

BAB I

PENDAHULUAN

A. Latar Belakang

Energi Listrik adalah bagian dari bentuk energi yang berperan penting dalam kehidupan bermasyarakat sekarang. Seiring dengan kemajuan teknologi, berbagai hasil riset terbaru yang dihasilkan memerlukan listrik berfungsi sebagai salah satu sumber energi utamanya. Dengan demikian, kebutuhan Listrik dalam kehidupan sangat penting semakin meningkat. Kebutuhan listrik di setiap daerah bervariasi dari waktu ke waktu, tergantung pada pemakaian listrik di daerah tersebut. Oleh karena itu, penyediaan tenaga listrik dan alokasi pembangkit yang digunakan juga berbeda antara satu daerah dengan daerah lainnya.

Permintaan tenaga kelistrikan setiap waktu nya berbeda-beda, oleh karena itu Perusahaan Listrik Negara (PT. PLN) sebagai Perusahaan BUMN penyedia Listrik satu-satunya harus mampu mengalokasikan dan menganalisa kebutuhan penggunaan Listrik setiap hari. . Prediksi dan analisa dari kebutuhan energi listrik juga dapat membantu penyelenggara jasa penyediaan energi listrik (PT. PLN) agar kebutuhan akan energi listrik dan ketersediaan Listrik dapat merata dan terpenuhi . Sehingga terjadi efisiensi biaya operasional pembangkitan listrik yang disebabkan oleh daya yang disalurkan dari pembangkit ke pelanggan sesuai dengan kebutuhan energi Listrik yang dibutuhkan. Dengan demikian, dapat mengantisipasi terjadi padam meluas (Black out) keseluruhan yang disebabkan oleh tidak mampunya

operasional pembangkit Listrik menanggung beban yang digunakan Masyarakat / pelanggan setiap waktunya. yang mana, jika hal-hal tersebut terjadi di kemudian hari, dapat merugikan baik pihak penyedia maupun pihak konsumen.

Gardu distribusi merupakan bagian bagian utama sebagai penyalur energi Listrik ke pelanggan. Sistem distribusi ini berfungsi untuk menyalurkan tenaga Listrik yang awalnya 20 kV dari pembangkit dan transmisi menjadi 220 V / 380 V untuk digunakan ke peralatan Listrik konsumen. Adapun tujuan atau pun fungsi dari Gardu distribusi menurunkan tegangan karena peralatan Listrik yang digunakan konsumen umumnya hanya pada tegangan 220 V dan 380 V.

Pengawasan sistem kelistrikan digunakan untuk mengawasi, dan mengendalikan status aktif atau nonaktif dari sebuah perangkat. Pentingnya pemantauan pada gardu distribusi adalah dapat memantau dan menganalisa perubahan beban yang mana setiap gardu distribusi mempunyai beberapa macam kapasitas dan masing-masing kapasitas gardu distribusi mempunyai nilai maksimum arus yang diperbolehkan. .

Dengan menggunakan system monitoring pada gardu distribusi PLN, diharapkan dapat memantau arus dan tegangan listrik pada gardu distribusi yang sedang dimonitoring. Oleh itu, penulis tertarik untuk mengangkat suatu judul penelitian **Prototype Sistem Monitoring Dan Data Logger Pada Gardu Distribusi Berbasis Mikrokontroler** yang mana nantinya arus dan tegangan listrik dapat dipantau secara real-time oleh perangkat *smartphone* dan data disimpan secara berkala untuk tujuan tinjauan lanjutan.

B. Rumusan Masalah

Rumusan masalah yang akan dikaji dalam penelitian ini adalah :

1. Bagaimana cara merancang perangkat dan sistem yang dapat memonitoring arus dan tegangan pada listrik ?
2. Alat dan sensor apa saja yang digunakan untuk tujuan penelitian?

C. Tujuan Penelitian

Tujuan penelitian adalah seperti berikut :

1. Merencanakan perangkat dan sistem yang dapat memonitoring arus dan tegangan pada listrik.
2. Alat dan sensor yang digunakan adalah *Arduino Uno*, *ESP32* dan *PZEM-004T*
3. Membuat Monitoring dan *data logger* arus dan tegangan listrik

D. Batasan Penelitian

Agar pembahasan materi dalam tugas akhir ini lebih fokus dan efektif dalam mencapai hasil yang diinginkan, dibuatlah beberapa batasan masalah sebagai berikut:

1. Alat dan sistem yang dibuat adalah *prototype*
2. Alat dan sistem akan dicoba pada Listrik rumah tangga dan bukan pada gardu distribusi sebagai *Prototype* penelitian
3. Data yang dimonitoring adalah arus dan tegangan listrik.

E. Manfaat Penelitian

Manfaat dari studi ini adalah:

1. Memberikan kemudahan bagi pihak-pihak terkait dalam memonitoring pemakaian listrik

2. Memudahkan masyarakat yang nantinya dapat menerapkan alat ini pada listrik untuk memantau pemakaian listrik sehari-harinya.
3. Untuk peneliti mencapai gelar strata 1 pada jurusan teknik elektro

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Gardu Distribusi



Gambar 2.1 Gardu Distribusi

Gardu Sistem distribusi merupakan istilah yang digunakan di sistem distribusi tenaga Listrik PLN. Yang mana, gardu distribusi terdiri dari beberapa bagian seperti trafo step down distribusi , kabel single core, panel hubung bagi ke pelanggan dan masih banyak lagi. Sistem distribusi ini berfungsi sebagai media penyalur energi Listrik dari Saluran Udara Tegangan Menengah (SUTM) yang awalnya 20 Kv, kemudian disalurkan ke Trafo distribusi step down dari 20 kV menjadi 220 V / 380 V dan kemudian masing-masing fasa R,S,T dibagi melalui Panel Hubung Bagi untuk disalurkan ke konsumen.

2. Listrik

Listrik merupakan suatu bentuk energi yang sangat penting dalam kehidupan kita (Sunawar & Daryanto, 2022). Energi ini terdiri dari muatan positif dan negatif. Arus listrik adalah aliran muatan yang bergerak dari area berpotensi tinggi ke area berpotensi rendah melalui penghantar listrik. Salah satu penghantar listrik yang umum adalah bahan logam, di mana elektron bebas bergerak dari satu atom ke atom logam lainnya. Sementara itu, dalam media air, elektron dibawa oleh elektrolit yang ada di dalamnya.

Terdapat dua jenis arus listrik, yaitu arus searah (DC) dan arus bolak-balik (AC). Arus searah (DC) adalah arus listrik yang mengalir di dalam satu arah secara kontinu. Arus DC sering digunakan dalam ranah industri yang melibatkan proses elektrolisa, seperti proses pembersihan dan penambahan lapisan atau penyepuhan logam.

Arus listrik AC adalah arus yang mengalir secara bolak-balik. Arus AC sering digunakan di rumah-rumah dan pabrik-pabrik, biasanya dengan tegangan 110 volt atau 220 volt. Arus bolak-balik (AC) dianggap lebih berpotensi bahaya dibandingkan dengan arus searah (DC).

3. Tegangan Listrik

Tegangan listrik adalah perbedaan potensial antara dua titik. Tegangan ini dihasilkan oleh pembangkit listrik. Berbagai macam jenis tegangan listrik, seperti tegangan sangat rendah, tegangan rendah, dan lainnya. Penting untuk memahami klasifikasi ini, karena kabel listrik memiliki batas tegangan tertentu yang dapat mereka tangani. Beberapa pengelompokan tegangan listrik mencakup :

- a. Tegangan Sangat Rendah mencakup tegangan hingga 50 V. Sementara itu, Tegangan Rendah (*Low Voltage* = LV) berkisar antara 50 volt hingga 1000 volt (1 KV). Jenis penghantar yang dipakai harus mampu menangani tegangan dalam rentang 50 volt hingga 1000 volt (1 KV).
- b. Tegangan menengah (*Medium Voltage* = MV) adalah tegangan yang berada dalam kisaran 1000 Volt (1 KV) hingga 36.000 Volt (36 KV). Kabel yang digunakan untuk tegangan menengah harus mampu beroperasi dalam rentang tegangan tersebut, yaitu antara 1 KV hingga 36 KV.
- c. Tegangan tinggi (*High Voltage* = HV) berada dalam kisaran 36 KV hingga 150.000 Volt (150 KV). Kabel yang digunakan untuk aplikasi ini juga harus mampu menangani tegangan di atas 36 KV hingga 150 KV
- d. Tegangan ekstra tinggi (*High Extra Voltage* = HEV) mencakup rentang tegangan antara 150 KV hingga 750 KV. Kabel yang digunakan juga harus dirancang untuk menahan tegangan di atas 150 KV hingga 750 KV..
- e. Tegangan ultra tinggi (*High Ultra Voltage* = HUV) merujuk pada tegangan yang melebihi 750 KV. Kabel yang digunakan harus dirancang untuk menangani tegangan di atas 750 KV tersebut.

4. Arus Listrik

Arus listrik adalah pergerakan elektron dari satu atom ke atom lain dalam sebuah penghantar, yang bergerak dengan tempo tertentu selama masa tertentu. Arus listrik muncul karena adanya beda potensial di kedua ujung penghantar, yang terjadi akibat adanya tenaga yang menggerakkan elektron-elektron tersebut untuk beralih lokasi.

Pergerakan elektron akan bergerak menuju area dengan tekanan yang tingkatannya lebih rendah. Perbedaan dalam ukuran arus listrik yang mengalir ditentukan oleh sumber listrik yang memproduksi tenaga tersebut.

Daya pendorong listrik diperlukan bertujuan supaya manusia dapat memanfaatkan sumber daya energi listrik dengan baik, tetapi tenaga ini harus mencukupi dan sesuai dengan jumlah. Oleh karena itu, arus listrik harus dapat mengalir dan terputus dan pergerakan yang konsisten.

Kecepatan aliran arus listrik dikenal sebagai laju aliran, yang dilambangkan oleh I dan diukur dalam satuan daya Ampere. Arus listrik ini terbentuk ketika muatan listrik mengalir setiap detik, sehingga ada hubungan antara muatan listrik, arus listrik, dan waktu.

Dua jenis arus listrik, yaitu arus searah dan arus bolak-balik. Berikut ini adalah penjelasan tentang kedua jenis arus listrik beserta contohnya:

- a. Arus searah atau *Direct current* adalah arus listrik yang nilainya tetap dan tidak berubah, hanya memiliki polaritas positif atau negatif, serta memiliki nilai yang konstan sepanjang waktu.
- b. Arus bolak-balik atau *alternating current* adalah jenis arus listrik yang arahnya berubah-ubah secara bergantian. Gelombang arus AC berbentuk sinusoidal, yang memungkinkan distribusi energi secara efisien. Penggunaan arus AC sangat umum dalam kehidupan sehari-hari, termasuk dalam berbagai perangkat elektronik yang memenuhi kebutuhan manusia. Arus ini dihasilkan oleh pembangkit listrik yang dikenal sebagai generator.

5. *Arduino Uno*



Gambar 2.2 *Arduino Uno*

Arduino Uno adalah papan mikrokontroler yang berbasis data ATmega328 (datasheet) (Wicaksono, 2017). Papan ini memiliki dengan 14 pin masukan / keluaran *digital*, di mana 6 diantaranya dapat dipergunakan untuk keluaran PWM dan 6 pin lainnya sebagai *input analog*. Arduino Uno juga memiliki osilator kristal 16 MHz, koneksi *USB*, *jack power*, *header ICSP*, dan *keypad reset*. Untuk mengelola mikrokontroler, Anda cukup menyambungkan *board Arduino Uno* ke komputer melalui penghubung *USB* atau menggunakan penghubung AC-DC atau baterai untuk sumber listrik. Berikut adalah spesifikasi dari Arduino Uno.

Tabel 2.1 Spesifikasi *Arduino Uno*

Mikrokontroler	<i>ATmega328</i>
Tegangan Operasi	5V
<i>Input Voltage</i> (disarankan)	7-12V
<i>Input Voltage</i> (limit)	6-20V
Pin Digital <i>I/O</i>	14 (6 pin digunakan sebagai output <i>PWM</i>)
Pins Input Analog	6
Arus DC per pin <i>I/O</i>	40 mA

<i>Flash Memory</i>	32KB (<i>ATmega328</i>) 2KB digunakan oleh <i>Bootloader</i>
<i>SRAM</i>	2 KB (<i>ATmega328</i>)
<i>EEPROM</i>	1KB (<i>ATmega328</i>)

Arduino Uno menyediakan berbagai sarana untuk berhubungan dengan komputer, *Arduino Uno* lainnya, atau mikrokontroler yang lain. Mikrokontroler *ATMega328* pada *board* ini mendukung komunikasi serial UART TTL (5V) melalui pin digital 0 (RX) dan 1 (TX). Semua dari 14 pin digital yang ada pada *Arduino Uno* dapat digunakan sebagai *input* atau *output*, dengan memakai kemampuan `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Operasi input/output dilakukan pada tegangan 5 volt, di mana setiap pin dapat mengalirkan atau mendapatkan nilai maksimal 40 mA dan dilengkapi dengan resistor pull-up internal 20-50 k Ω .

6. Modul PZEM-004T



Gambar 2.3 PZEM-004T

PZEM-004T adalah rangkaian elektronik yang memiliki kemampuan untuk menghitung berbagai parameter, termasuk *Volt, Current, Power, frequency, energy*, dan juga *Power factor*. Dengan fitur-fitur tersebut, modul PZEM-004T cukup sesuai untuk dipakai dalam proyek atau eksperimen sebagai alat untuk mengukur daya pada sistem kelistrikan, seperti di rumah atau gedung. Berikut adalah fungsi-fungsi dari modul ini:

- a. Fungsi pengukuran (*voltage / tegangan, current / arus, active power*).
- b. *Power button clear / reset Energy (PZEM-004T V2.0)*
- c. *Power-down data storage function (cumulative power down before saving)*
- d. Komunikasi Serial *TTL*
- e. Pengukuran Daya / Daya : 0 ~ 9999kW
- f. Pengukuran Tegangan / Tegangan : 80 ~ 260VAC
- g. Pengukuran Arus / Arus : 0 ~ 100A

Spesifikasi dari modul adalah seperti berikut :

- a. *Working voltage: 80 ~ 260VAC*
- b. *Rated power: 100A / 22000W*
- c. *Working Frequency: 45-65Hz*
- d. *Measurement accuracy: 1.0*

Modul PZEM-004T begitu sederhana diprogram dan dipakai dengan beberapa jenis *board* mikrokontroler, seperti *Arduino, ESP8266, STM32, WeMos, NodeMCU*, dan *Raspberry Pi*, berkat penggunaan komunikasi serial *TTL*.

7. *Arduino IDE*



Gambar 2.4 *Arduino IDE*

IDE, singkatan dari *Integrated Development Environment*, adalah lingkungan terintegrasi yang dipakai dalam keperluan pengembangan *software*. Dalam hal ini, Arduino IDE ini adalah alat di mana pemrograman dilakukan untuk mengatur fungsi-fungsi perangkat dengan memakai *syntax* pengaplikasian. Arduino mengimplementasikan bahasa pemrograman yang mirip dengan C, tetapi telah dimodifikasi (dikenal sebagai *Sketch*) untuk mempermudah pemula. Sebelum Arduino dijual, IC mikrokontrolernya dilengkapi dengan program dengan nama *Bootloader*, yang bertindak sebagai perantara diantara *compiler Arduino* dan mikrokontroler.

Arduino IDE dibangun menggunakan bahasa pengkodean *JAVA*. IDE ini juga menyertakan rumus *C/C++* dan juga dikenal sebagai *Wiring*, yang mempermudah operasional *input/output*. *Arduino IDE* dikembangkan dari perangkat lunak *Processing* yang dimodifikasi menjadi IDE khusus untuk pemrograman Arduino. Program yang dibuat menggunakan *Arduino Software*

(IDE) disebut sketch, yang tulisan ditulis dengan editor teks dan disimpan dalam *file* dengan ekstensi tertentu.

Teks *editor* dalam perangkat lunak *Arduino* menyediakan atribut yakni pemotongan/penempelan, pencarian/penggantian dan mempermudah sebuah tulisan kode program. Di dalam *Arduino IDE*, terdapat kotak pesan berwarna hitam yang menampilkan status seperti pesan kesalahan, proses kompilasi, dan pengunggahan program. Di sudut kanan bawah IDE *Arduino*, ditunjukkan board yang terkonfigurasi beserta COM Ports yang digunakan. Ketika sebuah sketch berjalan pada board dan menerima konfigurasi atau data lain saat sketch pertama kali dimulai, pastikan software yang berkomunikasi menunggu satu detik setelah membuka koneksi sebelum mengirim data. Untuk pemrograman yang tepat, *Arduino IDE* harus terhubung dengan board *Arduino* yang telah diinstal pada port tertentu. Program dapat diunggah ke mikrokontroler menggunakan kabel USB sebagai media.

8. *ESP32*



Gambar 2.5 *ESP32*

ESP32 merupakan rangkaian papan mikrokontroler serba guna dengan menawarkan atribut komprehensif dan performa yang besar. Ini adalah modul penerus setelah *ESP8266*, yang dikenal sebagai modul *WiFi* terkenal. *ESP32* terdiri dari dua prosesor: satu berguna sebagai pengelola konektivitas *Wi-Fi* dan *Bluetooth*, dan selanjutnya berguna untuk memulai aplikasi. Selain itu, modul ini difasilitasi oleh kapasitas penyimpanan *RAM* yang cukup maksimal berfungsi sebagai penyimpanan datanya.

Fitur-fitur bermanfaat seperti TCP/IP, HTTP, dan FTP tersedia. Selain itu, *ESP32* menawarkan pemrosesan data sinyal analog, *support* sebagai sensor, serta batuan untuk peranti masukan/keluaran (I/O) digital. Selain itu, *ESP32* mendukung konektivitas *Bluetooth*, memungkinkan pengendalian perangkat yang terhubung melalui *Bluetooth*.

ESP32 ideal untuk aplikasi IoT (*Internet of Things*). *Board* tersebut memudahkan koneksi peranti terhubung internet dan juga menangani pemrosesan sinyal analog serta perangkat I/O digital. Tersedia dalam bentuk modul terpisah atau papan sirkuit terpadu (PCB) yang siap pakai, *ESP32* menawarkan kemudahan penggunaan untuk berbagai proyek.

ESP32 adalah mikrokontroler *Wi-Fi* dirancang oleh *Espressif Systems*, menawarkan berbagai macam kegunaan komprehensif dan performa maksimal. Modul ini merupakan penyempurnaan dari *ESP8266*. Dikenal dengan dua sistem prosesor yang mana beroperasi pada pergerakan 80 MHz dan 160 MHz, *ESP32* juga dilengkapi dengan berbagai tambahan peranti seperti *ADC*, *DAC*, *I2C*, *I2S*, *SPI*, dan *UART* untuk mendukung berbagai aplikasi.

9. *Android*

Android merupakan operasi sistem yang dibuat oleh *Google* khususnya untuk perangkat pintar dan juga *tablet*. Berbagai produsen perangkat telah mengadopsi *Android* untuk produk mereka. *Android* juga memiliki toko aplikasi yang melayani milyaran pengguna aktif tiap bulannya, pada Januari 2018. Sejak peluncurannya pada tahun 2008, *Google* telah merilis beberapa versi, dengan "*Pie*" sebagai versi terbaru (Imaduddin & Permana, 2018).

10. *Flutter*

Platform Flutter yang memungkinkan pengembang membuat aplikasi lintas platform hanya dengan menggunakan satu basis kode..Dengan *Flutter*, aplikasi yang dibuat dapat digunakan di berbagai platform, termasuk *Android*, *iOS*, *web*, dan *desktop*. *Flutter* terdiri dari dua komponen utama: *Software Development Kit* (SDK) dan framework antarmuka pengguna.

- a. SDK (*Software Development Kit*) merupakan seperangkat alat yang dibuat untuk membantu dalam pengembangan aplikasi sehingga dapat berfungsi di berbagai *platform*.
- b. Framework UI adalah kumpulan komponen UI, seperti teks, tombol, navigasi, dan elemen lainnya, yang bisa dikustomisasi sesuai kebutuhan Anda.

Flutter adalah platform yang bebas digunakan dan sumber terbuka. Untuk mengembangkannya, Perlu menguasai sistem operasi *dart*, yang fungsinya tidak sama seperti framework front-end lain yang biasanya dipakai untuk JavaScript.

Flutter dikembangkan oleh Google sejak tahun 2015 dan diluncurkan secara resmi pada Desember 2018. Pada tahun 2019, Flutter semakin populer dan banyak pengembang mulai menggunakannya.

11. *Dart*

Dart merupakan pemrograman bahasa yang dibuat oleh *Google* dan menjadi bahasa utama untuk Flutter, sebuah toolkit UI dan platform aplikasi yang juga dibuat oleh Google. Flutter sudah diaplikasikan oleh berbagai Perusahaan besar, karena memungkinkan pengembangan aplikasi di berbagai platform dengan satu codebase, sehingga menghemat waktu dan usaha dibandingkan membuat codebase terpisah untuk setiap *platform*.

- a. *Dart* adalah bahasa yang harus dikuasai untuk mengembangkan aplikasi *Flutter*.
- b. Dengan menggunakan *Dart*, Anda hanya memerlukan satu basis kode untuk membuat program di platform, termasuk *web*, *Android*, dan *iOS*.
- c. Bahasa *Dart* dirancang agar mirip dengan bahasa pemrograman lainnya, sehingga memudahkan bagi mereka yang sudah berpengalaman dengan bahasa pemrograman lain maupun bagi pemula yang baru memulai sebagai *developer*.
- d. Penggunaan bahasa *Dart* tidak memerlukan biaya (*open source*) dan dikembangkan oleh komunitas pengembang berpengalaman yang aktif dan transparan.

- e. *Dart* adalah bahasa yang dirancang untuk memungkinkan pengembangan antarmuka pengguna secara cepat dan efisien di berbagai platform.

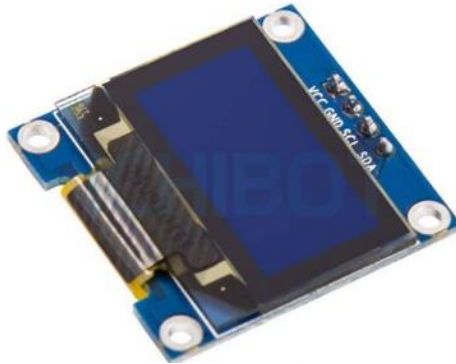
12. Node.JS

Node.js adalah sebuah platform perangkat lunak untuk server dan program *networking*. Platform ini ditulis dalam bahasa *JavaScript* dan dapat dijalankan di berbagai macam jenis prosesor, tanpa perlu mengubah sandi dari program. Node.js mempunyai beberapa *library server HTTP* bawaan, yang dapat membantu anda menjalankan atau mengoperasikan server web tidak lain memerlukan perangkat lunak server web seperti *Apache* atau *Lighttpd*.

Kelebihan Node JS:

1. kode *JavaScript* membuat pembelajaran sebuah server lebih mudah, terutama jika sudah handal di *Javascript* sebelumnya. Anda pun dapat belajar dengan tempo yang singkat disebabkan oleh bahasa yang digunakan sama dengan yang digunakan di sisi klien.
2. Ada peralihan teks kode diantara pengguna dan *server*, yang dikenal sebagai *server-side rendering* dalam kerangka kerja *JavaScript*.
3. Terdapat sarana sebagai bantuan media menciptakan sebuah aplikasi waktu yang realistis (*realtime application*).
4. Berasal dari sumber yang *open source*, yang mana pemakai dapat memahami bagaimana cara aplikasi digunakan, diubah, dan tidak terdapat biaya saat ingin dipakai.

13. *OLED 0.96*



Gambar 2.6 *OLED 0,96*

OLED LCD merupakan salah satu opsi untuk media tampilan pada modul Arduino atau *controller* lainnya. Kelebihannya terletak pada kontras piksel yang sangat tajam dan tidak memerlukan pencahayaan latar belakang, sehingga lebih hemat daya. *OLED* ini cukup menarik untuk digunakan sebagai pengganti LCD biasa. Dengan dukungan komunikasi I2C, hanya membutuhkan 2 PIN saja untuk mengoperasikan *OLED* ini.

B. Penelitian Terdahulu

Penelitian ini terkait dengan studi-studi sebelumnya yang telah dilakukan sebagai bahan perbandingan dan analisis..

Rancang Bangun Alat Monitoring Daya Listrik Berbasis IOT Menggunakan Firebase Dan Aplikasi Android oleh (Widodo, Kholis, & Rakhmaati, 2022) , kesimpulan dari penelitian ini adalah :

Penelitian ini menyimpulkan bahwa alat yang dirancang ini mampu mendeteksi angka tegangan, arus, daya, dan energi listrik yang mengalir ke beban. Alat ini juga dapat mengirimkan data secara *real-time* dan dipantau secara *online* melalui aplikasi MDL.

Penelitian yang dilaksanakan melebihi dari sepuluh kali pada sensor menunjukkan bahwa level nilai *error* dalam pengambilan nilai tegangan memiliki nilai umum *error* kurang lebih 0,35%. Sementara itu, nilai hasil ukur pada arus menampilkan level *error* pengukuran arus mencapai nilai umum kurang lebih 6,7%.

Penilaian hasil ukur daya telah diperoleh dari alat pendeteksi menampilkan adanya kekeliruan dengan nilai *error* kurang lebih 6,6%. Sementara itu, hasil pengambilan nilai ukur energi yang telah diuji selama 5 menit pada setiap arus menampilkan kekeliruan nilai kurang lebih 7,8%. Dalam Gambaran besar, alat pendeteksi ini berfungsi secara maksimal dan memberikan hasil relative tepat .

Sistem Monitoring Daya Listrik dan Pengontrolan Perangkat Elektronik Berbasis IOT oleh (Rendy, 2020). Kesimpulan dari penelitian ini adalah :

Sesuai dengan data yang telah diambil dapat ditarik Kesimpulan bahwa :

1. program yang dirancang mampu mengatur dan memantau alat-alat yang membutuhkan listrik di sektor konsumen melewati hubungan antara NodeMCU dan aplikasi Blynk di *smartphone*.
2. Level ketepatan pengambilan nilai data sensor *PZEM004-T* sebesar 99.9%.
3. Perubahan konektivitas *WiFi* bisa dikerjakan tanpa menggunakan, memodifikasi dan mengatur ulang program pada aplikasi *Arduino IDE*.
4. Sistem akan berfungsi dengan baik jika transformator menghasilkan tegangan 12 V dan modul penurun tegangan mengeluarkan 3,218 V, yang sesuai dengan kebutuhan NodeMCU sebesar 3 V.

5. Pengambilan nilai sensor *PZEM004-T* menunjukkan bahwa arus yang digunakan oleh bohlam adalah 15,90 Watt, dan tegangan yang digunakan pada bohlam itu mencapai 237,200 Volt (dengan beban lampu 15W).

BAB III

METODE PENELITIAN

A. Lokasi dan Waktu Penelitian

Penelitian ini dikerjakan di PLN UP3 Pinrang ,ULP Watang Sawitto dan waktu yang dipergunakan untuk pelaksanaan penelitian tersebut kurang lebih ($\pm$)2 bulan.

B. Jenis Penelitian

Berikut adalah beberapa bentuk penelitian oleh penulis diantaranya penelitian lapangan (*Field Research*) dengan pengamatan maupun tanya jawab langsung mengenai data-data yang sedang teliti dan kajian kepustakaan (*Literature Study*) dilakukan secara review mulai dari catatan yang ada di perkuliahan, beberapa buku, serta browsing *internet* yang berhubungan dengan permasalahan tersebut

C. Metode Pengumpulan Data

Metode yang diterapkan oleh penulis untuk mengumpulkan data dalam penelitian ini mencakup beberapa pendekatan, yaitu:

1. Observasi adalah cara yang melibatkan pemantauan dan penulisan dengan cara seksama, cermat, serta terstruktur berkaitan dengan kejadian yang sedang berlangsung. Proses ini dilakukan dengan pergi langsung ke lokasi penelitian.
2. Dokumentasi adalah proses mengumpulkan informasi dalam bentuk gambar, dokumen, atau catatan tertulis.

3. Wawancara adalah dengan melakukan wawancara terhadap tempat penelitian tersebut.

D. Alat dan Bahan Penelitian

Dalam melaksanakan penelitian, penting untuk memiliki alat dan bahan yang menyokong proses ini. Beberapa alat dan bahan yang dibutuhkan meliputi:

Alat	Bahan
1. <i>Laptop / PC</i>	1. <i>PZEM-004T</i>
2. <i>Smartphone</i>	2. <i>Kabel Jumper</i>
	3. <i>ESP832</i>
	4. <i>Arduino Uno</i>

Software yang digunakan dalam pembuatan alat adalah :

1. *Arduino IDE*

E. Tahapan Penelitian

Langkah-langkah penelitian yang dimaksud dalam penelitian kali ini terdiri dari beberapa langkah yakni :

1. Pengumpulan data

Mengumpulkan data dan informasi sebagai referensi dalam merancang dan membangun alat.

2. Analisis Sistem

Menemukan dan memahami masalah yang ada, lalu mencari solusi alternatif.

3. Analisis

Setelah masalah ditentukan dan analisis data sudah dilakukan maka perlu dilakukan pembangunan atau mendesain system tentang masalah yang temukan.

4. Perancangan

Peneliti kemudian merancang aplikasi dengan mempertimbangkan berbagai solusi yang mungkin, serta mendesain tampilan dan alur penggunaan aplikasi.

5. Pengujian

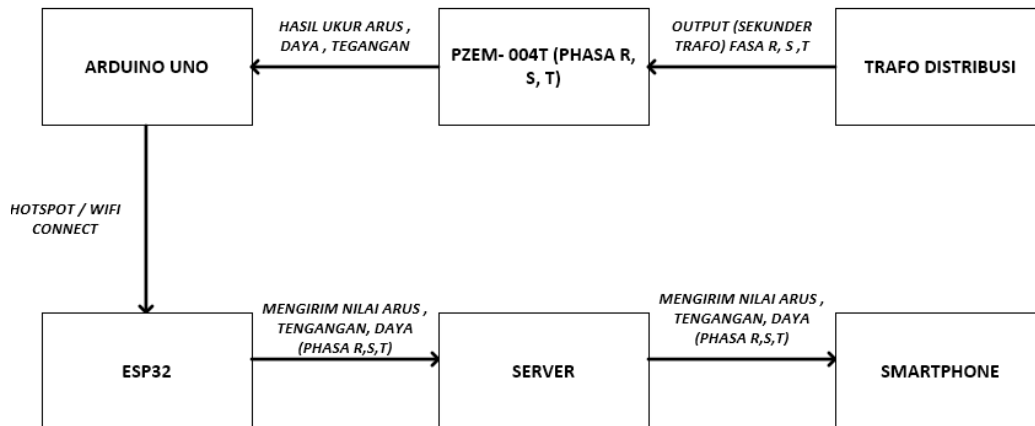
Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan yang telah dibuat menggunakan *black box*, *white box* dan pengujian dan sistem.

Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

6. Implementasi

Sesudah memastikan bahwa desain sudah dalam kondisi aman, alat dan sistem siap untuk digunakan.

F. Rancangan Alat Penelitian



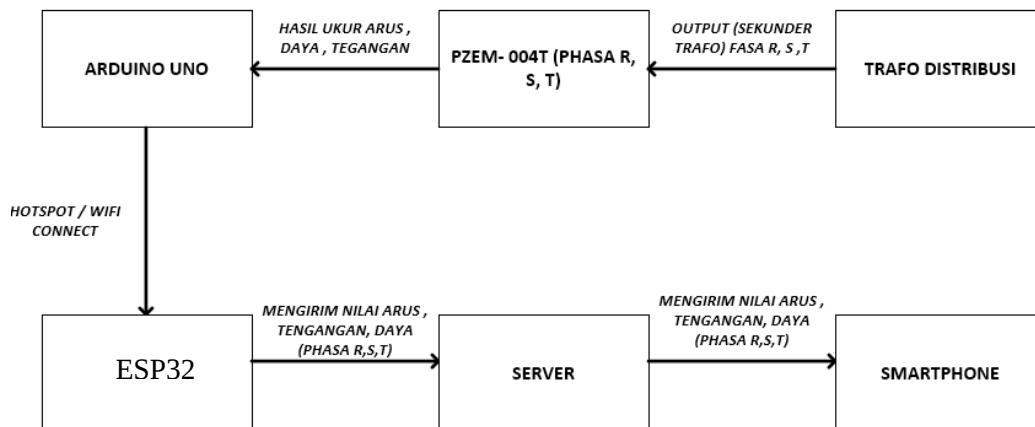
Gambar 3.1 Rancangan Alat Penelitian.

Diatas merupakan sistem yang diusulkan dimana pada bagian keluaran trafo distribusi *phasa R, S* dan *T* yang mana nilai tegangan , daya dan arus listrik akan dibaca oleh modul *PZEM-004T* yang kemudian nantinya nilai pengukuran dari modul *PZEM-004T* tersebut akan dikirimkan ke *Arduino Uno*, Disebabkan keterbatasan *Arduino Uno* untuk mengirimkan data secara *online* maka *arduino Uno* akan mengirim nilai tegangan, daya dan arus ke modul *ESP32* , namun perlu support dari wifi / hotspot yang tersedia di lokasi gardu distribusi.. Kemudian nilai hasil pengukuran tersebut akan diterima oleh server. Server kemudian akan menyimpan log data tersebut berdasarkan waktu penerimaan dan kemudiannya akan mengirimkan data tersebut kepada *smartphone*.

BAB IV

HASIL DAN PEMBAHASAN

A. Sistem Yang Diusulkan



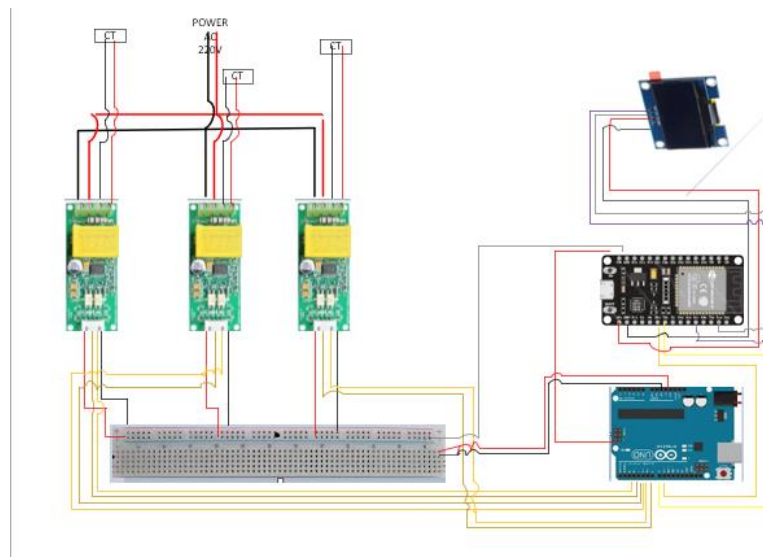
Gambar 4.1 Sistem Yang Diusulkan.

Diatas merupakan sistem yang diusulkan dimana pada trafo dikirimkan fasa R, S dan T yang mana nanti nilai tegangan dan arus listrik akan dibaca oleh modul *PZEM* yang kemudian nantinya nilai tersebut akan dikirimkan ke *arduino Uno*, Disebabkan keterbatasan *arduino Uno* untuk mengirimkan data secara *online* maka *arduino Uno* akan mengirim nilai tegangan dan arus kepada *ESP32* yang kemudian nilai tersebut akan diterima oleh server. Server kemudian akan menyimpan log data tersebut berdasarkan waktu penerimaan dan kemudiannya akan mengirimkan data tersebut kepada *smartphone*.

Adapun alasan mengapa menggunakan 2 mikrokontroler dikarenakan kelebihan dan kelemahan kontroler itu masing-masing. *Arduino uno* memiliki 6 pin analog namun tidak memiliki modul *wifi* untuk mengirimkan nilai sensor yang

didapatkan, manakala *esp32* memiliki *module wifi* untuk mengirimkan nilai sensor yang didapatkan namun hanya memiliki 1 pin *analog*. Penelitian peneliti mengharuskan untuk menggunakan 3 sensor *pzem-004*, maka peneliti memilih *arduino uno* untuk mengambil nilai sensor dikarenakan pin *analog* yang mencukupi, kemudian peneliti dengan menggunakan *serial communication* menggunakan *esp32* untuk mengirimkan data tersebut ke *server* yang kemudian ditampilkan ke aplikasi.

B. Perancangan Hardware



Gambar 4.2 Perancangan Hardware

Tabel 4.1 Koneksi Uno ke PZEM-004 (T)

Arduino Uno	PZEM 004 T (1)
D2	RX
D3	TX
5V	5V
GND	GND

Tabel 4.2 Koneksi Uno ke PZEM-004 (S)

Arduino Uno	PZEM 004 T (1)
D4	RX
D5	TX
5V	5V
GND	GND

Tabel 4.3 Koneksi Uno ke PZEM-004 (R)

Arduino Uno	PZEM 004 T (1)
D6	RX
D7	TX
5V	5V
GND	GND

Tabel 4.4 Koneksi Arduino Uno Ke ESP32

Arduino Uno	ESP32
D8	RX2
D9	TX2
5V	VIN
GND	GND

Tabel 4.5 Koneksi ESP32 ke OLED 0.96

ESP32	OLED 0.96
D23	SDA
D21	SCL
3.3V	VCC
GND	GND

Diatas merupakan koneksi antara *arduino uno*, *PZEM 004-T v3*, *esp32* dan *oled 0,96*. Dimana ketiga *pzem 004-t* terhubung ke *arduino uno* yang mana *uno* melakukan pembacaan *voltage*, *current*, *power*, *energy* dan *power factor*. Nilai ini kemudian dikirim ke *esp32* melalui *serial communication* untuk menutupi kekurangan *uno* berkomunikasi ke dunia luar, ini dengan cara menghubungkan *arduino uno* dan *esp32*. *Esp32* terhubung kepada *oled 0.96* untuk memberikan indikator bahwa *wifi* maupun nilai yang didepnteksi oleh *uno* melalui *pzem* dapat

terdeteksi dan ditampilkan. *Esp32* kemudiannya mengirimkan nilai tersebut kepada server yang nantinya ditampilkan ke aplikasi *mobile* yang telah dibuat oleh penulis.

C. Perancangan Software



Gambar 4.3 Perancangan Software

Diatas adalah Gambaran dari aplikasi yang telah dibuat oleh penulis. Aplikasi ini menampilkan nilai untuk tiga fasa yang dimonitoring yaitu *phasa R*,

fasa S dan *fasa T*. Pada tampilan diatas terlihat bahwa *fasa R* menampilkan nilai yang telah terdeteksi oleh alat yang telah dibuat oleh penulis sebelumnya, yang mana *fasa S* dan *fasa T* menunjukkan kegagalan mengambil data.

D. Data Logger Sistem

Sistem ini juga menampilkan data logger yang mana berfungsi untuk menyimpan data monitoring fasa listrik secara berkala, berikut gambar di bawah merupakan tampilan aplikasi yang dapat diakses melalui *smartphone*, yang mana menampilkan sepuluh data logger terakhir yang diinput oleh alat ke database.

Data LogFase R

Waktu	Voltage	Current	Power	Energy	PF
2024-06-14 02:11:50	-	-	-	-	-
2024-06-14 01:11:51	-	-	-	-	-
2024-06-14 00:11:49	218.70 V	0.10 A	0.00 W	0.01 KWh	49.90 %
2024-06-14 00:10:44	218.50 V	0.10 A	0.00 W	0.01 KWh	50.00 %
2024-06-14 00:10:44	218.50 V	0.10 A	0.00 W	0.01 KWh	50.00 %
2024-05-31 23:47:29	-	-	-	-	-
2024-05-31 10:47:33	-	-	-	-	-
2024-05-31 10:46:27	214.30 V	0.09 A	0.60 W	0.01 KWh	50.00 %
2024-05-31 10:41:38	-	-	-	-	-
2024-05-31 10:38:30	213.00 V	0.09 A	0.60 W	0.01 KWh	49.90 %

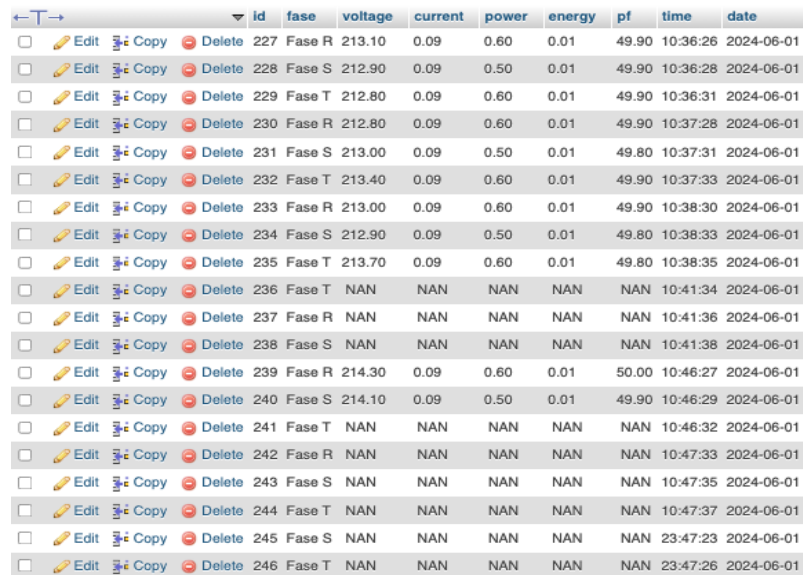
Data LogFase S

Waktu	Voltage	Current	Power	Energy	PF
2024-06-14 02:11:50	-	-	-	-	-
2024-06-14 01:11:46	-	-	-	-	-
2024-06-14 00:11:44	-	-	-	-	-
2024-06-14 00:10:39	-	-	-	-	-
2024-06-14 00:10:39	-	-	-	-	-
2024-05-31 23:47:23	-	-	-	-	-
2024-05-31 10:47:35	-	-	-	-	-
2024-05-31 10:46:29	214.70 V	0.09 A	0.50 W	0.01 KWh	49.90 %
2024-05-31 10:41:38	-	-	-	-	-

Gambar 4.4 Data Logger Aplikasi

Data logger yang ditampilkan diatas disimpan ke dalam *database* yang telah dibuat oleh penulis sebelumnya menggunakan *database MySQL*. Dibawah

merupakan tampilan *Phpmyadmin database* yang mana menyimpan *data logger monitoring* listrik.



				id	fase	voltage	current	power	energy	pf	time	date
☐	Edit	Copy	Delete	227	Fase R	213.10	0.09	0.60	0.01	49.90	10:36:26	2024-06-01
☐	Edit	Copy	Delete	228	Fase S	212.90	0.09	0.50	0.01	49.90	10:36:28	2024-06-01
☐	Edit	Copy	Delete	229	Fase T	212.80	0.09	0.60	0.01	49.90	10:36:31	2024-06-01
☐	Edit	Copy	Delete	230	Fase R	212.80	0.09	0.60	0.01	49.90	10:37:28	2024-06-01
☐	Edit	Copy	Delete	231	Fase S	213.00	0.09	0.50	0.01	49.80	10:37:31	2024-06-01
☐	Edit	Copy	Delete	232	Fase T	213.40	0.09	0.60	0.01	49.90	10:37:33	2024-06-01
☐	Edit	Copy	Delete	233	Fase R	213.00	0.09	0.60	0.01	49.90	10:38:30	2024-06-01
☐	Edit	Copy	Delete	234	Fase S	212.90	0.09	0.50	0.01	49.80	10:38:33	2024-06-01
☐	Edit	Copy	Delete	235	Fase T	213.70	0.09	0.60	0.01	49.80	10:38:35	2024-06-01
☐	Edit	Copy	Delete	236	Fase T	NAN	NAN	NAN	NAN	NAN	10:41:34	2024-06-01
☐	Edit	Copy	Delete	237	Fase R	NAN	NAN	NAN	NAN	NAN	10:41:36	2024-06-01
☐	Edit	Copy	Delete	238	Fase S	NAN	NAN	NAN	NAN	NAN	10:41:38	2024-06-01
☐	Edit	Copy	Delete	239	Fase R	214.30	0.09	0.60	0.01	50.00	10:46:27	2024-06-01
☐	Edit	Copy	Delete	240	Fase S	214.10	0.09	0.50	0.01	49.90	10:46:29	2024-06-01
☐	Edit	Copy	Delete	241	Fase T	NAN	NAN	NAN	NAN	NAN	10:46:32	2024-06-01
☐	Edit	Copy	Delete	242	Fase R	NAN	NAN	NAN	NAN	NAN	10:47:33	2024-06-01
☐	Edit	Copy	Delete	243	Fase S	NAN	NAN	NAN	NAN	NAN	10:47:35	2024-06-01
☐	Edit	Copy	Delete	244	Fase T	NAN	NAN	NAN	NAN	NAN	10:47:37	2024-06-01
☐	Edit	Copy	Delete	245	Fase S	NAN	NAN	NAN	NAN	NAN	23:47:23	2024-06-01
☐	Edit	Copy	Delete	246	Fase T	NAN	NAN	NAN	NAN	NAN	23:47:26	2024-06-01

Gambar 4.5 Database Data Logger

User dapat mencari lebih spesifik *data monitoring* yang lalu dengan memilih tanggal data yang ingin ditampilkan. Dibawah merupakan tampilan data pencarian yang dilakukan oleh *user*

The screenshot shows a mobile application interface with a green header and footer. The main content area displays three tables of power log data, each with a title and a date filter set to '2024-06-01'.

Data LogFase R

Waktu	Voltage	Current	Power	Energy	PF
2024-05-31 10:36:26	213.70 V	0.09 A	0.60 W	0.01 KWh	49.90 %
2024-05-31 10:37:28	212.80 V	0.09 A	0.60 W	0.01 KWh	49.90 %
2024-05-31 10:38:30	213.00 V	0.09 A	0.60 W	0.01 KWh	49.90 %
2024-05-31 10:41:36	-	-	-	-	-
2024-05-31 10:46:27	214.30 V	0.09 A	0.60 W	0.01 KWh	50.00 %
2024-05-31 10:47:33	-	-	-	-	-
2024-05-31 23:47:28	-	-	-	-	-

Data LogFase S

Waktu	Voltage	Current	Power	Energy	PF
2024-05-31 10:36:28	212.90 V	0.09 A	0.50 W	0.01 KWh	49.80 %
2024-05-31 10:37:31	213.00 V	0.09 A	0.50 W	0.01 KWh	49.80 %
2024-05-31 10:38:33	212.90 V	0.09 A	0.50 W	0.01 KWh	49.80 %
2024-05-31 10:41:38	-	-	-	-	-
2024-05-31 10:46:29	214.70 V	0.09 A	0.50 W	0.01 KWh	49.90 %
2024-05-31 10:47:35	-	-	-	-	-
2024-05-31 23:47:23	-	-	-	-	-

Data LogFase T

Waktu	Voltage	Current	Power	Energy	PF
2024-05-31 10:36:31	212.80 V	0.09 A	0.60 W	0.01 KWh	49.90 %
2024-05-31 10:37:33	213.40 V	0.09 A	0.60 W	0.01 KWh	49.90 %
2024-05-31 10:37:33	213.70 V	0.09 A	0.60 W	0.01 KWh	49.80 %

Gambar 4.6 Tampilan Pencarian *Data Logger*

E. Syntax Alat dan Aplikasi

1. Arduino Uno

```
#include <PZEM004Tv30.h>
#include <SoftwareSerial.h>
```

Diatas merupakan inisialisasi *pzem004t* dan juga *software serial* yang digunakan untuk komunikasi ke *esps32*

```
PZEM004Tv30 pzem(6, 7); //fasa r
PZEM004Tv30 pzem2(4, 5); //fasa s
PZEM004Tv30 pzem3(2, 3); //fasa t
SoftwareSerial fromesp32(8, 9);
```

Diatas merupakan pinout untuk tiga *pzem004t* dan juga *software serial* yang digunakan untuk komunikasi ke *esp32*

```
void setup()
```

```

{
  Serial.begin(19200);
  fromesp32.begin(9600);
  Serial.println("..initialized");
}

```

Diatas merupakan fungsi setup yang mana mengatur *serial monitor* untuk *arduino uno* dan juga *serial monitor* komunikasi antara *uno* dan *esp32*

```

void checkPzem(String status,float voltage, float current, float power, float energy, float frequency,
float pf)
{
  String ini = "";
  ini += status + ",";
  if (voltage != NAN)
  {
    Serial.print("Voltage: ");
    Serial.print(voltage);
    Serial.println("V");
    ini += String(voltage) + ",";
  }
  else
  {
    Serial.println("Error reading voltage");
    ini += "err,";
  }
  if (current != NAN)
  {
    Serial.print("Current: ");
    Serial.print(current);
    Serial.println("A");
    ini += String(current) + ",";
  }
  else
  {
    Serial.println("Error reading current");
    ini += "err,";
  }
  if (power != NAN)
  {
    Serial.print("Power: ");
    Serial.print(power);
    Serial.println("W");
    ini += String(power) + ",";
  }
}

```

```

else
{
    Serial.println("Error reading power");
    ini += "err,";
}
if (energy != NAN)
{
    Serial.print("Energy: ");
    Serial.print(energy, 3);
    Serial.println("kWh");
    ini += String(energy) + ",";
}
else
{
    Serial.println("Error reading energy");
    ini += "err,";
}
if (frequency != NAN)
{
    Serial.print("Frequency: ");
    Serial.print(frequency, 1);
    Serial.println("Hz");
    ini += String(frequency) + ",";
}
else
{
    Serial.println("Error reading frequency");
    ini += "err,";
}
if (pf != NAN)
{
    Serial.print("PF: ");
    Serial.println(pf);
    ini += String(pf);
}
else
{
    Serial.println("Error reading power factor");
    ini += "err,";
}
ini += "\n";
fromesp32.print(ini);

Serial.println();
delay(2000);
}

```

Diatas merupakan fungsi untuk mengambil nilai yang dibaca oleh *pzem-004t* dan dikirimkan ke *esp32* untuk dikirimkan ke *server*

```
void loop()
{
  checkPzem("Fasa R",pzem.voltage(), pzem.current(), pzem.power(), pzem.energy(),
pzem.frequency(), pzem.pf());
  checkPzem("Fasa S",pzem2.voltage(), pzem2.current(), pzem2.power(), pzem2.energy(),
pzem2.frequency(), pzem2.pf());
  checkPzem("Fasa T",pzem3.voltage(), pzem3.current(), pzem3.power(), pzem3.energy(),
pzem3.frequency(), pzem3.pf());
  // delay(1500);
}
```

Diatas merupakan script *arduino uno* yang mana pada script ini terjadinya pendeteksian arus dan tegangan listrik, Nilai yang dideteksi merupakan *voltage*, *current*, *power*, *energy* dan *power factor*. Semuanya dilakukan dengan menggunakan *library PZEM-004T-v30* dan *Software Serial* digunakan untuk komunikasi dengan *esp32* dan mengirim nilai yang telah terdeteksi oleh *arduino uno*.

2. ESP32

```
#include <Arduino.h>
#include <WiFi.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <HTTPClient.h>
```

Diatas merupakan inisialisai untuk *arduino*, *wifi*, *wire*, *adafruit display* dan juga *httpclient*.

```
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
```

Diatas merupakan penentuan *Screen display* yaitu $128 * 64$

```
// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
```

Kemudian *library adafruit* dipanggil dengan *library wire* agar *display* dapat digunakan.

```
HardwareSerial fromUno(2);
```

Diatas merupakan *hardware serial* dari *arduino uno* ke *esp32*

```
char c;
String dataIn;

const char *ssid = "KARAN";
const char *password = "12345679";
// const char *ssid = "iPhone (2)";
// const char *password = "12345678";

HTTPClient http;
// String url = "http://192.168.20.45:3001/";
String url = "https://electric-coba.kicap-karan.com/";
```

Diatas merupakan inisialisasi variabel seperti *wifi* dan juga *domain server*.

```
void displayData(String data)
{
    display.clearDisplay();

    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0, 10);
    display.println(data);
    display.display();
}
```

Diatas merupakan fungsi untuk menampilkan teks ke *display*.

```
String ifElse(String data, String name, String stat)
{
    if (data != "NAN")
    {
        Serial.println(name + " : " + data + " " + stat);
    }
}
```

```

    return name + " : " + data + " " + stat + "\n";
}
else
{
    Serial.println("Error get " + name);
    return "Error get " + name + "\n";
}
}

```

Diatas merupakan fungsi if *else* yang mana parameter merupakan data pzem-004 dari *arduino uno*.

```

void separator(String data)
{
    char charArray[data.length() + 1]; // Create a char array to store the string
    data.toCharArray(charArray, sizeof(charArray)); // Convert String to char array

    char *token = strtok(charArray, ","); // Tokenize the string based on comma delimiter
    int count = 0;

    String dataForDisplay = "";
    String jsonSend = "{\"fasa\":\"";
    while (token != NULL)
    {
        switch (count)
        {
            case 1:
                dataForDisplay += String(token) + "\n";
                Serial.println(String(token));
                jsonSend += "\"" + String(token) + "\", ";
                break;
            case 2:
                dataForDisplay += ifElse(String(token), "Voltage", "V");
                jsonSend += "\"voltage\":\"" + String(token) + "\", ";
                break;
            case 3:
                dataForDisplay += ifElse(String(token), "Current", "A");
                jsonSend += "\"current\":\"" + String(token) + "\", ";
                break;
            case 4:
                dataForDisplay += ifElse(String(token), "Power", "W");
                jsonSend += "\"power\":\"" + String(token) + "\", ";
                break;
            case 5:
                dataForDisplay += ifElse(String(token), "Energy", "kWh");
                jsonSend += "\"energy\":\"" + String(token) + "\", ";

```

```

        break;
    case 6:
        dataForDisplay += ifElse(String(token), "Power Factor", "pf");
        jsonSend += "\"pf\":\\"" + String(token) + "\"";
        break;
    default:
        dataForDisplay += ifElse(String(token), "Wifi", "");
        break;
    }
    token = strtok(NULL, ","); // Get the next token
    count++;
}
jsonSend += "}";

displayData(dataForDisplay);
Serial.println(jsonSend);
http.begin(url);
http.addHeader("Content-Type", "application/json");
int httpResponseCode = http.POST(jsonSend);

if (httpResponseCode > 0) {
    Serial.print("HTTP Response code: ");
    Serial.println(httpResponseCode);
    String payload = http.getString();
    Serial.println(payload);
} else {
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
}
}
}

```

Diatas merupakan fungsi *separator* yang memisahkan data yang diterima dari *arduino uno* kemudian mengirimkan data tersebut ke *server*.

```

void setup()
{
    Serial.begin(19200);
    fromUno.begin(9600); // ini dari esp32
    if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C))
    { // Address 0x3D for 128x64
        Serial.println(F("SSD1306 allocation failed"));
        for (;;)
            ;
    }
}

```

```

}

// Connect to WiFi
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED)
{
    delay(1000);
    Serial.println("Connecting to WiFi...\n");
    fromUno.print("Connecting to WiFi...\n");
    displayData("Connecting to WiFi...");
}
Serial.println("Connected to WiFi\n");
fromUno.print("Connected to WiFi\n");
displayData("Connected to WiFi");
}

```

Diatas merupakan fungsi sertu yang mana terdapat koneksi ke wifi dan juga monitor *speed*.

```

void loop()
{
    String wifiStatus = "";
    if (WiFi.status() == WL_CONNECTED)
    {
        Serial.println("WiFi Connected\n");
        fromUno.print("WiFi Connected\n");
        wifiStatus += "Connected";
        // delay(2000);
    }
    else
    {
        // toUnoWifiState.print("WiFi Disconnected\n");
        Serial.println("WiFi Disconnected\n");
        fromUno.print("WiFi Disconnected\n");
        wifiStatus += "Disconnected";
        // delay(2000);
    }
}

while (fromUno.available())
{
    c = fromUno.read();
    if (c != '\n')
    {
        dataIn += c;
    }
    else

```

```

    {
      break;
    }
  }

  if (c == '\n')
  {
    // Serial.println(dataIn);
    separator(wifiStatus + "," + dataIn);
    // displayData(wifiStatus);
    c = 0;
    dataIn = "";
  }

  delay(2000);
}

```

Diatas merupakan script *esp32* yang berfungsi sebagai kontroler untuk mengirimkan nilai yang telah terdeteksi oleh *arduino uno* sebelumnya. *Esp32* terhubung dengan *display oled 0.96* untuk menampilkan nilai seperti koneksi wifi, *voltage*, *current power*, *energy* dan *power factor* yang dikirim oleh *arduino uno*.

3. Server

```

const express = require('express');
const http = require('http');
const cors = require('cors');
const dotenv = require('dotenv');
const socket = require('./socket');
const app = express();
const server = http.createServer(app);
const io = socket.init(server);
const iosend = socket.getIO();

dotenv.config();

app.use(express.json());
app.use(express.urlencoded({ extended: true }));
// app.use(fileUpload());
app.options('*', cors());
app.use(cors());

// Handle POST requests to '/post'

```

```

app.post('/', (req, res) => {
  // Access the POST data sent in the request body
  const postData = req.body;

  // Process the received data
  console.log('Received data:', postData);
  io.send.emit('datanya', postData);

  // Respond with a simple message
  res.send('Received your POST request!');
});

// app error handler
app.use((err, req, res, next) => {
  console.log(err);
  res.status(500).send('Something broke!');
});

io.on('connection', (socket) => {
  let userID = socket.id;
  console.log('A user connected: ' + userID);

  socket.on('scan_dia', (data) => {
    console.log('Received scan_dia event: ' + data);
  });

  socket.on('disconnect', () => {
    console.log('User disconnected: ' + userID);
  });
});

module.exports = {
  app,
  server,
  io
};

const port = process.env.PORT || 3001;

// Start the server
server.listen(port, () => {
  console.log('Server is running on http://localhost:${port}');
});

```

Diatas merupakan script server yang menerima nilai yang telah dikirim oleh *esp32* sebelumnya. Nilai ini kemudian dikirimkan kembali kepada aplikasi mobile melalui *socket io* yang mana jika terjadi perubahan nilai pada alat maka akan langsung mengirimkan nilai kepada aplikasi secara *real time*.

4. Mobile

```
import 'package:flutter/material.dart';
import 'package:stacked/stacked.dart';

import '../..../app/themes/app_colors.dart';
import '../..../app/themes/app_text.dart';
import '../..../model/electric_model.dart';
import '../monitoring_view_model.dart';
```

Diatas merupakan import library yang digunakan untuk aplikasi

```
class MonitoringView extends StatelessWidget {
  const MonitoringView({super.key});

  @override
  Widget build(BuildContext context) {
    return ViewModelBuilder<MonitoringViewModel>.reactive(
      viewModelBuilder: () => MonitoringViewModel(),
      onViewModelReady: (MonitoringViewModel model) async {
        await model.init();
      },
      builder: (
        BuildContext context,
        MonitoringViewModel model,
        Widget? child,
      ) {
        return Scaffold(
          appBar: AppBar(
            backgroundColor: mainColor,
            title: const Row(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                Image(
                  image: AssetImage("assets/logo.png"),
                  width: 30,
                  height: 30,
                ),
                SizedBox(width: 10),
              ],
            ),
          ),
        );
      },
    );
  }
}
```

```

        Text(
          'Electric Monitoring',
          style: TextStyle(
            color: backgroundColor,
          ),
        ),
      ],
    ),
  ),
  body: SingleChildScrollView(
    child: Padding(
      padding: const EdgeInsets.all(20.0),
      child: Column(
        children: [
          MyContainer(
            electricModel: model.fasaR,
            color: Colors.red,
            fontColor: Colors.white,
          ),
          const SizedBox(height: 20),
          MyContainer(
            electricModel: model.fasaS,
            color: Colors.yellow,
            fontColor: Colors.black,
          ),
          const SizedBox(height: 20),
          MyContainer(
            electricModel: model.fasaT,
            color: Colors.green,
            fontColor: Colors.white,
          ),
        ],
      ),
    ),
  ),
);
}
};
}
}

```

```

class MyContainer extends StatelessWidget {
  const MyContainer({
    super.key,
    required this.electricModel,
    required this.color,
    required this.fontColor,
  }) : super();
}

```

```

});

final ElectricModel? electricModel;
final Color color;
final Color fontColor;

@override
Widget build(BuildContext context) {
  return Container(
    padding: const EdgeInsets.all(10.0),
    width: double.infinity,
    decoration: BoxDecoration(
      color: color,
      borderRadius: BorderRadius.circular(10.0),
    ),
    child: electricModel == null
      ? Center(
          child: CircularProgressIndicator(
            color: fontColor,
          ),
        )
      : Column(
          mainAxisAlignment: MainAxisAlignment.min,
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text(
              electricModel?.fasa ?? "",
              style: boldTextStyle.copyWith(
                color: fontColor,
                fontSize: 15,
                fontStyle: FontStyle.italic,
              ),
            ),
            const SizedBox(height: 5),
            MyText(
              title: 'Voltage : ',
              text: electricModel?.voltage != 'NAN'
                ? "${electricModel?.voltage} V"
                : "Error getting voltage",
              fontColor: fontColor,
            ),
            MyText(
              title: 'Current : ',
              text: electricModel?.current != 'NAN'
                ? "${electricModel?.current} A"
                : "Error getting current",
              fontColor: fontColor,
            ),
          ],
        ),
      ),
    ),
  );
}

```

```

    ),
    MyText(
      title: 'Power : ',
      text: electricModel?.power != ' NAN'
        ? "${electricModel?.power} W"
        : "Error getting power",
      fontColor: fontColor,
    ),
    MyText(
      title: 'Energy : ',
      text: electricModel?.energy != ' NAN'
        ? "${electricModel?.energy} kWh"
        : "Error getting energy",
      fontColor: fontColor,
    ),
    MyText(
      title: 'Power Factor : ',
      text: electricModel?.pf != ' NAN'
        ? "${electricModel?.pf} %"
        : "Error getting power factor",
      fontColor: fontColor,
    ),
  ],
),
);
}
}

```

```

class MyText extends StatelessWidget {
  const MyText({
    super.key,
    required this.title,
    required this.text,
    required this.fontColor,
  });

```

```

  final String title;
  final String text;
  final Color fontColor;

```

```

  @override
  Widget build(BuildContext context) {
    return RichText(
      text: TextSpan(
        children: [
          TextSpan(
            text: title,

```

```

        style: boldTextStyle.copyWith(
          color: fontColor,
          fontSize: 15,
        ),
      ),
      TextSpan(
        text: text,
        style: italicTextStyle.copyWith(
          color: fontColor,
          fontSize: 15,
        ),
      ),
    ],
  ),
);
}
}

```

Diatas merupakan script untuk aplikasi *mobile* yang dibuat oleh penulis, pada script ini aplikasi melakukan pemanggilan data ke server yang berlaku secara *real time* melalui *socket io*. Nilai yang didapat kemudian ditampilkan ke aplikasi agar pengguna dapat memonitoring listrik yang mana alat telah diletakan sebelumnya.

F. Pengujian Alat



Gambar 4.7 Alat Prototype Sistem Monitoring dan data Logger

Diatas merupakan pengujian alat yang dilakukan oleh penulis yang mana jika dilihat pada sebelah kiri gambar terlihat *esp32* yang terhubung ke *oled 0,96* . Di sini *esp32* berfungsi untuk menerima nilai *pzem-004-t* yang kemudian mengirimkan nilai ke server dan menampilkannya ke *display oled 0,96*. Kemudian pada tengah gambar terlihat *arduino uno* yang terhubung dengan tiga *pzem-004t* .Dimana *pzem-004t* ini kemudiannya tercolok pada colokan listrik yang digunakan untuk mendeteksi *voltage*, *current*, *power*, *energy* dan *power factor* dari listrik yang mengalir.

Adapun rumus dasar dari Pengujian alat diatas adalah $P = V \times I$, yang mana *P* (Daya), *V* (Tegangan) dan *I* (Arus). Rumus ini menghubungkan

Tabel 4.6 Pengujian Alat

<i>Voltage</i>	<i>Current</i>	<i>Power</i>	<i>Energy</i>	<i>Power Factor</i>
222.30 V	0.10 A	0.70 W	0.00 kWh	50.00 %

Diatas merupakan tabel pengujian alat yang mana colokan Perumpamaan dari sumber sekunder sisi trafo distribusi) dideteksi *Voltage* dengan 222.30 V, *current* dengan 0.08 A , *Power* dengan 16,90 W, *Energy* dengan 0.00 kWh dan *power factor* dengan 50.00%

Tabel 4.7 Pengujian Alat

<i>Voltage</i>	<i>Current</i>	<i>Power</i>	<i>Energy</i>	<i>Power Factor</i>
222.30 V	0.08 A	16.90 W	0.00 kWh	50.00 %

Diatas merupakan tabel pengujian alat yang mana colokan telah ditambahkan kipas alat dan dideteksi *Voltage* dengan 222.30 V, *current* dengan 0.08 A , *Power* dengan 16.90 W, *Energy* dengan 0.00 kWh dan *power factor* dengan 50.00% , Terdapat pengurangan dari *current* sebesar 0.02 A manakala *Power* mengalami kenaikan yang signifikan sebesar 11.20 W.

Adapun dari hasil diatas , dapat dihitung dengan rumus sebagai berikut :

$$\text{Dik: } P = V \times I$$

$$220,3 \times 0,08$$

$$P = 17,624 \text{ W}$$

$$\text{Dik: } V = \frac{P}{I}$$

$$= \frac{17,624}{0,08}$$

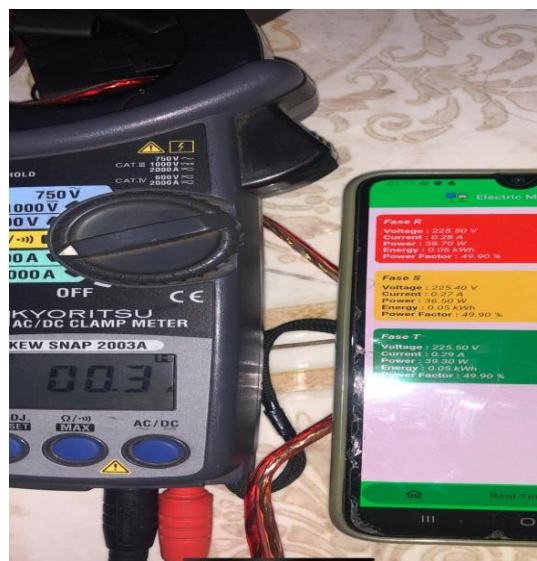
$$V = 220,3$$

$$\begin{aligned} \text{Dik: } I &= \frac{P}{V} \\ &= \frac{17,624}{220,3} \\ I &= 0,08 \end{aligned}$$

Tabel 4.8 Persentase Error Perhitungan rumus dan Perhitungan alat

Voltage (Pengujian Alat)	Voltage(Perhitungan Rumus)	Persentase Error
222.30 V	222.30 V	0%
Current (Pengujian Alat)	Current (Perhitungan Rumus)	Persentase Error
0.08 A	0.08 A	0%
Power (Pengujian Alat)	Power (Perhitungan Rumus)	Persentase Error
16,90 A	17,624 A	0,72 %

Tabel diatas menunjukkan bahwa dari hasil perhitungan dan hasil pengujian pada alat masih terdapat persentase kesalahan (Error) pada Pengukuran Power (Daya).,Yang mana nilai Persentase Error pada alat <5%. Adapun indikator yang mengalami kesalahan / selisih adalah indikator pengukuran pada Daya atau *Power*.

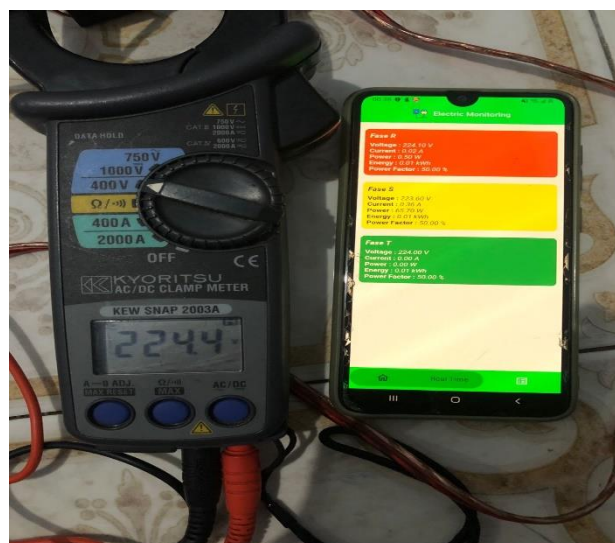


Gambar 4.8 Pengujian alat (Indikator Arus)

Gambar Diatas merupakan hasil pengujian Alat menggunakan metode perbandingan. Adapun media yang digunakan adalah *Digital Clamp Meter* sebagai media alat ukur pembanding dalam melakukan pengujian alat tersebut. Adapun sebagai beban (Arus) adalah peralatan rumah tangga seperti Setrika, *Rice cooker*, dan *Charger Laptop*. Adapun hasil pengujian alat berdasarkan hasil nilai pengukuran Arus (Ampere) adalah :

Tabel 4.9 Perbandingan hasil pengujian Alat (Nilai arus)

Hasil Ukur (Alat)	Hasil Ukur (Tang Ampere)	Persentase Nilai Error (%)
0,28 A	0,3 A	6,67%



Gambar 4.9 Pengujian alat (Indikator Tegangan)

Gambar diatas merupakan hasil Pengujian alat menggunakan metode perbandingan dengan membandingkan hasil pengukuran indikator Tegangan (*Voltage*). Adapun media yang digunakan sebagai pembanding adalah menggunakan *Digital Clamp Meter*. Adapun hasil perbandingannya adalah :

Tabel 4.10 Perbandingan hasil pengujian Alat (Nilai tegangan)

Hasil Ukur (Alat)	Hasil Ukur (Tang Ampere)	Persentase Nilai Error (%)
223,6 V	224,4 V	0,36 %

Setelah melakukan Pengujian alat dengan membandingkan hasil pengukuran antara alat yang dirancang penulis dan alat untuk mengukur tegangan dan arus (*Digital Clamp Meter*) dan didapatkan hasil persentase nilai error kesalahan alat yang dirancang penulis adalah <10 % (kurang dari sepuluh persen). Kemudian penulis melakukan pengambilan sampling data, yang mana alat tersebut di setting untuk mengirim nilai hasil ukur alat yang dirancang penulis dengan interval waktu yang berbeda, dan hari yang berbeda untuk mengukur keakuratan dan kegunaan dari alat tersebut.

id	fase	voltage	current	power	energy	pf	time	date
227	Fase R	213.10	0.09	0.60	0.01	49.90	10:36:26	2024-06-01
228	Fase S	212.90	0.09	0.50	0.01	49.90	10:36:28	2024-06-01
229	Fase T	212.80	0.09	0.60	0.01	49.90	10:36:31	2024-06-01
275	Fase R	225.50	0.36	67.40	0.07	49.90	01:29:41	2024-07-19
276	Fase S	225.40	0.36	67.20	0.06	49.90	01:29:44	2024-07-19
277	Fase T	225.40	0.36	67.20	0.07	49.90	01:29:46	2024-07-19
282	Fase T	220.20	1.41	303.00	0.60	49.90	15:51:09	2024-07-20
283	Fase R	220.30	2.63	561.10	0.79	49.90	15:51:11	2024-07-20
284	Fase S	220.30	1.40	301.40	0.60	49.90	15:51:14	2024-07-20
285	Fase T	221.60	1.42	308.00	0.61	50.00	15:53:10	2024-07-20
286	Fase R	221.50	2.47	534.80	0.80	50.00	15:53:13	2024-07-20
287	Fase S	221.40	1.42	306.40	0.61	50.00	15:53:16	2024-07-20
288	Fase T	221.60	1.43	308.90	0.63	49.80	15:55:11	2024-07-20
289	Fase R	221.60	2.48	536.00	0.82	49.90	15:55:13	2024-07-20
290	Fase S	221.40	1.43	308.50	0.62	49.80	15:55:16	2024-07-20
291	Fase T	222.00	1.41	304.80	0.63	49.80	15:57:15	2024-07-20
292	Fase R	221.70	2.55	548.90	0.84	49.90	15:57:18	2024-07-20
293	Fase S	221.80	1.41	305.40	0.63	49.90	15:57:20	2024-07-20
294	Fase T	222.60	1.42	306.90	0.64	50.00	15:59:16	2024-07-20
295	Fase R	222.50	2.52	545.60	0.86	50.00	15:59:19	2024-07-20
296	Fase S	222.40	1.42	307.20	0.64	50.00	15:59:22	2024-07-20
297	Fase T	222.80	1.41	305.80	0.66	50.00	16:01:18	2024-07-20
298	Fase R	222.70	2.50	542.60	0.88	50.00	16:01:21	2024-07-20
299	Fase S	222.60	1.40	303.60	0.65	50.00	16:01:23	2024-07-20
300	Fas	222.90	2.50	542.80	0.88	50.00	16:01:28	2024-07-20
301	Fase T	222.80	1.41	306.70	0.67	49.90	16:03:19	2024-07-20
302	Fase R	222.90	2.17	470.70	0.89	49.90	16:03:21	2024-07-20
303	Fase S	222.80	1.42	307.20	0.66	49.90	16:03:23	2024-07-20
304	Fas	220.30	1.42	304.50	0.67	49.90	16:03:37	2024-07-20
305	Fase T	221.80	1.42	307.50	0.68	50.00	16:05:19	2024-07-20
306	Fase R	221.40	2.17	468.80	0.91	50.00	16:05:21	2024-07-20
307	Fase S	221.60	1.42	306.90	0.67	50.00	16:05:24	2024-07-20
308	Fas	220.90	1.41	304.00	0.68	50.00	16:05:47	2024-07-20

Gambar 4.10 Pengambilan sampling data

Gambar diatas adalah hasil *screenshoot* Data logger yang di simpan di server data base dan kemudian di ekspor ke file *pdf* untuk memudahkan dalam pembacaan data. Gambar diatas dapat dilihat hasil pengukuran alat yang dirancang penulis menampilkan Indikator Arus , Tegangan , *Power factor*, dengan interval waktu , tanggal , jam yang berbeda -beda tiap waktu nya. Yang mana alat yang dirancang penulis selain dapat memonitoring secara *realtime*, juga telah berfungsi sebagai *Datalogger*.

KODE GARDU	ALAMAT / LOKASI / NOMOR-SERI	DATA TRAF0		DATA PENGUKURAN				
		KAPASITAS		TANGGAL UKUR	JAM	ARUS (ampe		
		1 F kVA	3 F kVA			R	S	T
GOT.S	KP. TASSOKOE 1	50		08 Nop 2021	18:58	157	0	0
GOT.T	KP. AWANG-AWANG	50		08 Nop 2021	19:14	82	0	0
GOT.BJ	KP.AWANG-AWANG		25	08 Nop 2021	19:21	7	2	23
GOT.U	KP. BUA-BUA	50		08 Nop 2021	19:31	0	159	0
GOT.V	KP. BANGA-BANGA		100	08 Nop 2021	19:41	59	73	70
GOT.EQ	KP. CAPAKALA	16		30 Nop 2021	10:01	0	0	0
GOT.BK	KP. AWANG-AWANG		100	08 Nop 2021	19:02	104	84	99
GOT.BV	KP. BUA-BUA II	25		08 Nop 2021	19:34	0	81	0
GOT.CO	BTN GRIYA YAMANI		50	08 Nop 2021	18:39	16	48	13

Gambar 4.11 Data pengukuran Gardu di Internal PLN UID
(Sumber : Data Report Transformer PLN ULP Sawitto)

Gambar diatas adalah hasil rekap data yang diukur manual menggunakan Avo meter dan dilakukan di beberapa titik gardu distribusi oleh petugas PLN. Setelah di bandingkan dengan data yang di ambil menggunakan alat monitoring dan data logger, tampaknya hampir sama menampilkan beberapa indikator seperti arus dan tegangan pada tiap fasa R,S dan T dari keluaran gardu distribusi. Namun, perbedaannya adalah data dari PLN hanya menampilkan satu kali interval waktu pengukuran arus dan tegangan gardu distribusi , dan juga tidak dapat dimonitoring secara *realtime*.

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah diuraikan pada bab-bab sebelumnya, Dihasilkan sebuah alat dan aplikasi Prototype Sistem Monitoring dan *datalogger* pada gardu distribusi berbasis Mikrokontroler yang dibuat sudah bebas dari kesalahan . Maka dapat ditarikan kesimpulan seperti berikut :

1. Penelitian ini berhasil merancang dan membangun sebuah prototipe Sistem Monitoring Dan Data Logger Pada Gardu Distribusi Berbasis Mikrokontroler (*Arduino Uno*) .
2. Kontroler yang digunakan adalah *Arduino uno* dan *esp32* yang mana *arduino uno* berfungsi untuk mengambil nilai yang dikirim oleh *pzem-004t* dan kemudiannya mengirimkan nilai tersebut ke *esp32* untuk dikirimkan ke server.
3. Sensor yang digunakan adalah *pzem-004t* yang berfungsi untuk mendeteksi *voltage, current, power,energy* dan *power factor* dari listrik fasa R, S dan T.
4. Alat tersebut sudah diuji berdasarkan hasil perhitungan dan pengujian perbandingan menggunakan *Digital Clamp Meter* Atau Tang ampere.
5. Alat tersebut sudah dapat berjalan sesuai fungsinya, yaitu sebagai monitoring dan *Datalogger*

B. Saran

Setelah mengambil beberapa kesimpulan, penulis menyarankan perlu adanya pengembangan terkait alat tersebut. Penulis berharap kedepannya alat tersebut dapat di kembangkan dan diperbaiki untuk mencapai hasil yang optimal dan dapat diaplikasikan untuk mempermudah monitoring data khususnya di instansi PT PLN (Persero).

DAFTAR PUSTAKA

- Yovi, M. (2016). *Sejarah Android (Online)*.
<http://woocara.blogspot.kr/2015/02/sejarah-android-dan-nama-nama-versi-android.html>.
- Widodo, A., Kholis, N., & Rakhmaati, L. (2022, Januari 06). Rancang Bangun ALat Monitoring Data Listrik Berbasis IOT Menggunakan Firebase Dan Aplikasi Android. *Rancang Bangun ALat Monitoring Data Listrik Berbasis IOT Menggunakan Firebase Dan Aplikasi Android*, 11(1).
- Rendy, F. (2020, Desember 01). Sistem Monitoring Daya Listrik Dan Pengontrolan Perangkat Elektronik Berbasis IOT. *Sistem Monitoring Daya Listrik Dan Pengontrolan Perangkat Elektronik Berbasis IOT*.
- Sunawar, A., & Daryanto. (2022). *Teknik Pengerjaan Listrik*. Bumi Aksara.
- Wicaksono, M. F. (2017). *Aplikasi Arduino Dan Sensor*. Bi Obses 2.
- Sukma Wirawan (2019). Perancangan Aplikasi *Monitoring* Gardu Distribusi
- Raharjo, B. (2022). *Pemrogramman Flutter Edisi 2*. Informatika.
- Saepudin Nirwan, MS Hafidz (2020). Rancang Bangun Aplikasi Untuk Prototipe Sistem Monitoring Konsumsi Energi Listrik Pada Peralatan Elektronik.
- Indar Chaerah Gunadin, Zity Aida Bahtiiar, Muhammad AL Izzza Hisbullah (2021). Perancangan Sistem Online Monitorinf Pembebanan Transformator Distribusi Menggunakan Mikrokontroller
- Muhammad Faisal Kartakusumah, Ayi Purbasari,DS (2019) *Prototype Sistem Monitoring Gardu Melalui SMS Gateway Berbasis Mikrokontroler*