



**REAL TIME SYSTEM AND INTERNET OF THINGS FINAL PROJECT REPORT  
DEPARTMENT OF ELECTRICAL ENGINEERING  
UNIVERSITAS INDONESIA**

**AUTOMATIC OUTLET**

**GROUP B2**

<b>Ahmad Rifqi Fadhlurrahman</b>	<b>2106731301</b>
<b>Zaki Ananda</b>	<b>2106705474</b>
<b>Alifya Zhafira Ananda</b>	<b>2106704111</b>
<b>Cecilia Inez Reva Manurung</b>	<b>2106636994</b>

## KATA PENGANTAR

Dengan rendah hati dan rasa syukur yang mendalam, kami ingin menyampaikan puji kepada Tuhan Yang Maha Esa atas anugerah-Nya, kami berhasil menyelesaikan laporan proyek akhir dalam rangka mata kuliah Internet of Things yang berjudul "Automatic Outlet" dengan tepat waktu. Laporan ini bertujuan untuk memberikan pemahaman kepada pembaca tentang penerapan sistem real-time dalam solusi inovatif yang secara signifikan memengaruhi kehidupan sehari-hari.

Proyek ini bukan hanya implementasi dari pengetahuan dan keterampilan yang kami peroleh selama mengikuti mata kuliah Internet of Things, tetapi juga merupakan upaya kolaboratif tim yang melibatkan dukungan dari dosen pembimbing kami, F. Astha Ekadiyanto, S.T., M.Sc., serta kerja sama yang erat dengan asisten laboratorium dan teman-teman sejawat. Meskipun kami telah dengan seksama menyusun laporan ini, kami menyadari bahwa terdapat ruang untuk perbaikan dan penyempurnaan.

Dengan tulus, kami mengundang masukan dan saran konstruktif dari pembaca, karena kami percaya bahwa kritik konstruktif tersebut akan menjadi landasan bagi peningkatan kualitas laporan proyek akhir "Automatic Outlet" ini. Harapannya, proyek ini tidak hanya menjadi pencapaian akademis, tetapi juga menjadi sumber inspirasi untuk proyek-proyek di masa depan dalam bidang teknologi smart home. Terima kasih atas perhatian dan waktu yang telah diberikan.

Depok, December 05, 2023

Group B2

## TABLE OF CONTENTS

<b>CHAPTER 1.....</b>	<b>4</b>
<b>INTRODUCTION.....</b>	<b>4</b>
1.1    PROBLEM STATEMENT.....	4
1.3    ACCEPTANCE CRITERIA.....	5
1.4    ROLES AND RESPONSIBILITIES.....	5
1.5    TIMELINE AND MILESTONES.....	5
<b>CHAPTER 2.....</b>	<b>7</b>
<b>IMPLEMENTATION.....</b>	<b>7</b>
2.1    HARDWARE DESIGN AND SCHEMATIC.....	7
2.2    SOFTWARE DEVELOPMENT.....	7
2.3    HARDWARE AND SOFTWARE INTEGRATION.....	8
<b>CHAPTER 3.....</b>	<b>9</b>
<b>TESTING AND EVALUATION.....</b>	<b>9</b>
3.1    TESTING.....	9
3.2    RESULT.....	9
3.3    EVALUATION.....	10
<b>CHAPTER 4.....</b>	<b>11</b>
<b>CONCLUSION.....</b>	<b>11</b>

# CHAPTER 1

## INTRODUCTION

### 1.1 PROBLEM STATEMENT

Automatic Outlet merupakan sebuah perangkat dalam sistem otomatisasi rumah yang mengoptimalkan platform Blynk untuk mengontrol dan memantau outlet adaptor. Tujuan utama dari proyek ini adalah memberikan pengguna kemampuan untuk mengelola perangkat listrik yang terhubung ke adaptor secara efisien melalui aplikasi Blynk pada smartphone. Dengan memanfaatkan komponen-komponen canggih seperti sensor PZEM-004T, relay modul, dan ESP32, proyek ini dirancang sebagai solusi yang dapat diakses dan ramah pengguna.

Proyek ini dibuat untuk menanggapi kebutuhan akan solusi yang dapat diakses dan ramah pengguna yang memungkinkan individu mengubah socket tradisional menjadi perangkat pintar yang terkoneksi dengan Internet of Things (IoT). Hal ini karena seiring dengan kemajuan teknologi, terjadi peningkatan permintaan terhadap solusi otomatisasi rumah yang efisien dan mudah digunakan. Stop kontak listrik tradisional kurang memiliki kemampuan untuk beradaptasi dengan kebutuhan yang terus berkembang di rumah-rumah modern. Ketergantungan pada kendali manual membatasi potensi integrasi yang mulus dengan teknologi modern, menghambat kemampuan untuk mengelola dan memantau perangkat listrik secara jarak jauh.

### 1.2 PROPOSED SOLUTION

Dengan ini, kelompok B2 membuat Automatic Outlet untuk mengatasi kesenjangan ini dengan menggunakan komponen yang mudah didapatkan seperti mikrokontroler ESP32, sensor PZEM-004T, relay modul, dan platform Blynk yang dapat disesuaikan dan terhubung dengan Internet of Things (IoT), berkontribusi pada evolusi teknologi *smart home*. Keberadaan sensor PZEM-004T memungkinkan pengukuran dan pemantauan konsumsi daya perangkat. Selain itu, relay modul berperan sebagai switch dan ESP32 berfungsi sebagai otak sistem, memungkinkan pengguna untuk mengatur jadwal penggunaan, menyalakan, dan mematikan perangkat dari jarak jauh. Selain itu, perangkat dilengkapi dengan layar OLED

untuk memberikan feedback kepada pengguna dan menampilkan informasi seperti nama outlet, status (nyala atau mati), konsumsi daya harian, dan konsumsi daya total. Pengguna juga dapat mengontrol outlet secara remote, termasuk mengatur timer untuk mengaktifkan atau mematikan outlet. Hal ini secara langsung mendukung konsep Internet of Things (IoT) dengan menyatukan perangkat rumah dalam ekosistem terkoneksi.

### 1.3 ACCEPTANCE CRITERIA

Pembuatan proyek akhir ini memenuhi kriteria-kriteria sebagai berikut:

1. Proyek automatic outlet berhasil jika outlet dapat melakukan pengontrolan perangkat elektronik secara remote dan melalui time-controller
2. Proyek automatic outlet memiliki desain fisik yang mirip dengan outlet adaptor
3. Keberhasilan proyek automatic outlet diukur dari kemampuan mengintegrasikan PZEM-004T sensor untuk memantau dan mengukur konsumsi daya perangkat terhubung.
4. Proses switching menggunakan relay dapat dioperasikan dengan baik dan perhitungan penggunaan daya dari outlet dilakukan secara akurat
5. Proyek automatic outlet berhasil jika outlet terhubung ke internet dan melalui aplikasi Blynk dapat menampilkan informasi-informasi dari proyek

### 1.4 ROLES AND RESPONSIBILITIES

Peran dan tanggung jawab yang diberikan kepada anggota kelompok adalah sebagai berikut:

Roles	Responsibilities	Person
Rangkaian	Bertugas menyusun rangkaian dan menjalankan proses penyolderan.	Zaki Ananda, Ahmad Rifqi Fadhlurrahman, Alifya Zhafira Ananda
Pembuatan kode Hardware	Bertanggung jawab untuk membuat kode untuk implementasi ESP32.	Ahmad Rifqi Fadhlurrahman, Zaki Ananda

Pembuatan Blynk	Bertanggung jawab untuk membuat <i>interface</i> pada Blynk	Ahmad Rifqi Fadhlurrahman, Cecilia Inez Reva
Menyiapkan Alat	Bertanggung jawab untuk membeli dan menyiapkan alat yang dibutuhkan	Zaki Ananda, Cecilia Inez Reva, Ahmad Rifqi Fadhlurrahman
Pembuatan Laporan Proyek Akhir	Bertanggung jawab untuk membuat laporan akhir	Cecilia Inez Reva, Alifya Zhafira Ananda
Membuat User Guidebook	Bertanggung jawab untuk membuat user guidebook	Alifya Zhafira Ananda
Presentasi	Bertanggung jawab untuk membuat presentasi Proyek Akhir	Zaki Ananda, Ahmad Rifqi Fadhlurrahman, Alifya Zhafira Ananda, Cecilia Inez Reva

Table 1. Roles and Responsibilities

## 1.5 TIMELINE AND MILESTONES

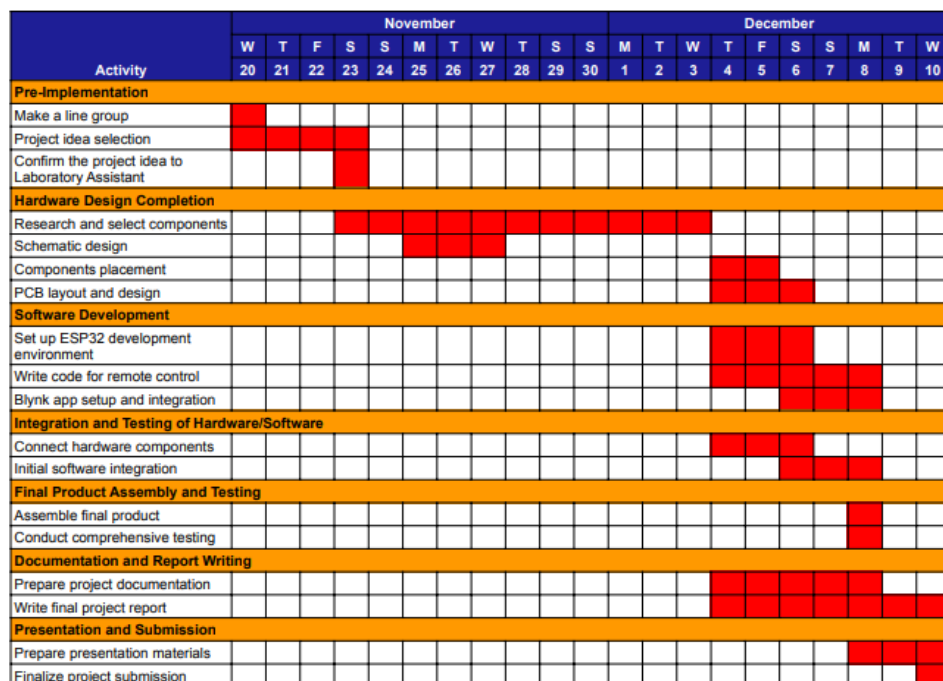


Table 2. Gantt Chart

## CHAPTER 2

## IMPLEMENTATION

### 2.1 HARDWARE DESIGN AND SCHEMATIC

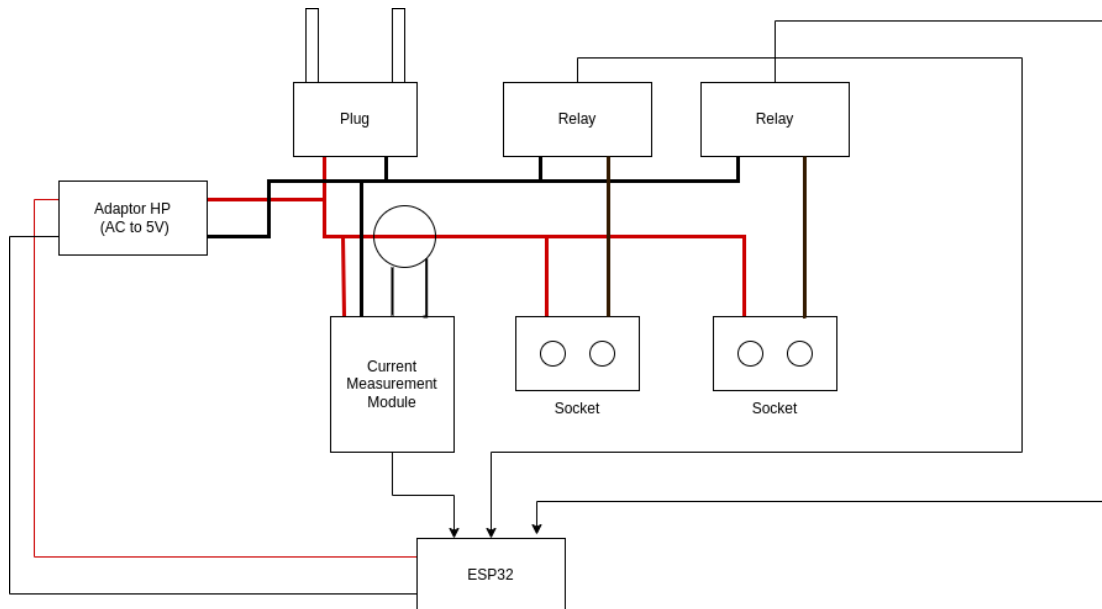


Fig 1. Skematik Rangkaian

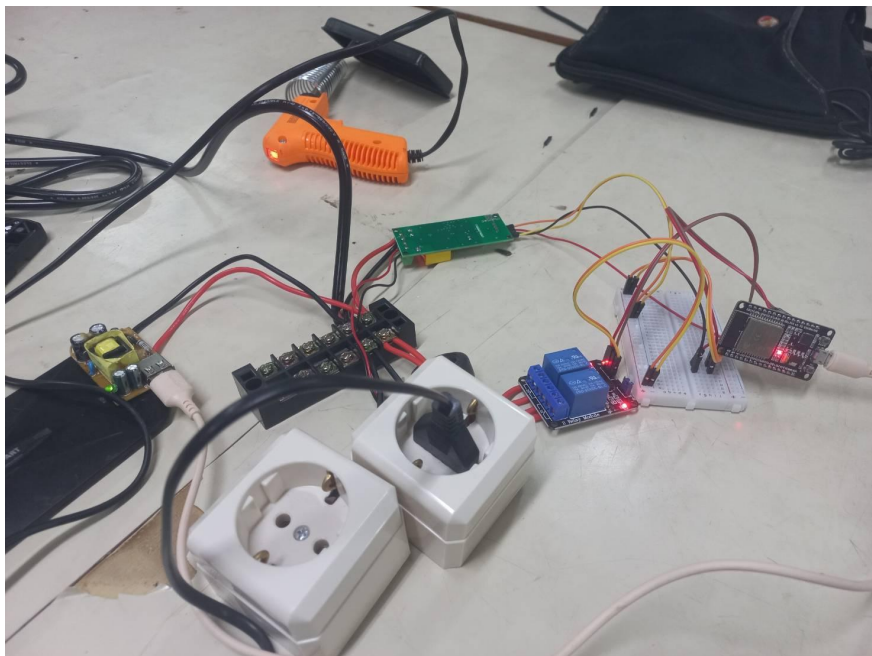


Fig 2. Rangkaian

Perancangan perangkat keras dimulai dengan pembuatan prototipe sebagai langkah awal, hal ini bertujuan untuk memudahkan kelompok dalam menyusun perangkaian alat. Proses ini melibatkan penyesuaian pin-pin yang akan digunakan dan komponen-komponen terlibat sebelum implementasi pada perangkat asli. Langkah ini membantu meminimalisir risiko kerusakan pada komponen asli akibat kesalahan rangkaian yang belum terintegrasi secara optimal.

Komponen yang dibutuhkan dalam merangkai skema alat serta fungsinya dalam rangkaian, antara lain adalah:

1. ESP32

ESP32 adalah mikrokontroler berbasis WiFi dan Bluetooth yang digunakan sebagai otak utama dalam proyek IoT ini. Fungsi ESP32 untuk mengelola logika pengendalian dan berkomunikasi dengan server Blynk untuk menerima instruksi dari pengguna. ESP32 juga berperan dalam mengontrol relay untuk memutus atau mengalirkan daya ke socket.

2. Relay modul

Relay adalah saklar elektronik yang dikendalikan oleh sinyal listrik. Dalam proyek ini, digunakan relay 2 channel untuk mengendalikan daya pada socket. Relay berfungsi sebagai switch yang dapat dikendalikan oleh ESP32. Ketika relay aktif, sirkuit listrik di socket terputus, mematikan daya untuk perangkat yang terhubung. Sebaliknya, ketika relay dimatikan, daya ke perangkat dinyalakan.

3. PZEM

PZEM-004T adalah sensor yang digunakan untuk mengukur parameter daya seperti tegangan, arus, daya, dan energi. PZEM-004T berperan dalam memonitor konsumsi daya perangkat yang terhubung ke socket pintar. PZEM memberikan informasi penting tentang penggunaan energi dan memungkinkan pengguna untuk melacak dan mengelola konsumsi daya perangkat.

4. Socket dan Plug

Socket adalah tempat di mana perangkat listrik dapat dihubungkan, sementara plug adalah konektor yang digunakan untuk menghubungkan perangkat ke socket. Socket



dan plug digunakan sebagai antarmuka fisik untuk menghubungkan perangkat listrik ke dalam sistem. Socket ini yang akan dikendalikan oleh relay.

## 5. Adaptor

Adaptor digunakan untuk memberikan daya ke ESP32, memastikan bahwa mikrokontroler dan komponen lainnya mendapatkan pasokan daya yang stabil untuk beroperasi.

## 2.2 SOFTWARE DEVELOPMENT

Pengembangan perangkat lunak difokuskan pada implementasi berbagai fitur yang menjadi kebutuhan utama. Mulai dari integrasi antar perangkat keras seperti ESP32, PZEM dan relay, hingga fungsionalitas dengan aplikasi Blynk.

Dalam proses pengembangan firmware ESP32, kami mengimplementasikan logika sistem menggunakan bahasa pemrograman C++. Firmware pada ESP32 bertanggung jawab atas kontrol keseluruhan, termasuk konfigurasi antarmuka Blynk, pemantauan parameter listrik dengan sensor PZEM004T (tegangan, arus, daya, dan energi), dan kontrol relay untuk mengatur status perangkat.

```
// Definisi Template Blynk dan Token Otentikasi
#define BLYNK_TEMPLATE_ID "TMPL6Z3pkF4mf"
#define BLYNK_TEMPLATE_NAME "Test"
#define BLYNK_AUTH_TOKEN "gPoFT9pxs7VIZdtQET7TV1_ORvA2b1Ko"

// Penggunaan Library dan Deklarasi Objek
#include <BlynkSimpleEsp32.h>
#include <PZEM004Tv30.h>
#include <WiFi.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <WiFiManager.h>

#ifdef ESP32
PZEM004Tv30 pzem(Serial2, 16, 17);
#else
PZEM004Tv30 pzem(Serial2);
#endif

// Definisi Pin Blynk dan Variabel Global
#define onOffPin V0
#define putHours V1
#define displayAV V2
#define displayWJ V3
#define putMin V4

SemaphoreHandle_t xSemaphore;
TimerHandle_t ntpUpdateTimer;
WiFiManager wifiManager;
const int relayPin = 26;
const int relayPin2 = 27;

static float current = 0;
static float voltage = 0;
static float power = 0;
static float energy = 0;
static float frequency = 0;
static float pf = 0;
```

Fig 3. Potongan Kode untuk Perangkat Keras

Kode di atas merupakan implementasi firmware untuk perangkat ESP32. Pertama, program melakukan inisialisasi template Blynk, token otentikasi, dan objek yang diperlukan seperti PZEM, WifiManager, NTPClient, serta relay. Deklarasi objek mencakup penggunaan PZEM004T untuk membaca data sensor dan WiFi Manager untuk mengelola koneksi WiFi. Objek PZEM004Tv30 diinisialisasi berdasarkan platform yang digunakan, yaitu ESP32 atau

bukan. Pada tugas 'setupWiFi()', dilakukan upaya koneksi WiFi menggunakan WiFi Manager. Jika berhasil, waktu klien NTP diinisialisasi untuk mendapatkan waktu saat ini.

```
// Konfigurasi NTP dan Inisialisasi Objek NTPClient
const char *ntpServer = "pool.ntp.org";
const long gmtOffset_sec = 3600 * 7;
const int daylightOffset_sec = 0;
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, ntpServer, gmtOffset_sec, daylightOffset_sec);
```

Fig 4. Potongan Kode untuk konfigurasi NTP dan objek NTPClient

Objek NTPClient diinisialisasi dengan menggunakan alamat server 'pool.ntp.org' melalui koneksi UDP menggunakan WiFiUDP. Penggunaan objek 'timeClient' memungkinkan perangkat ESP32 memperoleh dan memperbarui waktu dari server NTP. Tahapan ini dirancang untuk menjamin keakuratan waktu perangkat ESP32 serta untuk menyelaraskan waktu dengan standar global yang ditentukan oleh server NTP.

Fungsi Update NTP Time bertanggung jawab atas pembaruan waktu dan perbandingannya dengan waktu mati yang telah ditentukan sebelumnya. Jika waktu mati yang telah ditentukan tercapai, fungsi ini akan mengubah status perangkat menjadi aktif atau non-aktif sesuai kondisi yang ditentukan.

```
// Fungsi-fungsi Blynk Write
BLYNK_WRITE(onOffPin) {
  if (param.asInt() == 1) {
    if (!isOn) {
      isOn = true;
      Serial.println("BLYNK WRITE V0 : device on");
    } else {
      Serial.println("BLYNK WRITE V0 : cannot on");
    }
  } else {
    if (isOn) {
      isOn = false;
      Serial.println("BLYNK WRITE V0 : device off");
    } else {
      Serial.println("BLYNK WRITE V0 : cannot off");
    }
  }
}

BLYNK_WRITE(putHours) {
  // ...
}
```

Fig 5. Potongan Kode untuk Fungsi Blynk Write

Fungsi 'BLYNK\_WRITE(onOffPin)' dirancang untuk mengontrol status Automatic Outlet berdasarkan nilai input yang diterima dari pengguna pada pin Blynk yang terkait dengan kontrol daya (V0). Apabila nilai input adalah 1, program akan memeriksa apakah perangkat sudah dalam keadaan menyala ('isOn'). Jika belum, perangkat akan dihidupkan,

dan statusnya akan dicetak di Serial. Sebaliknya, jika nilai input adalah 0 dan perangkat dalam keadaan menyala, perangkat akan dimatikan, dan statusnya akan dicetak di Serial.

Selain itu, terdapat fungsi `BLYNK\_WRITE(putHours)` dan `BLYNK\_WRITE(putMin)` yang berfungsi menerima input waktu dari pengguna dan menyimpannya untuk pengaturan waktu pada perangkat. Fungsi-fungsi ini memungkinkan pengguna untuk menentukan jam (`putHours`) dan menit (`putMin`) pada aplikasi Blynk ketika perangkat akan dimatikan secara otomatis. Input waktu yang diterima akan disimpan dalam variabel yang sesuai untuk pengolahan selanjutnya dalam mengatur timer operasi perangkat.

```
// Fungsi Kontrol Relay
void relayCtr(void *pvParameters) {
    while (1) {
        if (isOn) {
            digitalWrite(relayPin, LOW);
            digitalWrite(relayPin2, LOW);
            Serial.println("relayCtr: device on");
        } else {
            digitalWrite(relayPin, HIGH);
            digitalWrite(relayPin2, HIGH);
            Serial.println("relayCtr: device off");
        }
    }
}
```

```
// Fungsi Membaca Sensor
void readSensor(void *pvParameters) {
    while (1) {
        if (isOn) {
            if (xSemaphoreTake(xSemaphore, portMAX_DELAY) == pdTRUE) {
                voltage = pzem.voltage();
                current = pzem.current();
                power = pzem.power();
                energy = pzem.energy();
                xSemaphoreGive(xSemaphore);
            }
        }
    }
}
```

Fig 6. Potongan Kode untuk Relay dan Sensor PZEM

Task relayCtr beroperasi secara kontinu dalam sebuah loop (while loop) untuk mengontrol status relay berdasarkan variabel isOn. Jika isOn bernilai true, relay akan diaktifkan (LOW), sedangkan jika isOn bernilai false, relay akan dimatikan (HIGH). Di sisi lain, task readSensor juga berjalan secara terus-menerus dan bertugas membaca data dari sensor PZEM-004T, mencakup informasi tegangan, arus, daya, dan energi. Proses membaca ini hanya dilakukan jika perangkat dalam kondisi aktif (isOn), dan hasilnya akan diteruskan ke dalam task `sendData` untuk dikirimkan ke server Blynk. Berikut adalah tampilan antarmuka yang diimplementasikan dalam aplikasi Blynk yang digunakan untuk mengontrol dan memantau perangkat.



Fig 7. Aplikasi Blynk

Untuk memahami lebih lanjut alur dari cara kerja alat, maka berikut adalah flowchart cara kerja “Automatic Outlet”:

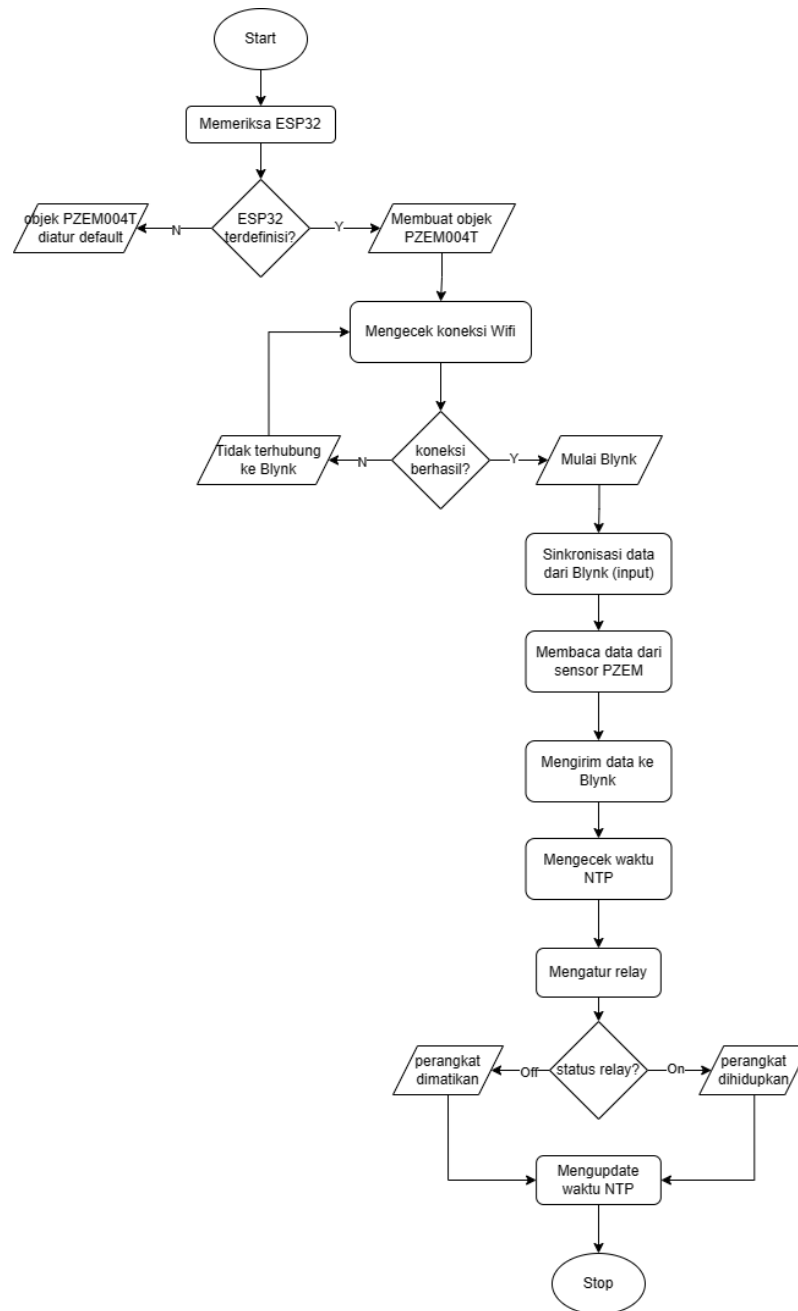


Fig 8. Flowchart

Sequence diagram juga dapat memberikan gambaran visual tentang interaksi antarmuka pengguna (UI) Blynk dengan perangkat keras. Diagram ini menunjukkan proses pengontrolan dan pemantauan outlet secara real-time, termasuk operasi manual, kontrol jarak jauh, dan pengaturan timer. Dengan menambahkan diagram ini, pemahaman tentang alur kerja sistem menjadi lebih jelas dan mendukung dokumentasi fungsionalitas utama Automatic Outlet.

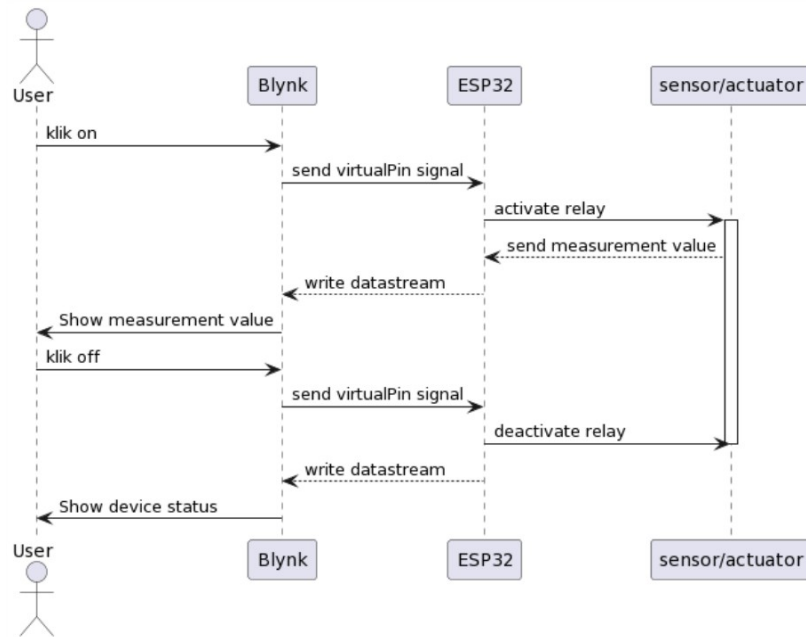


Fig 9. Sequence Diagram

State Diagram memberikan visualisasi yang jelas tentang transisi dan kondisi perangkat. Diagram status ini mencakup situasi seperti daya hidup, daya mati, dan mode otomatis. Dengan menggunakan simbol-simbol seperti lingkaran dan panah, diagram memberikan pemahaman instan tentang cara perangkat berinteraksi dengan pengguna dan lingkungannya.

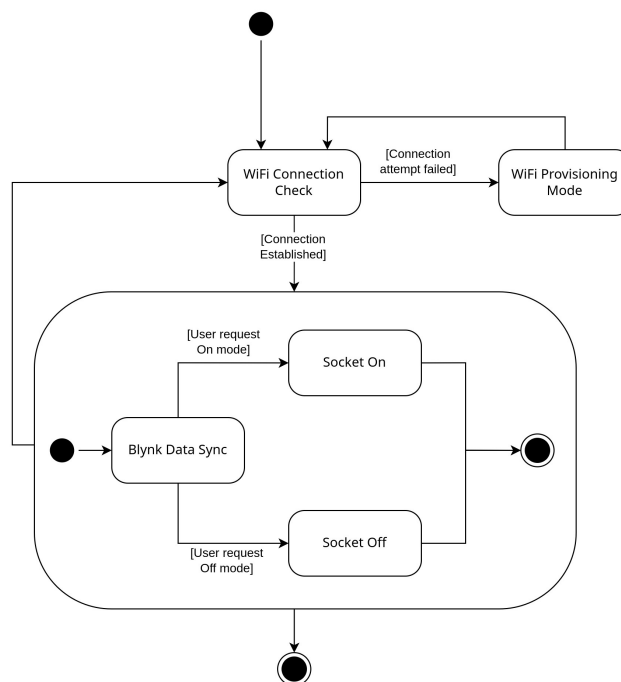


Fig 10. State Diagram

## 2.3 HARDWARE AND SOFTWARE INTEGRATION

Pengembangan software untuk proyek "Automatic Outlet" mengintegrasikan beberapa komponen kunci, seperti ESP32, relay 2 Channel, dan PZEM-004T Sensor. Pertama, software dirancang dengan platform Blynk untuk mengontrol dan memantau smart socket. Outlet ini memungkinkan pengguna untuk menyalakan dan mematikan perangkat elektronik secara remote atau melalui time-controller.

Perangkat ini menggunakan relay modul yang terhubung ke ESP32 untuk mengontrol aliran daya listrik ke perangkat tertentu. Relay berfungsi sebagai switch kontrol daya, memungkinkan outlet dimatikan atau dinyalakan baik secara langsung maupun melalui time-controller. Ketika relay aktif, sirkuit listrik di outlet terbuka, mematikan daya untuk perangkat yang terhubung. Sebaliknya, ketika relay dimatikan, daya ke perangkat dinyalakan. Selain itu, outlet dilengkapi dengan PZEM-004T sensor yang mengukur dan memonitor konsumsi daya perangkat yang terhubung ke automatic outlet. PZEM-004T dihubungkan ke sumber daya yang diukur, yaitu antara relay modul dan perangkat yang terhubung ke automatic outlet.

Dengan adanya layar OLED pada perangkat, pengguna dapat menerima *feedback* langsung terkait konsumsi daya dan status outlet. Koneksi internet memungkinkan outlet ini terhubung dan dimonitor melalui aplikasi Blynk. Aplikasi Blynk menampilkan informasi seperti nama outlet, status outlet (nyala atau mati), konsumsi daya harian, dan konsumsi daya kumulatif.

Setelah berhasil mengimplementasikan kode perangkat lunak pada ESP32, dilakukan pengujian pada *automatic outlet* untuk memverifikasi kesesuaian interaksi antara perangkat keras dan perangkat lunak. Proses pengujian melibatkan metode '*trial and error*', memungkinkan kami untuk melakukan penyesuaian pada kode sehingga memenuhi kriteria proyek dengan optimal.

## CHAPTER 3

### TESTING AND EVALUATION

#### 3.1 TESTING

Dalam fase pengujian sistem, kami melaksanakan serangkaian test untuk menegaskan bahwa program beroperasi sesuai dengan kriteria yang telah ditetapkan sebelumnya.

Saat dilakukan pengujian tanpa menggunakan *interface* Blynk, melalui serial monitor, smart outlet dapat berhasil dijalankan. Task serialTask membaca input dari Serial. Jika input adalah '1', perangkat dinyalakan (isOn diubah menjadi true), dan jika input adalah '0', perangkat dimatikan (isOn diubah menjadi false). Ketika input serial monitor adalah "1" maka relay modul menyala sehingga menyalakan smart outlet yang berhasil menyalakan perangkat yang terhubung. Pada kelompok kami, pengujian berhasil ditunjukkan seperti gambar berikut:

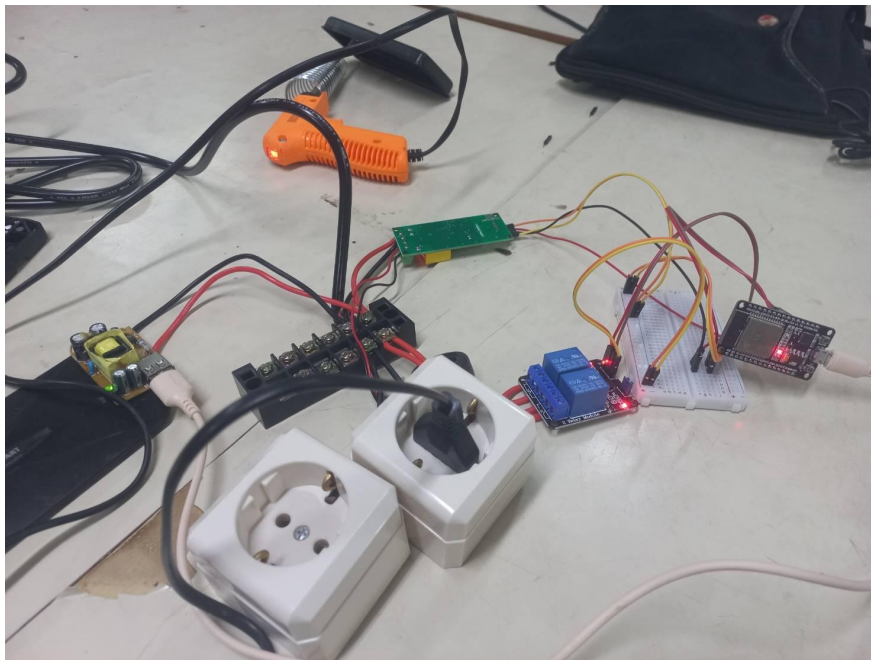


Fig 7. Hasil Testing tanpa menggunakan Blynk

Kemudian, dilakukan pengujian dengan menggunakan interface Blynk. Pada kelompok kami, pengujian berhasil ditunjukkan seperti gambar berikut:





Fig 8. Hasil Testing menggunakan Blynk



Fig 9. Hasil Testing menggunakan Blynk

Pada *interface* Blynk, bagian atas menampilkan status perangkat "on/off" untuk memberi informasi apakah ESP32 terhubung ke internet atau tidak. Sementara itu, bagian bawah menampilkan sebuah slider yang memungkinkan pengguna mengatur waktu untuk

mengaktifkan atau menonaktifkan outlet. Dengan kata lain, jika waktu yang diatur oleh pengguna telah tiba, misalnya pada tampilan bagian atas menunjukkan pukul 11.43, maka outlet akan beralih ke status yang berbeda. Jika sebelumnya berada dalam mode "off," outlet akan diaktifkan (on) pada waktu tersebut. Sebaliknya, jika sebelumnya dalam mode "on," outlet akan dinonaktifkan (off) pada waktu yang sama.

Dari hasil pengujian yang dilakukan, terdapat beberapa hal yang menjadi catatan penting. Diantaranya:

- a. Fungsi serialTask memberikan respons yang sesuai terhadap input dari Serial Monitor. Jika input adalah '1', perangkat berhasil dihidupkan (isOn = true), dan jika input adalah '0', perangkat berhasil dimatikan (isOn = false).
- b. Relay berhasil berperan sebagai switch untuk mengatur daya pada smart outlet. Pengujian menunjukkan bahwa melalui kontrol dari serial monitor atau input pengguna, relay mampu menentukan apakah smart outlet dihidupkan (relay aktif) atau dimatikan (relay non-aktif).
- c. Sensor PZEM-004T memberikan data yang akurat dan responsif. Pemantauan sensor hanya berlangsung saat perangkat dalam keadaan hidup (isOn = true), sesuai dengan logika program.
- d. Data konsumsi daya dan status perangkat berhasil dikirimkan ke server Blynk. Informasi yang ditampilkan di aplikasi Blynk sesuai dengan kondisi aktual perangkat.
- e. Fungsi-fungsi terkait waktu dan NTP berjalan dengan baik. Waktu sinkronisasi berfungsi sesuai harapan, dan perangkat dapat dimatikan secara otomatis sesuai dengan waktu yang telah diatur oleh pengguna melalui aplikasi Blynk.

### **3.2 RESULT**

Hasil dari pengujian diperoleh melalui eksekusi langsung pada komponen rangkaian asli dengan memanfaatkan platform Blynk. Dalam pengujian tersebut, fungsi dan respons sistem dievaluasi secara langsung untuk memastikan keberfungsan optimal dalam penggunaan aplikasi Blynk.

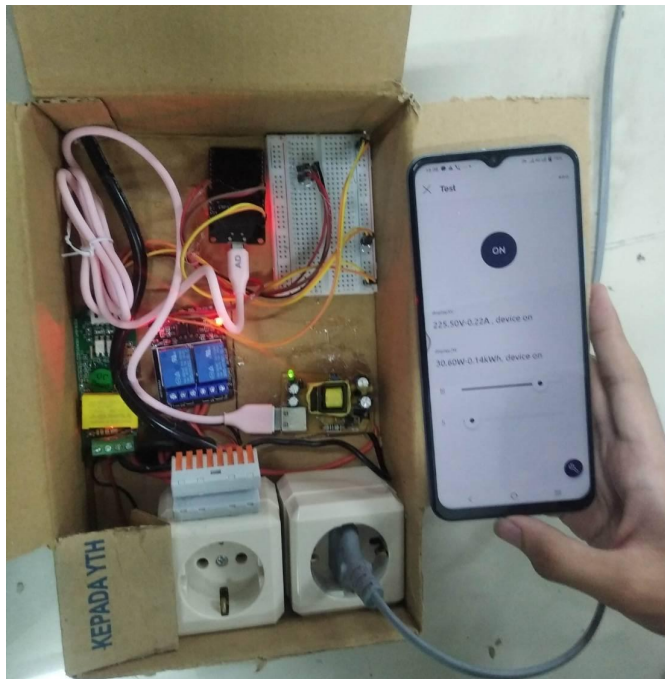


Fig 9. Hasil Testing

Berdasarkan hasil pengujian, proyek "Automatic Outlet" berhasil dalam mengintegrasikan perangkat keras dan perangkat lunak untuk menciptakan smart outlet yang responsif dan dapat dikontrol secara efisien. Keberhasilan implementasi relay, respons dari fungsi SerialTask, pemantauan sensor, pengiriman data ke Blynk, dan pengaturan waktu menunjukkan bahwa proyek ini mencapai tujuan fungsionalitas yang diinginkan. Evaluasi lebih lanjut dapat dilakukan untuk memperbaiki efisiensi atau menambah fitur tambahan sesuai dengan kebutuhan dan umpan balik pengguna.

### 3.3 EVALUATION

Evaluasi proyek "Automatic Outlet" mencakup analisis komprehensif terhadap integrasi perangkat keras dan perangkat lunak. Fase pengujian bertujuan untuk memvalidasi kinerja sistem terhadap kriteria yang telah ditetapkan sebelumnya, dan hasilnya mencerminkan keberhasilan dan fungsionalitas yang tercapai.

Kondisi pengujian melalui Serial Monitor berhasil dengan menunjukkan kemampuan sistem untuk mengoperasikan smart outlet. `serialTask` dengan efektif membaca input dari Serial Monitor, merespons dengan tepat terhadap '1' dengan menghidupkan perangkat (mengatur `isOn` menjadi true) dan terhadap '0' dengan mematikan perangkat (mengatur

`isOn` menjadi false). Modul relay berfungsi sesuai yang diinginkan, memungkinkan smart outlet dikontrol langsung melalui input pengguna.

Pengujian berikutnya dengan interface Blynk lebih lanjut untuk memverifikasi keberhasilan proyek, menunjukkan bahwa integrasi dengan platform Blynk efektif. Pengguna dapat mengontrol smart outlet secara remote, memantau konsumsi daya, dan menerima *feedback* waktu nyata mengenai status outlet melalui aplikasi Blynk. Berhasilnya pengiriman data yang efektif ke server Blynk, dan fungsionalitas terkait waktu yang dapat diandalkan menegaskan keberhasilan proyek secara keseluruhan. Pencapaian ini sesuai dengan tujuan proyek, menunjukkan smart outlet yang responsif dan dapat dikontrol dengan efisien.

Hasil evaluasi menunjukkan bahwa proyek "Automatic Outlet" telah mencapai tujuan fungsionalnya dan mendemonstrasikan integrasi yang efektif antara komponen perangkat keras dan perangkat lunak.

## **CHAPTER 4**

### **CONCLUSION**

Dengan mengimplementasikan mikrokontroler ESP32, relay 2 Channel, dan sensor PZEM-004T, proyek "Automatic Outlet" menyediakan solusi yang efisien untuk memantau dan mengontrol aliran daya listrik ke perangkat elektronik di dalam ruangan. Sistem ini beroperasi secara terus menerus, memeriksa konsumsi daya perangkat dan status relay untuk mengelola daya yang masuk. Ketika relay diaktifkan, outlet dapat dihidupkan, sedangkan ketika relay dimatikan, perangkat otomatis dimatikan. Integrasi sensor PZEM-004T memberikan pemantauan konsumsi daya yang akurat, memungkinkan pengguna untuk melacak penggunaan daya perangkat mereka.

Proyek "Automatic Outlet" menawarkan solusi yang responsif dan efektif. Relay berperan sebagai saklar kontrol daya yang dapat diandalkan, memungkinkan pengguna untuk mengelola perangkat mereka dengan mudah melalui antarmuka Blynk. Selain itu, sensor PZEM-004T memberikan data konsumsi daya yang dapat diakses melalui aplikasi Blynk, memberikan pengguna informasi real-time tentang penggunaan daya perangkat mereka.

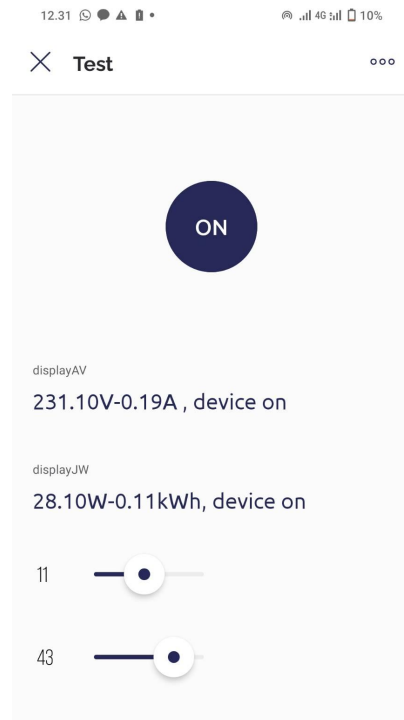
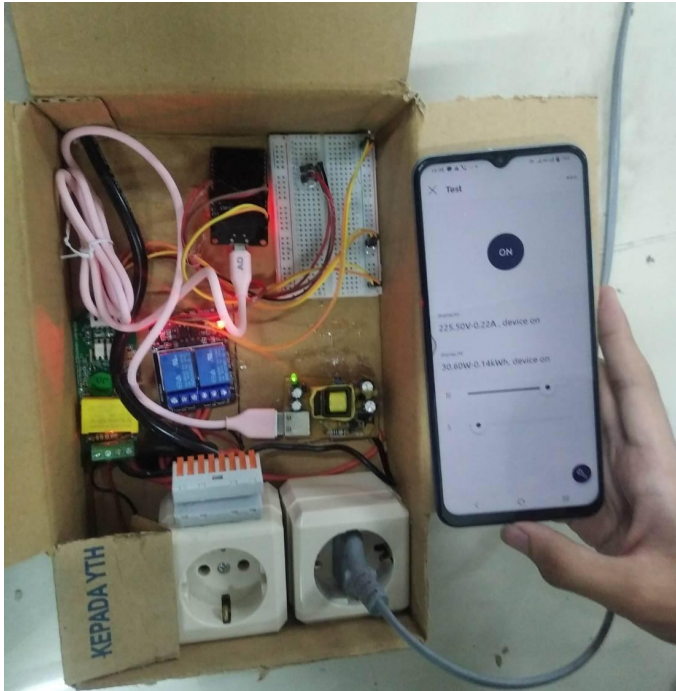
Dengan antarmuka Blynk, proyek ini memberikan kontrol jarak jauh dan pemantauan waktu nyata, memastikan kenyamanan dan keamanan pengguna. Meskipun proyek ini sederhana, integrasi yang baik antara perangkat keras dan perangkat lunak menciptakan solusi yang efisien dan mudah diakses untuk mengelola daya listrik di dalam ruangan.

## REFERENCES

- [1] “Smart Socket ESP8266 Blynk Home Automation Tutorial,” *Viral Science - The home of Creativity*. Sept, 13 2020.  
<https://www.youtube.com/watch?v=HGWTvZah3k4> [Accessed: Dec 10, 2023]
- [2] “The Dark Secret of Smart Sockets and How I Fixed It! (DIY or Buy),” *GreatScott!*. 2022. <https://www.youtube.com/watch?v=nIB3kWqqAzE> [Accessed: Dec 10, 2023]
- [3] Admin, “Getting Date and Time with ESP32 on Arduino IDE (NTP Client),” *Random Nerd Tutorials*,  
<https://randomnerdtutorials.com/esp32-ntp-client-date-time-arduino-ide/> [Accessed: Dec 10, 2023]
- [4] Admin, “ESP32 WiFiManager - Easy WiFi Provisioning,” *DroneBot Workshop*,  
<https://dronebotworkshop.com/wifimanager/> [Accessed: Dec 10, 2023]

## APPENDICES

### Appendix A: Project Schematic



### Appendix B: Documentation

