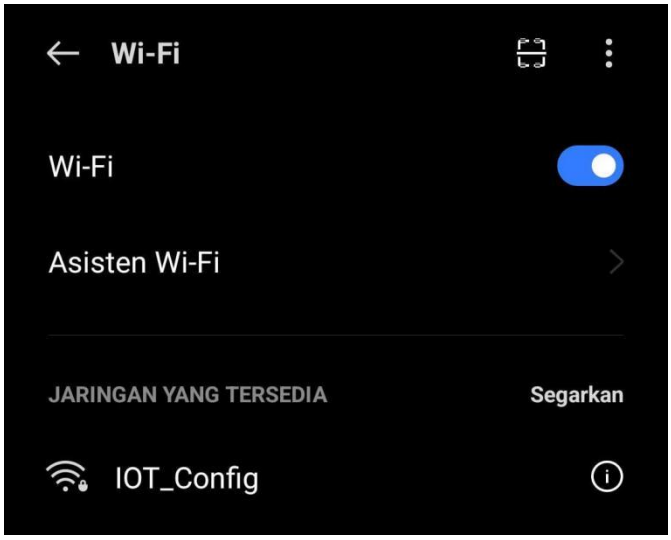
Proses ini dilakukan untuk memastikan apakah ESP32 sudah terhubung ke laptop atau belum, yang dapat dicek melalui aplikasi Arduino IDE. Jika ESP32 berhasil terhubung, maka port akan terdeteksi secara otomatis, sedangkan jika tidak terhubung, port tidak akan muncul. Pendeteksian port ini penting agar program dapat diunggah ke ESP32 melalui Arduino IDE.

2. Uji Coba Upload Kode Program Ke Mikrokontroler Esp32

Proses ini dilakukan untuk memastikan bahwa ESP32 dapat menerima, menyimpan, dan menjalankan kode program dengan baik. Program yang diunggah berisi instruksi dasar, seperti perintah untuk menghubungkan ke jaringan Wi-Fi, membaca data dari sensor, dan mengontrol perangkat. Setelah memastikan tidak ada kesalahan dalam kode, tombol Upload pada Arduino IDE ditekan untuk memulai proses unggah atau compiling sketch. Jika proses berhasil, akan muncul notifikasi "Done Uploading" di bagian bawah Arduino IDE, dan Esp32 akan mulai menjalankan program tersebut sebagaimana ditunjukkan pada gambar.

3. Uji Coba Menghubungkan Mikrokontroler Esp32 Ke Jaringan Wi-Fi

Setelah mikrokontroler Esp32 terhubung ke sumber daya listrik dan program berhasil diunggah melalui Arduino IDE, Esp32 akan otomatis memunculkan jaringan Wi-Fi bernama iot_config sebagai bagian dari proses konfigurasi awal dapat dilihat pada gambar 4.29. berikut:



Gambar 4.29. Konfigurasi Awal Wi-Fi.

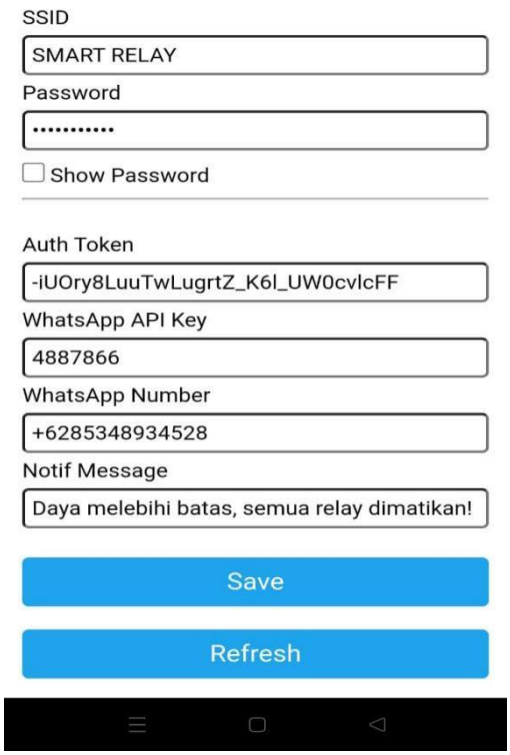
Pengguna kemudian akan diarahkan ke halaman Wi-Fi Manager setelah memilih jaringan iot_config, di mana dapat menekan tombol Configure Wi-Fi untuk melanjutkan dilihat pada gambar 4.30. berikut:



Gambar 4.30. Tampilan Wi-Fi Manager

Pada halaman konfigurasi ini, pengguna dapat memilih jaringan Wi-Fi (SSID) yang tersedia dan memasukkan password yang sesuai. Setelah

memasukkan kredensial Wi-Fi, tombol Save ditekan untuk menyimpan pengaturan, dan Esp32 akan mencoba terhubung ke jaringan Wi-Fi yang dipilih. Jika berhasil terhubung, Esp32 akan memperoleh alamat IP dari router yang digunakan, dan alat tersebut dapat mulai berkomunikasi dengan aplikasi Blynk dilihat pada gambar 4.31. berikut:



Gambar 4.31. Tampilan Halaman Konfigurasi

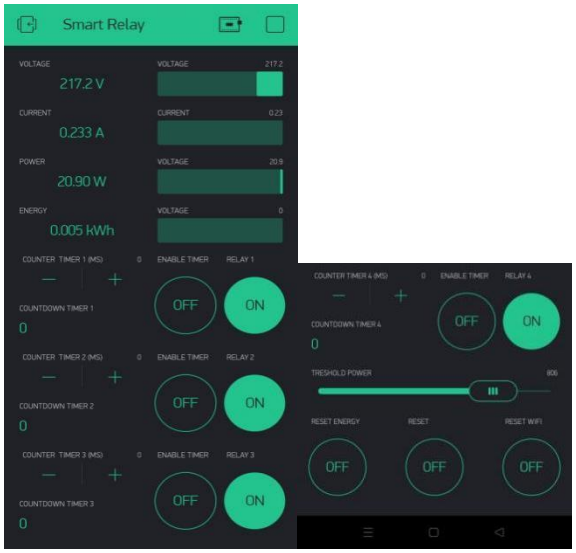
a. Uji Coba *Relay*

Relay berperan sebagai saklar otomatis yang mengatur aliran listrik ke peralatan rumah tangga berdasarkan perintah dari aplikasi Blynk. Dalam pengujian ini, masing-masing *relay* diuji menggunakan beban nyata berupa kipas angin, speaker GMC, charger laptop, dan charger HP. Ketika tombol *ON* ditekan melalui aplikasi Blynk, *relay* secara langsung mengaktifkan perangkat yang

terhubung. Sebaliknya, ketika tombol *OFF* ditekan, *relay* memutus aliran listrik sehingga perangkat mati.

Selain dapat dikontrol secara manual melalui aplikasi, sistem *relay* juga dapat bekerja otomatis melalui fitur timer dan threshold power. Dengan fitur ini, perangkat dapat diatur untuk menyala atau mati secara otomatis sesuai jadwal waktu tertentu, atau saat konsumsi daya melebihi ambang batas yang telah ditetapkan. Hasil pengujian menunjukkan bahwa semua *relay* berfungsi dengan baik dan responsif, baik dalam mode manual maupun otomatis.

Pada Gambar 4.32. dibawah ini antarmuka aplikasi Blynk yang menampilkan status *relay* aktif. Dalam kondisi ini, *relay* mengendalikan perangkat-perangkat seperti kipas angin, speaker GMC, charger laptop, dan charger HP sesuai pengaturan yang telah dibuat di aplikasi. Data seperti daya total juga ditampilkan secara real-time.



Gambar 4.32. Antarmuka Aplikasi Blynk Saat *Relay* Aktif

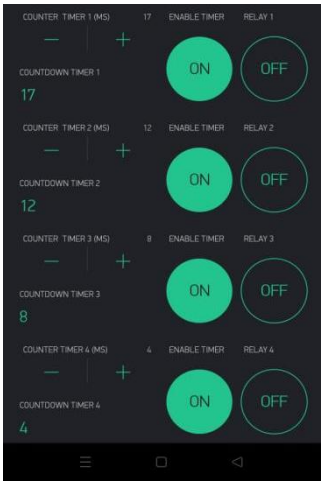
Sementara itu, pada Gambar 4.33, tampilan Oled display menunjukkan kondisi *relay* aktif disertai informasi kelistrikan yang terukur, seperti tegangan

(Volt), arus (Ampere), daya (Watt), dan energi (kWh). Informasi ini menunjukkan bahwa alat mampu memonitor parameter kelistrikan secara real-time untuk semua beban yang terhubung.



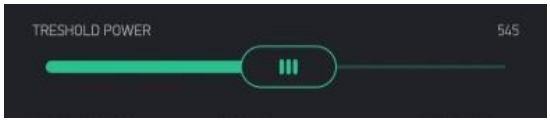
Gambar 4.33. Tampilan Oled Saat *Relay* Aktif

Pada Gambar 4.34 terlihat antarmuka aplikasi Blynk dengan fitur enable timer aktif, di mana keempat *relay* (*Relay* 1, *Relay* 2, *Relay* 3, dan *Relay* 4) dalam keadaan menyala. Masing-masing relay menampilkan penghitung waktu (timer) yang sedang berjalan. Kondisi ini menunjukkan bahwa perangkat yang terhubung pada prototype, yaitu kipas angin, speaker GMC, charger laptop, dan charger HP, sedang beroperasi dan akan mati secara otomatis ketika timer mencapai angka nol. Aplikasi tetap menampilkan nilai daya dari masing-masing perangkat sesuai pengaturan yang diterapkan.



Gambar 4.34. Fitur *Enable Timer*

Pada gambar 4.35. antarmuka aplikasi Blynk dibawah ini, terdapat fitur threshold power yang diatur pada nilai 545. Ketika daya listrik melebihi ambang batas tersebut, maka relay akan secara otomatis mati.



Gambar 4.35. Fitur Threshold Power

b. Uji Coba Sensor Arus

Sensor arus pada sistem ini berfungsi untuk membaca dan menampilkan parameter kelistrikan seperti tegangan (Volt), arus (Ampere), daya (Watt), dan energi (kWh). Pada tahap awal pengujian tanpa adanya beban yang terhubung, sensor menunjukkan tegangan sebesar ±207 Volt, arus 0,000 Ampere, dan daya 0,000 Watt, yang mengindikasikan bahwa sensor berada dalam kondisi siaga *(standby)* namun tetap aktif membaca nilai dasar dari jaringan listrik.

c. Uji Coba Step Down DC

Step down DC pada sistem ini berfungsi untuk menurunkan tegangan dari adaptor 12 Volt menjadi 5 Volt agar sesuai dengan kebutuhan mikrokontroler Esp32. Berdasarkan hasil pengujian, modul step down mampu menghasilkan output sebesar 5,03 Volt secara stabil, yang diukur menggunakan multimeter digital. Tegangan ini digunakan untuk menyuplai daya ke Esp32, sensor arus PZEM-004T, dan modul *relay* 4 channel. Selama pengujian, tidak ditemukan gangguan seperti perangkat mati mendadak, kesalahan pembacaan sensor, maupun kegagalan koneksi WiFi, yang menandakan bahwa suplai daya yang diberikan dalam kondisi stabil dan memadai. Dengan demikian, dapat disimpulkan bahwa modul step down DC berfungsi dengan baik dan pengujian terhadap komponen ini dinyatakan berhasil.

4.2 Pembahasan

4.2.1 Rangkaian

Mikrokontroler Esp32 merupakan komponen utama yang digunakan sebagai pusat kendali dalam sistem monitoring dan pengendalian konsumsi energi listrik rumah tangga. Esp32 dilengkapi dengan dual-core processor, kapasitas ram dan flash memory yang besar, serta konektivitas Wi-Fi dan Bluetooth yang sudah terintegrasi. Dalam sistem ini, Esp32 digunakan untuk membaca data dari sensor arus PZEM-004T, mengendalikan modul *relay*, mengirimkan data secara real time ke aplikasi blynk melalui koneksi Wi-Fi, serta menampilkan informasi konsumsi daya listrik ke display oled. Dengan kemampuannya membaca data digital maupun analog secara akurat, Esp32 mampu bekerja dengan efisien dalam

menjalankan logika otomatis seperti pengaturan waktu *(timer)* dan pemutusan otomatis aliran listrik ketika daya yang digunakan melebihi batas yang telah ditentukan.

4.2.2 Pemrograman Perangkat Lunak

Mikrokontroler Esp32 diluncurkan pada september 2016 sebagai pengembangan dari Esp8266. Mikrokontroler ini memiliki spesifikasi seperti prosesor dual-core Xtensa LX6 dengan kecepatan hingga 240 MHz, memori 520 KB SRAM, serta mendukung konektivitas Wi-Fi dan Bluetooth, sehingga menjadikannya pilihan populer untuk proyek *Internet of Things* (IoT). Dalam proses perancangan dan pemrograman, ditemukan beberapa kendala teknis, terutama saat menginstal board Esp32 di Arduino IDE yang memerlukan pengaturan tambahan dan pemilihan jenis board yang tepat. Selain itu, beberapa library sensor tidak dapat digunakan secara langsung karena banyak yang dibuat untuk Arduino Uno atau Mega. Untuk mengatasi hal ini, dilakukan pencarian library alternatif melalui GitHub dan penyesuaian kode program, termasuk pemetaan pin dan pengaturan komunikasi, agar semua sensor dan aktuator dapat berfungsi dengan baik.

4.2.3 Alat

Pengukuran konsumsi energi listrik pada rumah tangga di Kota Palangka Raya, tepatnya di Jl. Temanggung Tilung 22, Kelurahan Menteng, dilakukan pada tanggal 10 Mei 2025 menggunakan *prototype* alat monitoring konsumsi energi listrik berbasis IoT. Alat ini menggunakan mikrokontroler Esp32 sebagai pusat kendali sistem yang terhubung dengan, *relay*, sensor arus dan modul step down

DC. Berdasarkan hasil pengujian, diperoleh data bahwa total daya listrik dari komponen yang terhubung, yaitu kipas angin, speaker GMC, mejikom, dan setrika, adalah sekitar 20.90 watt,

Sistem ini dirancang untuk merespons secara otomatis berdasarkan batas daya yang sudah ditentukan, yaitu 545 Watt. Dari hasil pengujian yang dilakukan, alat berhasil memantau konsumsi daya listrik dari beberapa peralatan rumah tangga seperti kipas angin, speaker GMC, mejikom, dan setrika, dengan total daya sekitar 20.90 Watt. Selama daya masih di bawah batas tersebut, *relay* tetap menyala dan listrik tetap mengalir. Namun, jika konsumsi daya melebihi 545 Watt, *relay* akan otomatis mati sebagai bentuk perlindungan agar tidak terjadi kelebihan beban. Semua data penggunaan listrik dikirim secara real time ke aplikasi Blynk lewat koneksi Wi-Fi dan bisa langsung dipantau lewat *smartphone.*

4.2.4 Analisis Konsumsi Energi

Pengukuran konsumsi energi listrik dilakukan menggunakan kWh meter PLN yang terpasang di lokasi pengujian. Data dari kWh meter ini digunakan sebagai pembanding untuk mengetahui perbedaan konsumsi energi listrik sebelum dan sesudah penggunaan alat smart relay berbasis IoT. Pengujian dilakukan dalam dua kondisi, yaitu sebelum dan sesudah alat digunakan, masing-masing selama 3 (tiga) hari berturut-turut, dimulai pada tanggal 14 Juni 2025. Melalui pengujian ini diharapkan dapat terlihat perbedaan konsumsi energi listrik antara penggunaan sistem manual dan sistem otomatis dengan smart relay.

a. Kondisi Sebelum Menggunakan Smart *Relay*

Pada kondisi ini, semua peralatan listrik di lokasi pengujian digunakan secara manual tanpa bantuan alat pengendali otomatis. Perangkat dinyalakan dan dimatikan langsung oleh pengguna sesuai kebutuhan, sehingga memungkinkan terjadinya pemborosan energi karena ada perangkat yang tetap menyala meskipun sudah tidak digunakan lagi. Pengukuran konsumsi energi dilakukan dengan mencatat nilai sisa token atau pulsa listrik pada meteran prabayar PLN, di mana nilai awal sisa token digunakan sebagai acuan penghitungan selama pengujian. Pengujian ini dilakukan selama 3 (tiga) hari berturut-turut pada tanggal 14–17 Juni 2025. Tampilan nilai awal token sebelum dilakukan pengujian ditunjukkan pada gambar berikut..



Gambar 4.36. Token Sebelum Smart *Relay*

b. Kondisi Sesudah Menggunakan Smart *Relay*

Pada kondisi ini, penggunaan listrik sudah dilengkapi dengan sistem pengaturan otomatis yang memungkinkan perangkat menyala dan mati secara

terkontrol sesuai waktu dan batas daya yang telah ditentukan. Dengan adanya pengaturan ini, penggunaan energi listrik menjadi lebih efisien karena aliran listrik ke perangkat dapat dihentikan secara otomatis saat tidak lagi diperlukan. Pengukuran konsumsi energi dilakukan dengan mencatat nilai sisa token atau pulsa listrik pada meteran prabayar PLN, di mana nilai awal sisa token digunakan sebagai acuan penghitungan selama pengujian. Pengujian ini dilakukan selama 3 (tiga) hari berturut-turut pada tanggal 17–20 Juni 2025. Tampilan nilai awal token setelah penerapan sistem pengaturan otomatis ditunjukkan pada gambar berikut.



Gambar 4.37. Token Sesudah Smart *Relay*

c. Hasil Pengukuran

Pengukuran konsumsi energi listrik dilakukan untuk mengetahui selisih penggunaan energi sebelum dan sesudah menggunakan alat smart *relay*. Data diperoleh dari pencatatan nilai awal dan akhir sisa token atau pulsa listrik pada meteran prabayar PLN selama pengujian berlangsung. Hasil pengukuran tersebut disajikan pada tabel berikut:

Tabel 4.3. Tabel Perbandingan Konsumsi Energi Listrik

No	Tahap Pengujian	Nilai Awal Token (kWh)	Nilai Akhir Token (kWh)	Total Konsumsi Energi (kWh)
1	Sebelum Menggunakan Smart *Relay*	6.30	0.00	6.30
2	Sesudah Menggunakan Smart *Relay*	169.62	165.19	4.43

Berdasarkan hasil pengukuran yang dilakukan selama 6 hari, yaitu sebelum penggunaan smart relay pada tanggal 14–17 Juni 2025 dan sesudah penggunaan smart relay pada tanggal 17–20 Juni 2025, diperoleh perbedaan konsumsi energi listrik yang signifikan. Total konsumsi energi listrik sebelum menggunakan alat adalah sebesar 6,30 kWh, sedangkan setelah menggunakan alat tercatat sebesar 4,43 kWh, sehingga terdapat selisih konsumsi energi sebesar 1,87 kWh. Jika dihitung dalam bentuk persentase, penghematan energi listrik setelah penggunaan alat smart relay mencapai 29,68%. Hasil ini menunjukkan bahwa penerapan sistem smart relay berbasis IoT mampu membantu mengurangi konsumsi energi listrik secara otomatis dan lebih terkontrol dibandingkan dengan penggunaan manual.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem Smart *Relay* berbasis IoT menggunakan Esp32, maka diperoleh beberapa kesimpulan sebagai berikut:

a. Sistem berhasil dirancang dan diimplementasikan untuk memantau dan mengendalikan konsumsi energi listrik rumah tangga secara efisien. Dengan memanfaatkan koneksi internet melalui modul Esp32, pengguna dapat mengontrol perangkat listrik dari jarak jauh melalui aplikasi Blynk pada *smartphone* Android.
b. Fitur *timer* otomatis yang diintegrasikan ke dalam sistem memungkinkan pengguna untuk menjadwalkan waktu nyala dan mati perangkat secara mandiri, sehingga mendukung efisiensi energi serta mengurangi risiko lupa mematikan alat listrik.
c. Sistem dilengkapi pemantauan konsumsi energi secara real time, termasuk pembacaan parameter kelistrikan seperti tegangan, arus, dan daya. Seluruh data ditampilkan melalui oled dan aplikasi Blynk, serta dilengkapi notifikasi ketika konsumsi daya melebihi batas ambang yang telah ditentukan.
d. Manfaat dari sistem ini dirasakan baik oleh pengguna, peneliti, maupun institusi. Bagi pengguna, sistem ini memberikan kemudahan dalam mengelola konsumsi listrik dan menekan biaya tagihan. Bagi peneliti, menambah pengalaman dalam pengembangan IoT. Bagi STMIK

Palangkaraya, penelitian ini menjadi kontribusi nyata dalam mendorong teknologi inovatif yang aplikatif.

5.2 Saran

a. Disarankan agar sistem dilengkapi dengan fitur proteksi tambahan berupa pemutus arus otomatis *(circuit breaker)* yang akan aktif ketika terdeteksi arus berlebih. Hal ini bertujuan untuk mencegah kerusakan pada perangkat serta menghindari risiko korsleting listrik.
b. Sistem dapat dikembangkan lebih lanjut dengan menambahkan fitur pencatatan data historis penggunaan listrik. Dengan adanya riwayat data, pengguna dapat melihat pola konsumsi harian, mingguan, atau bulanan yang berguna untuk evaluasi dan pengelolaan daya listrik secara lebih baik.
c. Penggunaan energi alternatif seperti panel surya dapat menjadi pilihan yang ramah lingkungan sekaligus efisien dalam hal konsumsi daya. Dengan menambahkan modul solar charge controller dan baterai, sistem tetap dapat beroperasi secara mandiri meskipun terjadi gangguan atau pemadaman listrik dari PLN.
d. Karena sistem pemantauan dan pengendalian ini bekerja secara real-time melalui koneksi internet berbasis jaringan Wi-Fi, maka kestabilan serta kekuatan sinyal menjadi faktor penting. Oleh karena itu, disarankan agar perangkat dipasang pada lokasi yang memiliki jangkauan sinyal Wi-Fi yang kuat dan stabil, agar sistem dapat berjalan tanpa gangguan.

DAFTAR PUSTAKA

Asmazori, M. (2021) 'Rancang Bangun Alat Pendeteksi NOx dan CO Berbasis Mikrokontroler ESP32 dengan Notifikasi Via Telegram dan Suara', *JITCE (Journal of Information Technology and Computer Engineering)*, 5(02), pp. 57–62. Available at: https://doi.org/10.25077/jitce.5.02.57-62.2021.

Budiyanto, A., Pramudita, G.B. and Adinandra, S. (2020) 'Kontrol Relay dan Kecepatan Kipas Angin Direct Current (DC) dengan Sensor Suhu LM35 Berbasis Internet of Things (IoT)', *Techné : Jurnal Ilmiah Elektroteknika*, 19(01), pp. 43–54. Available at: https://doi.org/10.31358/techne.v19i01.224.

Dwi, N., Putri, I. and Surabaya, U.M. (2023) 'Manfaat Internet Of Things dan Potensinya dalam Sektor Logistik dan Penjelasan tentang IOT dan Pengaruhnya terhadap Bisnis dan Perekonomian Manfaat Internet Of Things dan Potensinya dalam Sektor Logistik dan Transportasi Oleh : Neysa Dwi Ivana Putri Manaj', (March).

Firdausi, M.T. and Rohmah, R.N. (2023) 'Perancangan Sistem Otomatis Pengatur Suhu Dan Kelembaban Pada Kandang Jangkrik Di Daerah Masaran Sragen Berbasis Mikrokontroller Arduino', *Jurnal Cahaya Mandalika*, 3(2), pp. 254–269.

Giawa, F.A., Siambaton, Z. and Haramaini, T. (2024) 'E-Monitoring pada Alat Penyortiran Buah Jeruk Nipis Otomatis Berdasarkan Ukuran dan Jenis Warna Berbasis Internet of Things', *sudo Jurnal Teknik Informatika*, 3(1), pp. 41–59. Available at: https://doi.org/10.56211/sudo.v3i1.406.

Gunawan, R., Yusuf, A.M. and Nopitasari, L. (2021) 'Rancang Bangun Sistem Presensi Mahasiswa Dengan Menggunakan Qr Code Berbasis Android', *Elkom: Jurnal Elektronika dan Komputer*, 14(1), pp. 47–58.

Handayani, Y.S. and Kurniawan, A. (2020) 'Rancang Bangun Prototipe Pengendali Pintu Air Berbasis SMS (Short Message Service) Untuk Pengairan Sawah Menggunakan Arduino', *Jurnal Amplifier : Jurnal Ilmiah Bidang Teknik Elektro Dan Komputer*, 10(2), pp. 34–41. Available at: https://doi.org/10.33369/jamplifier.v10i2.15330.

Ibrahim, Ridyandhika Riza , Bekti Yulianti, S.M. (2022) 'Rancang Bangun Monitoring Pemakaian Arus Listrik Pln Berbasis Iot', *jurnal Teknologi Industri*, 11(1), pp. 43–51.

Kiswantono, A., Afianti, H. and Purwahyudi, B. (2024) 'Proteksi Tegangan Berbasis IoT : Sistem Monitoring Cerdas dan Responsif', pp. 43–48.

Kiswantono, A. and S, Y.A. (2024) 'Pengukuran Energi Listrik dengan Modul Single on Circuit (SOC)', 1(3).

Meisak, D., Hendri and Agustini, S.R. (2022) 'Penerapan Metode Prototype Pada

Perancangan Sistem Informasi Penjualan Mediatama Solusindo Jambi', *STORAGE: Jurnal Ilmiah Teknik dan Ilmu Komputer*, 1(4), pp. 1–11. Available at: https://doi.org/10.55123/storage.v1i4.1066.

Oktodinata, W. and Purnomo, Y. (2024) 'Perangkat Jam Portabel dengan Fungsi Pembaca Suhu dan Pelacakan Suara Melalui Buzzer Menggunakan Modul Nrf Berbasis Arduino', *Jurnal Teknik Indonesia*, 3(7), pp. 115–128. Available at: https://doi.org/10.58860/jti.v3i7.446.

Rahmad Timor, A. and Kesuma, D. (2023) 'Alat Pengukur Denyut Nadi Dengan Tampilan OLED Berbasis Arduino', *Juni*, 2(1), pp. 92–97.

Shelemo, A.A. (2023) 'Monitoring Dan Pengendalian Beban Listrik Berbasis Internet Of Things (Iot) Menggunakan Esp32', *Jurnal Elektro*, 13(1), pp. 104–116.

Surahmat, A. (2023) 'Rancang Bangun Aplikasi Sistem Penjualan Pada Percetakan Cubic Art', *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(1), pp. 81–86. Available at: https://doi.org/10.36040/jati.v7i1.6064.

Suriana, I.W., Setiawan, I.G.A. and Graha, I.M.S. (2022) 'Rancang Bangun Sistem Pengaman Kotak Dana Punia berbasis Mikrokontroler NodeMCU ESP32 dan Aplikasi Telegram', *Jurnal Ilmiah Telsinas Elektro, Sipil dan Teknik Informasi*, 4(2), pp. 75–84. Available at: https://doi.org/10.38043/telsinas.v4i2.3198.

Sutarti, S., Triyatna, T. and Ardiansyah, S. (2022) 'Prototype Sistem Absensi Siswa/I Dengan Menggunakan Sensor Rfid Berbasis Arduino Uno', *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer*, 9(1), pp. 76–85. Available at: https://doi.org/10.30656/prosisko.v9i1.4744.

Sutiari, D.K. (2024) 'Prototipe Pengontrolan Nyala Dan Padamnya Lampu Berbasis Iot (Internet of Things)', 1, pp. 1–14.

Tri, N. *et al.* (2023) 'Perancangan Sistem Monitoring Ketersediaan Air Otomatis Menggunakan Applikasi Blynk Berbasis Internet of Things (IoT)', *Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI)*, 6, pp. 154–164.

Tri Sulistyorini, Nelly Sofi and Erma Sova (2022) 'Pemanfaatan Nodemcu Esp8266 Berbasis Android (Blynk) Sebagai Alat Alat Mematikan Dan Menghidupkan Lampu', *Jurnal Ilmiah Teknik*, 1(3), pp. 40–53. Available at: https://doi.org/10.56127/juit.v1i3.334.

Yusri, M., Maulita, Y. and Sembiring, H. (2024) 'Penerapan IoT dalam Monitoring dan Pengendalian Kualitas Air', 2(4).

Zahwa, M.A. *et al.* (2022) 'Adaptor Mesin Pencacah Sampah Plastik', *Community Services and Social Work Bulletin*, 1(1), p. 39. Available at: https://doi.org/10.31000/cswb.v1i1.5730.

L A M P I R A N

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangka Raya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

SURAT TUGAS

No.258/STMIK-3.C.2/KP/IX/2024

Ketua Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya menugaskan nama- nama tersebut di bawah ini :

1. Nama : Herkules, S.Kom., M.Cs.
   NIK : 198510042010106
   Sebagai Pembimbing I Dalam Pembuatan Program
2. Nama : Suparno, M.Kom.
   NIK : 196901041995105
   Sebagai Pembimbing II Dalam Penulisan Tugas Akhir

Untuk membimbing Tugas Akhir mahasiswa :

Nama	:	Aldya Saputra
NIM	:	C2155201085
Program Studi	:	TEKNIK INFORMATIKA (55201)
Tanggal Daftar	:	30 Agustus 2024
Judul Tugas Akhir	:	Rancang Bangun Smart Relay Berbasis IoT Menggunakan ESP32 Untuk Pengendalian Konsumsi Energi Listrik Rumah Tangga

Demikian surat ini dibuat agar dapat dipergunakan sebagaimana mestinya dan dilaksanakan dengan penuh tanggung jawab.

Palangka Raya, 10 September 2024
Ketua Program Studi Teknik Informatika,

Lili Rusdiana, M.Kom.
NIK. 198707282011007

Tembusan :
1. Pembimbing I dan II
2. Mahasiswa yang bersangkutan
3. Arsip

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangkaraya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

KARTU KEGIATAN KONSULTASI
TUGAS AKHIR

Nama Mahasiswa : ALDYA SAPUTRA
NIM : C2155206085
Tanggal Persetujuan Judul : .....
Judul Tugas Akhir : RANCANG BANGUN SMART RELAY BERBASIS IOT MENGGUNAKAN ESP 32 UNTUK PENGENDALIAN KONSUMSI ENERGI LISTRIK RUMAH TANGGA

No.	Tanggal Konsultasi		Uraian	Tanda Tangan
	Terima	Kembali		
1	16/11 24	10/12 24	- Kutipan dibuat paling sedikit dari dua ahli/sumber, diberi nasri dari sendiri	
			- Aturan penulisan disesuaikan dgn pedoman	
2	31/12 24	31/12 24	- Penyesuaian penulisan judul Obj dan Tabel, No. Halam sesuai pedoman	
3	14/01 25	14/01 25	- Sub anak sub bab yg ada dibawah paling bawah di pindah ke halaman selanjutnya.	
4		11/12 24	latar belakang dan Rumusan di perbaiki	
5		15/1 25	Analisis kebutuhan Alat dan Data listrik di perjelas	
6		17/1 25	Acc Seminar	
7		18/1 25	Acc Seminar	
8	050625	050625	- Layout tabel disesuaikan agar lebih rapi	
			- Kesimpulan harus menjawab pernyataan tujuan dan manfaat yg dinyatakan	
9	110625	110625	- Teknik penulisan selesai, lanjut ke Pembimbing I	

No.	Tanggal Konsultasi		Uraian	Tanda Tangan
	Terima	Kembali		
		20/05	Bab III desain interface & uji coba	
			Bab IV foto prototype & hasil implementasi,	
			uji coba ekspresi listrik pakai Smart Relay	
		27/05	Halaman kelengkapan ikuti laporan,	
			perbaiki penulisan	
		11/06	Acc Sidang	
		13/06/25	Acc Sidang	

Menyetujui :

Dosen Pembimbing I,

Hafrizal

Dosen Pembimbing II,

Supriyono

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
STMIK PALANGKARAYA
Jl. G. Obos No. 114 ~ Telp. 0563-3224593 ~ Fax. 0536-3225515 Palangka Raya
Posel: humas@stmikplk.ac.id ~ Laman: www.stmikplk.ac.id

Nomor : 079/STMIK-02/I.AX/XII/2024
Lampiran : -
Perihal : Permohonan Izin Penelitian dan Pengumpulan Data untuk Tugas Akhir

Kepada
Yth. **Pemilik Barak Rumah Tangga**
Jl. Temanggung Tilung 22, Kelurahan Menteng, Kota Palangka Raya

Dengan hormat,

Sehubungan dengan penyusunan Tugas Akhir mahasiswa sebagai persyaratan kelulusan Program Studi Teknik Informatika (S1) pada STMIK Palangkaraya, maka dengan ini kami sampaikan permohonan izin penelitian dan pengumpulan data bagi mahasiswa kami berikut:

Nama	:	Aldya Saputra
NIM	:	C2155201085
Prodi (Jenjang)	:	Teknik Informatika (S1)
Thn. Akad. (Semester)	:	2024/2025 (7)
Lama Penelitian	:	01 Oktober 2024 s.d 01 November 2024
Tempat Penelitian	:	Barak Rumah Tangga, Jl. Temanggung Tilung 22, Kelurahan Menteng, Kota Palangka Raya, Provinsi Kalimantan Tengah

Dengan judul Tugas Akhir:

RANCANG BANGUN SMART RELAY BERBASIS IOT MENGGUNAKAN ESP32 UNTUK PENGENDALIAN KONSUMSI ENERGI LISTRIK RUMAH TANGGA

Adapun ketentuan dan aturan pemberian informasi dan data yang diperlukan dalam penelitian tersebut menyesuaikan dengan ketentuan/peraturan pada instansi Bapak/Ibu.

Demikian permohonan ini disampaikan, atas perhatian dan kerja samanya diucapkan terima kasih.

Palangka Raya, 01 Oktober 2024

Ketua,

Suparno, M.Kom.
NIK. 196901041995105

SURAT KETERANGAN TELAH MELAKUKAN PENELITIAN

Yang bertanda tangan di bawah ini:

Nama	: Resiana
Sebagai	: Pemilik Barak
Alamat	: Jl. Temanggung Tilung 22, Kelurahan Menteng, Kota Palangka Raya

Dengan ini menerangkan bahwa:

Nama	: Aldya Saputra
NIM	: C2155201085
Program Studi	: Teknik Informatika
Universitas	: STMIK Palangkaraya

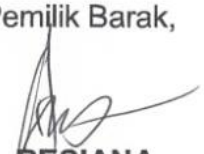
Telah melakukan penelitian dengan judul **"Rancang Bangun Smart Relay Berbasis IoT Menggunakan ESP32 untuk Pengendalian Konsumsi Energi Listrik Rumah Tangga"** pada lokasi penelitian di Jl. Temanggung Tilung 22, Kelurahan Menteng, Kota Palangka Raya dari tanggal 01 oktober 2024 hingga 01 november 2024.

Penelitian ini dilakukan dalam rangka penyusunan Tugas Akhir/Skripsi guna memenuhi salah satu syarat akademik dalam menyelesaikan studi di STMIK Palangkaraya.

Demikian surat keterangan ini dibuat untuk dapat digunakan sebagaimana mestinya.

Palangka Raya, 01 Oktober 2024

Pemilik Barak,



RESIANA

LEMBAR WAWANCARA

Judul Tugas Akhir:

Rancang Bangun Smart Relay Berbasis IoT Menggunakan ESP32 untuk Pengendalian Konsumsi Energi Listrik Rumah Tangga

Identitas Narasumber:

Nama: ...Resiana.....

Alamat: ...Jl. Temanggung Tilung XXII

Nomor HP: ...0857-5191-2796

Tanggal Wawancara: ...10 Oktober 2024

Pertanyaan Wawancara dan Jawaban:

1. Apakah Anda menggunakan perangkat otomatisasi rumah (smart home) saat ini? Jika ya, perangkat apa saja yang digunakan?
    - Sebelumnya, rumah belum dipasang teknologi IoT.
2. Bagaimana pola penggunaan listrik di rumah Anda sehari-hari?
    - Penggunaan listrik bervariasi tergantung kebutuhan, dengan konsumsi terbesar pada malam hari.
3. Apakah Anda mengalami kendala dalam mengontrol penggunaan listrik di rumah? Jika ya, kendala apa saja yang sering terjadi?
    - Ya, sering lupa mematikan perangkat listrik dan sulit memantau konsumsi listrik secara real-time.
4. Apakah Anda mengetahui tentang teknologi smart relay berbasis IoT?
    - Sebelumnya belum mengetahui, tetapi tertarik untuk mempelajari lebih lanjut.
5. Jika tersedia, apakah Anda tertarik menggunakan perangkat smart relay untuk mengontrol konsumsi listrik di rumah? Mengapa?
    - Ya, karena dapat membantu menghemat listrik dan memudahkan pengendalian perangkat listrik.
6. Fitur apa saja yang Anda harapkan dari perangkat smart relay berbasis IoT?
    - Pengendalian jarak jauh melalui aplikasi, pemantauan konsumsi listrik, dan pengaturan otomatis.
7. Bagaimana pendapat Anda tentang pengendalian listrik jarak jauh menggunakan aplikasi di smartphone?

- Sangat menarik dan membantu dalam mengontrol listrik tanpa harus berada di rumah.

8. Menurut Anda, apakah perangkat ini dapat membantu menghemat konsumsi listrik rumah tangga? Mengapa?
    - Ya, karena dapat menghindari pemborosan listrik dengan pengendalian otomatis.
9. Apakah Anda memiliki saran atau masukan terkait perangkat ini agar lebih sesuai dengan kebutuhan pengguna?
    - Sebaiknya memiliki fitur notifikasi jika konsumsi listrik melebihi batas yang ditentukan.

Catatan Tambahan:

.....

.....

Terima kasih atas waktu dan kesediaan Anda untuk mengikuti wawancara ini.

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangka Raya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

SURAT TUGAS
PENGUJI SEMINAR PROPOSAL TUGAS AKHIR

No.14/STMIK-3.C.2/KP/I/2025

Ketua Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya menugaskan kepada nama-nama berikut :

1. Nama : Amaya Andri Damaini, S.Kom., M.T
   NIK : 199309042018001
   Sebagai Ketua
2. Nama : Herkules, S.Kom., M.Cs.
   NIK : 198510042010106
   Sebagai Sekretaris
3. Nama : Suparno, M.Kom.
   NIK : 196901041995105
   Sebagai Anggota

Tim Penguji Seminar Proposal Tugas Akhir :

Nama	:	Aldya Saputra
NIM	:	C2155201085
Hari/Tanggal	:	Selasa, 21 Januari 2025
Waktu	:	11.00 WIB
Judul Proposal	:	Rancang Bangun Smart Relay Berbasis IoT Menggunakan ESP32 Untuk Pengendalian Konsumsi Energi Listrik Rumah Tangga

Demikian surat ini dibuat agar dapat dipergunakan sebagaimana mestinya dan dilaksanakan dengan penuh tanggung jawab.

Palangka Raya, 18 Januari 2025
Ketua Program Studi Teknik Informatika

Lili Rusdiana, M.Kom
NIK. 198707282011007

Tembusan :
1. Dosen Penguji
2. Mahasiswa yang Bersangkutan
3. Arsip Prodi

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangka Raya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

BERITA ACARA
SEMINAR PROPOSAL TUGAS AKHIR

Periode (Bulan) : Januari Tahun 2025

1. Hari/Tanggal Seminar : Selasa / 21/1/25
2. Waktu (Jam) : 11.00 WIB sampai dengan ..... WIB
3. Nama Mahasiswa : Acha Saputra
4. Nomor Induk Mahasiswa : C2185201085
5. Program Studi : Teknik Informatika
6. Tahun Angkatan : 2021
7. Judul Tugas Akhir : Rancang bangun Smart Relay Berbasis IoT menggunakan ESP32 untuk Pengendalian konservasi energi Contoh Rumah tangga
8. Dosen Penguji :

	Nama		Nilai		Tanda Tangan
1.	Amaya A.D.	=	.....	(	.....)
2.	Herlale	=	.....	(	.....)
3.	Suparno	=	.....	(	.....)

9. Hasil Ujian : **LULUS** / ~~**TIDAK LULUS** *~~) NILAI = 73,35
   Dengan Perbaikan/ Tanpa Perbaikan *)
10. Catatan Penting :
    1. Lama Perbaikan : 7 hari (Maks. 15 hari)
    2. Jika lebih dari 15 hari s/d 1 (satu) bulan dikenakan sanksi berupa denda sebesar Rp. 300.000,- (Tiga ratus ribu rupiah), dan jika lebih dari 1 (satu) bulan dikenakan denda Rp. 600.000,- (Enam Ratus ribu rupiah) per bulan dari tanggal ujian
    3. Jika lebih dari 3 (tiga) bulan dari tanggal ujian maka hasil ujian dibatalkan dan wajib mengajukan judul dan pembimbing baru. Wajib membayar Denda dan membayar biaya seminar ulang.

Palangka Raya, 21 Januari 2025

Mengetahui :
Ketua Program Studi Teknik Informatika,

Lili Rusdiana, M.Kom.
NIK. 198707282011007

Ketua Penguji,

Amaya Andri D. MT
NIK. 199.30.904201.8001

Tembusan :
1. Arsip Prodi Teknik Informatika
2. Mahasiswa yang bersangkutan

Dibawa saat konsultasi perbaikan dengan dosen penguji

**) Coret yang tidak perlu*

DAFTAR HADIR PESERTA
SEMINAR PROPOSAL TUGAS AKHIR

1. Nama Penyaji : ALDYA SAPUTRA
2. Hari/ Tanggal : Selasa / 21 / 1 / 2025
3. Waktu : 10.00 WIB
4. Judul Proposal : Rancang Bangun Smart Relay Berbasis IOT menggunakan ESP32 Untuk pengendalian Konsumsi Energi Listrik Rumah Tangga

No.	Nama Mahasiswa	NIM	Tanda Tangan
1	Taufikur Rahman	C2155201086	
2	AMRULLAH	C2155201113	
3	WILDAN HAMDANI MAHMUDA	C2155201098	
4	AHMAD HOSSAM	C2155201105	
5	Nazar	C2155201092	
6	IMANUEL NATALIS T.	C2155201092	
7	Jeki	C2155201001	
8	Rendi Febrianto	C2157201055	
9	Feri Purwanto	C2355201022	
10	Sanjiwo	C2157201085	
11	Lola	C2157201005	
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			

Palangka Raya, 21 Januari 2025

Mengetahui :
Ketua Tim Penguji,

AMALYA ANDRI O.

Mahasiswa Penyaji,

ALDYA SAPUTRA

LEMBAR PENGUJIAN BLACK BOX

Uji Coba Prototype Pengendalian Energi Listrik Pada Rumah Tangga

No	Pengujian	Keterangan	Hasil
1	Mengoneksikan Board Mikrokontroler ESP32 ke Laptop	Board ESP32 dapat terkoneksi dengan baik ke laptop dan terdeteksi di perangkat.	✓
2	Uji Coba Upload Kode Program ke Mikrokontroler ESP32	Proses upload program berjalan lancar tanpa ada error.	✓
3	Uji Coba Menghubungkan Mikrokontroler ESP32 ke Jaringan Wi-Fi	Mikrokontroler berhasil terhubung ke jaringan Wi-Fi tanpa masalah.	✓
4	Uji Coba *Relay*	Semua *relay* dapat berfungsi dengan baik sesuai perintah yang diberikan.	✓
5	Uji Coba Sensor Arus	Sensor arus dapat mendeteksi dan membaca arus listrik dengan akurat.	✓
6	Uji Coba Modul Step Down DC	Modul step down DC bekerja dengan baik dalam menurunkan dan menstabilkan tegangan.	✓

Palangkaraya, 6 Juni 2025
Penguji Black Box

Abdul Hadi, S.T., M.Kom.
NIK. 198505102021101