BAB I

PENDAHULUAN

A. Latar Belakang

Energi listrik sebagai sumber energi merupakan sebuah sarana yang dapat membantu meningkatkan kesejahteraan masyarakat. Energi listrik adalah kebutuhan penting bagi masyarakat dan merupakan sumber energi ekonomis yang dapat digunakan untuk berbagai kegiatan masyarakat. Hampir semua aktifitas manusia ditopang oleh adanya energi listrik. Pemanfaatan energi listrik didasarkan pada pasokan energi yang murah dan pemanfaatan energi yang lebih sederhana dibandingkan dengan energi lainnya. Energi listrik juga berperan penting dalam mendukung segala jenis kegiatan baik pada bagian industri, bangunan komersil, transportasi maupun rumah tangga.

Perusahaan Listrik Negara (PLN) merupakan Badan Usaha Milik Negara yang diamanatkan untuk mengurus, mengatur dan memenuhi kebutuhan listrik bagi rakyat Indonesia. Tenaga listrik memiliki peranan penting dan cukup strategis dalam mewujudkan tujuan pembangunan nasional, maka penyediaan tenaga listrik dikuasai dan dikelola oleh negara melalui Badan Usaha Milik Negara dan penyediaannya harus ditingkatkan, sejalan dengan perkembangan pembangunan agar ketersediaan tenaga listrik dalam jumlah yang cukup, merata dan bermutu dapat tercapai demi memenuhi kebutuhan tenaga listrik bagi rakyat Indonesia.

Seiring meningkatnya kebutuhan Masyarakat untuk daya listrik yang lebih besar dalam waktu tertentu, PT. PLN (Persero) Perusahaan Listrik Negara menyediakan layanan penyambungan sementara untuk kebutuhan tersebut. Adapun batasan daya penyambungan sementara 1 (satu) phasa yaitu daya 2200 VA ,3500 VA, 4400 VA, 5500 VA, 7700 VA, dan 11000 VA. Dengan adanya batasan daya pada layanan penyambungan sementara tersebut berarti PLN harus menyiapkan Alat pembatas dalam hal ini adalah MCB (*Miniature Circuit Breaker*) yang memiliki arus nominal yang berbeda-beda.

Penyambungan Sementara dapat diberikan kepada Pelanggan PLN maupun bukan pelanggan PLN. Penyambungan sementara dimaksudkan untuk memenuhi permintaan pasar yang semakin berkembang, antara lain adanya kebutuhan energi sesaat (*intermittern-usage,spot-demand*) tanpa harus terdaftar sebagai Konsumen. Penyambungan Sementara dilaksanakan dengan mekanisme sebagai berikut :

1. Pendaftaran;
2. Penerbitan Jawaban;
3. Pembayaran Biaya;
4. Pelaksanaan Penyambungan Sementara; dan
5. Pelaksanaan Pembongkaran Penyambungan Sementara.

Dalam pelaksanaan pembongkaran Penyambungan Sementara terkadang petugas datang terlambat kelokasi untuk melakukan penormalan. Hal-hal yang kadang membuat petugas datang terlambat kelokasi untuk penormalan Penyambungan sementara antara lain adalah lokasi yang jauh, waktu penormalan

yang bersamaan dengan Penyambungan Sementara yang lain, Kendaraan petugas bermasalah, petugas tidak masuk karena sakit.

Pemasangan penyambungan sementara tidak menggunakan kWh (*kilo Watt hour)* meter sehingga besarnya pemakaian energi listrik pada masa penyambungan sementara tidak diketahui. Dikarenakan Penyambungan Sementara memerlukan MCB (*Miniature Circuit Breaker*) yang berbeda-beda setiap dayanya, pelaksanaan penormalan penyambungan sementara yang tidak tepat waktu, dan tidak diketahuinya besar pemakaian energi listrik dari penyambungan sementara menjadi awal munculnya ide sistem kendali dan monitoring penyambungan sementara PLN berbasis *mikrokontroler ESP32*.

Dengan 1 (satu) alat kendali dan *monitoring* Penyambungan Sementara tersebut dapat digunakan untuk semua batas daya penyambungan sementara 1(satu) phasa. Dengan alat kendali dan monitoring Penyambungan Sementara tersebut batas waktu penyambungan dapat diatur sehingga kWh (*kilo Watt hour*) meter dirumah Pelanggan dapat mengukur kembali seperti semula apabila telah mencapai batas waktu penormalan. Dengan alat kendali dan *monitoring* Penyambungan Sementara tersebut besarnya arus listrik yang mengalir dan besarnya pemakaian energi listrik pada periode penyambungan sementara dapat ditampilkan melalui LCD (*Liquid Kristal Display*) sehingga besarnya arus dan energi listrik dapat diketahui.

B. Rumusan Masalah

Berdasarkan indentifikasi permasalahan yang telah diuraikan pada latar belakang, mendapatkan penyelesaian masalah sebagai berikut :

1. Bagaimana cara membuat pembatas dan pemutus Arus listrik Digital sebagai pengganti MCB (*Miniature Circuit Breaker*) berbasis *Mikrokontroler ESP32*?
2. Bagaimana cara merancang sistem kendali pengaturan batas Arus nominal dan waktu penormalan layanan Penyambungan Sementara PLN berbasis *Mikrokontroler ESP32*?
3. Bagaimana cara merancang sistem *monitoring* pemakaian tenaga listrik berbasis *Mikrokontroler ESP32*?

C. Tujuan Penelitian

Tujuan penelitian berdasarkan rumusan masalah yang telah diuraikan diatas dan hendaknya dapat dicapai dalam pembuatan tugas akhir ini yaitu :

1. Untuk mengetahui cara membuat pembatas dan pemutus Arus listrik digital sebagai pengganti MCB (*Miniature Circuit Breaker*) berbasis *Mikrokontroler ESP32*.
2. Untuk mengetahui cara merancang sistem kendali pengaturan batas Arus listrik dan waktu penormalan layanan Penyambungan Sementara berbasis *Mikrokontroler* ESP32.
3. Untuk mengetahui cara merancang sistem *monitoring* pemakaian tenaga listrik berbasis *Mikrokontroler* ESP32.

D. Batasan Masalah

Untuk memfokuskan pada masalah – masalah yang akan dibahas, maka ada beberapa batasan masalah yang perlu diperhatikan sebagai berikut:

1. Batas pengujian arus penyambungan pada 1 A, 1.6 A, 2 A, 2.5 A, & 3 A
2. Komunikasi *Web Server mode Access Point Mikrokontroler* ESP32
3. *Monitoring* pemakaian arus dan energi listrik pada LCD (*Liquid Cristal Display*)

E. Manfaat Penelitian

Peneliti mempunyai sasaran atau tujuan yang ingin dicapai dalam Penelitian ini antara lain:

1. Meminimalisir penggunaan MCB (*Miniature Circuit Breaker*) dalam pemasangan Layanan Penyambungan Sementara.
2. Meminimalisir adanya Susut Energi listrik akibat keterlambatan penormalan Layanan Penyambungan Sementara.
3. Mengetahui pemakaian energi listrik Penyambungan Sementara.
4. Meningkatkan citra perusahaan PT PLN (Persero).

BAB II

KAJIAN PUSTAKA

A. Dasar Teori

Dasar teori yang digunakan pada penelitian ini merupakan pengambilan dari teori – teori terkait, berupa artikel,jurnal, makalah tugas akhir, dan buku yang ada. Adapun pembahasan teori – teori tersebut sebagai berikut :

1. Listrik

a. Pengertian Listrik

Listrik adalah aliran elektron dari atom kepada penghantar. Pada masa modern sekarang ini, listrik telah menjadi kebutuhan penting/pokok bagi ummat manusia, karena sangat dibutuhkan dalam kehidupan sehari-hari (Yosua Erick;2021).

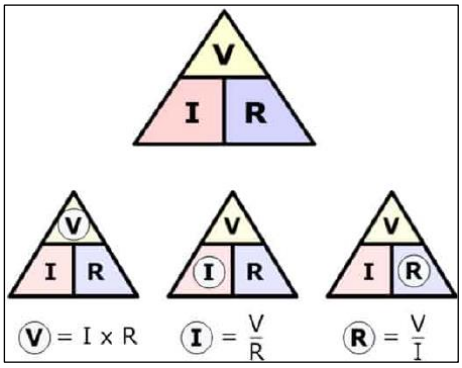
b. Sejarah Penemuan Listrik

Listrik ditemukan oleh Thales, seorang peneliti Yunani pada tahun 640 – 646 SM. Saat itu, Thales menggosok amber (elektron) dengan sentuhan bulu kucing dan menciptakan listrik yang dihasilkan melalui muatan gesekan. Penelitian mengenai kekuatan gerak kemudian dilanjutkan oleh seorang peneliti asal Inggris bernama William Gilbert pada tahun 1733. Kemudian pada tahun 1739, Charles du Fay menemukan hal baru. Charles menyadari bahwa kekuatan dari pengungkapan masa lalu terdiri dari poros Positif (+) dan Negatif (-). Penelitian kemudian dilakukan oleh Benjamin Franklin yang menunjukkan bahwa petir adalah salah satu jenis kekuatan alami listrik. Penelitian tentang listrik terus

berlanjut hingga akhirnya Michael Faraday menemukan Listrik magnet, yaitu elektromagnetik. Dengan cara ini, listrik dapat diperoleh dari perak dan magnet.

c. Besaran Listrik

Besaran – besaran dasar terkait listrik yaitu arus (I) dengan satuan *Ampere*,tegangan (V) dengan satuan *Volt* dan hambatan (R) dengan satuan *ohm*. Berikut ini adalah rumus dari Tegangan,Arus,dan Hambatan atau sering disebut dengan rumus segetiga daya.



Gambar 2.1 Rumus Segetiga Daya

d. Daya Listrik

Daya listrik yaitu kemampuan suatu alat listrik untuk melakukan usaha karena adanya perubahan kerja dan perubahan muatan listrik per satuan waktu. Besarnya daya listrik yang dialirkan oleh peralatan listrik dipengaruhi oleh adanya tegangan listrik, kuat aliran listrik, dan hambatan listrik pada suatu rangkaian listrik tertutup, serta kondisinya dalam jangka panjang. Ketiga besaran listrik ini menentukan berapa banyak daya listrik yang diburuhkan peralatan listrik agar berfungsi secara baik, berikut adalah rumus daya Listrik.

$P = V \text{ x } I$ ……..(1)

Keterangan:

P = Daya (*Watt* / W); I = Arus (*Ampere* / A); dan *V* = Beda potensial (*Volt* / V)

Hukum joule dapat digabungkan dengan hukum ohm untuk menghasilkan persamaan tambahan sebagai berikut;

$$P = I^2 \text{x } R = \frac{V^2}{R} \quad \ldots\ldots\ldots\

b. Penyambungan Sementara dilaksanakan dengan mekanisme sebagai berikut:

1) Pendaftaran;
2) Penerbitan jawaban;
3) Pembayaran biaya;
4) Pelaksanaan Penyambungan Sementara; dan
5) Pelaksanaan Pembongkaran Penyambungan Sementara.

3. Susut Energi Listrik

Berdasarkan Keputusan Direksi PT PLN (Persero) No.217-1.JK/DIR/2005 tentang Pedoman Penyusunan Laporan Neraca Energi (kWh), jenis susut (*losses*) energi listrik dapat dibedakan menjadi dua berdasarkan sifatnya yaitu Susut Teknis dan Susut *Non* Teknis. Susut Non Teknis adalah hilangnya energi listrik yang digunakan oleh pelanggan dan bukan pelanggan yang disebabkan tidak terhitung dalam penjualan. Ada beberapa hal yang menyebabkan adanya susut non Teknik antara lain yaitu pencurian Listrik, ketidak akuratan pembacaan meter, kesalahan alat ukur dan lain-lain. Berdasarkan penjelasan diatas Susut (*Losses*) yaitu suatu bentuk kehilangan energi listrik yang bersumber dari selisih jumlah energi listrik yang tersedia dengan jumlah energi listrik yang terjual.

4. *Kilo Watt Hour* (kWh) Meter

Pengukuran merupakan suatu siklus pengukuran yang pada hakekatnya merupakan upaya menyatakan sifat-sifat suatu zat atau barang sebagai suatu

angka atau nilai. Alasan pemberian angka dalam pengukuran dimungkinkan dilakukan dengan cara membandingkan alat yang diperkirakan dengan instrumen tertentu yang dipandang sebagai suatu standar atau membandingkan besaran yang diukur dengan skala yang telah ditera.

Kebenaran suatu hasil pengukuran bergantung pada instrumen yang digunakan sebagai korelasi atau penunjuk dan orang yang melakukan pengukuran, termasuk bagaimana alat ukur tersebut dipasang. Peralatan yang digunakan dalam pengukuran ini disebut instrumen pengukur. Perangkat ini menunjukkan nilai dari jumlah yang diukur. Hasil pengukuran adalah tanda-tanda langsung yang dapat dibaca dengan teliti.

Kilo Watt Hour (KWH) meter merupakan alat pengukuran energi aktif yang memanfaatkan instrumen komputasi dan menggunakan asas induksi. KWH meter adalah suatu instrumen untuk menghitung berapa kerja listrik (*Watt hour*) dalam periode tertentu. KWH meter dilengkapi dengan plat alumunium dan alat komputasi yang disebut dengan penghitung mekanis. Instrumen ini terdiri dari kumparan arus yang terhubung secara seri dengan beban dan tegangan yang terhubung dengan beban secara paralel. Besarnya jumlah kerja listrik pada suatu beban untuk waktu tertentu (Surya Darma,dkk, 2019).

5. *Miniature Circuit Breaker* (MCB)

Miniature Circuit Breaker (MCB) diperuntukkan untuk membatasi arus listrik dan sebagai tindakan pengamanan pada instalasi listrik. MCB sebagai pengaman arus lebih dan pengaman hubung singkat. MCB akan berfungsi sebagai pemutus beban apabila arus yang melewati lebih dari arus nominal yang telah

ditetapkan pada MCB. Batas arus pada MCB yaitu 2A, 4A, 6A,10A, 16A, 20A, 25A, 35A, 50 A dan lain sebagainya (Muhammad saleh,2017).



Gambar 2.2 Bentuk MCB
Sumber : Muhammad saleh ; 2017

MCB yaitu suatu alat pengaman otomatis yang berguna sebagai pembatas arus listrik yang menuju ke peralatan listrik. MCB juga berfungsi sebagai saklar. Pada penggunaannya, MCB disesuaikan dengan daya listrik yang ada pada instalasi listrik, supaya energi listrik yang digunakan sesuai dengan kebutuhan (Wahyu Zainal Riyadi, 2018).

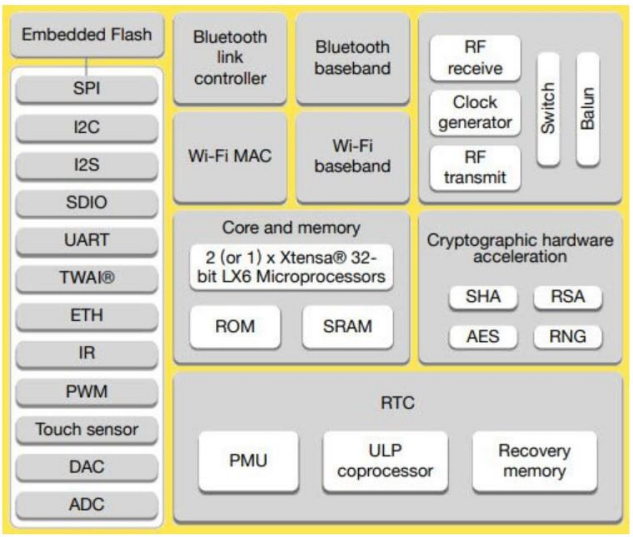
6. *Mikrokontroller ESP32*

a. Pengertian Mikrokontroler ESP32

ESP32 yaitu mikrokontroler System on Chip (SoC) berbiaya rendah dari Espressif Frameworks, yang merupakan perancang SoC ESP8266 terkenal dengan NodeMCU. ESP32 adalah pengganti SoC ESP8266 yang menggunakan Mikroprosesor Xtensa LX6 32-bit Tensilica yang dilengkapi Wi-Fi dan Bluetooth.

Keunggulan ESP32, seperti ESP8266, yaitu komponen RF yang terintegrasi antara lain Power Amplifier, Low-Noise Receive Amplifier, Antena

Switch, dan Filter. Hal ini yang membuat perancangan perangkat keras ESP32 menjadi mudah karena hanya memerlukan beberapa komponen luar.



Gambar 2.3 Hardware Mikrokontroler ESP32
Sumber: Muhammad Iqbal,2022

Hal penting yang perlu diketahui pada ESP32 yaitu dibuat menggunakan inovasi berdaya super rendah 40nm dari TSMC. Dengan cara ini, baterai ini dapat digunakan dengan baik dengan baterai biasa seperti yang digunakan pada peralatan suara, gadget pengecekan, jam tangan pintar, dan sebagainya.

b. Spesifikasi ESP32

ESP32 mempunyai banyak fitur dibandingkan ESP8266. Berikut adalah daftar spesifikasi penting dari ESP32 :

Table 2.1 Spesifikasi Teknis Mikrokontroler ESP32

No	Nama	Deskripsi
1	Tegangan INPUT	5 Volt
2	Tegangan kerja	5 Volt
3	ADC PIN	18 Buah
4	DAC PIN	2 Buah
5	Flash Memori	128 KB
6	SRAM	320 KB
7	Clock Speed	240 MHz
8	Komunikasi	WIFI, Bluetooth. I2C, SPI,Serial
9	Ukuran board	58,6 mm x 29 mm
10	Berat	25gram

c. Software Pendukung

Perangkat keras yang baik seperti ESP32 akan lebih mudah dipahami apabila bisa diprogram dengan lebih dari satu perangkat lunak pendukung. ESP32 dapat didukung oleh beberapa pemrograman seperti Arduino IDE, LUA, *MicroPython*, *Espressif* IDF, dan JavaScript.

d. Komunikasi Wired

ESP32 dapat berkomunikasi/bertukar data lewat kabel dengan memakai beberapa protokol:

- I2C (*Inter-Integrated Circuit*) bisa digunakan pada ESP32 melalui pin SCL (22) dan SDA (42)

- SPI (*Serial Peripheral Interface*) adalah Komunikasi SPI pada ESP32 memakai pin MOSI (23), MISO (19), SCK (18), dan CS (2)
- UART (*universal asynchronous receiver-transmitter*), ESP32 dapat menggunakan komunikasi UART dengan pin Tx (1) dan Rx (3)

e. Komunikasi Nirkabel

Komunikasi tanpa kabel pada ESP32 adalah *Wifi*, *Wifi* melibatkan signal radio sebagai bagian utama untuk melakukan komunikasi. Frekuensi radio yang paling banyak digunakan adalah frekuensi 2.4GHz dan 5GHz. *Router* atau *Access* Point dapat menerima informasi dari internet dan kemudian mengkomunikasikannya ke dalam transmisi radio yang kemudian akan dikirim dari antena *Wi-Fi* dan dikirim ke perangkat penerima. Istilah untuk *Wifi* adalah sebagai berikut;

- Access Point (AP) merupakan suatu perangkat yang tugasnya menghubungkan perangkat-perangkat *Wifi* yang berbeda-beda sehingga memungkinkan saling terkoneksi satu sama lain ke jaringan lokal atau internet.
- *Service Set Identifier* (SSID) sering disebut ID *Network* yaitu nama jaringan nirkabel (*Wifi)* yang dikomunikasikan agar dapat dikenali oleh berbagai *device*.
- *Password* adalah pengaman *wifi* supaya tidak dimanfaatkan orang lain. *Wifi* diamankan dengan enkripsi yang berbeda seperti WEP, WPA, WPA2-PSK, dan sebagainya.

- *Received signal strength indication* (RSSI) adalah data kekuatan sinyal yang didapat oleh *device Wifi.* Kisaran RSSI yaitu -10dBm hingga -100 dBm. semakin dekat -10, maka semakin baik.
- *Internet Protocol* (IP) yaitu nomor yang dijadikan alamat agar dapat terkoneksi ke jaringan internet maupun lokal.

f. Wifi ESP32 (*Mode Access Point*)

ESP32 menggunakan *Wifi* 802.11 b/g/n hingga 150 Mbps yang terintegrasi dengan *board,* sehingga lebih mudah dalam melakukan IoT. *WiFi* pada ESP32 memiliki kualitas yang sangat stabil karena mampu menjangkau sinyal sampai jarak 25 meter. *Wifi* ESP32 dapat dimanfaatkan pada mode *Access Point* (AP).

Dalam *Mode Access Point,* ESP32 berfungsi sebagai stasiun atau penerima sinyal *Wifi* yang ditransmisikan oleh *device* lain *(Router, Access Point* dsb.) ESP32 harus mengubah SSID dan *password* sesuai *router* yang dihubungkan. SSID dan *password* dapat diatur lewat program yang diupload ke ESP32 sesuai *access point* yang akan dikaitkan. ESP32 akan mendapatkan IP dari *router*. *Mode* ini lebih mudah karena digunakan ketika ESP32 berfungsi sebagai penyedia data, atau sebagai pengirim data ke internet maupun jaringan lokal. (Muhammad Iqbal,2022).

7. Sensor PZEM 004T

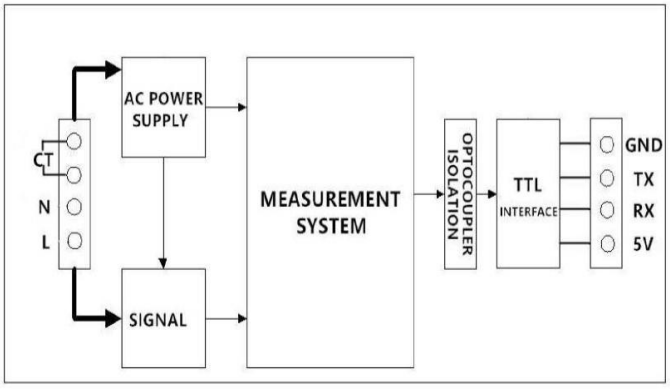
1. Pengertian Sensor PZEM-004T

PZEM-004T merupakan sensor yang bisa dimanfaatkan untuk mengukur tegangan, arus, daya dan frekuensi yang dapat dikkoneksikan melalui Arduino maupun perangkat *opensource* yang lain. (Ririn Rosmanty Silaban,2020).



Gambar 2.4 Sensor PZEM-004T
Sumber: Ririn Rosmanty Silaban,2020

Dengan kemampuan/fitur tersebut, modul PZEM-004T cocok digunakan menjadi alat pengukur daya pada jaringan listrik. Modul PZEM-004T ada *type* 10 Ampere dan 100 Ampere. PZEM-004T yang menggunakan Trafo Arus *type split core* mempunyai kelebihan karena mudah digunakan karena cenderung langsung dipasang pada kabel listrik yang umum dipasang tanpa melepas kabel tersebut (Ririn Rosmanty Silaban,2020).



Gambar 2.5 Blok Diagram PZEM-004T
*Sumber**:** Ririn Rosmanty Silaban,2020*

Berikut merupakan spesifikasi dari PZEM-004T:

1. Fungsi pengukuran (Arus, tegangan, daya aktif, frekuensi).
2. *Power button clear / reset Energy* (PZEM-004T V2.0)
3. *Power-down data storage function* (*cumulative power down before saving*)
4. Komunikasi Serial TTL
5. Pengukuran Daya: 0 ~ 9999 kW
6. Pengukuran Tegangan: 80 ~ 260VAC
7. Pengukuran Frekuensi: 45 Hz ~ 65 Hz
8. Pengukuran Arus: 0 ~ 100 A

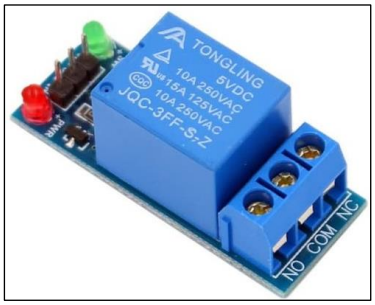
8. Modul Relay

a. Pengertian Modul Relay

Modul relay merupakan komponen elektronik sebagai saklar dengan aliran listrik sebagai pengontrolnya. Modul relay digunakan mengendalikan beban AC dengan rangkaian kendali DC dari sumber tegangan yang berbeda antara tegangan beban dengan tegangan rangkaian kendali. Modul relay diperlukan pada rangkaian

elektronika sebagai pelaksana dan titik penghubung antar beban dan sistem kendali elektronik dengan sistem catu daya berbeda. *(Made Toby Sathya Pratika;2021).*

Modul relay merupakan suatu alat yang bekerja berdasarkan standar elektromagnetik untuk menggerakkan kontaktor agar berpindah posisi *ON* ke *Off* atau dengan menggunakan tenaga listrik. Peristiwa penutupan dan pembukaan kontaktor ini terjadi karena dampak induksi magnet yang muncul dari kumparan induksi listrik. Hal mendasar yang membedakan antara saklar dan relay yaitu saat perpindahannya dari posisi *ON* ke *Off*. Saklar perpindahannya secara manual sedangkan relay secara otomatis memanfaatkan aliran listrik. *(Aldy Razor;2020).*



Gambar 2.6 *Modul Relay*
Sumber: Aldy Razor;2020

b. Fungsi Modul Relay

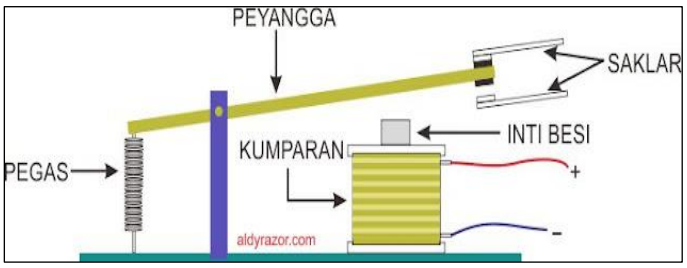
Pada hakikatnya modul relay berfungsi sebagai saklar listrik, yang mana ketika diberikan perintah logika perpindahan akan bekerja secara otomatis. Pada umumnya relay DC 5volt digunakan untuk membuat proyek yang salah satu bagiannya memerlukan tegangan tinggi atau AC (*Alternating Current*). Adapun fungsi relay yang lebih jelas sebagai berikut:

- Menjalankan perintah logika dari mikrokontroler, seperti ESP32

- Wadah untuk mengontrol tegangan tinggi dengan menggunakan tegangan rendah
- Meminimalisir terjadinya drop tegangan
- Dapat berfungsi untuk penundaan waktu atau fungsi *time delay function*
- Mengamankan komponen lain dari *over voltage* penyebab *korsleting*.
- Menyederhanakan rangkaian supaya lebih *simple*.

c. Cara Kerja Relay

Beberapa komponen pada *Relay* seperti Penyangga (*Armature*), Kumparan (*Coil*), Pegas (*Spring*), Saklar (*Switch Contact*), dan Inti Besi (*Iron Core*). Penempatan letak komponennya seperti pada gambar dibawah ini:



Gambar 2.7 Struktur Relay
Sumber: Aldy Razor;2020

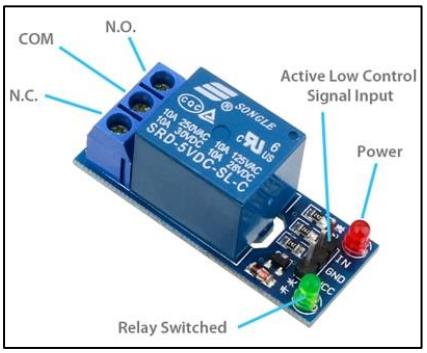
Berdasarkan gambar relay diatas, dapat dipahami bahwa relay akan bekerja karena gaya elektromagnetik. Ini terbuat dari bagian tengah besi yang dililitkan kawat kumparan dan dialiri aliran listrik. Ketika kumparan teraliri listrik maka inti besi akan berubah jadi magnet dan menggerakkan penyangga sehingga merubah kondisi dari tertutup menjadi terbuka. Sementara itu, ketika kumparan

tidak lagi mempunyai daya, maka pegas akan menarik ujung penyangga dan menyebabkan kondisi terbuka semula menjadi tertutup.

Kondisi atau posisi pada *relay* terbagi menjadi dua,yaitu:

- ***Normally Close* (NC)**, merupakan posisi awal atau posisi dimana *relay* dalam posisi tertutup dikarenakan tidak mendapatkan arus listrik.
- ***Normally Open* (NO)**, merupakan posisi dimana *relay* dalam kondisi terbuka dikarenakan mendapatkan arus listrik.

d. Skema Relay



Gambar 2.8 Skema Kerja Relay
Sumber: Aldy Razor;2020

Berdasarkan gambar skematik *relay* di atas, dibawah ini merupakan penjelasan dari ketiga pin yang sangat penting kita ketahui:

- **COM (*Common*),** merupakan pin yang harus dikoneksikan pada salah satu dari dua ujung kabel yang akan digunakan.
- **NO (*Normally Open*),** merupakan pin tempat mengkoneksikan kabel yang lain apabila menghendaki kondisi posisi awal terbuka atau arus listrik yang terputus.

- **NC (*Normally Close*),** merupakan pin tempat mengkoneksikan kabel yang lain apabila menghendaki kondisi posisi awal tertutup atau arus Listrik yang tersambung.

9. *Liquid Crystal Display* (LCD)

Liquid Cristal Display (LCD) adalah salah satu jenis tampilan layar elektronik yang dirancang dengan teknologi CMOS logic yang dapat bekerja dengan tidak menghasilkan cahaya melainkan memantulkan cahaya disekitarnya ke arah *front-lit* atau mentransmisikan cahaya dari *back-lit.* Kemampuan LCD sebagai tampilan informasi berupa karakter, huruf, angka atau grafik. (M. Natzir, dkk, 2019).

LCD merupakan lapisan kombinasi alami lapisan kaca bening dengan elektroda transparan indium oksida sebagai tampilan *seven-segment* dan lapisan elektroda pada kaca belakang. Apabila elektroda diaktifkan dengan medan listrik (tegangan), maka molekul organik yang panjang akan menyesuaikan dengan elektroda dari segmen. Lapisan sandwich memiliki *polarizer* Cahaya vertikal depan dan polarizer cahaya datar belakang diikuti oleh lapisan reflektor. Cahaya cermin tidak dapat menembus molekul-molekul yang telah menyesuaikan dan segmen yang telah diaktifkan akan tampak gelap dan membentuk karakter

informasi yang akan ditampilkan.



Gambar 2.9 Liquid Crystal Display (LCD) 20x4
Sumber : M. Natzir,dkk ; 2019

10. Modul *Power Supply*

Modul *power supply* AC ke DC 5V/12V adalah *device e*lektronik yang dibuat untuk mengubah arus listrik dari sumber daya listrik AC dengan tegangan 220V menjadi arus DC dengan tegangan 5V/12V. Modul ini biasanya digunakan untuk memberikan catu daya stabil dan aman kepada perangkat elektronik yang membutuhkan tegangan 5V, seperti mikrokontroler, sensor, dan perangkat elektronik kecil lainnya.



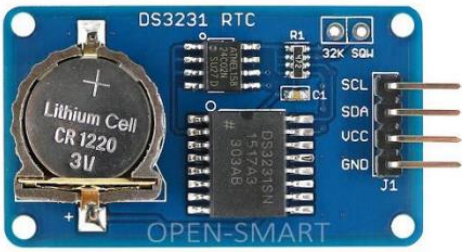
Gambar 2.10 Modul Power Supply

Modul *power supply* AC ke DC 5V/12V umumnya terdiri dari beberapa komponen, termasuk transformator untuk mengubah tegangan AC menjadi tegangan DC yang lebih rendah, serta rangkaian penyearah dan penyaring

berfungsi merubah arus AC menjadi arus DC yang stabil dan bersih. Beberapa modul *power supply* AC ke DC 5V/12V juga dilengkapi dengan regulator tegangan untuk memastikan keluaran tegangan tetap pada 5V/12V meskipun terjadi fluktuasi pada tegangan input. Modul ini biasanya memiliki beberapa fitur keamanan seperti perlindungan arus pendek dan perlindungan terhadap lonjakan tegangan. Hal ini membantu melindungi perangkat elektronik yang terhubung dari kerusakan yang disebabkan oleh lonjakan atau gangguan pada pasokan listrik.

11. Modul *Real Time Clock* DS3231

Modul RTC DS3231 merupakan Jam Waktu Nyata dengan kompensasi suhu osilator kristal yang terhubung (TCX0). TCX0 memberikan jam referensi yang stabil dan tepat serta menjaga presisi RTX sekitar ±2 menit dalam setahun. Modul DS3231 dilengkapi waktu dan jadwal dengan dua waktu peringatan setiap hari dan hasil gelombang persegi yang dapat diprogram. Waktu dan kalender memberikan data tentang detik, menit, jam, hari, tanggal, bulan, tahun yang terdapat dalam register dan memungkinkan untuk diakses dengan menggunakan komunikasi *Inter Integrated Circuit* (I2C).



Gambar 2.11 Modul RTC DS3231

12. *Hyper Text Markup Language* (HTML)

Hyper Text Markup Language (HTML) merupakan bahasa markup yang berfungsi untuk membuat halaman situs dan menampilkan data yang berbeda dalam program *Web*. Berawal dari bahasa yang sebelumnya banyak digunakan dalam bidang penerbitan dan pencetakan yang dikenal dengan SGML (*Standard Generalized Markup Language*), HTML merupakan standar yang umumnya digunakan untuk menampilkan halaman *website*. HTML adalah sebagai kode yang memerintahkan *browser* program untuk membuat tampilan sesuai yang dikehendaki. File yang berupa dokumen HTML dapat dibuka menggunakan *browser* seperti *Mozilla Firefox, Microsoft Internet Explorer,Opera, Google Chrome, Safari* dan sebagainya.

B. Kajian Terdahulu

Pada penelitian kali ini, peneliti mendapat *referensi* dari beberapa peneliti yang terdahulu yang mengunakan system yang hampir sama. Adapun kajian terdahulu sebagai berikut;

Penelitian yang dilakukan oleh Fadil Muhammad Noor, Sunarto, Yoseph Santosa (2022), yang berjudul "**Rancang Bangun Sistem Pengendali Beban Listrik Skala Rumah Tinggal Berbasis *Mikrokontroler* Atmega 328P".** Pada penelitian tersebut sistem pengendali menggunakan mikrokontroler Atmega328P yang dipergunakan sebagai pengendali dengan melakukan perintah dengan modul relay sebagai pengendali on/off pada beban listrik yang melibihi batas arus dan menggunakan sensor PZEM-004T. Sensor arus diatur pada arus 2A untuk kapasitas daya 450V, 4A untuk kapasitas daya 900 VA dan 6A untuk kapasitas

daya 1300 VA. Alat juga dilengkapi dengan modul RTC untuk memberi informasi waktu secara real time dan LCD untuk menampilkan data hasil pembacaan sensor.

Penelitian yang dilakukan oleh Rahma Farah Ningrum, Herman Bedi Agtriadi, & Novita Rose Marisi Pardede (2018), yang berjudul **"Prototype Alat Pembatas dan Pemutus Arus Listrik Pascabayar pada Rumah Tangga Berbasis *Smarthpone*"** pada penelitian tersebut menggunakan Sensor Arus TA12-200 untuk mendeteksi jumlah arus dan Sensor tegangan ZMPT101B Ultra Micro Voltage Transformer untuk mendeteksi jumlah tegangan. Data arus dan tegangan akan dikirim terbaca pada serial Arduino, kemudian dikirimkan kedalam database pada laptop/PC dengan menggunakan serial USB. Melalui aplikasi smartphone bisa memilih ruangan yang akan dimatikan.

Penelitian yang dilakukan oleh Rio Fernando, Ignatius Prasetya Dwi Wibawa, Cahyantari Ekaputri (2018), yang berjudul "**Sistem Kendali Dan *Monitoring* Penggunaan Daya Listrik Pada Perangkat Listrik Rumah Berbasis IOT".** Pada penelitian tersebut menggunakan sensor tegangan ZMPT101B untuk mendeteksi jumlah tegangan dan Sensor Arus ACS712 untuk mendeteksi besar arus yang mengalir. Hasil pembacaan diproses melalui *Mikrokontroller* ESP8266 dan memberikan perintah *ONN/OFF* ke Relay.

Penelitian yang dilakukan oleh Abdul Hanif (2022), yang berjudul "**Rancangan Sistem Kontrol Dan *Monitoring* Instalasi Listrik Berbasis Internet Of Things".** Pada penelitian tersebut menggunakan ESP8266 sebagai Mikrokontrollernya, dan Sensor PZEM 004-T sebagai pembaca arus dan tegangannya. Hasil percobaan rancangan *monitoring* instalasi listrik menggunakan

aplikasi OS android Blynk dan dapat dilihat melalui website sebagai media pemantauannya.

Penelitian yang dilakukan oleh Edy Supriyadi dan Siti Dinaryati (2020), yang berjudul **"Rancang Bangun System *Monitoring* dan Kendali Listrik Rumah Tangga Berbasis ESP8266 Node MCU"** Pada penelitian tersebut menggunakan *Mikrokontroller* ESP8266 sebagai mikrokontrollernya. Sistem wifi menggunakan Wemos D1 Mini. Adapun sensor arus menggunakan Modul ACS712 dan sensor tegangan menggunakan Modul ZMPT101B. Akses internet dibutuhkan agar dapat dipantau secara jarak jauh oleh pengguna, baik menggunakan *smartphone* yang mudah dibawa ataupun akses *web server* melalui PC dimanapun pengguna berada.

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Pada penilitian ini metode yang digunakan adalah (R&D) *Research and Development* yang mengacu pada referensi terdahulu. Karena metode ini dianggap efektif dalam melaksanakan penelitian ini.

Menurut Prof. Dr. Sugiono dalam bukunya "Metode Penelitian dan Pendidikan", beliau menyebutkan bahwa metode penelitian dan pengembangan atau dalam bahasa inggrinya *Research and Development* (R&D) adalah metode penelitian yang digunaan untuk menghasilkan produk tertentu dan menguji keefektifan produk tersebut.

Dengan menggunakan metodeologi penelitian R&D, nantikanya akan menghasilkan sebuah alat Kendali dan monitoring Penyambungan Sementara PLN Berbasis *Mikrokontroler* ESP32. Tahapan yang akan digunakan meliputi study leteratur, perancangan, pengujian, dan analisis hasi dari penelitian.

B. Lokasi Dan Waktu

Penelitian ini dilakukan di kab. Sidrap Sulawesi Selatan sedangkan waktu penelitian ini dimulai pada bulan Juni 2024 sampai Juli 2024, dimulai dari melakukan pengamatan pada proses yang berlangsung dan menganalisa permasalahan yang terjadi.

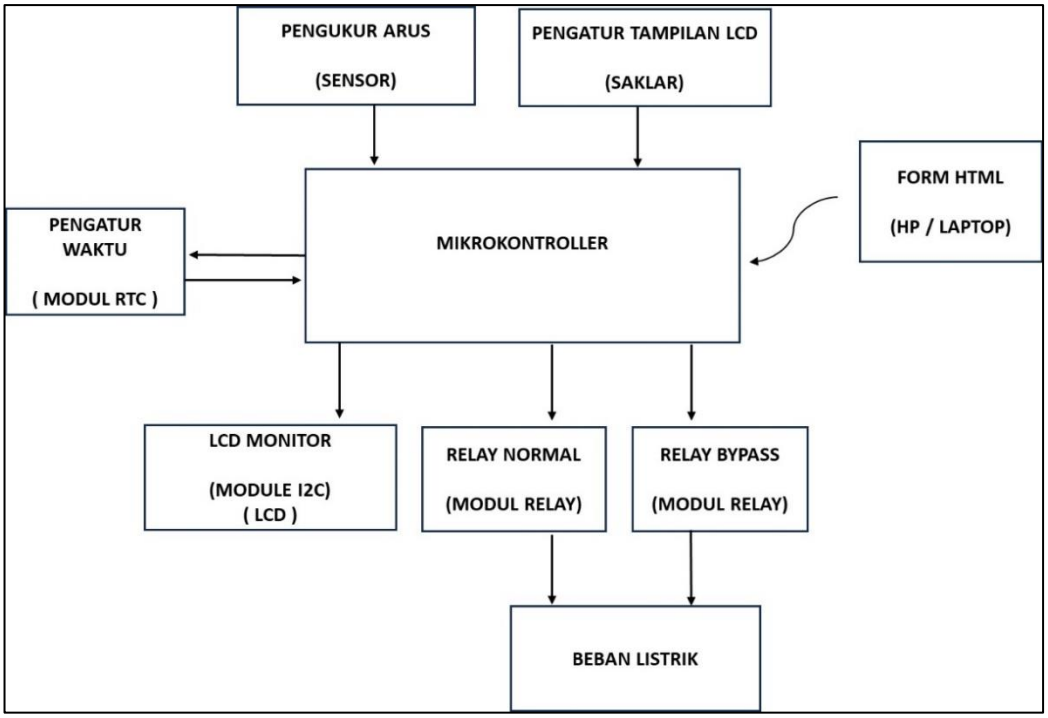
C. Alat Dan Bahan

Adapun alat dan bahan yang digunakan dalam melaksanakan penelitian ini adalah sebagai berikut:

Tabel 3.1 Alat dan Bahan Penelitian

NO	Alat dan bahan	Jumlah
1	Mikrokontoler ESP32	1 bh
2	Modul Relay 5volt 30 A	2 bh
3	Sensor PZEM-004T 100 A	1 bh
4	LCD 4x20	1 bh
5	Modul I2C	1 bh
6	Saklar	1 bh
7	RTC DS3231	1 bh
8	Power Suply 5 v	1 bh
9	Kabel Jumper	1 set
10	kWh (kilo Watt hour) meter	1 bh
11	MCB (Miniature Circuit Breaker)	1 bh
12	Smart Phone/Laptop	1 unit

D. Rancangan Penelitian



Gambar 3.1 Diagram Blok Rancangan Sistem Kendali dan *Monitoring*

Pada gambar diatas menjelaskan bagaimana system kendali dan monitoring berbasis *mikrokontroler ESP32* akan bekerja. *Sensor PZEM 004T* akan berfungsi sebagai input yang akan membaca besarnya arus yang mengalir dan memberikan informasi besarnya arus yang mengalir ke *mikrokontroler ESP32*. Saklar berfungsi untuk merubah informasi pada LCD yaitu tampilan batas penyambungan dan tampilan Waktu, pemakaian Arus dan energi Listrik. *Form Input HTML* melalui *Webserver Mode Access Point (AP)* juga berfungsi sebagai input untuk memberikan perintah pengaturan batas arus nominal dan waktu penormalan Penyambungan sementara pada *mikrokontroler ESP32*. Dari hasil pemrosesan data melalui *mikrokontroler ESP32* selanjutnya akan diteruskan ke

Modul relay untuk *On* atau *Off*. Adapun data hasil pembacaan Arus, *kilo Watt hour* (kWh) dan batas arus penyambungan akan ditampilkan melaui *Liquid Crystal Display (LCD)*.

E. Teknik Pengumpulan Data

Teknik pengambilan data yang dilakukan pada penelitian ini terdiri dari beberapa tahap. Tahap pertama dilakukan dengan studi literatur, tahapan ini bertujuan agar dapat memahami konsep dasar sistem komunikasi *Webserver Access point (AP) Mikrokontroler ESP32* pada beberapa referensi artikel dan jurnal. Tahapan kedua dilakukan untuk menguji pengiriman perintah dari *Web* HTML ke *Microkontroler ESP32* melalui *Webserver Mode Access Point* (AP) untuk pengaturan batas Arus nominal dan batas waktu Penyambungan Sementara. Tahapan ketiga dilakukan untuk menguji waktu penormalan Penyambungan sementara yang telah di inputkan melalui *Web* HTML ke *Microkontroler* ESP32 melalui *Webserver Mode Access Point* (AP). Tahapan keempat dilakukan dengan pengujian pembacaan *Sensor PZEM004T* terutama pada pembacaan arus yang akan di bandingkan dengan sebuah alat ukur Ampere Meter. Tahapan kelima dilakukan untuk menguji batas arus nominal yang telah di atur melalui *Web* HTML ke *Microkontroler* ESP32 melalui *Webserver Mode Access Point (AP)*.

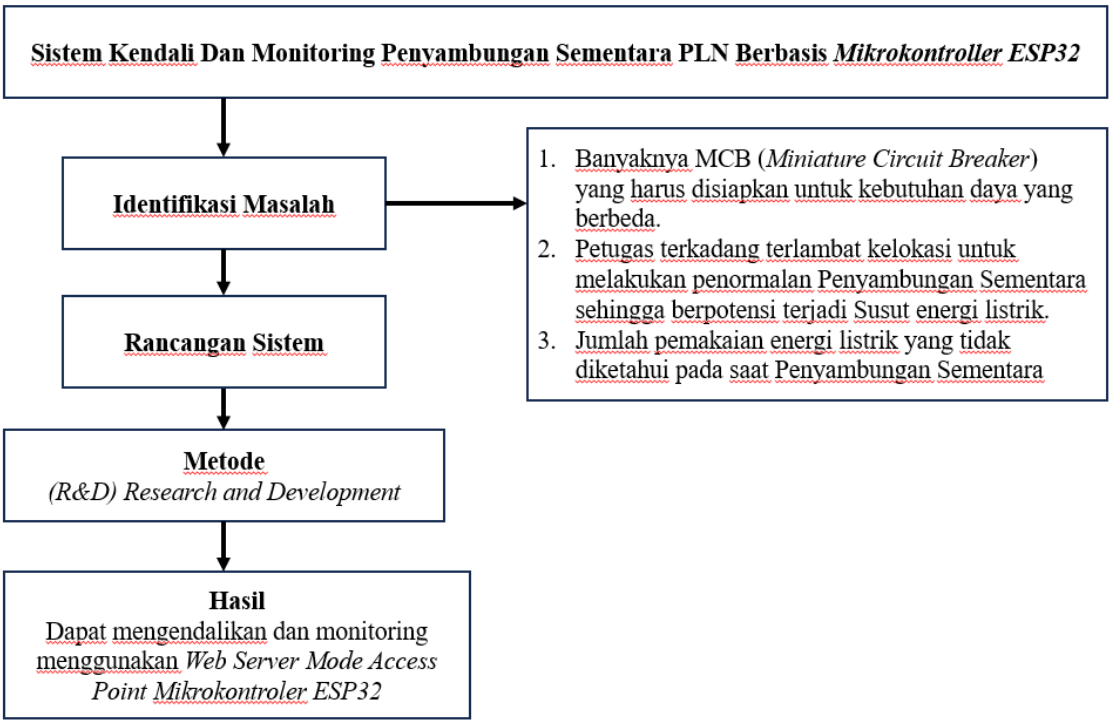
F. Teknik Analisa Data

Teknik analisa data yang dipergunakan adalah Statistik Deskriptif, teknik ini bertujuan untuk menganalisia data dengan cara mendiskripsikan atau menggambarkan data yang telah terkumpul berupa penyajian data melalui table, grafik, diagram, perhitungan dan sebagainya, dalam statistic deskriptif juga bisa

dilakukan untuk mencari besarnya hubungan antara variabel melalui analisis korelasi, melakukan prediksi dengan analisa korelasi, regresi atau membandingkan data-data sampel atau populasi.

G. Kerangka Pikir

Kerangka pikir yang digunakan dalam penelitian ini adalah sebagai berikut :



Gambar 3.2 Kerangka Pikir Sistem Kendali dan *Monitoring*

BAB IV

HASIL DAN PEMBAHASAN

A. Perancangan Sistem

1. Perancangan Perangkat Keras (*Hardware*)

Dalam Rangkaian Sistem kendali dan monitoring ini bermula dari *mikrokontroler* ESP32 sebagai *mikrokontroler*, yang memiliki tegangan input sebesar 5 V DC, oleh karena itu dalam perangkat ini dilengkapi dengan modul *power supply* ± 5V DC yang dikoneksikan dengan pin VIN dan GND pada *mikrokontroler* ESP32.

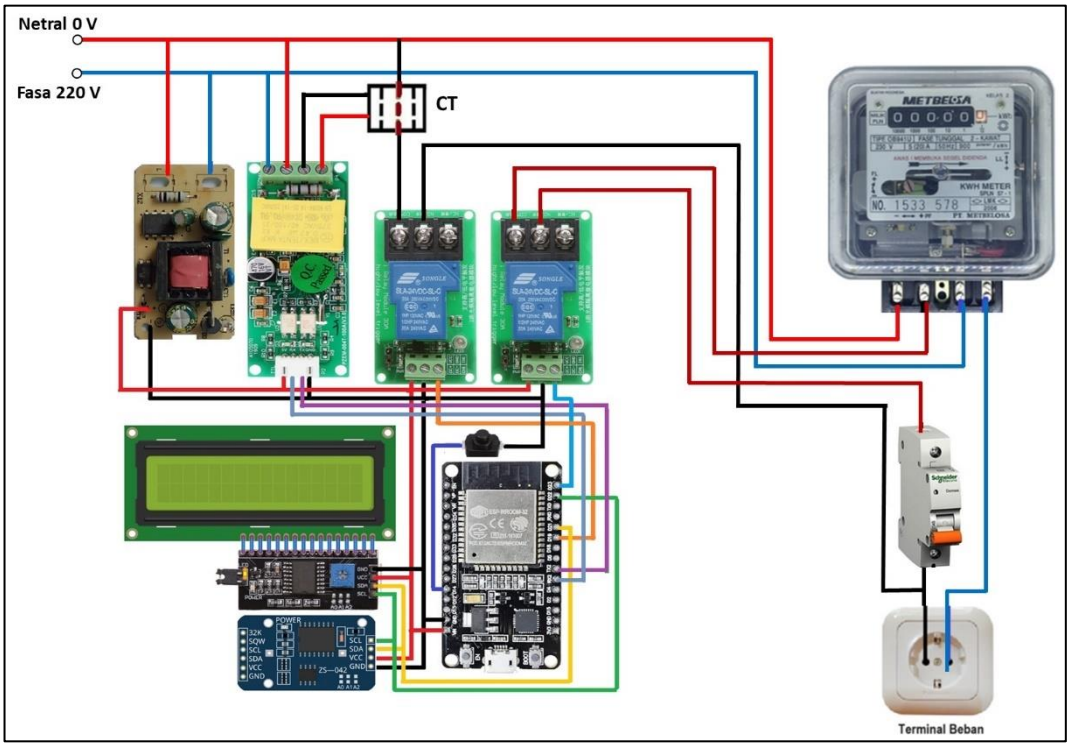
Kemudian untuk sensor Arus *PZM004T* pin RX dikoneksikan dengan pin RX2 pada *mikrokontroler* ESP32 ,pin TX dikoneksikan dengan pin TX2 pada *mikrokontroler* ESP32 adapun pin 5V dan GND dihubungkan dengan power supply.

Kemudian untuk Module relay 5 V yang pertama pin COM dikoneksikan dengan tegangan AC 220 V yang berfungsi sebagai Input ,kemudian pin NO yang berfungsi sebagai output dihubungkan ke output MCB pada APP (Alat Pengukur dan Pembatas) PLN yang terhubung ke instalasi listrik milik Pelanggan. Adapun pin IN1 pada Module relay terhubung pada pin D19 pada *mikrokontroler* ESP32, dan untuk pin VCC dan GND module relay terhubung pada power supply.

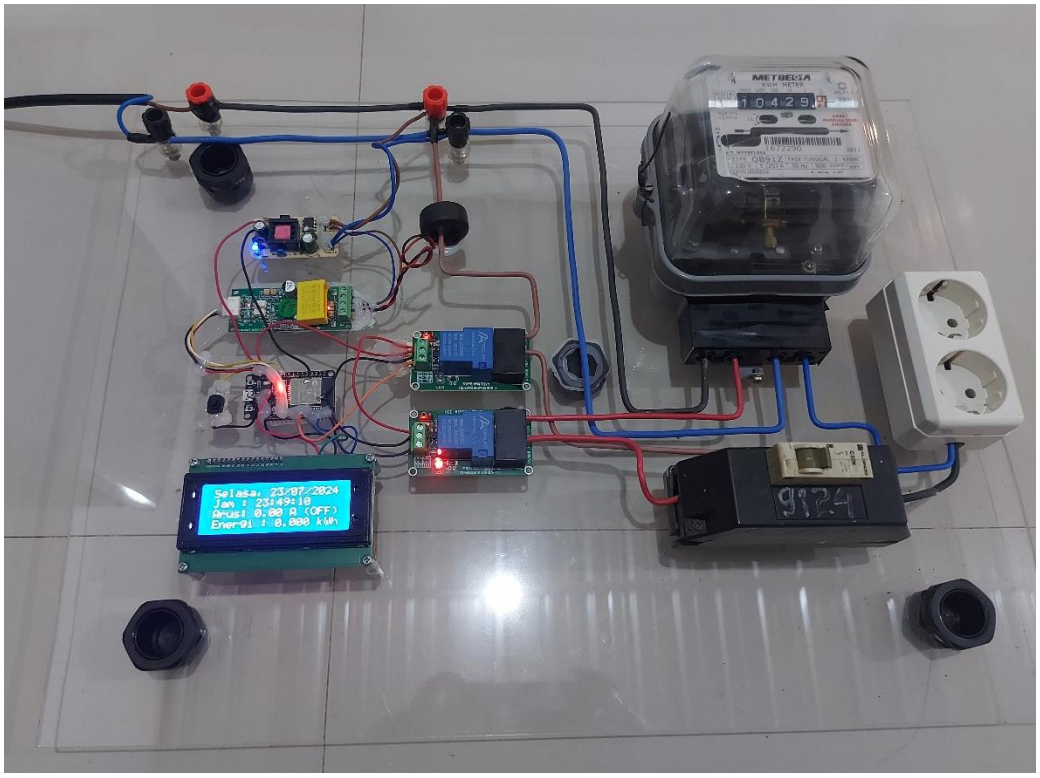
Untuk Module relay 5 V yang kedua pin COM dihubungkan dengan terminal 3 (tiga) atau phasa output dari kWh (kilo Watt hour) meter dan pin NO

dihubungkan ke input *MCB* (*Miniature Circuit Breaker*) pada APP (Alat Pengukur dan Pembatas) PLN. Adapun pin IN1 dihubungkan dengan pin D23 pada *mikrokontroller ESP32*, dan untuk pin VCC dan GND module relay terhubung pada *power supply*.

Untuk Module I2C pin SDA dihubungkan ke pin SDA module RTC DS3231 dan pin D21 pada *mikrokontroler* ESP32. Adapun vin SCL module I2C terhubung ke pin SCL module RTC DS3231 dan pin D22 pada *mikrokontroler* ESP32, dan untuk pin VCC dan GND module relay dan RTC terhubung pada *power supply*. Untuk lebih jelasnya dapat dilihat pada gambar *wiring* diagram dibawah ini.



Gambar 4.1 *Wiring* diagram Sistem kendali dan *monitoring* PS



Gambar 4.2 Prototype Sistem kendali dan *monitoring* PS

2. Perancangan Perangkat Lunak (*Software*)

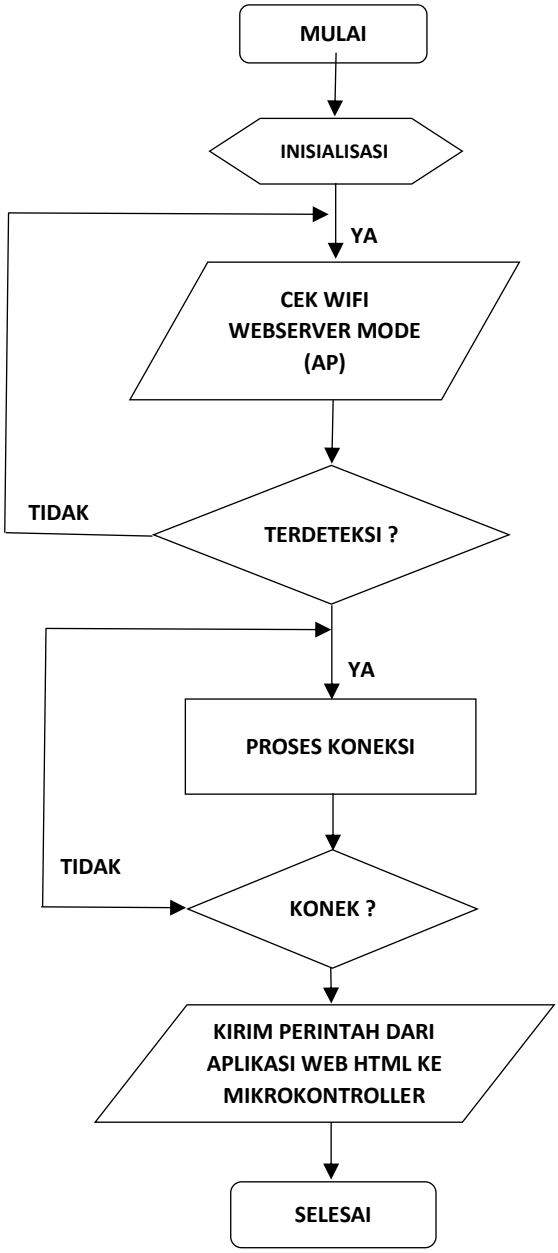
Pada perancangan perangkat lunak (*software*) ada beberapa hal yang akan di jabarkan, yaitu alur pemrograman dan pemrograman Sistem kendali dan monitoring.

a. Alur Pemrograman (*Flowchart*)

Alur pemograman adalah langkah-langkah logis yang harus diikuti dalam menyelesaikan suatu masalah atau tugas dengan menggunakan bahasa pemrograman. Alur pemrograman memberikan panduan tentang bagaimana program harus berjalan, dari awal hingga akhir.

1. Alur Pemrograman pengiriman perintah dari Form *HTML melalui komunikasi Webserver Mode Acces Point.*

Berikut adalah alur pengiriman perintah dari Form *HTML melalui komunikasi Webserver Mode Acces Point* yang berfungsi memberi perintah batas waktu penyambungan Penerangan Sementara berupa data Hari, Jam, Menit, Detik, dan batas Arus penyambungan.



Gambar 4.3 *Flowchart* komunikasi *Webserver Mode Acces Point*

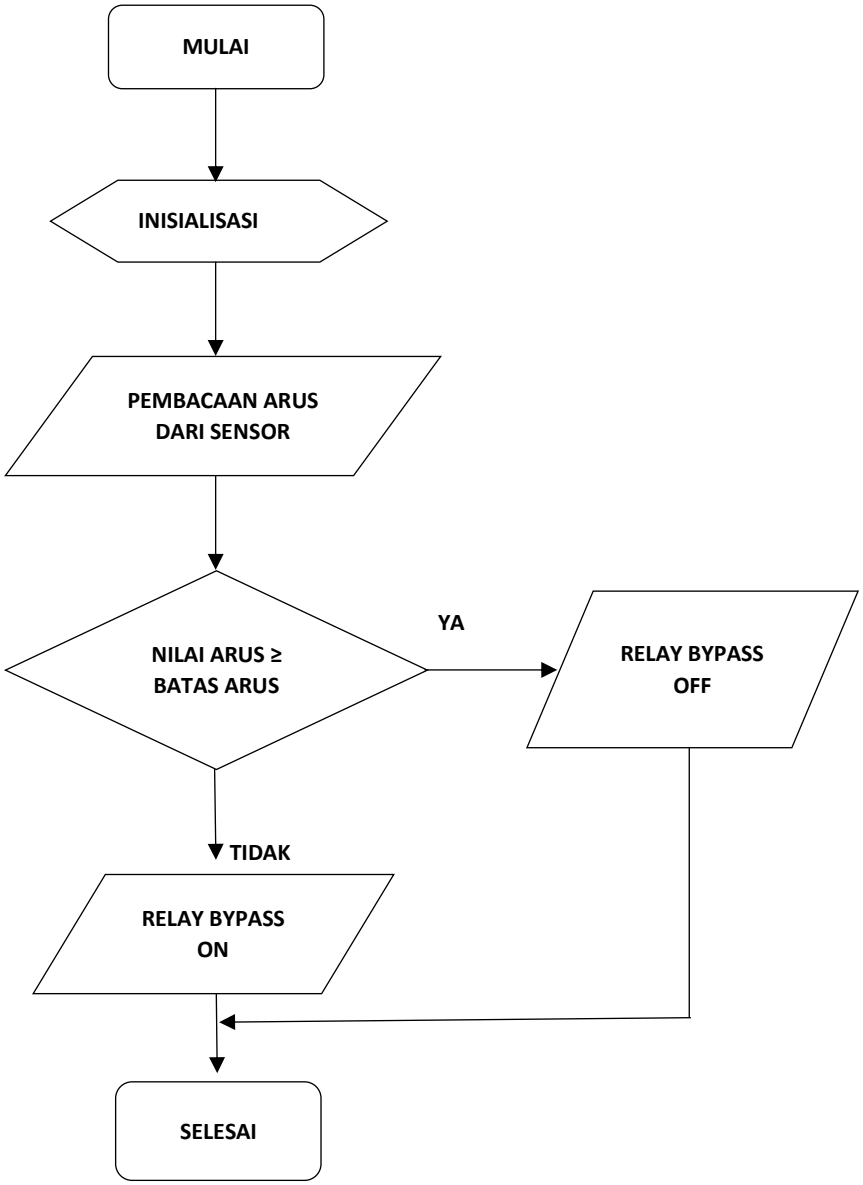
Gambar 4.3 merupakan alur pemrograman dari form *HTML* ke *Mikrokontroller* ESP32 melalui komunikasi *Webserver* mode *Access Point*. Langkah pertama adalah memulai sistem, dilanjutkan dengan inisialisasi setiap komponen yang telah digunakan, selanjutnya proses pendeteksian *wifi Wbserver Mode Acces Point*. Apabila *wifi* telah ditemukan kemudian lanjut pada proses

koneksi kejaringan dengan memasukka *password* yang telah ditentukan. Apabila telah terkoneksi pada jaringan yang telah ditentukan maka lanjut pada tahap berikutnya, namun apabila belum terkoneksi kejaringan maka akan kembali ke langkah pertama.

Setelah terkoneksi dengan jaringan *Webserver Mode Acces Point* maka dilanjutkan dengan membuka browser untuk memasukkan alamat *Web HTML* yang telah ditentukan. Setelah masuk ke halaman tersebut akan tampil form *HTML* untuk penginputan data batasan penyambungan berupa Hari , Jam, Menit, detik, dan batas arus penyambungan dan dilakukan submit. Setelah tekan submit atau mulai maka perintah akan dikirimkan dan diproses oleh *Mikrokontroller*.

2. Alur Pemrograman Batas Arus Penyambungan Sementara

Berikut adalah Alur pemrograman batasan arus Penyambungan Sementara yang bertujuan untuk membatasi nilai nominal arus yang telah ditentukan.



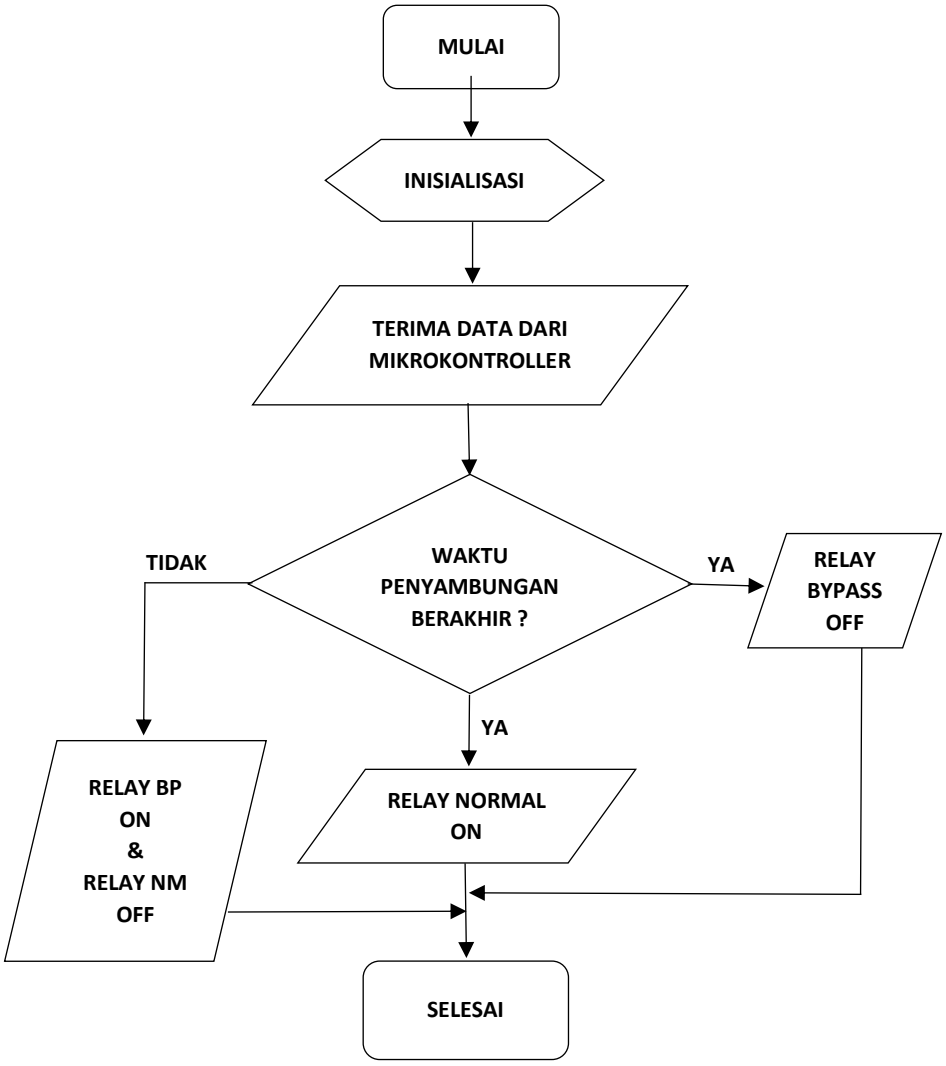
Gambar 4.4 *Flowchart* Batas Arus Penyambungan Sementara

Gambar 4.4 merupakan alur pemrograman batas Arus Penyambungan Sementara. Langkah pertama adalah memulai sistem, dilanjutkan dengan inisialisasi setiap komponen yang telah digunakan. Hasil pembacaan nilai arus oleh sensor *PZEM* 004T akan diproses oleh *mikrokontroler*. Apabila nilai arus dibawah dari batas arus yang telah ditentukan maka *relay bypass* akan ON. Dan sebaliknya apabila nilai arus yang terbaca melebihi dari batas arus yang telah

diatur maka mikrokontroller akan memerintahkan mdule *relay bypass* untuk *OFF* dan akan *ON* Kembali setelah durasi waktu duapuluh detik, dan apabila setelah proses *ON* Kembali dan sensor arus masih membaca arus lebih dari batas arus yang telah ditentukan maka module *relay bypass* akan *OFF* Kembali dan *ON* Kembali setelah durasi waktu duapuluh detik.

3. Alur Pemrograman Batas Waktu Penyambungan Sementara

Berikut adalah Alur pemrograman batas waktu Penyambungan Sementara yang bertujuan untuk membatasi durasi penyambungan sementara yang meliputi batas Hari dan Jam penyambungan.

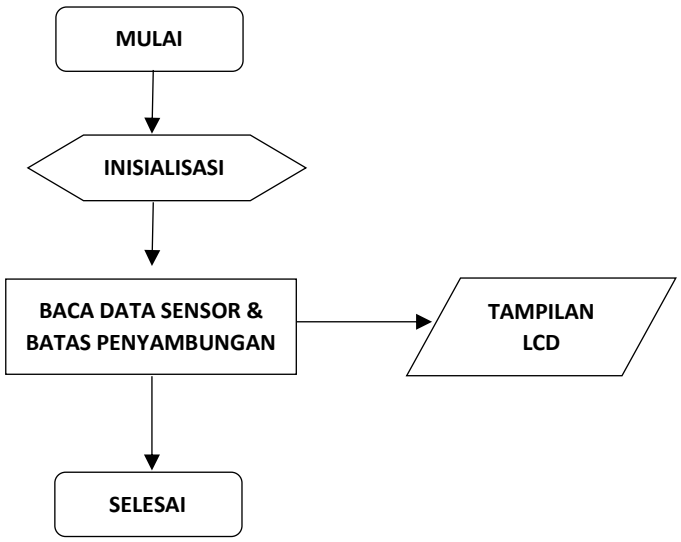


Gambar 4.5 *Flowchart* Batas waktu Penyambungan Sementara

Gambar 4.5 merupakan alur Pemrograman Batas Waktu Penyambungan Sementara. Langkah pertama adalah memulai sistem, dilanjutkan dengan inisialisasi setiap komponen yang telah digunakan. Adapun batas waktu penyambungan sementara sebelumnya sudah dilakukan pengaturan batas penyambungan melalui module *RTC*. Apabila waktu penyambungan belum berakhir maka sistem Penyambungan Sementara akan berfungsi sebagaimana mestinya dimana *Relay Bypass* akan *ON* dan Relay Normal akan *OFF*. Dan

sebaliknya apabila waktu penyambungan telah berahir maka Relay Bypass akan *OFF* dan relay Normal akan *ON*. Dengan relay Normal posisi ON dan *Relay Bypass* posisi *OFF* maka kWh (*kilo Watt hour*) meter di rumah pelanggan akan mengukur Kembali seperti semula.

4. Alur Pemrograman Fungsi Monitoring



Gambar 4.6 *Flowchart* fungsi Monitoring

Gambar 4.6 merupakan alur Pemrograman fungsi monitoring Penyambungan Sementara. Langkah pertama adalah memulai sistem, dilanjutkan dengan inisialisasi setiap komponen yang telah digunakan. Pada LCD terdapat beberapa parameter yang ditampilkan antara lain Hari, jam, menit, detik, pembacaan arus, Informasi Relay *OFF* dan *ON*. Tampilan pertama berupa tampilan Hari,Jam, menit, detik ,arus terbaca dan pemakaian energi. Sedangkan

tampilan kedua berupa batas waktu penyambungan yang menunjukkan Hari, Jam, menit, detik, dan Batas arus penyambungan. Adapun saat nilai arus melebihi batas yang telah ditentukan maka Relay akan *OFF* selama duapuluh detik dan akan menampilkan nilai arus lebih tersebut.

b. Pemrograman Pengiriman perintah dari Form *HTML ke Mikrokontroler ESP32* melalui *Webserver Acces Point*

1) Penyertaan *Library*

Pada pemrograman ini ada beberapa hal yang perlu diperhatikan seperti penggunaan *library* untuk komponen atau module yang digunakan. *Library* yang digunakan dalam program arduino terdiri dari 10 (sepuluh) *library* yaitu *Arduino.h, Wifi.h, AsyncTCP.h, SPIFFS.h, ESPAsyncWebServer.h, Preferences.h, PZEM004Tv30.h, LiquidCrystal_I2C.h, Wire.h dan RTClib.h.S.* Adapun penulisan *library* sebagai berikut :

```
#include <Arduino.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <SPIFFS.h>
#include <ESPAsyncWebServer.h>
#include <Preferences.h>
#include <PZEM004Tv30.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
```

2) Membuat Objek *AsyncWebServer*

Objek *AsyncWebServer* akan digunakan untuk menyiapkan server web ESP32. Ini akan meneruskan port HTTP default yaitu 80, sebagai masukan ke konstruktor yangi akan menjadi port dimana server akan mendengarkan permintaan.

```
AsyncWebServer server(80);
```

3) Mengatur Kredensial Jaringan

Perlu dibuat dua variabel global, satu untuk SSID dan satu lagi untuk kata sandi. Ini akan menyimpan kredensial jaringan yang akan digunakan untuk terhubung ke jaringan nirkabel kami. Di sini diguanakan nama jaringan "ESP32-Access-Point" dengan sandi "123456789"

```
const char* ssid     = "ESP32-Access-Point";
const char* password = "123456789";
```

4) Membuat Variabel Masukan

Diperlukan 5 (lima) variabel global untuk setiap kolom masukan. Semuanya akan digunakan untuk memantau apakah permintaan HTTP GET diterima dari kolom input. Jika demikian, maka nilai baru yang diperbarui akan disimpan di setiap variabel. Variabel-variabel ini akan diperbarui setiap kali pengguna memasukkan nilai di kolom input masing-masing dan menekan tombol kirim di server web.

Untuk menyimpan data Hari, jam, Menit, Detik dan batas Arus maka dibuat variabel parameternya sbb. :

```
const char* parameter_Hari = "input_Hari";
const char* parameter_Jam = "input_Jam";
const char* parameter_Menit = "input_Menit";
const char* parameter_Detik = "input_Detik";
const char* parameter_Arus = "input_Arus";
```

5) Membuat Setup

Di dalam fungsi setup, program akan membuka koneksi serial dengan baud rate 9600.

```
Serial.begin(9600);
```

Bagian kode berikut akan menghubungkan papan ESP32 ke jaringan lokal yang kredensial jaringannya telah ditentukan di atas. Pertama-tama kita akan mengatur modul ESP32 dalam *mode acces point* dan kemudian menggunakan fungsi *WiFi*.begin(). Argumennya adalah SSID dan kata sandi yang ditentukan sebelumnya di kode. Setelah koneksi berhasil dibuat, alamat IP akan ditampilkan di server web.

```
//=======  Set as an ACCES POINT ============
   Serial.print("Setting AP (Access Point)…");
   WiFi.softAP(ssid, password);
   IP = WiFi.softAPIP();
   Serial.print("AP IP address: http://"); ----
 Serial.println(IP);
  delay(1000);
```

6) Root URL

Permintaan URL root yang akan diterima oleh papan ESP dengan menggunakan metode send_P. Metode ini akan mengambil tiga parameter. Yang pertama adalah 200 yang merupakan kode status HTTP untuk 'ok'. Yang kedua adalah “teks/html” yang sesuai dengan tipe konten respons. Input ketiga adalah teks yang disimpan pada variabel index_html yang akan dikirimkan sebagai reespon.

```
 // Route for root / web page
  server.on("/", HTTP_GET, [](AsyncWebServerRequest * request)
{
     request->send(SPIFFS, "/index.html", String(), false,
processor);
  });
  server.on("/start", HTTP_GET, [](AsyncWebServerRequest
*request){
    relayState=1;
    preferences.putBool("state", relayState);
    pzem.resetEnergy(); //reset nilai kWh menjadi 0.00
    request->send(SPIFFS, "/index.html", String(), false,
processor);
  });
server.on("/stop", HTTP_GET, [](AsyncWebServerRequest
*request){
    relayState=0;
    preferences.putBool("state", relayState);
    request->send(SPIFFS, "/index.html", String(), false,
processor);
  });
```

7) Get URL

Setiap kali pengguna memasukkan data apa pun di kolom input dan menekan tombol submit, permintaan HTTP GET dikirim pada URL '/get'. Baris kode berikut akan memeriksa apakah permintaan yang diterima pada URL tertentu berisi parameter masukan yang awalnya kita tentukan. Pertama, kita akan mendefinisikan dua variabel string bernama *'input_message'* dan

'input_parameter.' Yang pertama akan menyimpan nilai input (*string*, int atau *float*) dan yang kedua akan menyimpan kolom *input*.

Melalui pernyataan *if-else if* kita akan memeriksa permintaan HTTP dibuat pada parameter *input* mana dari kelimanya (*input*_Hari, *input*_Jam, *input* Menit, **input** detik atau *input*_arus). Nilai yang dimasukkan oleh pengguna untuk setiap kolom masukan akan diperbarui dalam variabelnya masing-masing (*input*_parameter hari, jam, menit, detik, dan *input*_parameter_Arus). Selain itu, *'input_message'* yang merupakan nilai yang dimasukkan oleh pengguna di kolom input juga akan ditampilkan di monitor serial.

```
-server.on("/get", HTTP_GET, [] (AsyncWebServerRequest --
*request) {
     String inputMessage;
     if (request->hasParam(parameter_Hari)) {
      inputMessage = request->getParam(parameter_Hari)-
>value();
       write_file(SPIFFS, "/input_Hari.txt", -
inputMessage.c_str());
    }
     else if (request->hasParam(parameter_Jam)) {
      inputMessage = request->getParam(parameter_Jam)-
>value();
     write_file(SPIFFS, "/input_Jam.txt",
inputMessage.c_str());
    }

     else if (request->hasParam(parameter_Menit)) {
      inputMessage = request->getParam(parameter_Menit)-
>value();
       write_file(SPIFFS, "/input_Menit.txt",
inputMessage.c_str());
    }
     else if (request->hasParam(parameter_Detik)) {
      -inputMessage = request->getParam(parameter_Detik)-
>value();
       write_file(SPIFFS, "/input_Detik.txt",
inputMessage.c_str());
    }
     else if (request->hasParam(parameter_Arus)) {
       inputMessage = request->getParam(parameter_Arus)-
>value();
      -write_file(SPIFFS, "/input_Arus.txt",
inputMessage.c_str());
    }

     else {
      inputMessage = "No message sent";
    }
     Serial.println(inputMessage);
     request->send(200, "text/text", inputMessage);
  });
  server.begin();
```

8) Membuat Halaman Web

Pada halaman web dibuat variabel index_html untuk menyimpan teks HTML. Dimulai dengan judul halaman web. Tag <title> akan menunjukkan awal judul dan tag </tile> akan menunjukkan akhir. Di antara tag ini, kami akan menentukan "Penyambungan Sementara PLN" yang akan ditampilkan di bilah judul browser. <title> *Form* HTML untuk Input Data </title>.

Selanjutnya dibuatkan meta tag untuk memastikan server web kita tersedia untuk semua browser misalnya smartphone, laptop, komputer dll.

```
/* ========= File index.html =================

<!DOCTYPE HTML><html><head>

  <title>Penyambungan Sementara PLN</title>

  <meta name="viewport" content="width=device-width, initial-
scale=2">
```

Langkah menentukan badan halaman web HTML. Ini akan masuk ke dalam tag <body></body> yang menandai awal dan akhir skrip. Bagian ini akan mencakup judul halaman web dan kolom input. Judul halaman *web* berada di dalam tag <h2></h2> dan itu akan menjadi "*Form* HTML untuk Input Data"

<h2> Form HTML untuk Input Data </h2>

Di bawah judul, akan ditambahkan input pengguna yang perlu diisi. Ini akan bertipe 'formulir'. Mendefinisikan tindakannya sebagai "/get" yang berarti permintaan HTTP GET akan dikirim ketika pengguna memasukkan kolom dan menekan tombol kirim. Saat menampilkan lima kolom masukan pada halaman web, akan didefinisikan lima formulir terpisah. Setiap formulir terdiri dari kotak input dan tombol kirim (submit).

Pada dasarnya kita memiliki lima kolom input yang masing-masing bertipe teks yang memperbarui variabelnya masing-masing setiap kali nilai valid dimasukkan ke dalam kotak dan tombol submit ditekan.

Adapun alamat IP untuk penginputan *Form* HTML Penyambungan sementara ini yaitu 192.168.4.1. Pada halaman *Form* HTML terdapat beberapa parameter untuk menunjukkan batas waktu dan batas arus penyambungan antara lain parameter Hari, Jam, Menit, detik, dan batas arus penyambungan. Parameter Hari menunjukkan Batasan hari berahirnya masa penyambungan, begitupun dengan parameter jam, menit, detik menunjukkan waktu berahirnya masa penyambungan sementara. Parameter arus menunjukkan batasan arus maksimal yang dapat dilalui oleh alat tersebut. Dapat disimpulkan bahwa parameter pada *form* HTML seperti gambar dibawah ini merupakan pengaturan batasan penyambungan sementara.



Gambar 4.7 *Form* HTML Untuk Penginputan Data

c. Pemrograman Batas Arus Penyambungan

Pada pemrograman ini Relay bypass adalah relay yang bekerja saat periode penyambungan sementara masih berlangsung sedangkan Relay Normal adalah relay yang bekerja pada saat Penyambungan sementara telah melewati batas waktu yang telah ditentukan dan kWh *kilo Watt hour* (kWh) meter berfungsi kembali seperti semula. Pada pemrograman ini relay bypass akan *OFF* pada saat arus yang mengalir melebihi batas arus yang telah disubmit sebelumnya pada form HTML. Relay akan *ON* Kembali setelah 20 (duapuluh) detik.

Library PZEM004Tv30.h digunakan untuk implementasi sensor PZEM004TV30 untuk membaca arus, tegangan dan *energy* (pemakaian daya listrik).

```
#include <PZEM004Tv30.h>
PZEM004Tv30 pzem(Serial2, 16, 17); // RX, TX
```

Untuk menampilkan output di laya LCD yang dilengkapi dengan modul I2C, digunakan *library LiquidCrystal*_I2C.h. yang dapat digunakan untuk semua dimensi LCD alfanumerik.

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
```

Adapun pemrogramannya batas arus sebagai berikut :

```
//Kendalikan Relay_ByPass
  if(hari==Hari) {
     if(jam==Jam && mnt==Menit && dtk>=Detik) stateOFF();
     if(jam==Jam && mnt>Menit) stateOFF();
     if(jam>Jam) stateOFF();
  }
    if(!relayState) { digitalWrite(Relay_ByPass, 0);
                      digitalWrite(Relay_Normal, 1);
                      OnState = "OFF"; }
  else { if(current>=Arus) {
              digitalWrite(Relay_ByPass, 0);
              delay(20000);
          }
          else digitalWrite(Relay_ByPass, 1);
          digitalWrite(Relay_Normal, 0);
          OnState = "ON";
        }
  if(dtk%5==0) {
    Serial.println();
    Serial.println("Data Tersimpan : ");
    Serial.println("Hari : "+ dayOfWeek[Hari]);
    Serial.println("Jam : "+ String(Jam));
    Serial.println("Menit : "+String(Menit));
    Serial.println("Detik : "+String(Detik));
    Serial.println("Arus : "+String(Arus));
    Serial.println();
    Serial.print("AP IP address: http://");
Serial.println(IP);
    Serial.println();
  }
```

d. Pemrograman Batas Waktu Penyambungan

Pengaturan waktu hari, jam, menit dan detik menggunkan modul RTC DS3231 yang inisialisasinya membutuhkan *library*. Pada program ini memakai *library* Wire.h dan RTClib.h.

```
#include <Wire.h>
#include "RTClib.h"
RTC_DS3231 rtc;
```

Ada tiga pin I/O dari ESP32 yang perlu dideklarasikan yaitu pin 19 untuk modul relay ByPass, pin 23 untuk modul relay_Normal dan pin 14 untuk tombol pilih tampilan.

```
#define Relay_ByPass 19
#define Relay_Normal 23
#define SW 14
```

Agar kita dapat menampilkan nama hari dengan lebih sederhana maka digunkan variabel *array String dayOfWeek*. Semua data yang diproses melalui harus dinyatakan sebagai variabel string. Oleh sebab itu kita perlu membuatkan deklarasi variabel-variabel yang dimaksud sebagai berikut:

```
String dayOfWeek[]= {"Minggu", "Senin", "Selasa", "Rabu",
"Kamis", "Jum'at", "Sabtu"};
String sHari, sTgl, sBln, sThn, sJam, sMnt, sDtk;
String dspHari, dspJam, dspArus;
String dJam, dMnt, dDtk, dHari, dJM, dArus;
```

Selain variabel string, juga dibuat beberapa variabel numerik untuk menampung data dari RTC dan sensor.

```
int Hari, Jam, Menit, Detik;
float Arus;
uint8_t hari, tgl, bln, jam, mnt, dtk;
uint16_t thn;
bool relayState, tampilan;
String onState, OnState;
float voltage, current, energy, kWhAwal, kWhAhir;
```

Untuk batas waktu penyambungan telah diatur berdasarkan hari, jam, menit, detik. Apabila waktu telah melewati batas waktu penyambungan yang telah ditentukan sebelumnya maka secara otomatis relay Bypass akan bekerja (*OFF*) dan relay normal akan bekerja (*ON*) dan kWh *kilo Watt hour* (kWh) meter dirumah Pelanggan dapat beroperasi kembali seperti semula. Adapun pemrogramannya sebagai berikut :

```
//Kendalikan Relay_ByPass
  if(hari==Hari) {
     if(jam==Jam && mnt==Menit && dtk>=Detik) stateOFF();
     if(jam==Jam && mnt>Menit) stateOFF();
     if(jam>Jam) stateOFF();
  }
    if(!relayState) { digitalWrite(Relay_ByPass, 0);
                      digitalWrite(Relay_Normal, 1);
                      OnState = "OFF"; }
  else { if(current>=Arus) {
              digitalWrite(Relay_ByPass, 0);
              delay(20000);
          }
          else digitalWrite(Relay_ByPass, 1);

          digitalWrite(Relay_Normal, 0);
          OnState = "ON";
        }
  if(dtk%5==0) {
    Serial.println();
    Serial.println("Data Tersimpan : ");
    Serial.println("Hari : "+ dayOfWeek[Hari]);
    Serial.println("Jam : "+ String(Jam));
    Serial.println("Menit : "+String(Menit));
    Serial.println("Detik : "+String(Detik));
    Serial.println("Arus : "+String(Arus));
    Serial.println();
    Serial.print("AP IP address: http://");
Serial.println(IP);
    Serial.println();
  }
```

e. Pemrograman Tampilan Pada LCD

Pada saat alat dihidupkan, maka tampilan awal pada lcd diatur :

```
 void startLCD() {
  lcd.init(); lcd.backlight();
  lcd.setCursor(0,0);  lcd.print(F("SISTEM PENYAMBUNGAN "));
  lcd.setCursor(0,1);  lcd.print(F("   SEMENTARA PLN    "));
  lcd.setCursor(0,2);  lcd.print(F(" BERBASIS WEBSERVER "));
  lcd.setCursor(0,3);  lcd.print(F(" ACCES POINT  ESP32 "));
  delay(2000);
  lcd.clear();
```

Pada LCD akan tampil hari , jam, menit, detik dan nilai arus secara *realtime* serta Total energi Selama periode penyambungan. Ditampilan kedua akan ditampilkan batas Penyambungan berupa batas hari, jam, menit, detik dan batas arus. Begitupun saat arus melebihi batas arus yang ditentukan pada LCD akan tampil nilai arus lebih selama 20 (duapuluh detik) sampai relay Bypass ON kembali.

```
void loop() {
  Hari = read_file(SPIFFS, "/input_Hari.txt").toInt();
  Jam = read_file(SPIFFS, "/input_Jam.txt").toInt();
  Menit = read_file(SPIFFS, "/input_Menit.txt").toInt();
  Detik = read_file(SPIFFS, "/input_Detik.txt").toInt();
  Arus = read_file(SPIFFS, "/input_Arus.txt").toFloat();
bacaRTC();
  bacaSensor();
  if(digitalRead(SW)) {
    if(!tampilan) {lcd.clear(); tampilan=1;}
    tampilkanWaktu();
```

```
 if(hari==Hari) {
     if(jam==Jam && mnt==Menit && dtk>=Detik) stateOFF();
     if(jam==Jam && mnt>Menit) stateOFF();
     if(jam>Jam) stateOFF();
  }
  if(!relayState) { digitalWrite(Relay_ByPass, 0);
                    digitalWrite(Relay_Normal, 1);
                    OnState = "OFF"; }
  else { if(current>=Arus) {
             digitalWrite(Relay_ByPass, 0);
             delay(20000);
         }
         else digitalWrite(Relay_ByPass, 1);

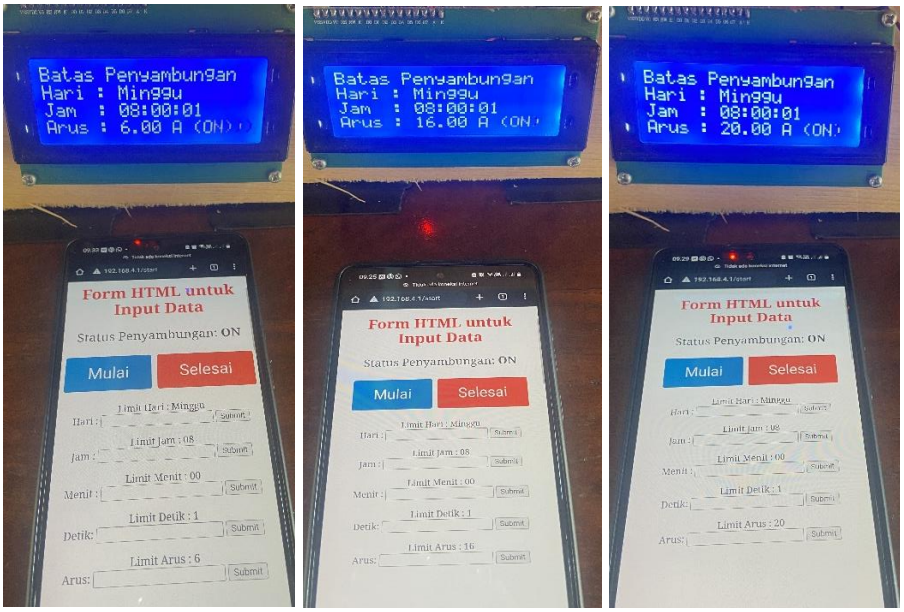
         digitalWrite(Relay_Normal, 0);
         OnState = "ON";
       }

  if(dtk%5==0) {
    Serial.println();
    Serial.println("Data Tersimpan : ");
    Serial.println("Hari : "+ dayOfWeek[Hari]);
    Serial.println("Jam : "+ String(Jam));
    Serial.println("Menit : "+String(Menit));
    Serial.println("Detik : "+String(Detik));
    Serial.println("Arus : "+String(Arus));
    Serial.println();
    Serial.print("AP IP address: http://");
Serial.println(IP);
    Serial.println();
  }
  //Menampilkan Data pada Layar Monitor
  Serial.println(Hari+"  "+Jam);
  Serial.println(Arus);
  delay(1000);
```

B. Pengujian Perangkat

1. Pengujian pengiriman perintah dari *Form input HTML* ke *Mikrokontroler ESP32* melalui *Web Server Mode Access Point (AP)*

Pengujian pengiriman perintah dari aplikasi *Web HTML* ke *Mikrokontroller* ESP32 melalui *webserver mode accespoint* dengan cara membandingkan perintah batas penyambungan yang diinputkan pada form *HTML* dengan tampilan LCD pada alat yang berupa data hari penormalan, waktu penormalan dan batasan arus penyambungan. Tahap ini dilakukan untuk mengetahui apakah penginputan data pada form *HTML* sesuai dengan data yang diterima pada *mikrokontroller* melalui *Webserver Acces Point.*



Gambar 4.8 Tampilan batas waktu penyambungan *Form* HTML dengan LCD

Tabel. 4.1 Pengujian pengiriman perintah dari *Web* HTML ke *Mikrokontroller* ESP32 melalui *Webserver Acces Point*

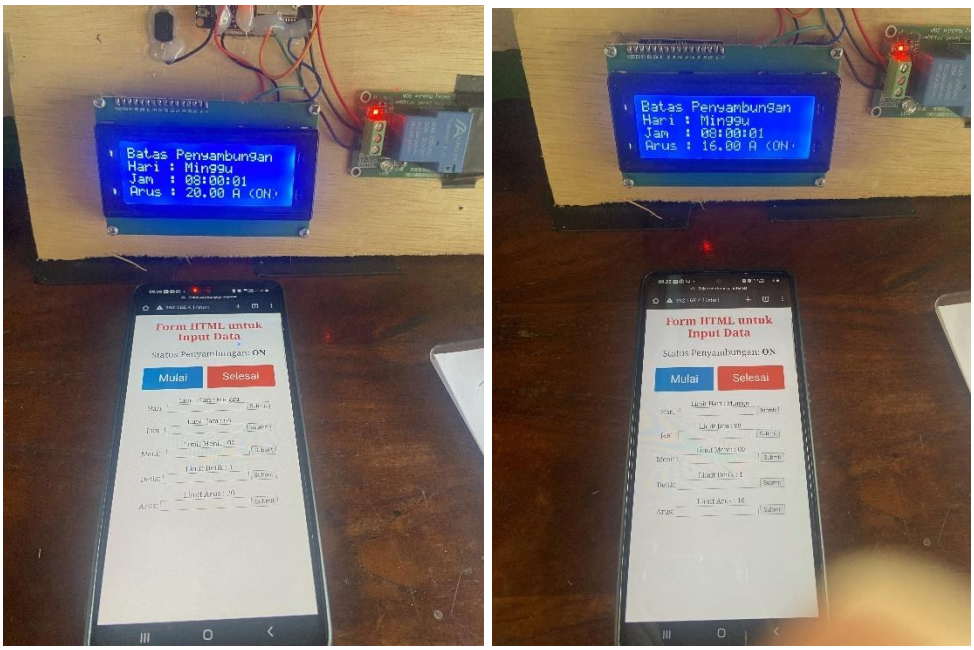
No.	Penginputan pada web HTML		Tampilan pada layar LCD		Ket.
	Hari/Jam	Batas Arus (A)	Hari/Jam	Batas Arus (A)	
1	Minggu/08:00:01	6	Minggu/08:00:	6	Sesuai
2	Minggu/08:00:01	10	Minggu/08:00:	10	Sesuai
3	Minggu/08:00:01	16	Minggu/08:00:	16	Sesuai
4	Minggu/08:00:01	20	Minggu/08:00:	16	Sesuai
5	Minggu/08:00:01	25	Minggu/08:00:	25	Sesuai
6	Rabu/18:30.01	3	Rabu/18:30.01	3	Sesuai
7	Kamis/15:20:01	20	Kamis/15:20:0	20	Sesuai
8	Kamis/10:35:01	20	Kamis/10:35:0	20	Sesuai
9	Jumat/16:00:01	16	Jumat/16:00:0	16	Sesuai
10	Selasa/14:00:01	20	Selasa/14:00:0	20	Sesuai

Pengujian dilakukan sebanyak 10 (sepuluh) kali dengan menginputkan data yang berbeda disetiap pengujiannya. Dari ke sepuluh hasil pengujian tersebut didapatkan data yang sesuai antara data yang diinputkan melalui Form *HTML* dengan data yang ditampilkan pada *LCD.*

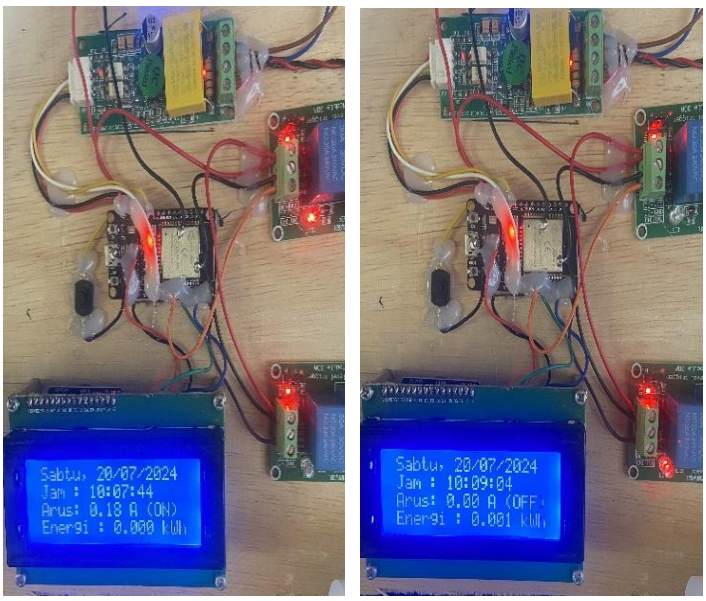
2. Pengujian Batas Waktu Penyambungan

Pengujian waktu penormalan penyambungan dilakukan dengan membandingkan setting batas waktu penyambungan dengan realisasi waktu penormalan penyambungan pada alat yang juga ditampilkan melalui LCD. Adapun data batas penyambungan berupa hari, jam, menit, detik, dan batas arus

penyambungan. Tahap ini dilakukan untuk mengetahui apakah data batas penyambungan yang telah diinputkan pada *form* HTML sesuai dengan realisasi waktu penormalan pada alat.



Gambar 4.9 Tampilan batas penyambungan pada *form* HTML dan LCD



Gambar 4.10 Tampilan posisi *ON* dan *OFF* pada LCD

Pengujian dilakukan sebanyak 10 (sepuluh) kali dengan sampel pengujian sebagai berikut:

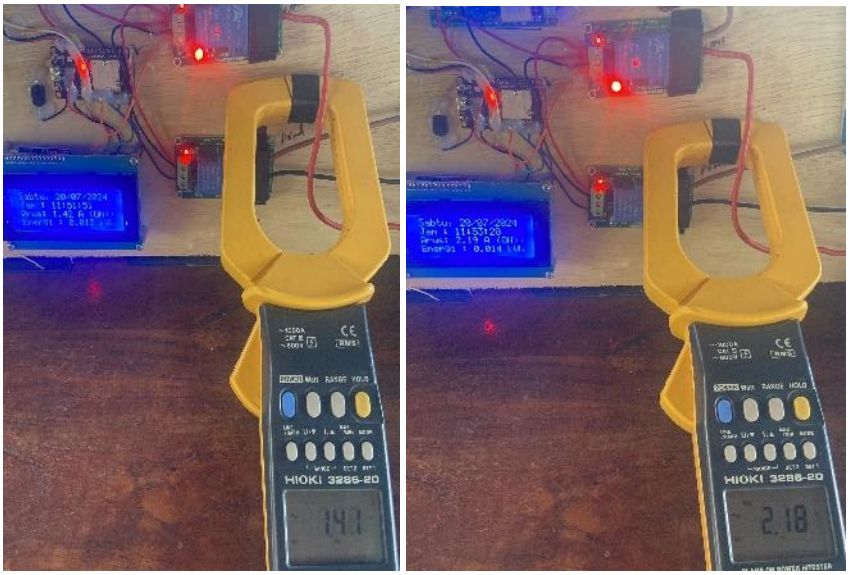
Tabel 4.2 Pengujian batas waktu penyambungan

No.	Pengaturan waktu penormalan pada form HTML		Realisasi waktu penormalan alat			Ket.
	Hari	Jam	Hari	Jam	Pemakaian kWh	
1	Sabtu	09:51:01	Sabtu	09:51:01	0,001	Sesuai
2	Sabtu	10:05:01	Sabtu	10:05:01	0,001	Sesuai
3	Sabtu	10:09:01	Sabtu	10:09:01	0,001	Sesuai
4	Sabtu	10:13:01	Sabtu	10:13:01	0,001	Sesuai
5	Sabtu	10:17:01	Sabtu	10:17:01	0,002	Sesuai
6	Sabtu	10:22:01	Sabtu	10:22:01	0,003	Sesuai
7	Sabtu	10:27:01	Sabtu	10:27:01	0,001	Sesuai
8	Sabtu	10:30:01	Sabtu	10:30:01	0,001	Sesuai
9	Selasa	13:32:01	Selasa	13:32:01	0,002	Sesuai
10	Rabu	13:35:52	Rabu	13:35:52	1,525	Sesuai

Pengujian dilakukan sebanyak 10 (sepuluh) kali dengan menginputkan data yang berbeda disetiap pengujiannya. Dari ke sepuluh hasil pengujian tersebut didapatkan data yang sesuai antara data batas waktu penyambungan yang diinputkan melalui *Form* HTML dengan realisasi waktu penormalan pada alat.

3. Pengujian Pembacaan *Sensor PZEM004T* Pada Pembacaan Arus

Pegujian pembacaan Arus dilakukan dengan membandingkan hasil pengukuran antara pembacaan arus sensor *PZEM* 004T dengan alat ukur Ampere Meter yang memiliki tingkat akurasi dasar Arus ±1.3 % rdg. ±3 dgt. (*at 45 – 66 Hz, True RMS*), kapasitas pengukuran arus antara 0 A – 1000 A, kapasitas pengukuran tegangan 0 V – 600 V. Tahap ini dilakukan agar kita dapat mengetahui apakah sensor arus yang digunakan dapat berfungsi dengan baik.



Gambar 4.11 Tampilan posisi *ON* dan *OFF* pada *LCD*

Pengujian dilakukan dengan menggunakan 10 (sepuluh) kali pembacaan arus dengan nilai peralatan Listrik dan nilai arus yang berbeda. Adapun hasil pengukurannya dituangkan dalam tabel pengujian dibawah ini.

Tabel 4.3 Pengujian arus pada sensor PZEM004T

No.	Beban	Sensor Arus (A)	Ampere Meter (A)	Selisih (A)	Error (%)
1	Kipas Angin	0.18	0.18	0	0
2	Bor Listrik	1.42	1.41	0.01	0.7
3	Blower	2.19	2.18	0.01	0.4
4	Hot M Glue	0.12	0.12	0	0
5	Setrika	1.58	1.56	0.02	1.2
6	Blender	0.89	0.88	0.01	1.1
7	Charger HP	0.14	0.14	0	0
8	Bor Listrik	0.95	0.93	2.1	2.1
9	Kipas Angin	0.44	0.43	0.01	2.3
10	Blower	1.96	1.93	0.03	1.5
Rata-rata persentase kesalahan					**0.93**

Berdasarkan tabel diatas, perhitungan persentase error dan presentase rata – rata error didapat pada persamaan berikut ini :

$$Error\ (\%) = \frac{(Arus\ 1 - Arus\ 2)}{Arus\ 2}\ x\ 100\% \quad \text{.....(1)}$$

$$Rata - rata\ Error\ (\%) = \frac{Jumlah\ Nilai\ Error}{Banyaknya\ Error\ yang\ Terjadi} \quad \text{.....(2)}$$

1. Pengambilan Data pada beban kipas angin
   - $Error\ (\%) = \frac{(0{,}18 - 0{,}18)}{0{,}18}\ x\ 100\% = 0\ \%$
2. Pengambilan Data pada beban bor Listrik
   - $Error\ (\%) = \frac{(1{,}42 - 1{,}41)}{1{,}41}\ x\ 100\% = 0{,}7\ \%$
3. Pengambilan Data pada beban Elektrik blower
   - $Error\ (\%) = \frac{(2{,}19 - 2{,}18)}{2{,}18}\ x\ 100\% = 0{,}4\ \%$

4. Pengambilan Data pada beban Hot Melt Glue Gun

- *Error (%)*$=\frac{(0,12-0,12)}{0,12}$ *x 100% = 0 %*

5. Pengambilan Data pada beban Setrika

- *Error (%)*$=\frac{(1,58-1,56)}{1,56}$ *x 100% = 1,2 %*

6. Pengambilan Data pada beban Blender

- *Error (%)*$=\frac{(0,89-0,88)}{0,88}$ *x 100% = 1,1 %*

7. Pengambilan Data pada beban Charger HP

- *Error (%)*$=\frac{(0,14-0,14)}{0,14}$ *x 100% = 0 %*

8. Pengambilan Data pada beban Bor Listrik

- *Error (%)*$=\frac{(0,95-0,93)}{0,93}$ *x 100% = 2,1 %*

9. Pengambilan Data pada beban Kipas Angin

- *Error (%)*$=\frac{(0,44-0,43)}{0,43}$ *x 100% = 2,3 %*

10. Pengambilan Data pada beban Blower

- *Error (%)*$=\frac{(1,96-1,93)}{1,93}$ *x 100% = 1,5 %*

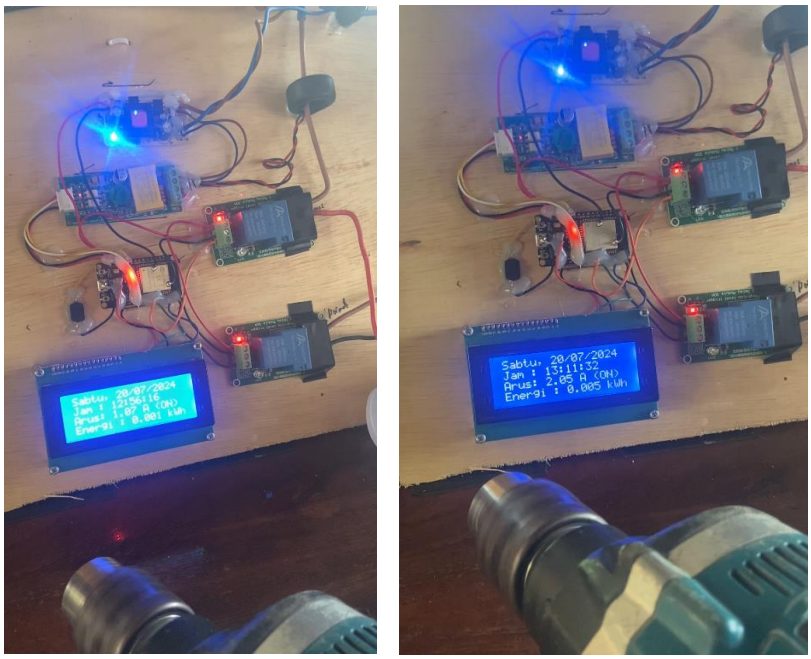
11. Rata-rata Error

Rata – rata Error(%)$=\frac{9,3}{10}$ *= 0,93 %*

Dari pengujian diatas didapatkan hasil *error* minimum yakni 0%, error maksimum sebesar 2,3 % dan rata – rata error 0,93%. Dari kemampuan msengukur arus, sensor tersebut layak untuk di gunakan guna melakukan penelitian yang lebih lanjut.

4. Pengujian Batas Arus

Pengujian batas arus dilakukan dengan menginputkan batas arus pada *form* HTML dengan batas arus 1 A, 1,6 A, 2 A, 2,5 A dan 3 A pada masing-masing pengujian. Peralatan Listrik yang digunakan sebagai beban Listrik untuk pengujian yaitu antara lain bor Listrik, setrika, Elektrik *Blower*, dan kipas angin. Tahap ini dilakukan untuk mengetahui alat tersebut dapat membatasi arus Listrik sesuai dengan nilai batas arus yang telah diinputkan pada *form* HTML.



Gambar 4.12 Pemberian beban listrik pada alat

Dibawah ini merupakan hasil pengujian batas arus dengan cara memberikan beban dibawah batas arus dan beban yang melebihi batas arus.

Tabel 4.4 Hasil pengujian batas arus 1 A

No.	Pembacaan Arus		Keterangan
	Arus (A)	**Status Relay (ON/OFF)**	
1	0,95	ON	Normal
2	0,92	ON	Normal
3	0,98	ON	Normal
4	0,73	ON	Normal
5	0,93	ON	Normal
6	1,07	OFF	Melebihi batas Arus
7	1,05	OFF	Melebihi batas Arus
8	1,06	OFF	Melebihi batas Arus
9	1,04	OFF	Melebihi batas Arus
10	1,03	OFF	Melebihi batas Arus

Tabel 4.5 Hasil pengujian batas arus 1,6 A

No.	Pembacaan Arus		Keterangan
	Arus (A)	**Status Relay (ON/OFF)**	
1	1,57	ON	Normal
2	1,34	ON	Normal
3	1,30	ON	Normal
4	1,47	ON	Normal
5	1,49	ON	Normal
6	1,84	OFF	Melebihi batas Arus
7	2,05	OFF	Melebihi batas Arus
8	1,80	OFF	Melebihi batas Arus
9	1,81	OFF	Melebihi batas Arus
10	1,76	OFF	Melebihi batas Arus

Tabel 4.6 Hasil pengujian batas arus 2 A

No.	Pembacaan Arus		Keterangan
	Arus (A)	**Status Relay (ON/OFF)**	
1	1,77	ON	Normal
2	1,76	ON	Normal
3	1,87	ON	Normal
4	1,79	ON	Normal
5	1,96	ON	Normal
6	2,05	OFF	Melebihi batas Arus
7	2,01	OFF	Melebihi batas Arus
8	2,23	OFF	Melebihi batas Arus
9	2,03	OFF	Melebihi batas Arus
10	2,14	OFF	Melebihi batas Arus

Tabel 4.7 Hasil pengujian batas arus 2,5 A

No.	Pembacaan arus ditampilan LCD		Keterangan
	Arus (A)	**Status (ON/OFF)**	
1	2,37	ON	Normal
2	2,20	ON	Normal
3	2,26	ON	Normal
4	2,43	ON	Normal
5	2,21	ON	Normal
6	2,79	OFF	Melebihi batas Arus
7	2,64	OFF	Melebihi batas Arus
8	2,75	OFF	Melebihi batas Arus
9	2,56	OFF	Melebihi batas Arus
10	2,74	OFF	Melebihi batas Arus

Tabel 4.8 Hasil pengujian batas arus 3 A

No.	Pembacaan arus ditampilan LCD		Keterangan
	Arus (A)	**Status (ON/OFF)**	
1	2,92	ON	Normal
2	2,67	ON	Normal
3	2,68	ON	Normal
4	2,72	ON	Normal
5	2,87	ON	Normal
6	3,41	OFF	Melebihi batas Arus
7	3,94	OFF	Melebihi batas Arus
8	3,08	OFF	Melebihi batas Arus
9	3,02	OFF	Melebihi batas Arus
10	3,31	OFF	Melebihi batas Arus

Dari 50 (lima puluh) kali pengujian batas arus diatas didapatkan hasil bahwa apabila arus yang mengalir lebih kecil dari setting batas arus maka alat tersebut akan berfungsi Normal (*ON*) untuk menghantarkan aliran Listrik ke beban dan sebaliknya apabila arus yang mengalir lebih besar dari setting batas arus maka alat tersebut akan berfungsi sebagai pemutus (*OFF*).

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan alat Sistem kendali dan *Monitoring* Penyambungan Sementara PLN Berbasis *Minkrokontroller ESP32* yang telah dirancang dapat diambil kesimpulan bahwa, Sistem tersebut dapat menggantikan MCB (*Miniature Circuit Breaker*) untuk membatasi dan memutus arus listrik berbasis *mikrokontroler* ESP32. Dengan alat tersebut pengaturan batas arus nominal dan waktu penyambungan sementara dapat dikendalikan dengan bebasis *Mikrokontroler* ESP32. Dengan alat tersebut pemakaian energi listrik pada periode penyambungan sementara dapat diketahui dengan berbasis *Mikrokontroler* ESP32.

B. Saran

Untuk penelitian selanjutnya penulis memberikan saran agar dapat dilakukan pengembangan antara lain:

1. Dapat melakukan penggantian nama jaringan dan password *webserver Acces Point* tanpa harus merubah program.
2. Meningkatkan keamanan pada saat mengakses jaringan *webserver Acces Point* dengan menambahkan *Username* dan *password* setelah terkoneksi ke *webserver*.

3. Menggunakan *module relay* arus yang lebih besar untuk kapasitas beban listrik yang lebih besar.

DAFTAR PUSTAKA

Christopher, dkk (2021). *Perancangan dan Implementasi Perangkat Akuisisi Data dan Regulasi Daya Pada Sistem.* Jurnal Electron.2(1):34-40.

Darma, Surya, dkk (2019). *Studi Sistem Peneraan kWH Meter.* Journal of Electrical Technology. 4(3):2598

Erick, Yosua. (2021). Pengertian Listrik, Jenis, Fungsi, Besaran, Sejarah, dan Manfaat. Diperoleh dari: https://stellamariscollege.org/listrik/. (Diakses 14 Agustus 2023).

Fernando, Rio, dkk. (2018). *Sistem Kendali Dan Monitoring Penggunaan Daya Listrik Pada Perangkat Listrik Rumah Berbasis IOT. e-Proceeding of Engineering*. 5(3):4235.

Hanif, Abdul. (2022). *Rancangan Sistem Kontrol Dan Monitoring Instalasi Listrik Berbasis Internet Of Things.* Tugas Akhir. Universitas Islam Negeri Ar-Raniry.

Iqbal, M. (2022). Pengertian *Mikrokontroler ESP32*. Diperoleh dari: https://miqbal.staff.telkomuniversity.ac.id/mikrokontroler-esp32. (Diakses 14 Agustus 2023).

Kustiawan, Eko. (2018).*Meningkatkan Efisiensi Peralatan Dengan Menggunakan Solid State Relay (SSR) Dalam Pengaturan Suhu Pack Pre-Heating Oven (PHO)*. Jurnal STT Yuppentek. 9(1):1-6.

Natzir, M, dkk. (2019). *Implementasi IOT Untuk Sistem Kendali AC Otomatis Pada Ruang Kelas Di Universitas Serang Raya*. Jurnal Prosisko. 6(1):2597-9922.

Ningrum, Rahma Farah, dkk. (2018). *Prototype Alat Pembatas dan Pemutus Arus Listrik Pascabayar Pada Rumah Tangga Berbasis Smartphone. Jurnal Teknik*. 7(2):43-52.

Noor, Fadil Muhammad, dkk (2022). *Rancang Bangun Sistem Pengendali Beban Listrik Skala Rumah Tinggal Berbasis Mikrokontroler ATmega 328P*. *Prosiding The 13th Industrial Research Workshop and National Seminar*

Pratika, Made Toby Sathya; *Rancang Bangun Wireless Relay Dengan Monitoring Daya Listrik Berbasis Internet Of Things.* Jurnal Ilmiah Teknologi dan Komputer. 2(3):2021.

Razor, Aldy. (2020). Modul Relay Arduino: Pengertian, Gambar, Skema, dan Lainnya. Diperoleh dari: https://www.aldyrazor.com/2020/05/modul-relay-arduino.html. (Diakses 14 Agustus 2023).

Saleh, Muhammad, dkk. (2017). *Rancang Bangun Sistem Keamanan Rumah Menggunakan Relay.* Jurnal Teknologi Elektro. 8(2):2086-9479.

Silaban, Ririn Rosmanty. (2020). *Perancangan Sistem Monitoring Dan Kendali Beban Generator Berbasis Internet Of Things*. Tugas Akhir. Universitas Pembangunan Panca Budi.

Supriyadi, Edi; Dinaryati, Siti. (2020). *Rancang Bangun System Monitoring dan Kendali Listrik Rumah Tangga Berbasis ESP8266 NodeMCU*. Sinusoida. 22(4):4593.