**RANCANG BANGUN SISTEM *MONITORING* DAN *CONTROLLING* PADA PERANGKAT *AUTOMATIC TRANSFER SWITCH* (ATS) DENGAN PEMANFAATAN *INTERNET OF THINGS* (IOT) BERBASIS *WEBSITE***

**TUGAS AKHIR**

**IRVAN TONNY WIDJAYA**

**119400018**



PROGRAM STUDI TEKNIK TELEKOMUNIKASI

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI SUMATERA

2025

**RANCANG BANGUN SISTEM *MONITORING* DAN *CONTROLLING* PADA PERANGKAT *AUTOMATIC TRANSFER SWITCH* (ATS) DENGAN PEMANFAATAN *INTERNET OF THINGS* (IOT) BERBASIS *WEBSITE***

**TUGAS AKHIR**

Karya tulis sebagai salah satu syarat untuk memperoleh gelar Sarjana dari Institut Teknologi Sumatera

IRVAN TONNY WIDJAYA

119400018



PROGRAM STUDI TEKNIK TELEKOMUNIKASI

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI SUMATERA

2025

# **HALAMAN PENGESAHAN**

Tugas Akhir Sarjana dengan judul **“Rancang Bangun Sistem *Monitoring* dan *Controlling* pada Perangkat *Automatic Transfer Switch* (ATS) dengan Pemanfaatan *Internet of Things* (IoT) berbasis *Website*”** adalah benar dibuat oleh saya sendiri dan belum pernah dibuat dan diserahkan sebelumnya, baik sebagian ataupun seluruhnya, baik oleh saya ataupun orang lain, baik di Institut Teknologi Sumatera maupun di institusi pendidikan lainnya.

|  |  |
| --- | --- |
| Lampung Selatan, 15 Juli 2025  Penulis,  Irvan Tonny Widjaya  119400018 |  |

Diperiksa dan disetujui oleh,

|  |  |
| --- | --- |
| Pembimbing I | Pembimbing II |
| Syamsyarief Baqaruzi, S.T., M.T.  NIP. 199009072019031015 | Muhammad Rizky Hikmatullah, S.T., M.T  NIP. 198812302024061001 |

Disahkan oleh,

Koordinator Program Studi Teknik Telekomunikasi

Novalia Pertiwi, S.Pd., M.T.

NIP. 199311202022032013

# **LEMBAR PERSETUJUAN**

Tugas Akhir Sarjana dengan judul **“Rancang Bangun Sistem *Monitoring* dan *Controlling* pada Perangkat *Automatic Transfer Switch* (ATS) dengan Pemanfaatan *Internet of Things* (IoT) berbasis *Website*”** adalah benar dibuat oleh saya sendiri dan belum pernah dibuat dan diserahkan sebelumnya, baik sebagian ataupun seluruhnya, baik oleh saya ataupun orang lain, baik di Institut Teknologi Sumatera maupun di institusi pendidikan lainnya.

|  |  |
| --- | --- |
| Lampung Selatan, 15 Juli 2025  Penulis,  Irvan Tonny Widjaya  119400018 |  |

Diperiksa dan disetujui oleh,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pembimbing I | : | Syamsyarief Baqaruzi, S.T., M.T.  NIP. 199009072019031015 | ………….. |  |
| Pembimbing II | : | Muhammad Rizky Hikmatullah, S.T., M.T  NIP. 198812302024061001 | ………….. |  |
| Penguji I | : | Purwono Prasetyawan, S.T., M.T.  NIP. 198505252024211022 | |  | | --- | | ………….. | |  |
| Penguji II | : | Afit Miranto, S.T., M.T.  NIP. 199105122022031007 | ………….. |  |

# **HALAMAN PERNYATAAN ORISINALITAS**

**Skripsi ini adalah karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan benar.**

|  |  |
| --- | --- |
| **Nama** | **: Irvan Tonny Widjaya** |
| **NIM** | **: 119400018** |
| **Tanggal** | **: 15 Juli 2025** |
| **Tanda Tangan** | **:** |

# **HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI**

Sebagai civitas akademik Institut Teknologi Sumatera, saya yang bertanda tangan di bawah ini:

Nama : Irvan Tonny Widjaya

NIM : 119400018

Program Studi : Teknik Telekomunikasi

Jurusan : Fakultas Teknologi dan Industri

Jenis karya : Ilmiah

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Teknologi Sumatera **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty Free Right*)** atas karya ilmiah saya yang berjudul:

**“Rancang Bangun Sistem *Monitoring* dan *Controlling* pada Perangkat *Automatic Transfer Switch* (ATS) dengan Pemanfaatan *Internet of Things* (IoT) berbasis *Website*”**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Institut Teknologi Sumatera berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di: Lampung Selatan

Pada tanggal: 15 Juli 2025

Yang menyatakan Irvan Tonny Widjaya

# **ABSTRAK**

Penggunaan energi listrik di Indonesia yang di kelola oleh PLN, setiap tahunnya mengalami peningkatan. Hal ini dapat menyebabkan gangguan pada jaringan listrik dan dapat mengakibatkan pemadaman, sehingga dibutuhkan sumber energi listrik lainnya seperti PLTS sebagai *backup* atau sumber listrik ke-2.

Untuk melakukan distribusi beban listrik antara sumber listrik PLN dan PLTS diperlukan sistem kontrol ATS yang dapat di *monitoring* dan *controlling* dari jarak jauh dengan memanfaatkan IoT. Sistem kontrol ATS yang dirancang menggunakan sensor ZMPT101B, sensor Tegangan DC, sensor ACS712, dan ADC ADS1115. Sistem *monitoring* dan *controlling* dibagun menggunakan *framework nuxt* dan *firebase realtime database* untuk menghasilkan sebuah halaman *website* yang dapat melakukan *monitoring* dan *controlling* pada ATS.

Dari hasil pengujian menunjukkan bahwa sistem yang dirancang dapat bekerja dengan baik untuk melakukan distribusi beban listrik antara sumber listrik PLN dan PLTS. Hasil dari pengujian sensor ZMPT101B menunjukkan kesalahan ukur sekitar 0,13% dan 1,39% dan pengujian sensor Tegangan DC menunjukkan kesalahan ukur sekitar 0,08%. Pengujian juga dilakukan pada sensor ACS712 yang menunjukkan kesalahan ukur berkisar 5,57% hingga 6,58% saat beban terhubung ke sumber listrik PLN dan 7,43% hingga 10,04% saat beban terhubung ke sumber listrik PLTS. Pengujian sistem *monitoring* yang dirancang juga dapat menampilkan data secara *realtime* saat terdapat perubahan data pada *database* setiap 2 sampai 3 detik sekali. Untuk sistem *controlling*, menunjukkan hasil dapat bekerja dengan baik sesuai dengan *flowchart* yang dibuat.

Kata Kunci: ATS, *monitoring*, *controlling*, *website*

# ***ABSTRACT***

*The usage of electrical energy in Indonesia managed by PLN, increases every year. This can cause disruption to the electricity network and can result in blackouts, so other sources of electrical energy are needed such as PLTS as a backup or 2nd source of electricity.*

*To distribute the electrical load between the PLN and PLTS electricity source, an ATS control system is needed that can be monitored and controlled remotely by utilizing IoT. The ATS control system designed uses the ZMPT101B sensor, DC Voltage sensor, ACS712 sensor, and ADC ADS1115. The monitoring and controlling system are built using the nuxt framework and firebase real time database to produce a website page that can monitor and control the ATS.*

*The test results show that the designed system can work well to distribute the electrical load between the PLN electricity source and PLTS. The results of the ZMPT101B sensor test showed a measurement error of around 0.13% and 1.39% and the DC Voltage sensor test showed a measurement error of around 0.08%. Testing was also carried out on the ACS712 sensor which showed a measurement error ranging from 5.57% to 6.58% when the load was connected to the PLN electricity source and 7.43% to 10.04% when the load was connected to the PLTS electricity source. Testing of the designed monitoring system can also display data in real time when there is a change in data in the database every 2 to 3 seconds. For the controlling system, it shows that the results can work well according to the created flowchart.*

*Keywords: ATS, monitoring, controlling, website*

# **MOTTO**

“Teruslah tersenyum bahagia dan selalu bersyukur walau hidup sedang tidak baik-baik saja”

# **PERSEMBAHAN**

Dengan diselesaikannya skripsi ini, Penulis mengucapkan rasa syukur kepada Allah SWT dan terima kasih kepada pihak-pihak yang telah membantu penulis dalam menyelesaikannya. Skripsi ini saya persembahkan kepada:

1. Kedua orang tua dan kakek nenek saya yang tak pernah lelah mendoakan, mendidik, dan mendukung penulis yang tak ada hingga nya.
2. Bapak Syamsyarief Baqaruzi, S.T., M.T. dan Bapak Muhammad Rizky Hikmatullah, S.T., M.T sebagai dosen pembimbing saya yang tidak pernah lelah membimbing penulis sampai penulis dapat menyelesaikan skripsi ini dengan baik.
3. Bapak Muhammad Rizky Hikmatullah, S.T., M.T selaku dosen wali saya.
4. Bapak dan Ibu dosen prodi Teknik Telekomunikasi yang saya cintai.
5. Teman-teman ku yang tak bisa di sebutkan satu per satu yang penulis cintai dan sayangi.
6. Anak-anak Kos Pakde Dakir yang selalu memberikan motivasi dan semangat kepada penulis.
7. Sayangku yang ada disana yang sedang menunggu penulis menyelesaikan skripsi ini (walaupun belum punya).
8. Waifu-waifu ku yang ada di dalam game.
9. Almamater Institut Teknologi Sumatera dan kampus Institut Teknologi Sumatera yang saya banggakan.

# **KATA PENGANTAR**

Puji syukur kita curahkan kepada Allah SWT yang telah melimpahkan rahmat, hidayah, dan inayah-Nya sehingga penulis bisa menyelesaikan tugas akhir mengenai **“Rancang Bangun Sistem *Monitoring* dan *Controlling* pada Perangkat *Automatic Transfer Switch* (ATS) dengan Pemanfaatan *Internet of Things* (IoT) berbasis *Website*”**.

Penulis juga berharap semoga dengan diajukan ini nantinya akan bermanfaat bagi semua orang. Penulis juga tidak lupa untuk mengucapkan terima kasih kepada:

1. Kedua orang tua dan kakek nenek saya yang tak pernah lelah mendoakan, mendidik, dan mendukung penulis yang tak ada hingga nya.
2. Bapak dan Ibu dosen yang selalu tak henti-hentinya memberi dukungan, bimbingan dan arahan kepada penulis.
3. Teman-teman dari prodi Teknik Telekomunikasi Institut Teknologi Sumatera.
4. Sahabat-sahabatku yang tidak bisa disebutkan satu persatu dan semua pihak yang telah membantu penulis dalam menyelesaikan tugas akhir ini.

Lampung Selatan, 15 Juli 2025

Irvan Tonny Widjaya

119400018

# **DAFTAR ISI**

[HALAMAN PENGESAHAN iii](#_Toc203659549)

[LEMBAR PERSETUJUAN iv](#_Toc203659550)

[HALAMAN PERNYATAAN ORISINALITAS v](#_Toc203659551)

[HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI vi](#_Toc203659552)

[ABSTRAK vii](#_Toc203659553)

[*ABSTRACT* viii](#_Toc203659554)

[MOTTO ix](#_Toc203659555)

[PERSEMBAHAN x](#_Toc203659556)

[KATA PENGANTAR xi](#_Toc203659557)

[DAFTAR ISI xii](#_Toc203659558)

[DAFTAR GAMBAR xv](#_Toc203659559)

[DAFTAR TABEL xviii](#_Toc203659560)

[DAFTAR SINGKATAN xix](#_Toc203659561)

[BAB I PENDAHULUAN 1](#_Toc203659562)

[1.1 Latar Belakang 1](#_Toc203659563)

[1.2 Rumusan Masalah 4](#_Toc203659564)

[1.3 Tujuan Penelitian 4](#_Toc203659565)

[1.4 Batasan Masalah 4](#_Toc203659566)

[1.5 Manfaat Penelitian 5](#_Toc203659567)

[1.6 Sistematika Penulisan 5](#_Toc203659568)

[BAB II TINJAUAN PUSTAKA 7](#_Toc203659569)

[2.1 Perangkat Listrik 7](#_Toc203659570)

[2.2 Pembangkit Listrik Tenaga Surya (PLTS) 8](#_Toc203659571)

[2.1.1 *Solar Charger Controller* (SCC) 9](#_Toc203659572)

[2.1.2 Baterai 10](#_Toc203659573)

[2.1.3 *Inverter* 11](#_Toc203659574)

[2.3 *Automatic Transfer Switch* (ATS) 12](#_Toc203659575)

[2.4 *Miniature Circuit Breaker* (MCB) 13](#_Toc203659576)

[2.5 Mikrokontroller ESP32 14](#_Toc203659577)

[*2.6* *Relay* 15](#_Toc203659578)

[2.7 Sensor ZMPT101B 16](#_Toc203659579)

[2.8 Sensor ACS712 17](#_Toc203659580)

[2.9 Sensor Tegangan DC 18](#_Toc203659581)

[2.10 ADS1115 *Converter* 19](#_Toc203659582)

[2.11 *Firebase* *Realtime Database* 20](#_Toc203659583)

[2.12 *Framework Nuxt* 21](#_Toc203659584)

[BAB III METODE PENELITIAN 23](#_Toc203659585)

[3.1 Metode Penelitian 23](#_Toc203659586)

[3.2 Perancangan Prototipe ATS 25](#_Toc203659587)

[3.2.1 Alat dan Bahan 25](#_Toc203659588)

[3.2.2 *Flowchart* Sistem ATS 25](#_Toc203659589)

[3.2.3 Rangkaian Prototipe ATS 29](#_Toc203659590)

[3.2.4 Rangkaian *Input* Sistem ATS 31](#_Toc203659591)

[3.2.5 Rangkaian *Output* Sistem ATS 32](#_Toc203659592)

[3.3 Perancangan Sistem *Monitoring* dan *Controlling* 33](#_Toc203659593)

[3.4 Perancangan PLTS 37](#_Toc203659594)

[3.4.1 Kebutuhan Beban 37](#_Toc203659595)

[3.4.2 Alat dan Bahan 38](#_Toc203659596)

[3.4.3 Rangkaian PLTS 39](#_Toc203659597)

[BAB IV HASIL DAN PEMBAHASAN 40](#_Toc203659598)

[4.1 Hasil Perancangan Prototipe ATS 40](#_Toc203659599)

[4.2 Pengujian Sensor Tegangan ZMPT101B 41](#_Toc203659600)

[4.3 Pengujian Sensor Tegangan DC 43](#_Toc203659601)

[4.4 Pengujian Sensor Arus ACS712 44](#_Toc203659602)

[4.5 Pengujian Sistem *Monitoring* dan *Controlling* 51](#_Toc203659603)

[BAB V KESIMPULAN DAN SARAN 57](#_Toc203659604)

[5.1 Kesimpulan 57](#_Toc203659605)

[5.2 Saran 58](#_Toc203659606)

[DAFTAR PUSTAKA 59](#_Toc203659607)

[LAMPIRAN 64](#_Toc203659608)

# **DAFTAR GAMBAR**

[Gambar 2.1 Proses Pembangkitan Arus Listrik Pada Sel Surya [10] 8](#_Toc203545596)

[Gambar 2.2 Panel Surya *Polycrystalline* [10] 9](#_Toc203545597)

[Gambar 2.3 *Solar Charger Controller* (SCC) [11] 10](#_Toc203545598)

[Gambar 2.4 Baterai *Acid* [12] 11](#_Toc203545599)

[Gambar 2.5 Struktur Baterai *Acid* [13] 11](#_Toc203545600)

[Gambar 2.6 *Inverter* DC to AC [16] 12](#_Toc203545601)

[Gambar 2.7 ATS Berbasis Mikrokontroller ESP32 13](#_Toc203545602)

[Gambar 2.8 *Miniature Circuit Breaker* (MCB) [18] 14](#_Toc203545603)

[Gambar 2.9 Bagian Mikrokontroller ESP32 [19] 14](#_Toc203545604)

[Gambar 2.10 *Pinout* Mikrokontroller ESP32 [20] 15](#_Toc203545605)

[Gambar 2.11 *Pinout* Relay [22] 15](#_Toc203545606)

[Gambar 2.12 *Relay* SPDT dan *Relay* DPDT 16](#_Toc203545607)

[Gambar 2.13 Skematik ZMPT101B [24] 16](#_Toc203545608)

[Gambar 2.14 Sensor ZMPT101B [25] 17](#_Toc203545609)

[Gambar 2.15 Skematik ACS712 [26] 17](#_Toc203545610)

[Gambar 2.16 Sensor ACS712 [27] 18](#_Toc203545611)

[Gambar 2.17 Sensor Tegangan DC [29] 19](#_Toc203545612)

[Gambar 2.18 *Diagram Block* ADS1115 [30] 19](#_Toc203545613)

[Gambar 2.19 ADS1115 ADC *Converter* [31] 20](#_Toc203545614)

[Gambar 2.20 *Firebase Realtime Database* [33] 21](#_Toc203545615)

[Gambar 2.21 Kode Halaman Monitoring 22](#_Toc203545616)

[Gambar 3.1 *Flowchart* Garis Besar Metode Penelitian 23](#_Toc198547767)

[Gambar 3.2 *Flowchart* Sistem ATS 27](#_Toc198547768)

[Gambar 3.3 *Diagram Block* Prototipe ATS 29](#_Toc198547769)

[Gambar 3.4 Skematik Rangkaian Prototipe ATS Menggunakan KiCad 30](#_Toc198547770)

[Gambar 3.5 Rangkaian *Input* Sistem ATS Menggunakan KiCad 31](#_Toc198547771)

[Gambar 3.6 Rangkaian *Output* Sistem ATS Menggunakan KiCad 32](#_Toc198547772)

[Gambar 3.7 Tampilan Halaman Sistem *Monitoring* 33](#_Toc198547773)

[Gambar 3.8 Tampilan Halaman Histori 35](#_Toc198547774)

[Gambar 3.9 Tampilan Halaman Sistem *Controlling* Kontrol Otomatis *On* 36](#_Toc198547775)

[Gambar 3.10 Tampilan Halaman Sistem *Controlling* Kontrol Otomatis *Off* 36](#_Toc198547776)

[Gambar 3.11 *Diagram Block* Rangkaian PLTS *Offgrid* 39](#_Toc198547777)

[Gambar 4.1 Hasil Perancangan Prototipe ATS 40](#_Toc198781911)

[Gambar 4.2 Hasil Pengujian Sensor ZMPT101B I 41](#_Toc198781912)

[Gambar 4.3 Contoh Pengujian Pengukuran Sensor ZMPT101B PLN 42](#_Toc198781913)

[Gambar 4.4 Hasil Pengujian Sensor ZMPT101B PLTS 42](#_Toc198781914)

[Gambar 4.5 Contoh Pengujian Pengukuran Sensor ZMPT101B PLTS 43](#_Toc198781915)

[Gambar 4.6 Hasil Pengujian Sensor Tegangan DC 44](#_Toc198781916)

[Gambar 4.7 Contoh Pengujian Pengukuran Sensor Tegangan DC 44](#_Toc198781917)

[Gambar 4.8 Hasil Pengujian Sensor Arus ACS712 I Terhadap Beban 45](#_Toc198781918)

[Gambar 4.9 Contoh Pengujian Sensor ACS712 I PLN 46](#_Toc198781919)

[Gambar 4.10 Contoh Pengujian Sensor ACS712 I PLTS 46](#_Toc198781920)

[Gambar 4.11 Hasil Pengujian Sensor Arus ACS712 II Terhadap Beban 47](#_Toc198781921)

[Gambar 4.12 Contoh Pengujian Sensor ACS712 II PLN 48](#_Toc198781922)

[Gambar 4.13 Contoh Pengujian Sensor ACS712 II PLTS 48](#_Toc198781923)

[Gambar 4.14 Hasil Pengujian Sensor Arus ACS712 III Terhadap Beban 49](#_Toc198781924)

[Gambar 4.15 Contoh Pengujian Sensor ACS712 III PLN 50](#_Toc198781925)

[Gambar 4.16 Contoh Pengujian Sensor ACS712 III PLTS 50](#_Toc198781926)

[Gambar 4.17 Hasil Pengujian Sistem Monitoring 51](#_Toc198781927)

[Gambar 4.18 Hasil Pengujian Halaman Histori (A) 52](#_Toc198781928)

[Gambar 4.19 Hasil Pengujian Halaman Histori (B) 52](#_Toc198781929)

# **DAFTAR TABEL**

[Tabel 3.1 Kebutuhan Alat dan Bahan Prototipe ATS 25](#_Toc203546395)

[Tabel 3.2 Konfigurasi Maksimal Beban Pada Relai 37](#_Toc203546396)

[Tabel 3.3 Beban Lampu Pijar 38](#_Toc203546397)

[Tabel 3.4 Kebutuhan Alat dan Bahan Perancangan PLTS 38](#_Toc203546398)

[Tabel 4.1 Hasil Pengujian *Switch On/Off* Sumber Listrik PLN dan PLTS 53](#_Toc198087254)

[Tabel 4.2 Hasil Pengujian Kontrol Otomatis 54](#_Toc198087255)

# **DAFTAR SINGKATAN**

|  |  |  |  |
| --- | --- | --- | --- |
| Singkatan | Kepanjangan | Satuan | Pemakaian Pertama Kali Pada Halaman |
| PLN | Perusahaan Listrik Negara |  | 1 |
| GWh | *Giga Watt Hour* |  | 1 |
| MW | *Mega Watt* |  | 1 |
| PLTS | Pembangkit Listrik Tenaga Surya |  | 2 |
| AC | *Alternating Current* |  | 2 |
| DC | *Direct Current* |  | 2 |
| ATS | *Automatic Transfer Switch* |  | 2 |
| IoT | *Internet of Things* |  | 2 |
| MCB | *Miniature Circuit Breaker* |  | 3 |
| VA | *Volt Ampere* | VA | 4 |
| W | *Watt* | W | 6 |
| Wh | *Watt Hour* | Wh | 6 |
| SCC | *Solar Charger Controller* |  | 8 |
| BCR | *Battery Charge Regulator* |  | 8 |
| Ah | *Ampere Hour* | Ah | 9 |
| Hz | *Hertz* | Hz | 11 |
| V | *Volt* | V | 11 |
| A | *Ampere* | A | 11 |
| GHz | *Giga Hertz* | GHz | 12 |
| ADC | *Analog to Digital Converter* |  | 12 |
| DAC | *Digital to Analog Converter* |  | 12 |
| NO | *Normally Open* |  | 13 |
| NC | *Normally Close* |  | 13 |
| SPDT | *Single Pole Double Throw* |  | 14 |
| DPDT | *Double Pole Double Throw* |  | 14 |
| VCC | *Voltage at the Common Collector* |  | 15 |
| GND | *Ground* |  | 15 |
| KΩ | *Kilo Ohm* | KΩ | 16 |
| PGA | *Programmable Gain Amplifier* |  | 17 |
| SPS | *Sample per Second* |  | 17 |
| mV | *mili Volt* | mV | 17 |
| *NoSQL* | *Non-Structured Query Language* |  | 17 |
| JSON | *Javascript Object Nation* |  | 17 |
| SFC | *Sigle File Component* |  | 19 |
| HTML | *Hypertext Markup Language* |  | 19 |
| CSS | *Cascading Style Sheets* |  | 19 |
| mA | *mili Ampere* | mA | 21 |
| VAC | *Volt Alternating Current* |  | 21 |
| VDC | *Volt Direct Current* |  | 21 |
| Wp | *Watt Power* | Wp | 32 |

# **BAB I PENDAHULUAN**

## **Latar Belakang**

Penggunaan energi listrik di Indonesia sebagian besar masih bergantung pada listrik yang di hasilkan oleh generator-generator pembangkit listrik berbahan bakar fosil (seperti minyak bumi atau batu bara) yang di kelola dan di distribusikan oleh PLN [1]. Energi listrik sendiri memiliki peran yang sangat penting dalam kehidupan manusia. Akan tetapi, seiring berjalannya waktu kebutuhan akan penggunaan energi listrik juga semakin meningkat. Berdasarkan statistik PLN pada tahun 2023, produksi energi listrik yang di hasilkan oleh PLN meningkat sebesar 15.156,46 GWhatau sebesar 12.20% dari tahun sebelumnya yang hanya 308.002,30 GWh menjadi 323.320,62 GWh dengan beban puncak yang meningkat sebesar 16.481,58 MW atau sebesar 39.43% menjadi 58.282,48 MW yang sebelumnya hanya 41.800,90 MW [2]. Dengan bertambahnya beban yang harus ditanggung oleh generator, sering kali menyebabkan gangguan pada jaringan kelistrikan yang ada dan dapat mengakibatkan pemadaman listrik karena keterbatasan *supply* energi listrik. Hal ini dapat menimbulkan kerusakan pada perangkat elektronik yang tidak memiliki cadangan listik pada rangkaiannya. Untuk mengatasi masalah ini, dibutuhkan sumber energi listrik lainnya yang dapat digunakan sebagai *backup* atau sumber energi listrik ke-2 untuk memenuhi kebutuhan akan energi listrik [3].

Untuk menghasilkan energi listrik, terdapat berbagai macam sumber energi yang terdapat di bumi untuk bisa menghasilkan energi listrik. Sumber energi sendiri terbagi menjadi 2, yaitu energi terbarukan (*renewable energy*) maupun energi tidak terbarukan (*non-renewable energy*). Di Indonesia, pemanfaat energi terbarukan untuk memenuhi kebutuhan energi listrik telah diatur dalam peraturan Menteri Energi dan Sumber Daya Mineral (ESDM) no 53 tahun 2018 tentang pemanfaatan sumber Energi Baru Terbarukan (EBT) untuk penyediaan tenaga listrik [4]. Dalam pemanfaatan EBT untuk memenuhi kebutuhan listrik, beberapa pembangkit listrik di Indonesia telah memanfaatkan energi cahaya matahari sebagai sumber energi utamanya dengan menggunakan panel surya atau yang sering kita sebut dengan PLTS [2].

Dengan seiring berjalannya waktu, pemanfaatan panel surya juga sudah banyak digunakan dan diterapkan di berbagai tempat. Baik pada skala besar seperti di sektor industri ataupun skala kecil seperti di sektor perumahan untuk mencukupi kebutuhan akan energi listrik. Panel surya sendiri merupakan komponen utama dari PLTS yang dapat menghasilkan energi listrik melalui sel surya yang bekerja dengan efek *photovoltaic*. Efek *photovoltaic* dapat mengubah atau mengkonversikan radiasi sinar foton matahari menjadi energi listrik. Arus listrik yang di hasilkan oleh panel surya sendiri adalah listrik searah atau DC yang kemudian akan diubah menjadi listrik bolak-balik atau AC dengan menggunakan *inverter* agar dapat digunakan pada peralatan elektronik sehari-hari [5].

Pada skala kecil seperti perumahan, PLTS sering digunakan sebagai sumber listrik cadangan saat sumber listrik utama mengalami pemadaman atau sebagai sumber listrik ke-2 untuk memenuhi kebutuhan energi listrik. Dalam mengatur *supply* energi listrik baik yang berasal dari sumber listrik utama (PLN) ataupun sumber listrik cadangan (PLTS) secara efektif dan responsif, maka dibutukan sistem pendukung yang disebut dengan ATS untuk melakukan pengaturan sumber atau perpindahan sumber secara otomatis [6]. Ditambah dengan perkembangan teknologi yang tumbuh semakin cepat, memungkinkan untuk melakukan *monitoring* dan *controlling* dari jarak jauh dengan pemanfaatan IoT. IoT sendiri memiliki peran yang sangat besar untuk menghubungkan manusia, sistem, perangkat, dan berbagai hal lainnya melalui jaringan internet agar dapat berkomunikasi satu sama lain.

Dalam perancangan sistem ATS, terdapat mikrokontroller ESP32 yang bertugas untuk memproses data yang diterima dari sensor-sensor yang digunakan seperti sensor tengangan AC ZMPT101B, sensor arus ACS712 dan sensor tegangan DC. Data-data hasil pengukuran tersebut kemudian akan diproses oleh mikrokontroller dan di simpan ke dalam *database* *firebase* yang kemudian akan mengatur posisi relai agar beban dapat terhubung dengan sumber energi listrik yang di inginkan sesuai dengan program yang telah dimasukkan ke dalam mikrokontroller tersebut. Sebelum beban terhubung langsung dengan sumber listrik yang di inginkan, terdapat *Miniature Circuit Breaker* (MCB) sebagai pengaman jika terjadi hubung singkat listrik ataupun beban berlebih (*overload*) agar tidak merusak komponen yang terdapat pada sistem ATS terutama pada komponen *inverter*. Data hasil pengukuran sensor yang disimpan ke dalam *database* kemudian akan di tampilkan ke dalam sebuah *website* yang akan digunakan untuk melakukan *monitoring* dan *controlling* perangkat ATS agar dapat berjalan sesuai yang diinginkan. Sehingga, berdasarkan hal tersebut penulis menyusun tugas akhir dengan judul **“Rancang Bangun Sistem *Monitoring* dan *Controlling* pada Perangkat *Automatic Transfer Switch* (ATS) dengan Pemanfaatan *Internet of Things* (IoT) berbasis *Website*”.**

## **Rumusan Masalah**

Berdasarkan latar belakang tersebut, penulis merumuskan masalah sebagai berikut:

1. Melakukan distribusi beban listrik yang terhubung antara sumber listrik PLN dan PLTS.
2. Melakukan *monitoring* penggunaan listrik serta melakukan *controlling* sumber listrik antara listrik PLN dan PLTS dari jarak jauh.

## **Tujuan Penelitian**

Penelitian ini dilakukan penulis untuk memahami beberapa hal, yaitu:

1. Merancang sistem ATS untuk distribusi beban listrik yang terhubung antara sumber listrik PLN dan PLTS.
2. Merancang sistem yang dapat melakukan *monitoring* dan *controlling* dari jarak jauh dengan memanfaatkan IoT melalui sebuah *website*.

## **Batasan Masalah**

Untuk lebih memfokuskan penelitian ini, terdapat beberapa batasan-batasan masalah yang ditentukan oleh penulis, yaitu:

1. Sumber listrik utama berasal dari sumber listrik PLN dengan kapasitas 450VA dan sumber listrik kedua berasal dari sumber listrik PLTS dengan kapasitas 220VA.
2. Sistem PLTS yang dirancang adalah sistem *offgrid* (tidak terhubung dengan PLN).
3. Mikrokontroller yang digunakan untuk memproses data adalah ESP32.
4. Parameter dari prototipe ATS bedasarkan dari hasil pembacaan tegangan listrik pada PLN, PLTS, dan baterai, dan arus listrik yang terhubung ke beban.
5. *Framework* yang digunakan untuk membangun sebuah *website* pada penelitian ini menggunakan *nuxt* dan *database* yang digunakan untuk menyimpan data pengukuran sensor menggunakan *firebase*.
6. Pembahasan terfokus pada fungsional alat yang dapat memindahkan sumber listrik sesuai dengan konfigurasi sumber listrik dan maksimal beban yang telah ditetapkan melalui *website*.
7. Pengujian alat dan pengambilan data pada prototipe ini dilakukan secara *real time*.

## **Manfaat Penelitian**

Dari penelitian yang dilakukan, manfaat yang dapat diberikan sebagai berikut:

1. Hasil penelitian diharapkan dapat dijadikan referensi dalam merancang sistem *monitoring* dan *controlling* pada perangkat IoT.
2. Hasil penelitian diharapkan dapat menambah wawasan dan pengetahuan dalam bidang PLTS.

## **Sistematika Penulisan**

Adapun sistematika penulisan yang merupakan urutan garis besar pada penulisan, yaitu:

**BAB I PENDAHULUAN**

Menjelaskan mengenai latar belakang pemilihan judul, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian dan sistematika penulisan.

**BAB II TINJAUAN PUSTAKA**

Memuat perihal dasar dari teori-teori yang akan menjadi fondasi dan memberikan arahan kepada penulis untuk mencapai tujuan penelitian.

**BAB III METODE PENELITIAN**

Menjelaskan mengenai alur dan metode penelitian yang digunakan, perancangan alat dan garis besar waktu pelaksanaan penelitian.

**BAB IV HASIL DAN PEMBAHASAN**

Menjelaskan mengenai hasil *output* dari pengujian yang dilakukan, dan membahas serta menganalisis data yang didapatkan.

**BAB V KESIMPULAN DAN SARAN**

Uraian yang berisikan rangkuman kesimpulan yang diperoleh dan juga saran-saran agar penelitian ini menjadi lebih baik.

# **BAB II TINJAUAN PUSTAKA**

## **Perangkat Listrik**

Perangkat listrik merupakan perangkat beban yang membutuhkan sumber energi listrik agar dapat beroperasi. Sumber energi listrik yang digunakan dapat berupa tegangan AC atau DC tergantung perangkat listrik yang digunakan [7]. Saat beban terhubung ke sumber listrik, terdapat daya aktif yang digunakan dalam satuan W. Untuk mendapatkan daya aktif pada saat beban terhubung ke sumber listrik, dapat digunakan Persamaan 2.1 dibawah ini [8].

|  |  |
| --- | --- |
|  | (2.1) |

Keterangan:

*P* = Daya Aktif (W)

*V* = Tegangan (V)

*I* = Arus (A)

*cos φ* = Faktor Daya

Persamaan (2.1) diatas merupakan persamaan untuk menghitung daya aktif yang digunakan saat beban terhubung ke sumber listrik dengan mengkalikan tegangan, arus, dah faktor daya. Beban listrik juga mengkonsumsi daya dalam waktu tertentu yang dinyatakan dalam satuan Wh. Untuk mendapatkan berapa jumlah konsumsi daya dalam waktu, dapat digunakan persamaan (2.2) [9].

|  |  |
| --- | --- |
|  | (2.2) |

Keterangan:

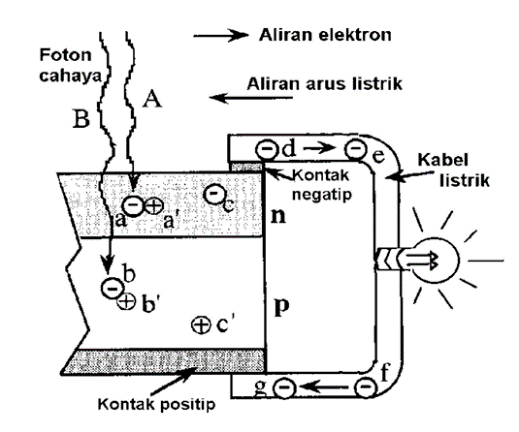
P = Daya Aktif (W)

t = Lama Pemakaian (hour)

Persamaan (2.2) merupakan persamaan untuk menghitung konsumsi daya yang digunakan oleh beban dengan mengkalikan daya aktif yang digunakan dan lama pemakaian dalam satu jam.

## **Pembangkit Listrik Tenaga Surya (PLTS)**

PLTS merupakan pembangkit listrik yang menggunakan energi cahaya matahari dengan memanfaatkan teknologi *Photovoltaic* (PV) atau sering disebut juga dengan sel surya. Sel surya sendiri terbuat dari lapisan tipis dari bahan semi konduktor, yang umumnya terbuat dari pengolahan *silicon* (Si), dengan ketebalan sekitar 0,3 mm dan dengan permukaan dari 100 hingga 225 . Timbulnya arus listrik pada sel surya yang disinari cahaya matahari adalah akibat proses absorpsi foton yang dapat menghasilkan pasangan *hole-electron*, serta pengumpulan dan pemisahan *hole* dan *electron* tersebut oleh p-n *junction* sehingga akan menghasilkan beda potensial diantara lapisan tipe-p dan tipe-n yang menghasilkan listrik searah atau DC [10].

****

Gambar 2.1 Proses Pembangkitan Arus Listrik Pada Sel Surya [10]

Panel surya yang digunakan pada PLTS memiliki berbagai jenis berdasarkan sel surya yang digunakan, pada penelitian ini digunakan panel surya dengan sel surya *polycrystalline* yang sering digunakan oleh masyarakat saat ini. Jenis sel surya ini memiliki warna biru tua yang terbuat dari beberapa batang kristal *silicon* yang dilebur atau dicairkan kemudian dituangkan dalam cetakan yang berbentuk persegi. Tampilannya nampak seperti ada motif pecahan kaca di dalamnya. Bentuknya yang persegi, jika disusun membentuk panel surya, akan rapat dan tidak akan ada ruangan kosong yang sia-sia. Sel surya jenis ini memiliki tingkat efisiensi yang rendah sekitar 13% - 16% [10].



Gambar 2.2 Panel Surya Polycrystalline [10]

Selain panel surya yang menjadi komponen utama dari PLTS, terdapat komponen-komponen pendukung lainnya yang juga tidak kalah penting dalam sistem pembangkit listrik ini yaitu:

### ***Solar Charger Controller* (SCC)**

SCC atau juga dikenal sebagai (BCR) adalah komponen yang digunakan pada PLTS untuk mengatur pengisian arus searah yang disebut dengan proses *charge* dan pengaturan penyaluran arus listrik dari baterai menuju beban listrik atau disebut dengan proses *discharge*. Fungsi utama dari SCCadalah untuk menjaga atau melindungi baterai saat menerima pengisian berlebih (*overcharge*) dari sel surya, dengan cara membatasi pengisian energi saat baterai dalam keadaan penuh dan melindungi baterai dari pengosongan berlebih (*overdischarge*) yang dikarenakan oleh beban yang digunakan, dengan cara memutuskan hubungan baterai dengan beban saat baterai dalam keadaan *low state of charge* dari panel surya ke baterai [10].



Gambar 2.3 Solar Charger Controller (SCC) [11]

### **Baterai**

Baterai merupakan salah satu komponen yang digunakan pada sistem PLTS yang dilengkapi dengan penyimpanan cadangan energi listrik. Baterai memiliki fungsi untuk menyimpan energi listrik yang dihasilkan oleh panel surya dalam bentuk energi arus searah. Satuan kapasitas energi yang disimpan pada baterai adalah *ampere hour* (Ah), yang diartikan arus maksimum yang dapat dikeluarkan oleh baterai selama satu jam. Dalam proses pengosongan (*discharge*), baterai tidak boleh dikosongkan hingga titik maksimumnya, hal ini dikarenakan agar baterai dapat bertahan lebih lama. Batas pengosongan dari baterai sering disebut dengan istilah *depth of discharge* (DOD), yang dinyatakan dalam satuan persen, biasanya ditentukan sebesar 80%.



Gambar 2.4 Baterai Acid [12]

Banyak tipe dan klasifikasi banterai yang diproduksi saat ini, yang masing-masing memiliki desain yang spesifik dan karakteristik performa berbeda sesuai dengan pengaplikasiannya. Pada sistem PLTS jenis baterai *lead-acid* lebih banyak digunakan, hal ini dikarenakan ketersediaan ukuran Ah yang lebih banyak [10].



Gambar 2.5 Struktur Baterai Acid [13]

### ***Inverter***

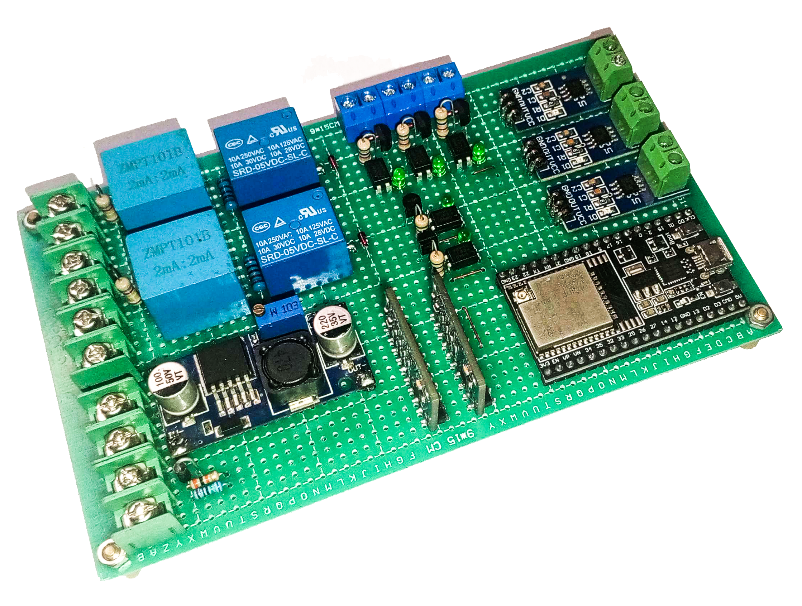
*Inverter* merupakan rangkaian elektronika yang memiliki fungsi untuk mengubah arus listrik searah atau DC yang dihasilkan oleh modul panel surya ataupun baterai menjadi arus listrik bolak balik atau AC agar dapat digunakan pada peralatan elektronik rumah [10]. Tegangan DC yang dihasilkan dari panel surya cenderung tidak konstan karena tegangan yang dihasilkan sesuai dengan tingkat radiasi matahari. Tegangan masukan DC yang tidak konstan ini akan diubah oleh *inverter* menjadi tegangan AC yang konstan dan siap digunakan atau disambungkan pada sistem yang ada [14]. Tanpa alat ini, arus searah yang dihasilkan oleh modul panel surya tidak akan dapat digunakan secara langsung pada peralatan elektronika yang umumnya membutuhkan arus bolak-balik sebagai pasokan daya utamanya [15].



Gambar 2.6 Inverter DC to AC [16]

## ***Automatic Transfer Switch* (ATS)**

ATS merupakan sakelar otomatis yang berfungsi untuk memindahkan beban dari sumber listrik utama (PLN) ke sumber listrik cadangan (PLTS) atau sebaliknya secara otomatis. Pada alat ini terdapat komponen utama seperti kontaktor atau relai yang berfungsi untuk memindahkan ke sumber daya yang diinginkan. Penggunaan kontaktor atau relai ini disesuaikan dengan besarnya kebutuhan, untuk kebutuhan skala besar, maka penggunaan kontaktor diperlukan, dan sebaliknya jika untuk kebutuhan skala kecil maka penggunaan relai sudah cukup [17].



Gambar 2.7 ATS Berbasis Mikrokontroller ESP32

## ***Miniature Circuit Breaker* (MCB)**

Bedasarkan IEC 60898 menjelaskan bahwa MCB merupakan alat yang berfungsi untuk melindungi instalasi listrik dari beban berlebih atau korsleting listrik pada suatu bangunan atau sejenisnya. Alat ini bekerja pada frekuensi 50Hz atau 60Hz dengan tegangan tidak lebih dari 440V, arus yang dilewati tidak lebih dari 125A, dan kapasitas hubung singkat tidak lebih dari 25A.

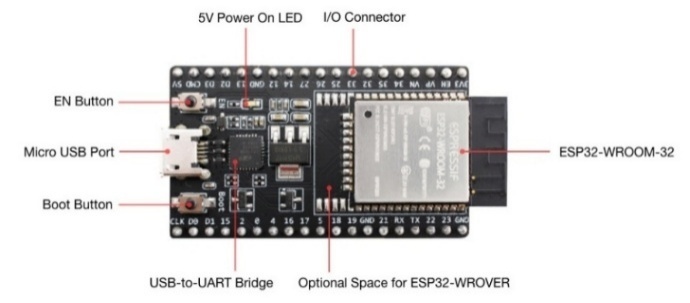
Saat terjadi kelebihan beban atau korsleting listrik, MCB secara otomatis memutus sirkuit listrik untuk mencegah kerusakan pada kabel dan menghindari risiko kebakaran. MCB dilengkapi dengan dua mekanisme pemutus: mekanisme pemutus termal tertunda untuk perlindungan kelebihan beban dan mekanisme pemutus magnetik untuk perlindungan hubungan arus pendek [18].



Gambar 2.8 Miniature Circuit Breaker (MCB) [18]

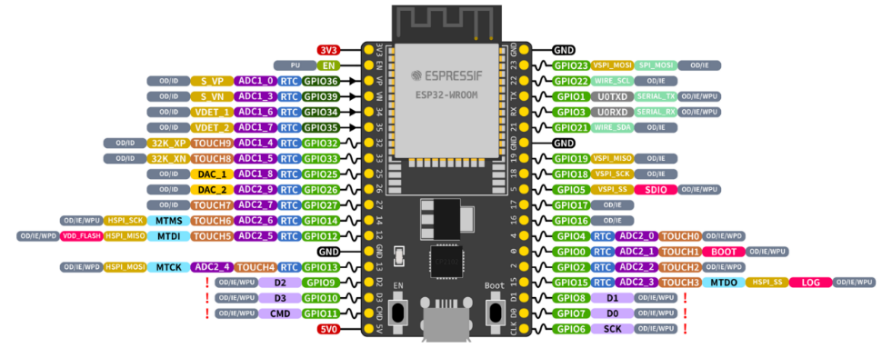
## **Mikrokontroller ESP32**

Mikrokontroller ESP32 merupakan salah satu produk dari *Espressif* yang memiliki banyak keunggulan seperti konsumsi daya yang rendah, memiliki intergrasi yang tinggi serta memiliki module Wi-Fi 2.4GHz dan *Bluetooth* di dalamnya sehingga tidak memerlukan module tambahan untuk dapat terhubung kedalam suatu jaringan *local* ataupun internet [19].



Gambar 2.9 Bagian Mikrokontroller ESP32 [19]

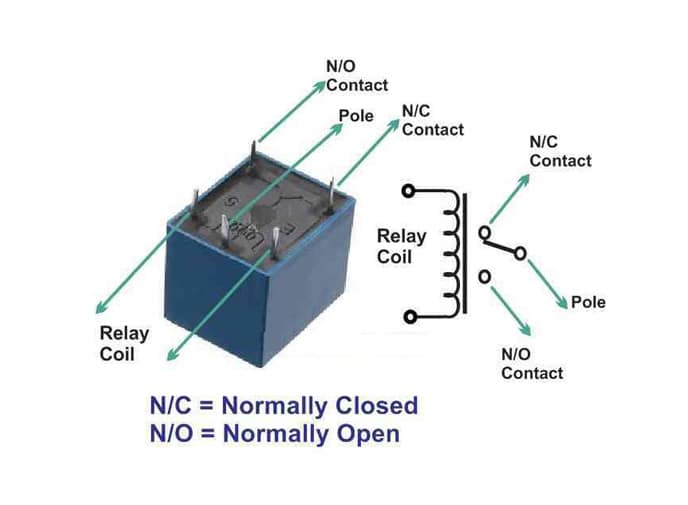
Mikrokontroller ini terdapat 38 pin *input/output* (I/O) yang dapat digunakan sesuai kebutuhan. Setiap pin memiliki fungsi yang berbeda-beda, seperti pin ADCyang berfungsi untuk mengubah sinyal input *analog* menjadi sinyal *digital*, pin DAC yang berfungsi untuk mengubah sinyal input *digital* menjadi sinyal *analog*, pin *serial* yang berfungsi untuk melakukan *debug* atau pemograman pada mikrokontroller, dan lain-lain [20].



Gambar 2.10 Pinout Mikrokontroller ESP32 [20]

## ***Relay***

*Relay* merupakan saklar elektrik yang beroperasi secara elektromagnetik dengan menggunakan sinyal elektrik. *Relay* merupakan komponen elektronika yang digunakan dalam melaksanakan logika *switching*. *Relay* membutuhkan daya yang sangat kecil untuk mengubah kontak *switch* nyadan mampu mengontrol perangkat yang membutuhkan daya yang lebih besar. Berdasarkan *contact point* nya, relai dibagi menjadi dua yaitu NO dan NC [21].



Gambar 2.11 Pinout Relay [22]

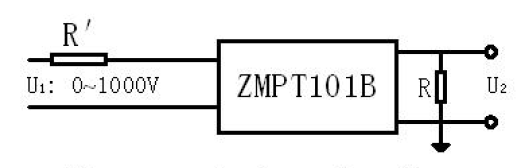
Setiap *relay* memiliki konfigurasi yang berbeda-beda tergantung penggunaannya pada rangkaian elektronika, Akan tetapi terdapat 2 istilah umum yang digunakan untuk menggambarkan konfigurasi yang terdapat pada relai yaitu: *Pole* dan *Throw*. *Pole* mengacu pada jumlah terminal umum dalam sakelar, sedangkan *Throw* mengacu pada jumlah posisi di mana sakelar dapat ditempatkan untuk membuat jalur sinyal atau koneksi. Misalnya pada relai SPDT memiliki 1 *pole* dan 2 *throw* dengan jumlah pin sebanyak 5, sedangkan pada relai DPDT memiliki 2 *pole* dan 2 *throw* dengan jumlah pin sebanyak 8 [23].



Gambar 2.12 Relay SPDT dan Relay DPDT

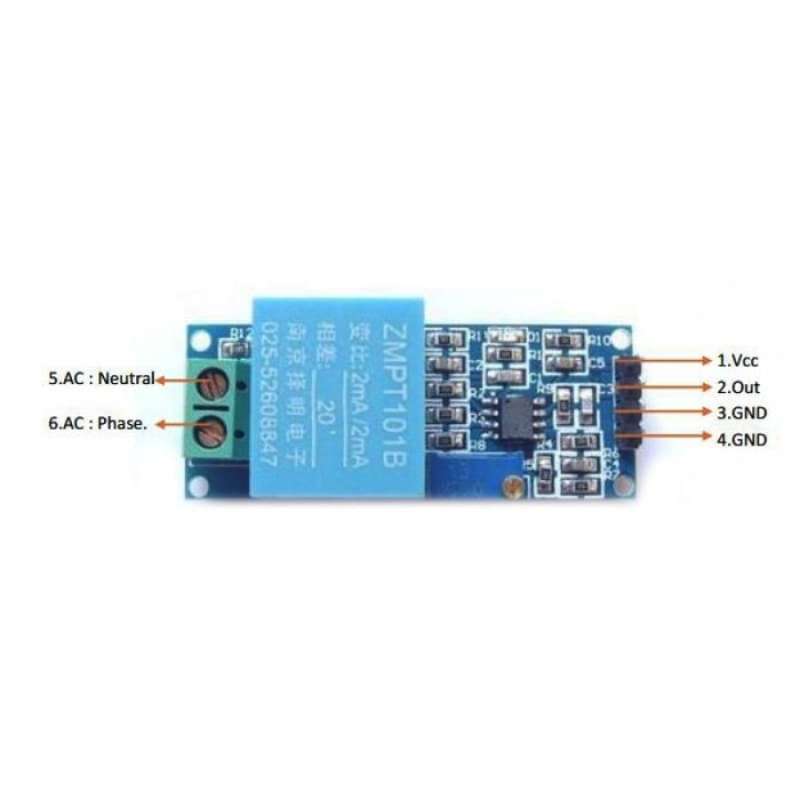
## **Sensor** **ZMPT101B**

Sensor ZMPT101B adalah sensor yang digunakan untuk mengukur tegangan listrik AC dengan presisi tinggi pada suatu rangkaian listrik yang bekerja pada tegangan 0V hingga 1000V. Sensor ini bekerja dengan cara menurunkan tegangan masukkan menggunakan *step down transformator* dengan rasio lilitan 1000:1000 yang kemudian akan disampling menggunakan resistor [24].



Gambar 2.13 Skematik ZMPT101B [24]

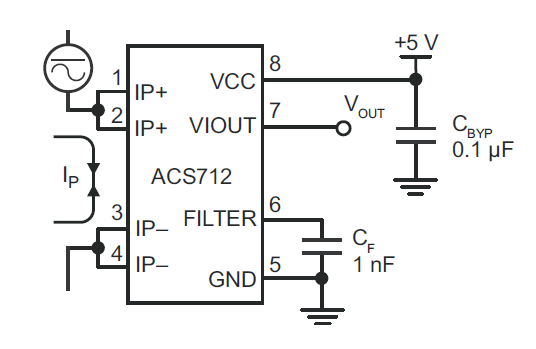
Sensor ini memiliki ukuran yang kecil. Dalam penggunaannya, pin *phase* dan *neutral* dihubungkan ke sumber listrik AC, sedangkan pin VCC dan pin GND dihubungkan ke *power supply* 5V, dan pin out dihubungkan ke pin *analog* mikrokontroller. Untuk mendapatkan output yang stabil pengaturan potensiometer dibutuhkan untuk mendapatkan gelombang sinus yang stabil.



Gambar 2.14 Sensor ZMPT101B [25]

## **Sensor ACS712**

Sensor ACS712 merupakan sensor yang digunakan untuk mengukur besar arus listrik yang mengalir pada suatu rangkaian listrik. Sensor ini menggunakan chip ACS712 yang dapat mengukur listrik AC atau DC yang terhubung pada pin 1~4. Sensor ini memiliki 3 varian: 5A, 20A dan 30A.



Gambar 2.15 Skematik ACS712 [26]

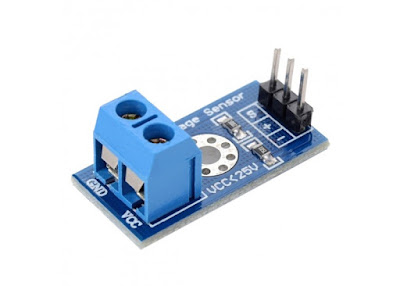
Sensor ACS712 bekerja dengan memanfaatkan efek Hall. Efek hall merupakan suatu peristiwa berbeloknya aliran listrik (elektron) dalam pelat konduktor karena pengaruh medan magnet. Sensor ini bekerja dengan memanfaatkan arus yang mengalir melalui kabel tembaga yang ada di dalam sensor tersebut yang menghasilkan medan magnet kemudian medan magnet tersebut ditangkap oleh *integrated* IC Hall dan diubah menjadi tegangan proporsional. Akurasi perangkat dioptimalkan melalui kedekatan sinyal magnetik dengan transduser Hall. Tegangan proporsional yang presisi disediakan oleh IC Hall BiCMOS dengan offset rendah dan stabil [26].



Gambar 2.16 Sensor ACS712 [27]

## **Sensor Tegangan DC**

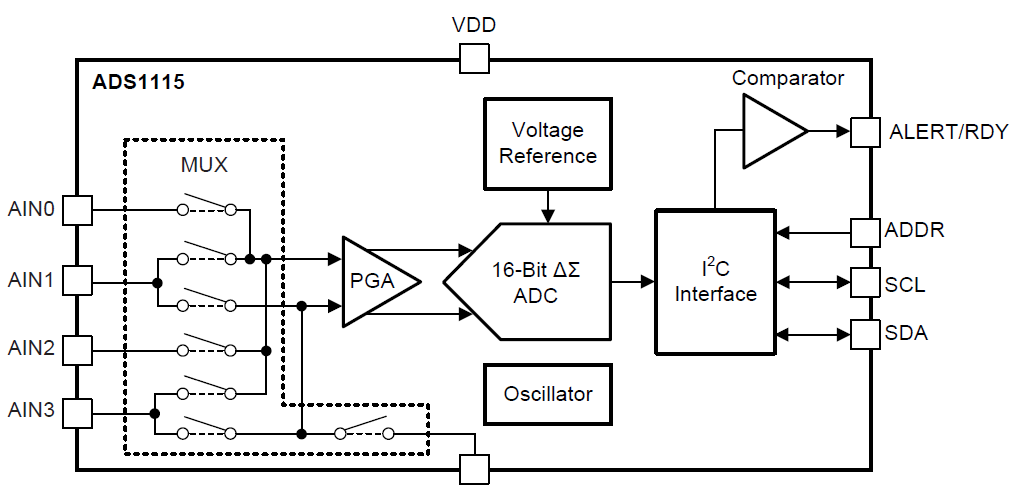
Sensor tegangan DC bekerja dengan prinsip rangkaian pembagi tegangan. Sensor ini terdapat 2 buah resistor yang bernilai 30KΩ dan 7,5KΩ yang dapat menerima *input* hingga 25V dengan *output* hingga 5V. Dengan kata lain nilai pembacaan dari sensor tegangan ini disesuaikan dengan tegangan masukan yang dapat diterima sensor [28].



Gambar 2.17 Sensor Tegangan DC [29]

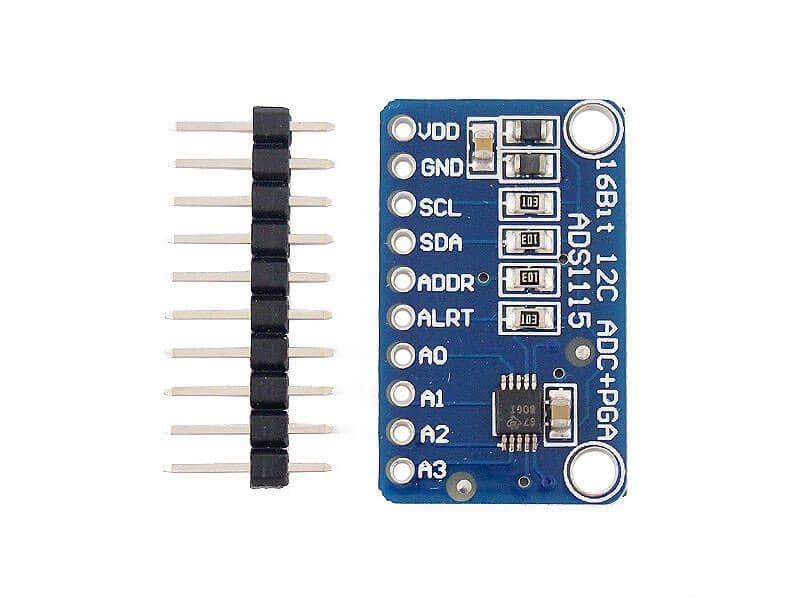
## **ADS1115 *Converter***

ADS1115 merupakan suatu *converter* yang dapat mengubah sinyal *analog* menjadi sinyal *digital* dengan presisi, *low-*power, 16-bit, dan kompatibel dengan I2C yang bekerja pada rentang tegangan 2V hingga 5.5V. Module ini bekerja dengan cara menggabungkan refrensi tegangan drift rendah dan osilator dan juga menggabungkan PGA dan pembanding digital. Dengan fitur-fitur tersebut membuat ADS1115 ini sangat cocok untuk digunakan pada aplikasi pengukuran sensor yang membutuhkan presisi yang tinggi.



Gambar 2.18 Diagram Block ADS1115 [30]

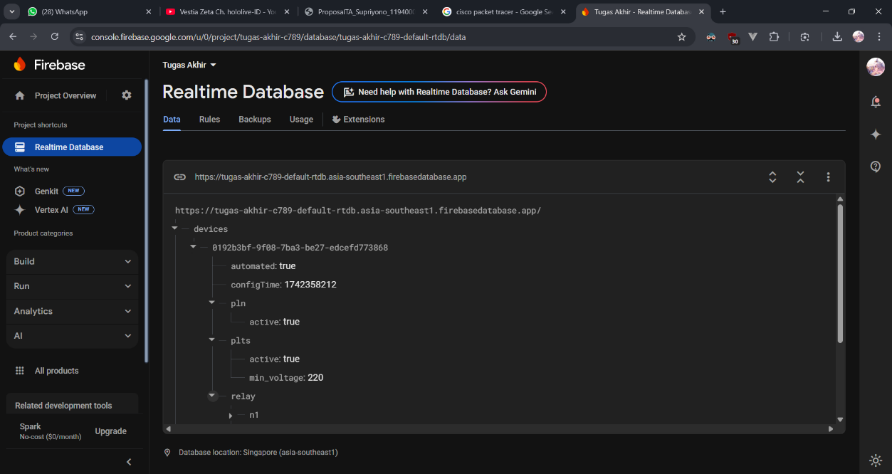
ADS1115 melakukan konversi dengan kecepatan hingga 860 SPS dengan PGA yang menawarkan rentang masukan dari ±256 mV hingga ±6,144V yang memungkinkan melakukan pengukuran sinyal besar maupun kecil secara presisi. ADS1115 juga dilengkapi dengan *input multiplexer* (MUX) yang memungkinkan dua pengukuran masukan diferensial atau empat pengukuran masukkan tunggal [30].



Gambar 2.19 ADS1115 ADC Converter [31]

## ***Firebase* *Realtime Database***

*Firebase* *Realtime Database* merupakan salah satu basis penyimpanan data *NoSQL* milik Google yang dapat menampilkan data kepada *client* secara *realtime*. *Database* ini menyimpan data dalam bentuk format JSON dan juga dapat melakukan sinkronisasi data setiap kali terdapat data yang berubah dan semua perangkat yang terhubung akan menerima perubahan tersebut dalam hitungan milidetik. Hal ini memungkinkan untuk membuat suatu aplikasi atau *website* *realtime* yang andal dan dapat melayani jutaan pengguna tanpa mengorbankan kemampuan *respons*. *Firebase* *realtime database* juga mendukung banyak platform seperti iOS, Android, Web, Flutter, Unity, C++, Admin, dan REST API [32].

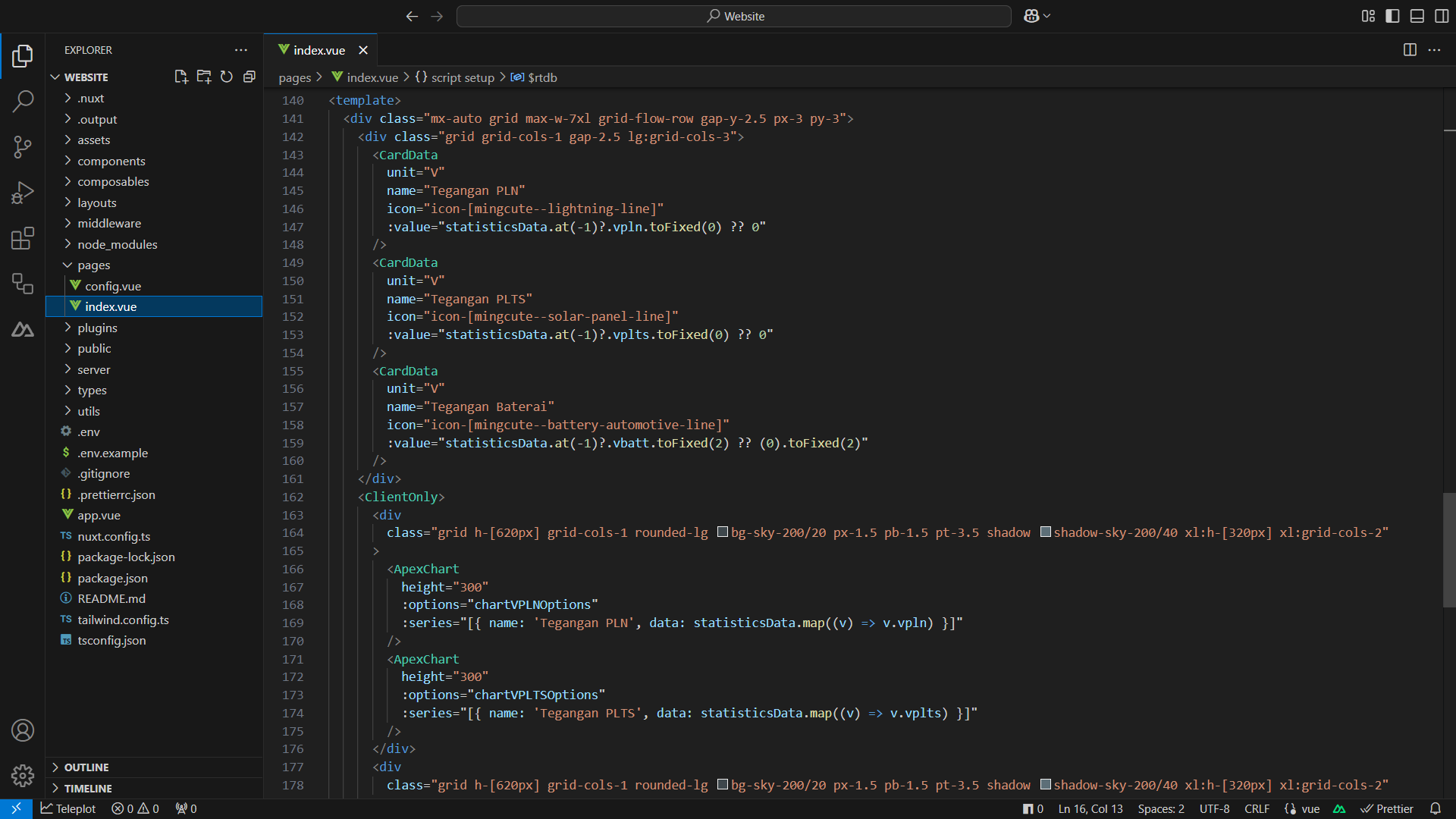


Gambar 2.20 Firebase Realtime Database [33]

## ***Framework Nuxt***

*Nuxt* adalah sebuah *framework open-source* untuk membangun sebuah halaman *website* yang dibangun dengan *Vue*. *Framework* ini juga menawarkan banyak fitur seperti *routing* untuk melakukan navigasi antar halaman, *layout* untuk mengatur tata letak agar terlihat menarik, *data fetching* yang berfungsi untuk mengambil data, *assets* dan *style* untuk menyimpan file grafis dan mempercantik tampilan website, *seo* dan *meta tags* yang berfungsi untuk menampilkan deskripsi singkat dari konten yang muncul di hasil pencarian mesin pencari, *middleware* untuk melakukan penyaringan permintaan, *state management* yang digunakan untuk mengelola, menyimpan, dan mengakses data secara terpusat. Dengan fitur-fitur yang telah disediakan dan diberikan tersebut, dapat memudahkan para *web developer* untuk membangun *website* yang *intuitive* dan *powerful* [34].

*Framework* ini menggunakan bahasa pemograman javascript dengan komponen vue yang menggunakan format berkas mirip seperti HTML yang dikenal sebagai SFC yang kemudian dikenal dengan berkas \*.vue. Dalam file ini dapat merangkum logika komponen (JavaScript), *template* (HTML), and *styles* (CSS) dalam satu file [35].



Gambar 2.21 Kode Halaman Monitoring

# **BAB III METODE PENELITIAN**

## **Metode Penelitian**

Adapun metode yang dilakukan oleh penulis untuk menyelesaikan penelitian ini dapat dilihat pada *flowchart* Gambar 3.1 dibawah ini.



Gambar 3.1 Flowchart Garis Besar Metode Penelitian

Bedasarkan pada Gambar 3.1 terdapat *flowchart* atau alur secara garis besar metode penelitian. Pada tahap pertama, penulis melakukan Studi Literatur untuk melakukan kajian teori yang mendukung topik pada penelitian ini. Kajian teori yang dijadikan acuan berasal dari buku, jurnal, dan referensi lain yang relevan dengan hal-hal yang berkaitan dengan penelitian yang akan dilakukan. Selanjutnya Perancangan Prototipe ATS dan Website, pada tahap ini penulis melakukan perancangan prototipe ATS bedasarkan referensi-referensi yang penulis dapatkan dan melakukan perancangan sebuah *website* dengan menggunakan *framework* nuxt dan firebase untuk dapat melakukan *monitoring* dan *controlling* pada prototipe ATS. Setelah perancangan prototipe ATS dan *Website* di lakukan, penulis melakukan Pengujian Prototipe ATS dan *Website* untuk melakukan *debug* dan *troubleshooting* pada prototipe ATS dan *website* untuk memastikan rancangan bekerja dengan baik dan meminimalisir kesalahan saat pengambilan data. Pada tahap ini sumber listrik PLTS akan digantikan sementara menggunakan sumber listrik PLN. Setelah rancangan bekerja dengan baik, penulis melakukan perancangan PLTS dan pengujian PLTS yang akan digunakan pada prototipe ATS sebagai sumber listrik ke dua. Jika hasil perancangan PLTS dapat bekerja dengan baik, penulis akan mengimplementasikan PLTS ke dalam prototipe ATS dan langkah terakhir penulis akan melakukan Pengambilan Data yang dilakukan secara *real-time* dan melakukan Analisis Hasil Data bedasarkan hasil pemantauan yang dilakukan.

## **Perancangan Prototipe ATS**

### **Alat dan Bahan**

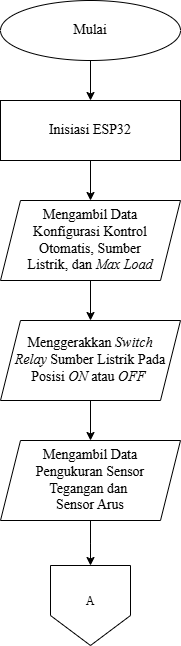
Berikut adalah alat dan bahan yang digunakan dalam perancangan prototipe ATS pada penelitian ini dapat dilihat pada Table 3.1 dibawah ini:

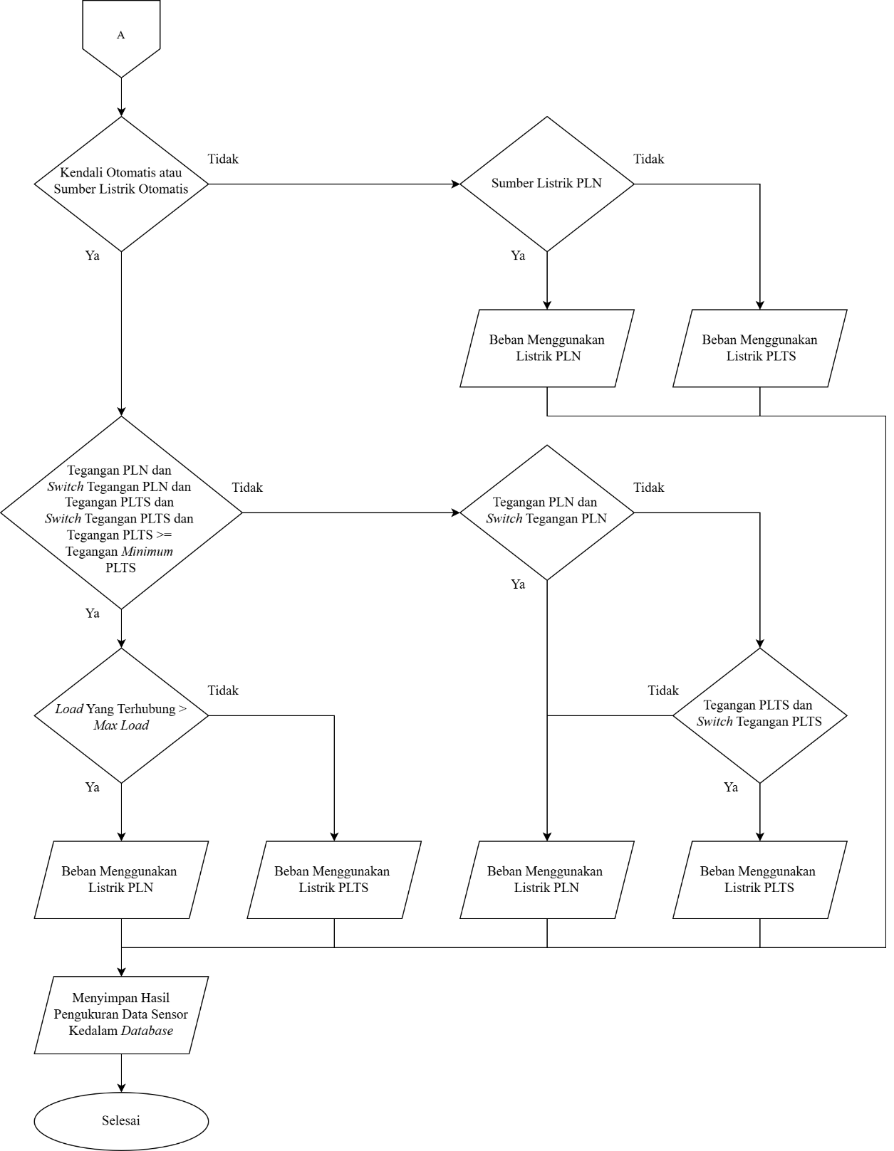
Tabel 3.1 Kebutuhan Alat dan Bahan Prototipe ATS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Kebutuhan Alat** | **Spesifikasi** | **Penggunaan Daya** | **Jumlah** |
| 1 | Mikrokontroller ESP32 | - | 40mA | 1 unit |
| 2 | Sensor Tegangan AC ZMPT101B | 0~1000VAC | 2mA | 2 unit |
| 3 | Sensor Arus ACS712 | 30A | 10mA | 3 unit |
| 4 | Sensor Tegangan DC | 0~25VDC | 1mA | 1 unit |
| 5 | ADC ADS1115 | 16-bit | 2mA | 2 unit |
| 7 | Relai 5V SPDT | 10A | 35mA | 2 unit |
| 8 | Relai 12V DPDT | 5A | 100mA | 3 unit |
| 9 | MCB 2A | - | - | 1 unit |
| 10 | MCB 1A | - | - | 1 unit |
| Total Penggunaan Daya | | | 447mA | |

### ***Flowchart* Sistem ATS**

*Flowchart* sistem merupakan tahapan proses kerja rangkaian yang akan dibuat. Proses kerja dari sistem ATS dapat dilihat pada *flowchart* Gambar 3.2 dibawah ini.





Gambar 3.2 Flowchart Sistem ATS

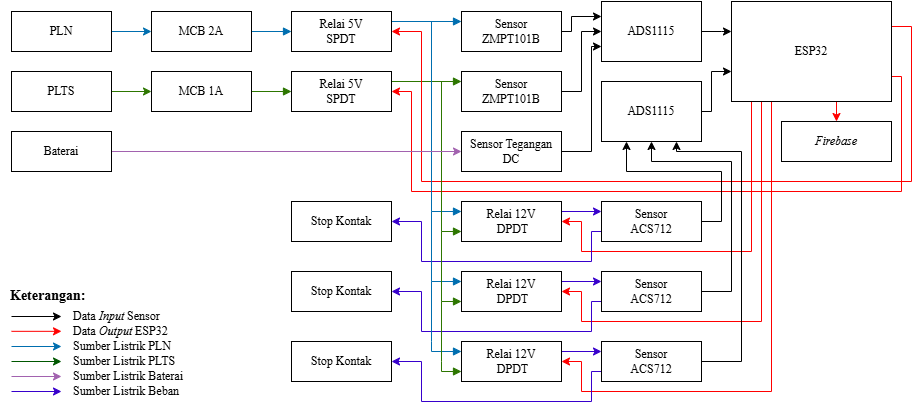
Berdasarkan *flowchart* pada Gambar 3.2 diatas terdapat alur kerja sistem prototipe ATS yang akan dirancang. Pertama mikrokontroller ESP32 akan menginisiasi program yang dibutuhkan. Selanjutnya, ESP32 akan mengambil data konfigurasi yang tersimpan di dalam *database* seperti kendali otomatis, *on/off* sumber listrik PLN, *on/off* sumber listrik PLTS, sumber listrik yang digunakan pada masing-masing relai, dan maksimal beban *load* pada masing-masing relai. Setelah mendapatkan data konfigurasi, data ini akan di simpan menggunakan *preferences* agar data konfigurasi tidak hilang jika perangkat kehilangan sumber listrik atau tidak dapat terhubung ke dalam jaringan. Selanjutnya perangkat akan menggerakkan relai *on/off* pada sumber listrik bedasarkan konfigurasi yang telah ditetepkan. Selanjutnya perangkat akan mengukur tegangan listrik PLN dan PLTS melalui sensor ZMPT101B, mengukur tegangan baterai yang digunakan pada PLTS, dan arus yang mengalir menggunakan sensor ACS712. Data pengukuran tersebut akan diproses oleh mikrokontroller untuk menggerakan relai ke sumber listrik yang akan digunakan.

Sistem yang dirancang memiliki dua kendali, yaitu kendali otomatis dan kendali sumber listrik. Pada sistem kendali otomatis, perangkat akan secara otomatis memindahkan sumber listrik yang digunakan berdasarkan konfigurasi maksimal beban *load* yang telah di tetapkan dan ketersediaan sumber listrik. Sedangkan pada sistem kendali sumber listrik, terdapat tiga konfigurasi yaitu sumber listrik otomatis, sumber listrik PLN, dan sumber listrik PLTS. Jika sumber listrik otomatis digunakan, maka perangkat akan bekerja seperti sistem kendali otomatis. Jika sumber listrik PLN digunakan maka perangkat akan menghubungkan beban ke sumber listrik PLN. Jika sumber listrik PLTS digunakan dan beban yang terhubung tidak melebihi dari konfigurasi maksimal *load*, maka perangkat akan menghubungkan beban ke sumber listrik PLTS. Dan jika sumber listrik PLTS digunakan dan beban yang terhubung melebihi dari konfigurasi maksimal *load*, maka perangkat akan mengubungkan beban ke sumber listrik PLN.

Setelah perangkat menggerakkan posisi relai ke sumber listrik yang ditentukan berdasarkan beban yang terhubung, perangkat akan mengirim data hasil pengukuran kedalam *database* firebase untuk disimpan dan ditampikan pada *website* yang telah dirancang untuk melakukan *monitoring* pada prototipe ATS.

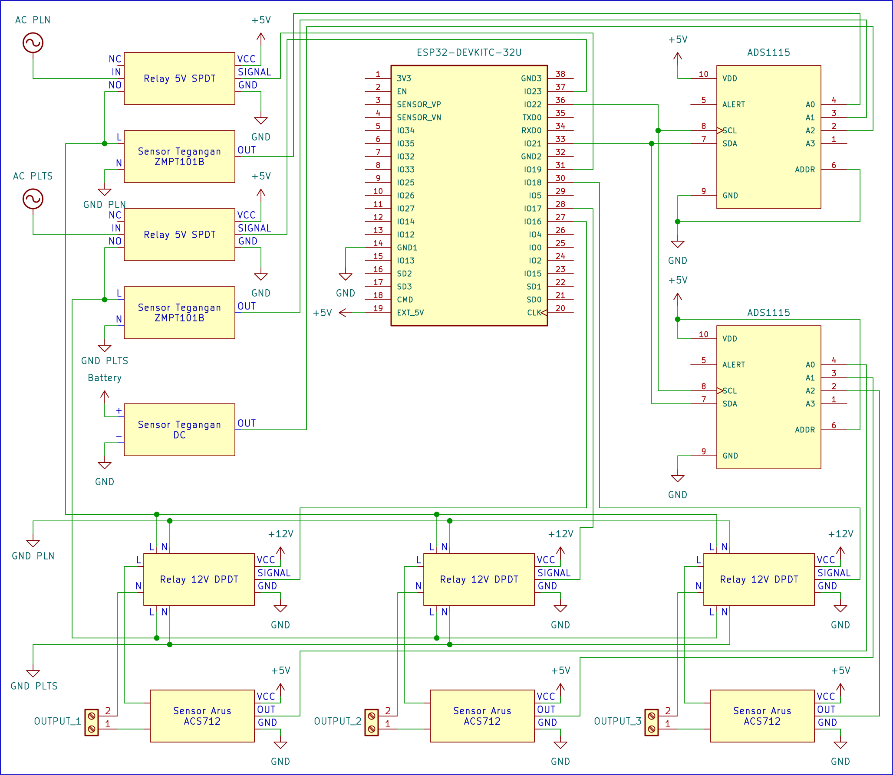
### **Rangkaian Prototipe ATS**

Perancangan prototipe ATS yang akan digunakan pada penelitian ini dapat dilihat pada *diagram block* Gambar 3.3 dibawah ini.



Gambar 3.3 Diagram Block Prototipe ATS

Pada rancangan prototipe yang ditunjukkan oleh *diagram block* pada Gambar 3.3 diatas, Secara garis besar rangkaian bekerja dengan cara membaca tegangan yang berasal dari PLN dan PLTS yang masuk kedalam rangkaian relai. Tujuan adanya rangkaian relai ini untuk digunakan sebagai *switch* *on*/*off* sumber listrik PLN dan PLTS. Kemudian *output* dari relai akan masuk ke dalam sensor tegangan ZMPT101B untuk dilakukan pembacaan tegangan listrik. *Output* tegangan yang berasal dari pembacaan sensor tegangan ZMPT101B ini kemudian akan masuk ke dalam ADC ADS1115 yang terhubung pada mikrokontroler ESP32 agar mendapatkan hasil yang akurat. Setelah melakukan pembacaan tegangan, rangkaian akan meneruskan sumber listrik PLN dan PLTS ke dalam relai yang kemudian *output* relai pada kabel fasa akan dimasukkan ke dalam sensor arus ACS712 untuk dilakukan pengukuran arus yang terhubung pada beban dan menentukan sumber daya listrik yang akan digunakan. Hasil pembacaan sensor arus ACS712 juga akan masuk ke dalam ADC ADS1115 untuk mendapatkan hasil yang akurat. Setelah melakukan pembacaan data yang berasal dari sensor dan mengatur posisi relai ke sumber listrik yang ditetapkan, data hasil pengukuran akan dikirim kedalam *database* Firebaseyang kemudian akan di tampikan melalui *website* yang telah dirancang. Dari hasil perancangan diatas di dapatkanlah skematik rangkaian menggunakan *software* KiCad pada Gambar 3.4 dibawah ini.



Gambar 3.4 Skematik Rangkaian Prototipe ATS Menggunakan KiCad

Pada Gambar 3.4 merupakan skematik rangkaian prototipe ATS yang telah dirancang bedasarkan *diagram block* pada Gambar 3.3 menggunakan *software* KiCad. Skematik rangkaian diatas berfungsi untuk mengetahui gambaran keseluruhan hubungan komponen yang akan digunakan.

### **Rangkaian *Input* Sistem ATS**

Rangkaian *input* dari sistem ATS yang digunakan pada penelitian ini yang dapat dilihat pada Gambar 3.5 dibawah ini.

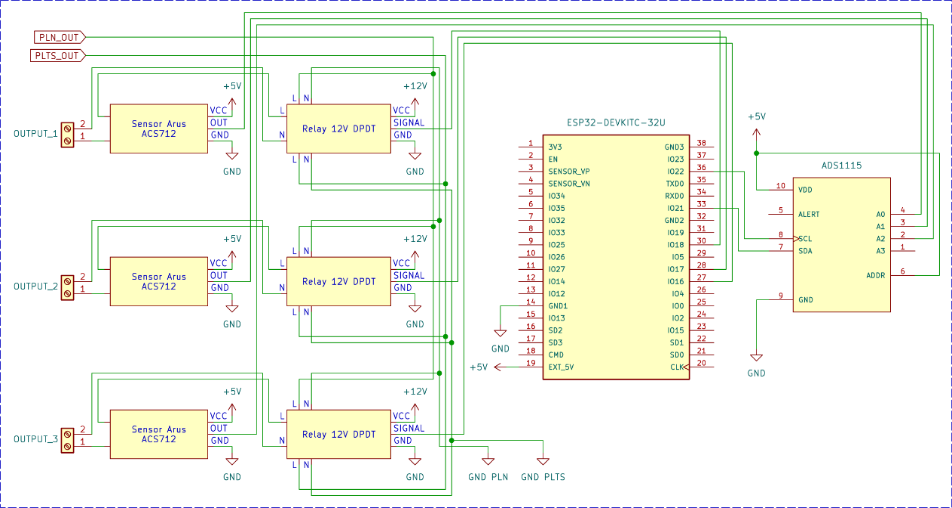


Gambar 3.5 Rangkaian Input Sistem ATS Menggunakan KiCad

Pada rangkaian *input* diatas menggunakan sensor tegangan ZMPT101B, sensor tegangan DC, ADC ADS1115, dan relai 5V SPDT. Komponen ini digunakan untuk melakukan *switch* *on/off* sumber listrik, melakukan pembacaan data tegangan PLN, tegangan PLTS, dan tegangan baterai yang kemudian data hasil pengukuran akan di proses oleh mikrokontroller ESP32. Komponen relai 5V SPDT yang berjumlah dua masing-masing akan terhubung pada *pin* 21 dan *pin* 19 ESP32. Sedangkan pada komponen sensor tegangan ZMPT101B dan sensor tegangan DC akan terhubung ke ADC ADS1115 yang masing-masing terhubung pada *pin* A0 dan *pin* A1 untuk sensor tegangan ZMPT101B dan *pin* A2 untuk sensor tegangan DC. *Output* dari ADC ADS1115 ini terhubung ke mikrokontroller ESP32 melalui *pin* SDA yang terhubung pada *pin* 23 dan *pin* SCL yang terhubung pada *pin* 22. Rangkaian ini membutuhkan input tegangan listrik PLN dan PLTS, tegangan DC 5V, dan tegangan baterai. Seluruh komponen sensor memerlukan tegangan DC 5V sebagai tegangan kerja, yang dimana seluruh *pin* VCC dan *pin* GND pada komponen sensor dihubungkan pada *pin* Vin dan *pin* GND.

### **Rangkaian *Output* Sistem ATS**

Rangkaian *output* dari sistem ATS yang digunakan pada penelitian ini yang dapat dilihat pada Gambar 3.6 dibawah ini.

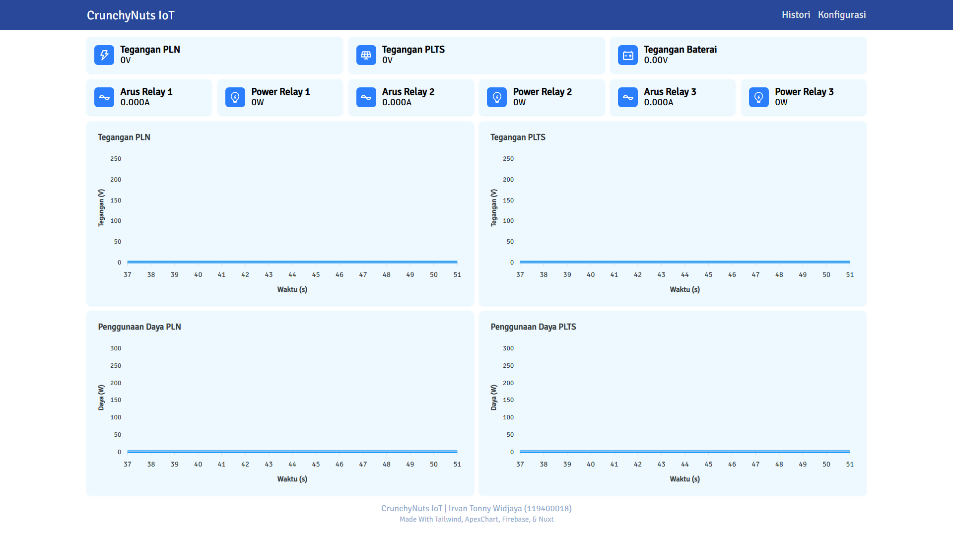


Gambar 3.6 Rangkaian Output Sistem ATS Menggunakan KiCad

Pada rangkaian *output* diatas menggunakan sensor arus ACS712, ADC ADS1115, dan relai 12V DPDT. Komponen ini digunakan untuk menghubungkan beban ke sumber listrik yang telah ditentukan oleh mikrokontroller ESP32. Komponen relai 12V DPDT yang berjumlah tiga masing-masing akan terhubung pada *pin* 18, *pin* 17, dan *pin* 16 ESP32. Sedangkan pada komponen sensor arus ADS1115 masing-masing akan terhubung ke ADC ADS1115 pada *pin* A0, *pin* A1, dan pin A2. *Output* dari ADC ADS1115 ini terhubung ke mikrokontroller ESP32 melalui *pin* SDA yang terhubung pada *pin* 23 dan *pin* SCL yang terhubung pada *pin* 22. Rangkaian ini membutuhkan *input* tegangan PLN dan PLTS pada relai, tegangan DC 5V, dan tegangan DC 12V. Komponen sensor ACS712 dan ADC ADS1115 memerlukan tegangan DC 5V sebagai tegangan kerja, sedangkan komponen relai memerlukan tegangan DC 12V sebagai tegangan kerja.

## **Perancangan Sistem *Monitoring* dan *Controlling***

Dalam melakukan *monitoring* dan *controlling* pada penelitian ini, penulis membangun sebuah halaman *website* dengan menggunakan *framework* *nuxt* dan *firebase* untuk menyimpan data yang dikirim dari sensor. Penulis membangun sendiri halaman *website* ini agar dapat mudah di kustomisasi sesuai dengan kebutuhan yang diperlukan. Hasil rancangan tampilan dapat dilihat pada Gambar 3.7 dibawah ini.



Gambar 3.7 Tampilan Halaman Sistem Monitoring

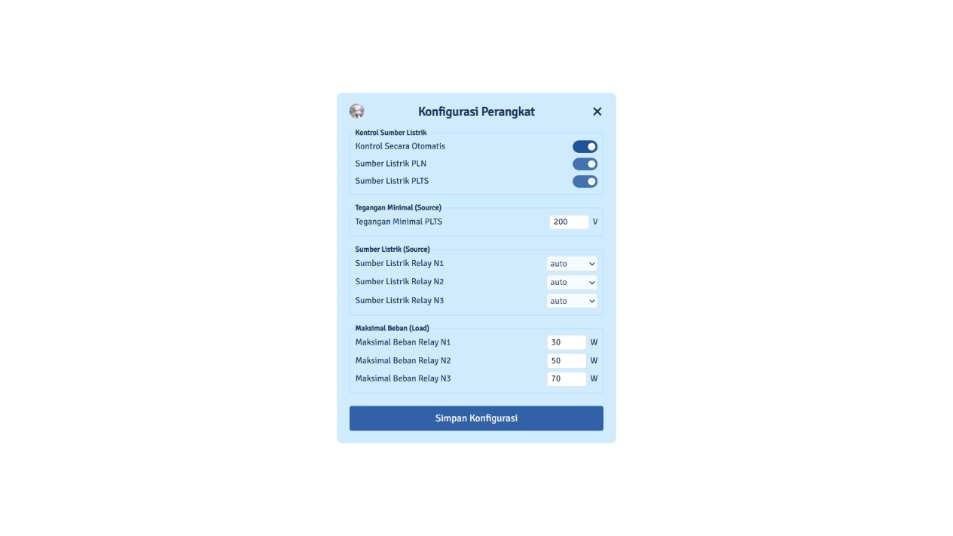
Pada Gambar 3.7 diatas merupakan tampilan sistem *monitoring* menggunakan *framework* *nuxt* yang terdapat bagian untuk menampilkan hasil pembacaan sensor tegangan listrik pada PLN, PLTS, dan beterai, arus yang mengalir dan penggunaan daya (*power*) pada masing-masing relai, grafik tegangan listrik PLN dan PLTS, dan grafik penggunaan daya listrik PLN dan PLTS yang akan ditampilkan dalam rentang waktu lima belas detik terakhir secara *real time* menggunakan konektifitas internet. Pada sistem monitoring juga terdapat fitur histori untuk melihat hasil pengukuran sensor dan penggunaan daya. Pada halaman ini terdapat *input* tanggal untuk dapat menampilkan grafik pada rentang waktu yang di inginkan dan grafik pengukuran sensor tegangan listrik PLN, PLTS, dan baterai serta grafik penggunaan daya PLN dan PLTS. Tampilan halaman histori hasil pengukuran sensor dan penggunaan daya dapat dilihat pada Gambar 3.8 dibawah ini.



Gambar 3.8 Tampilan Halaman Histori

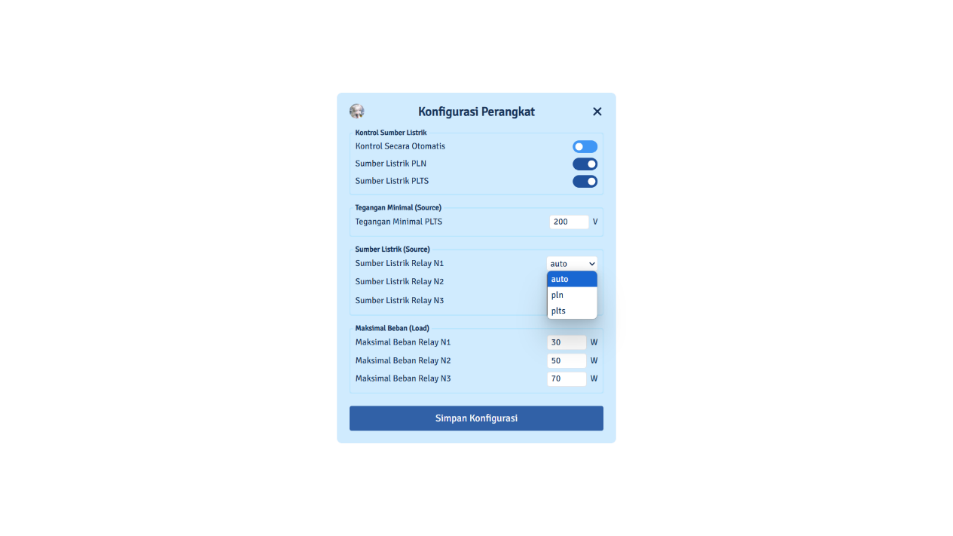
Selain dapat melakukan *monitoring*, halaman *website* ini juga dapat mengatur kontrol otomatis, kontrol sumber listrik, pengaturan sumber listrik yang digunakan pada masing-masing relai, dan maksimal beban pada masing-masing relai seperti yang

ditunjukkan pada Gambar 3.9 dibawah ini.



Gambar 3.9 Tampilan Halaman Sistem Controlling Kontrol Otomatis On

Pada Gambar 3.9 diatas merupakan tampilan halaman *controlling*, pada halaman tersebut terdapat *switch* untuk melakukan kontrol secara otomatis dan *switch* untuk melakukan *on/off* pada sumber listrik. Saat kontrol otomatis diaktifkan, maka tidak dapat melakukan perubahan pada masing-masing relai. Untuk dapat mengubah sumber listrik yang digunakan sesuai keinginan, perlu menonaktifkan kontrol otomatis terlebih dahulu agar dapat mengubah sumber listrik pada relai tertentu seperti yang ditunjukkan pada Gambar 3.10 dibawah ini.



Gambar 3.10 Tampilan Halaman Sistem Controlling Kontrol Otomatis Off

Pada Gambar 3.10 diatas terdapat opsi untuk mengubah sumber listrik yang terhubung ke beban sesuai keinginan. Saat opsi *auto* dipilih, maka prototipe akan menggerakan relai ke sumber listrik yang akan digunakan bedasarkan daya beban yang terhubung. Sedangkan jika opsi PLN atau PLTS dipilih, maka prototipe akan menggerakan relai sesuai konfigurasi tersebut. Untuk membatasi maksimal beban yang dapat terhubung ke sumber listrik PLTS, pada masing-masing relai terdapat konfigurasi maksimal beban yang dapat terhubung ke sumber listrik PLTS. Berikut adalah tabel konfigurasi maksimal beban pada masing-masing relai yang akan digunakan pada pengujian.

Tabel 3.2 Konfigurasi Maksimal Beban Pada Relai

|  |  |  |
| --- | --- | --- |
| **No** | **Relai** | **Daya (W)** |
| 1 | N1 | 30 |
| 2 | N2 | 50 |
| 3 | N3 | 70 |

Pada Tabel 3.2 diatas menunjukkan konfigurasi maksimal beban pada masing-masing relai dengan total 150W. Pembatasan beban ini dilakukan agar prototipe bekerja sesuai *flowchart* sistem yang telah dibuat.

## **Perancangan PLTS**

### **Kebutuhan Beban**

Agar prototipe dapat bekerja sesuai yang diinginkan, penentuan kebutuhan beban sangat penting untuk mengetahui spesifikasi minimum alat yang akan digunakan. Data-data yang digunakan berasal dari lampu pijar yang digunakan yang memiliki daya yang berbeda-beda. Berikut adalah tabel daya lampu pijar yang digunakan dalam pengujian.

Tabel 3.3 Beban Lampu Pijar

|  |  |  |
| --- | --- | --- |
| **No** | **Beban** | **Daya (W)** |
| 1 | Lampu Pijar | 10 |
| 25 |
| 40 |
| 60 |
| 1 | Lampu Pijar | 75 |
| 100 |

Tabel 3.3 diatas merupakan penggunaan beban yang akan digunakan untuk melakukan pengujian prototipe. Pengujian dilakukan untuk mengetahui apakah prototipe dapat mengubah sumber listrik yang terhubung dengan beban secara otomatis sesuai dengan konfigurasi yang telah ditetapkan.

### **Alat dan Bahan**

Berikut adalah alat dan bahan yang digunakan dalam perancangan PLTS pada penelitian ini dapat dilihat pada tabel 3.4 dibawah ini:

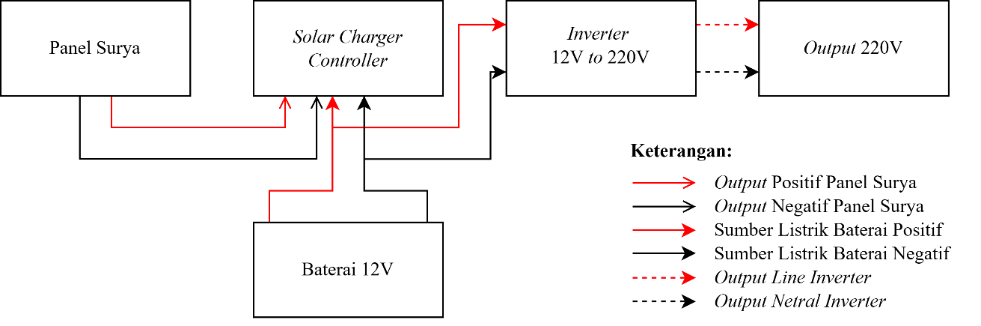
Tabel 3.4 Kebutuhan Alat dan Bahan Perancangan PLTS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Kebutuhan** | **Spesifikasi** | **Penggunaan Daya** | **Jumlah** |
| 1 | Panel Surya *Polycrystalline* | 10Wp | - | 1 unit |
| 2 | *Solar Charger Controller* (SCC) | 10A | 1mA | 1 unit |
| 3 | Baterai Accu | 12V 12Ah | - | 1 unit |
| 4 | Inverter DC 12V *to* AC 220V | 220W | 150mA | 1 unit |
| Total Penggunaan Daya | | | 151mA | |

Tabel 3.4 diatas menunjukkan kebutuhan alat dan bahan dalam merancang PLTS yang akan digunakan untuk pengujian prototipe. Kebutuhan tersebut juga sudah mencakup kebutuhan daya pada prototipe ATS dan PLTS sebesar ±7.2 Wh.

### **Rangkaian PLTS**

Agar menjadi suatu sistem, komponen tersebut perlu untuk dirangkai menjadi satu seperti ditunjukkan pada *diagram block* Gambar 3.11 dibawah ini.



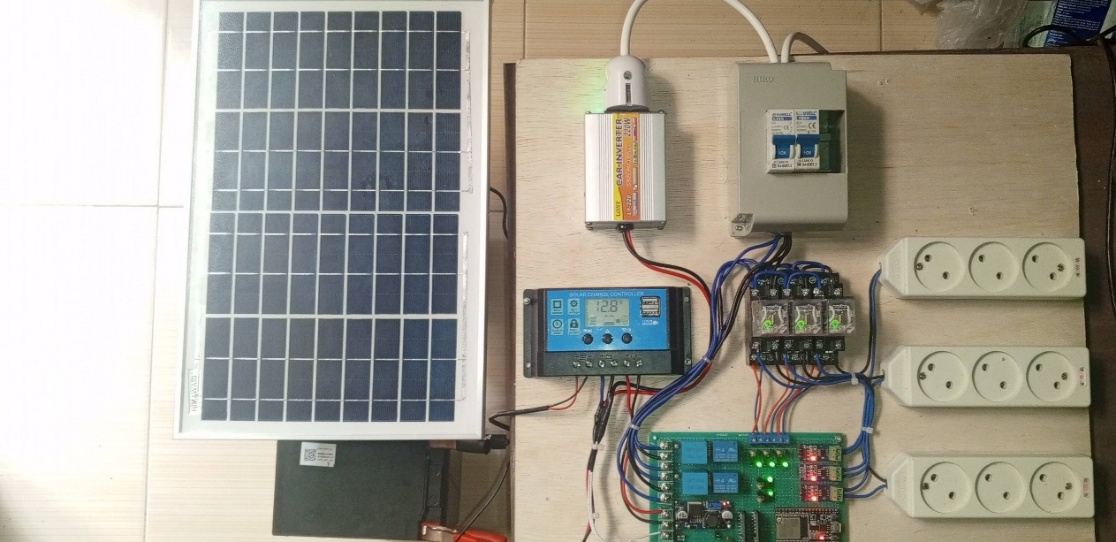
Gambar 3.11 Diagram Block Rangkaian PLTS Offgrid

Bedasarkan *diagaram block* rangkaian PLTS *offgrid* yang ditunjukkan pada Gambar 3.10 diatas, *Output* tegangan yang berasal dari panel surya dan baterai akan dihubungkan ke SCC. Kemudian *output* baterai juga akandihubungkan ke *inverter* agar dapat digunakan pada peralatan listrik yang membutuhkan sumber listrik AC.

# **BAB IV HASIL DAN PEMBAHASAN**

## **Hasil Perancangan Prototipe ATS**

Hasil perancangan prototipe ATS dan perancangan PLTS dapat dilihat pada Gambar 4.1 berikut.



Gambar 4.1 Hasil Perancangan Prototipe ATS

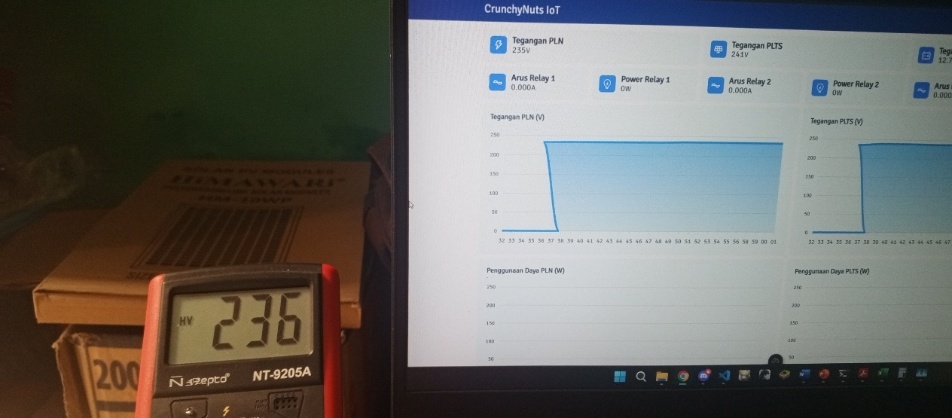
Berdasarkan Gambar 4.1 diatas, dapat terlihat seluruh komponen yang digunakan digabungkan menjadi satu-kesatuan. Prototipe ini terhubung dengan sumber listrik PLN dan PLTS untuk melakukan distribusi beban listrik yang telah terhubung dengan MCB untuk mencegah beban berlebih dan hubung singkat atau korsleting listrik dan prototipe ini memiliki *output* sebanyak tiga yang terhubung dengan sumber listrik PLTS. Agar rancangan ini dapat bekerja sesuai fungsinya untuk melakukan distribusi beban listrik antara sumber listrik PLN dan sumber listrik PLTS, setiap *output* memiliki konfigurasi maksimal beban yang berbeda-beda.

## **Pengujian Sensor Tegangan ZMPT101B**

Pengujian sensor tegangan ZMPT101B dilakukan untuk mengetahui sensitivitas sensor dan tingkat akurasi sensor ketika melakukan pembacaan tegangan dengan pembanding menggunakan alat ukur. Pada penelitian ini, penulis menggunakan sensor ZMPT101B sebanyak 2 buah untuk mengukur tegangan listrik AC pada sumber listrik PLN dan sumber listrik PLTS. Hasil dari pengujian pembacaan tegangan dari sensor ZMPT101B pertama yang terhubung ke sumber listrik PLN dapat dilihat berdasarkan grafik pada Gambar 4.2 berikut.

Gambar 4.2 Hasil Pengujian Sensor ZMPT101B I

Gambar 4.2 diatas merupakan pengujian pada sensor ZMPT101B pertama yang terhubung ke sumber listrik PLN dengan pembanding alat ukur dengan jumlah *sample* pengukuran sebanyak 10 didapat bahwa nilai rata-rata persentase *error* pada sensor ZMPT101B pertama yaitu sebesar 0,13%. Berikut adalah contoh gambar pengujian sensor ZMPT101B yang terhubung ke sumber listrik PLN.



Gambar 4.3 Contoh Pengujian Pengukuran Sensor ZMPT101B PLN

Hasil pengujian sensor ZMPT101B kedua yang terhubung ke sumber listrik PLTS dapat dilihat berdasarkan grafik pada Gambar 4.4 berikut.

Gambar 4.4 Hasil Pengujian Sensor ZMPT101B PLTS

Gambar 4.4 diatas merupakan pengujian pada sensor ZMPT101B yang kedua dengan pembanding alat ukur didapat bahwa nilai rata-rata persentase *error* pada sensor ZMPT101B kedua yaitu 1,92%. Berikut adalah contoh gambar pengujian sensor ZMPT101B yang terhubung ke sumber listrik PLTS.



Gambar 4.5 Contoh Pengujian Pengukuran Sensor ZMPT101B PLTS

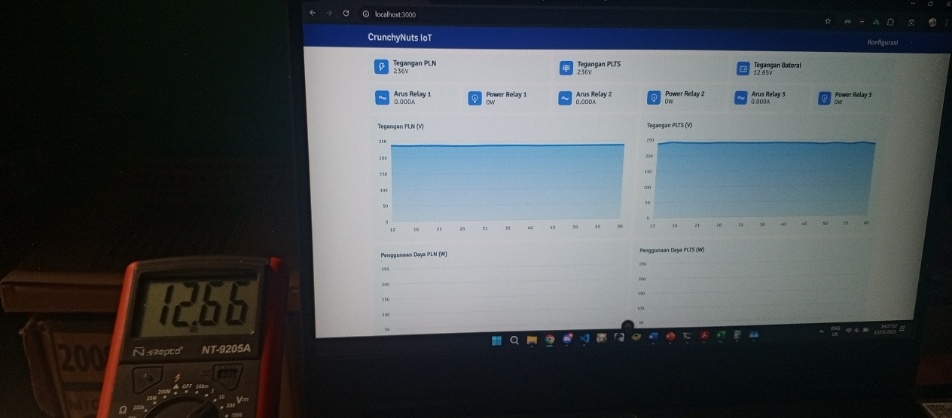
Berdasarkan hasil pengukuran kedua sensor ZMPT101B tersebut, didapat sensor tegangan ZMPT101B dapat disimpulkan kedua sensor tersebut memiliki tingkat akurasi yang baik dan presisi yang tinggi.

## **Pengujian Sensor Tegangan DC**

Pengujian sensor tegangan DC pada penelitian ini merupakan konfigurasi dari rangkaian pembagi tegangan. Pengujian ini dilakukan untuk mengetahui keakuratan nilai yang diperoleh dari pembacaan sensor dengan pembanding menggunakan alat ukur. Hasil dari pengujian pembacaan tegangan DC dari sensor yang terhubung ke sumber listrik baterai PLTS dapat dilihat berdasarkan grafik pada Gambar 4.6 berikut.

Gambar 4.6 Hasil Pengujian Sensor Tegangan DC

Gambar 4.6 diatas merupakan pengujian pada sensor ZMPT101B yang kedua dengan pembanding alat ukur dengan jumlah *sample* pengukuran sebanyak 10 didapat bahwa rata-rata persentase *error* pada sensor tegangan DC adalah 0,08%. Berikut adalah contoh gambar pengujian tegangan baterai yang dilakukan.



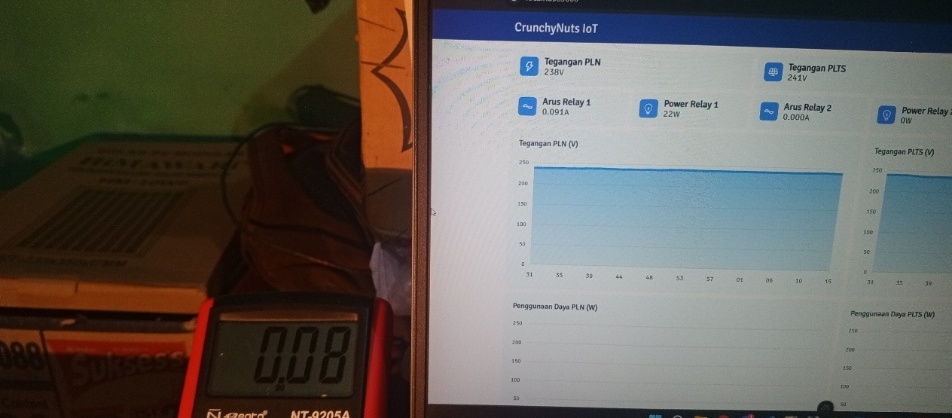
Gambar 4.7 Contoh Pengujian Pengukuran Sensor Tegangan DC

## **Pengujian Sensor Arus ACS712**

Pengujian sensor arus ACS712dilakukan untuk mengetahui sensitivitas sensor dan tingkat akurasi sensor ketika melakukan pembacaan arus dengan pembanding menggunakan alat ukur. Pada penelitian ini, penulis menggunakan sensor arus ACS712 sebanyak 3 buah yang memiliki batas pembacaan arus 30A. Pengujian dilakukan saat beban terhubung dengan sumber listrik PLN dan sumber listrik PLTS. Hasil dari pengujian pembacaan arus dari sensor pertama saat beban terhubung ke masing-masing sumber listrik dapat dilihat berdasarkan grafik pada Gambar 4.8 berikut.

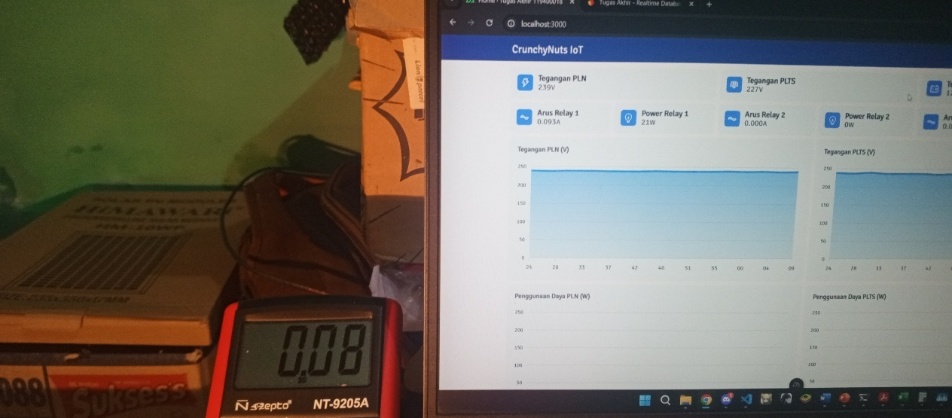
Gambar 4.8 Hasil Pengujian Sensor Arus ACS712 I Terhadap Beban

Gambar 4.8 diatas merupakan hasil pengujian pada sensor ACS712 yang pertama dengan pembanding alat ukur didapat bahwa nilai rata-rata persentase *error* pada sensor ACS712 pertama saat terhubung ke sumber listrik PLN adalah 6,58% dan saat terhubung ke sumber listrik PLTS adalah 10,04%. Berikut adalah contoh gambar pengujian sensor pertama yang dilakukan saat beban lampu 10W terhubung ke sumber listrik PLN.



Gambar 4.9 Contoh Pengujian Sensor ACS712 I PLN

Berikut adalah beberapa contoh pengujian sensor pertama yang dilakukan saat beban lampu 10W terhubung ke sumber listrik PLTS.

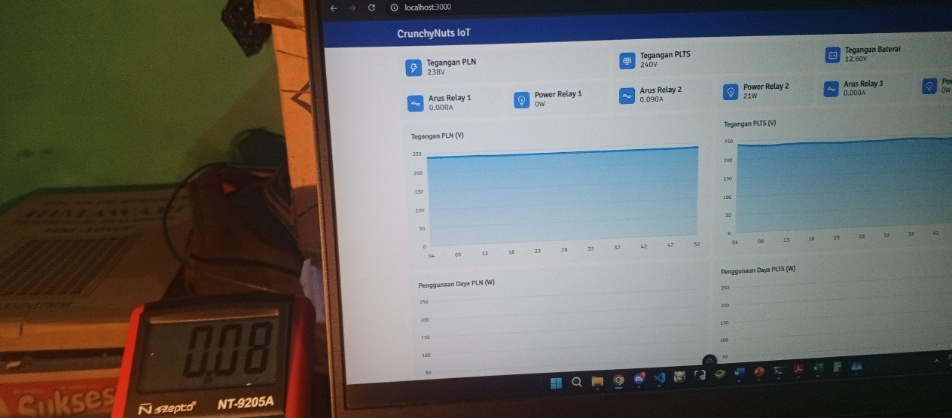


Gambar 4.10 Contoh Pengujian Sensor ACS712 I PLTS

Hasil dari pengujian pembacaan arus dari sensor kedua saat beban terhubung ke masing-masing sumber listrik dapat dilihat berdasarkan grafik pada Gambar 4.11 berikut.

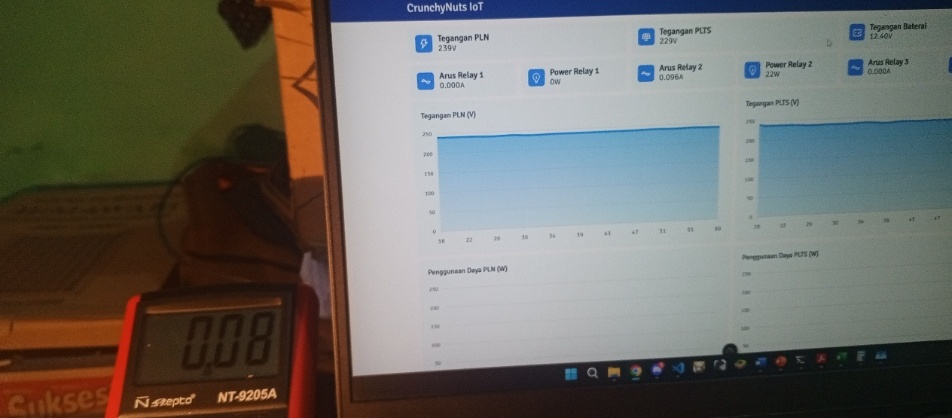
Gambar 4.11 Hasil Pengujian Sensor Arus ACS712 II Terhadap Beban

Gambar 4.11 diatas merupakan hasil pengujian pada sensor ACS712 yang kedua dengan pembanding alat ukur didapat bahwa nilai rata-rata persentase *error* pada sensor ACS712 kedua saat terhubung ke sumber listrik PLN adalah 5,57% dan saat terhubung ke sumber listrik PLTS adalah 9,20%. Berikut adalah contoh gambar pengujian sensor kedua yang dilakukan saat beban lampu 10W terhubung ke sumber listrik PLN.



Gambar 4.12 Contoh Pengujian Sensor ACS712 II PLN

Berikut adalah contoh gambar pengujian sensor kedua yang dilakukan saat beban lampu 10W terhubung ke sumber listrik PLTS.

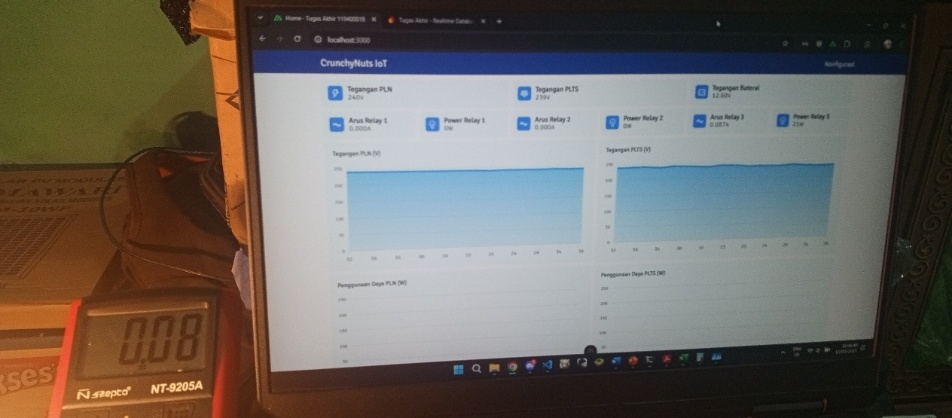


Gambar 4.13 Contoh Pengujian Sensor ACS712 II PLTS

Hasil dari pengujian pembacaan arus dari sensor pertama saat beban terhubung ke masing-masing sumber listrik dapat dilihat berdasarkan grafik pada Gambar 4.7 dibawah ini.

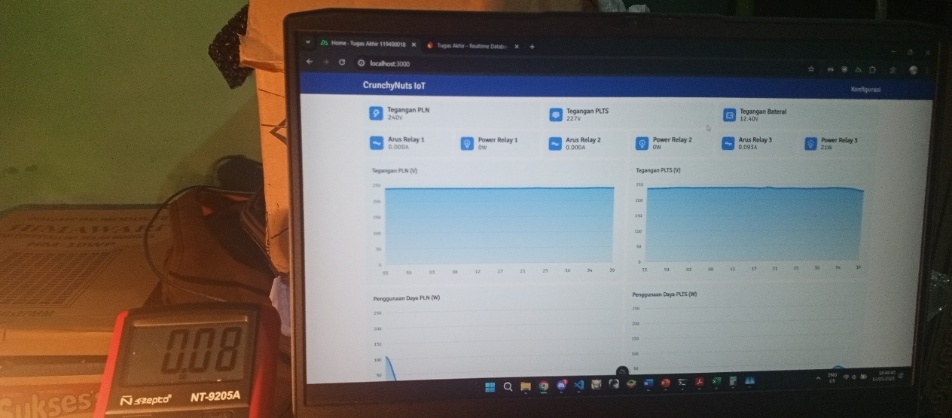
Gambar 4.14 Hasil Pengujian Sensor Arus ACS712 III Terhadap Beban

Pada Gambar 4.14 merupakan hasil pengujian pada sensor ACS712 yang ketiga dengan pembanding alat ukur didapat bahwa nilai rata-rata persentase *error* pada sensor ACS712 ketiga saat terhubung ke sumber listrik PLN adalah 6,50% dan saat terhubung ke sumber listrik PLTS adalah 7,43%. Berikut adalah contoh gambar pengujian sensor ketiga yang dilakukan saat beban lampu 10W terhubung ke sumber listrik PLN.



Gambar 4.15 Contoh Pengujian Sensor ACS712 III PLN

Berikut adalah contoh gambar pengujian sensor ketiga yang dilakukan saat beban lampu 10W terhubung ke sumber listrik PLTS.

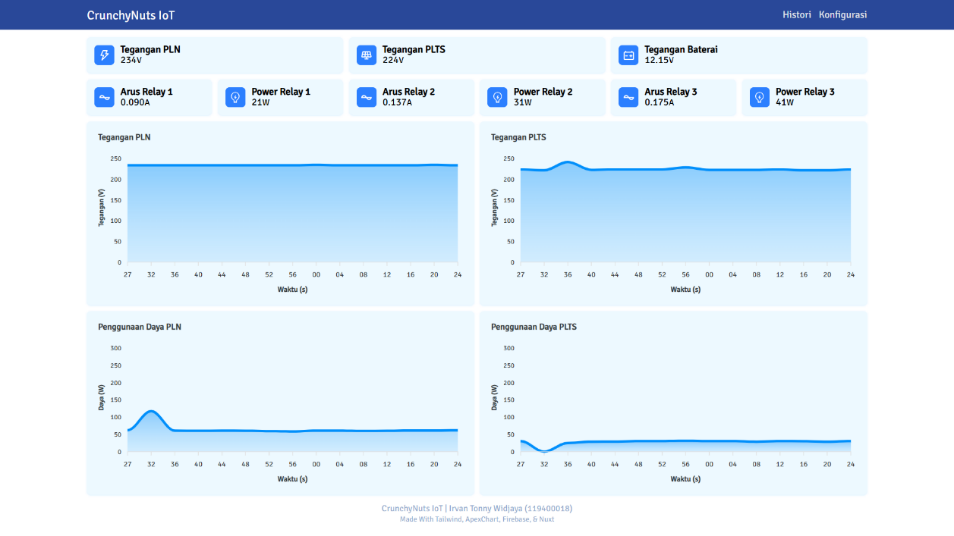


Gambar 4.16 Contoh Pengujian Sensor ACS712 III PLTS

Berdasarkan hasil pengukuran ketiga sensor tersebut, didapat ketiga sensor arus ACS712 yang digunakan memiliki persentase *error* yang berkisar dari 5% hingga 10%. Dari hasil tersebut ketiga sensor tersebut memiliki tingkat akurasi yang baik dan presisi yang tinggi.

## **Pengujian Sistem *Monitoring* dan *Controlling***

Pengujian sistem *monitoring* dan *controlling* berbasis *website* yang dirancang dengan menggunakan *framework nuxt* dan *database Firebase Realtime Database* bertujuan untuk mengetahui sistem yang telah dirancangdapat bekerja dengan baik. Hasil pengujian sistem *monitoring* dapat dilihat pada Gambar 4.17 berikut.

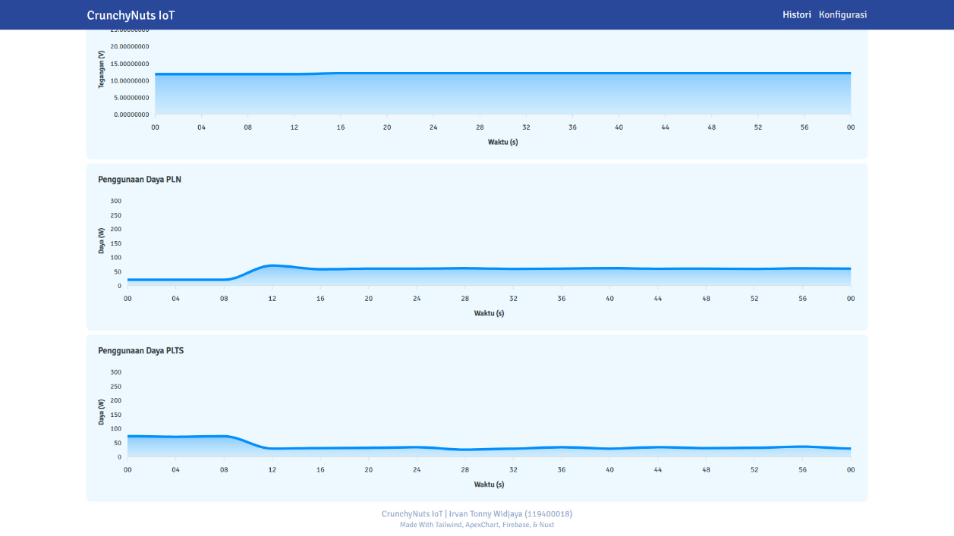


Gambar 4.17 Hasil Pengujian Sistem Monitoring

Pada Gambar 4.17 merupakan hasil dari pengujian sistem *monitoring* dan dapat terlihat bahwa sistem *monitoring* yang dapat berubah setiap terdapat perubahan data pada *database* *Firebase* secara *real time*. Pengujian juga dilakukan pada halaman histori yang dapat di lihat pada Gambar 4.18 dan Gambar 4.19 berikut.



Gambar 4.18 Hasil Pengujian Halaman Histori (A)



Gambar 4.19 Hasil Pengujian Halaman Histori (B)

Pada pengujian sistem *controlling*, dilakukan dengan menguji fungsi *switch* kontrol otomatis dan sumber listrik. Hasil pengujian dapat dilihat pada tabel 4.1.

Tabel 4.1 Hasil Pengujian Switch On/Off Sumber Listrik PLN dan PLTS

|  |  |  |  |
| --- | --- | --- | --- |
| Relai | Kondisi Awal Relai | *Switch* pada *Website* | Kondisi Akhir Relai |
| Sumber Listrik PLN | *HIGH* | Kontrol Otomatis *ON* | *LOW* |
| Kontrol Otomatis *OFF* | *LOW* |
| Sumber Listrik *ON* | *LOW* |
| Sumber Listrik *OFF* | *HIGH* |
| Sumber Listrik PLTS | Kontrol Otomatis *ON* | *LOW* |
| Kontrol Otomatis *OFF* | *LOW* |
| Sumber Listrik *ON* | *LOW* |
| Sumber Listrik *OFF* | *HIGH* |

Bedasarkan hasil pengujian pada tabel 4.1 diatas, karena menggunakan *trigger LOW* untuk menyalakan relai sehingga kondisi awal relai sumber listrik PLN dan sumber listrik PLTS berada diposisi *HIGH*. Pada saat prototipe dinyalakan dan terhubung ke internet, prototipe akan mengambil data dari *database* untuk mendapatkan kondisi *switch*. Jika prototipe tidak terhubung atau gagal terhubung ke internet, perangkat akan membaca konfigurasi yang disimpan pada mikrokontroller. Jika kondisi *switch* kontrol otomatis aktif, maka relai sumber listrik PLN dan sumber listrik PLTS akan berada diposisi *LOW*. Jika *switch* kontrol otomatis di matikan dan *switch* sumber listrik PLN atau PLTS aktif, maka relai akan berada diposisi *LOW*. Begitupun sebaliknya, Jika *switch* kontrol otomatis di matikan dan *switch* listrik PLN atau PLTS tidak aktif, maka relai akan berada diposisi *HIGH*. Jika prototipe tidak terhubung ke internet, maka prototipe akan menggerakkan posisi relai sumber listrik PLN dan sumber listrik PLTS ke posisi *LOW.*

Setelah melakukan pengujian kontrol otomatis dan sumber listrik, pengujian dilanjutkan dengan menguji kendali relai terhadap beban listrik. Pengujian ini bertujuan untuk memastikan prototipe dapat bekerja sesuai dengan *flowchart* yang telah dirancang. Pengujian dilakukan dengan konfigurasi tegangan minimal sumber listrik PLTS sebesar 200V. Hasil dari pengujian dapat dilihat pada tabel 4.2 dibawah ini.

Tabel 4.2 Hasil Pengujian Kontrol Otomatis

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Relai | Kondisi Awal Relai | Kontrol Otomatis | Konfigurasi Sumber Listrik | *Switch* Sumber Listrik PLN | *Switch* Sumber Listrik PLTS | Beban Yang Terhubung | Tegangan PLTS | Kondisi Akhir Relai | Sumber Listrik |
| Sumber Listrik Relai N1 | *LOW* | *ON / OFF* | *Auto* | *ON* | *ON* | <= 30W | Lebih Dari Minimal | *LOW* | PLTS |
| Kurang Dari Minimal | *HIGH* | PLN |
| > 30W | Lebih atau Kurang Dari Minimal | *HIGH* | PLN |
| *OFF* | *Auto* | *ON* | *OFF* | - | | *HIGH* | PLN |
| *OFF* | *ON* | - | | *LOW* | PLTS |
| PLN | - | | | | *HIGH* | PLN |
| PLTS | - | | | | *LOW* | PLTS |
| Sumber Listrik Relai N2 | *ON / OFF* | *Auto* | *ON* | *ON* | <= 50W | Lebih Dari Minimal | *LOW* | PLTS |
| Kurang Dari Minimal | *HIGH* | PLN |
| > 50W | Lebih atau Kurang Dari Minimal | *HIGH* | PLN |
| *OFF* | *Auto* | *ON* | *OFF* | - | | *HIGH* | PLN |
| *OFF* | *ON* | - | | *LOW* | PLTS |
| PLN | - | | | | *HIGH* | PLN |
| PLTS | - | | | | *LOW* | PLTS |
| Sumber Listrk Relai N3 | *ON / OFF* | *Auto* | *ON* | *ON* | <= 70W | Lebih Dari Minimal | *LOW* | PLTS |
| Kurang Dari Minimal | *HIGH* | PLN |
| > 70W | Lebih atau Kurang Dari Minimal | *HIGH* | PLN |
| *OFF* | *Auto* | *ON* | *OFF* | - | | *HIGH* | PLN |
| *OFF* | *ON* | - | | *LOW* | PLTS |
| PLN | - | | | | *HIGH* | PLN |
| PLTS | - | | | | *LOW* | PLTS |

Bedasarkan tabel 4.2 diatas, dapat terlihat bahwa saat kontroller aktif, kontroller akan menggerakan relai ke kondisi *LOW* untuk menghubungkan beban ke sumber listrik PLTS. Kemudian kontroller akan mengambil data konfigurasi, Apabila konfigurasi kontrol otomatis *on/off*, sumber listrik *auto*, *switch* sumber listrik PLN dan PLTS *on*, kontroller akan menggerakkan relai ke sumber listrik sesuai dengan batasan konfigurasi maksimal beban dan hasil pembacaan tegangan PLTS. Sehingga didapatkan 3 kondisi seperti berikut:

1. Apabila beban yang tehubung kurang dari maksimal beban yang ditetapkan dan tegangan PLTS lebih dari tegangan konfigurasi minimum PLTS, maka kontroller akan menggerakan relai ke kondisi *LOW* untuk menghubungkan beban ke sumber listrik PLTS.
2. Apabila beban yang tehubung kurang dari maksimal beban yang ditetapkan dan tegangan PLTS kurang dari tegangan konfigurasi minimum PLTS, maka kontroller akan menggerakan relai ke kondisi *HIGH* untuk menghubungkan beban ke sumber listrik PLN.
3. Apabila beban yang tehubung lebih dari yang ditetapkan dan tegangan PLTS lebih dari atau kurang dari tegangan konfigurasi minimum PLTS, maka kontroller akan menggerakan relai ke kondisi *HIGH* untuk menghubungkan beban ke sumber listrik PLN.

Apabila konfigurasi kontrol otomatis *off* dan sumber listrik *auto*, dan salah satu *switch* sumber listrik berada di kondisi *off*, maka didapatkan 2 kondisi seperti berikut.

1. Apabila *switch* sumber listrik PLN berada di kondisi *on* dan switch sumber listrik PLTS berada di kondisi *off*, maka kontroller akan menggerakan relai ke kondisi *HIGH* untuk menghubungkan beban ke sumber listrik PLN.
2. Apabila *switch* sumber listrik PLN berada di kondisi *off* dan switch sumber listrik PLTS berada di kondisi *on*, maka kontroller akan menggerakan relai ke kondisi *LOW* untuk menghubungkan beban ke sumber listrik PLTS (kondisi ini akan mengabaikan konfigurasi maksimal beban).

Apabila konfigurasi kontrol otomatis *off* dan konfigurasi sumber listrik di kondisi PLN, maka kontroller akan menggerakan relai ke kondisi *HIGH* untuk menghubungkan beban ke sumber listrik PLN dan apabila konfigurasi sumber listrik di kondisi PLTS, maka kontroller akan menggerakan relai ke kondisi *LOW* untuk menghubungkan beban ke sumber listrik PLTS (kondisi ini akan mengabaikan konfigurasi maksimal beban).

# **BAB V KESIMPULAN DAN SARAN**

## **Kesimpulan**

Berdasarkan hasil pengujian dan analisis pada prototipe yang telah dibuat, didapat kesimpulan sebagai berikut.

1. Sistem ATS yang dirancang bekerja dengan baik untuk melakukan distribusi beban listrik antara sumber listrik PLN dan PLTS berdasarkan konfigurasi *switch* sumber listrik, tegangan minimum PLTS, pengaturan sumber listrik dan maksimal beban pada masing-masing relai yang telah ditetapkan.
2. Pengukuran yang didapatkan melalui sensor-sensor yang digunakan seperti sensor ZMPT101B untuk mengukur tegangan AC PLN dan PLTS, sensor Tegangan DC untuk mengukur tegangan baterai, dan sensor ACS712 untuk mengukur arus saat beban terhubung ke sumber listrik PLN atau PLTS dengan pembanding alat ukur didapatkan semua sensor yang digunakan memiliki tingkat akurasi yang baik.
3. Sistem *monitoring* yang dirancang dapat menampilkan data yang tercatat pada *database firebase* secara *realtime* saat terjadi perubahan data setiap 4 sampai 5 detik sekali pada *database firebase*.
4. Sistem *controlling* yang dirancang dapat bekerja dengan baik sesuai dengan *flowchart* sistem yang telah dirancang dan parameter-parameter atau konfigurasi yang telah ditetapkan.

## **Saran**

Penelitian selanjutnya diharapkan dapat lebih mengembangkan penelitian ini, untuk itu penulis memberikan saran penelitian selanjutnya sebagai berikut.

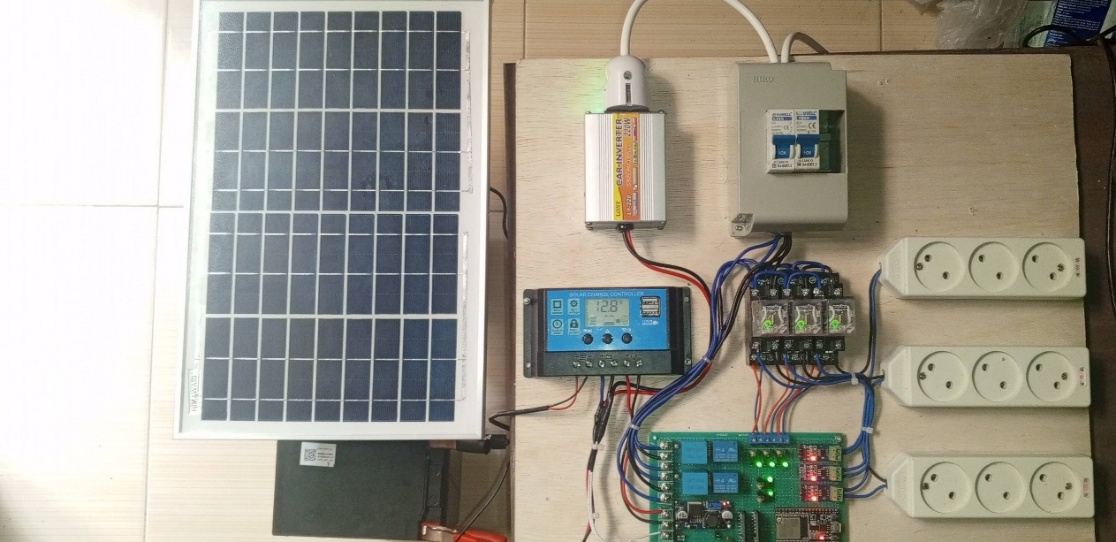
1. Melakukan pengujian prototipe dengan kondisi operasional yang lebih beragam.
2. Penambahan kontrol secara *analog* agar dapat melakukan kontrol disaat tidak ada konektivitas internet.
3. Penambahan kontrol waktu untuk menghubungkan beban ke sumber listrik yang di inginkan pada waktu tertentu.

# **DAFTAR PUSTAKA**

|  |  |
| --- | --- |
| [1] | N. P. Purba, "Variabilitas Angin dan Gelombang Laut Sebagai Energi Terbarukan di Pantai Selatan Jawa Barat," vol. 5, no. Jurnal Akuatika, pp. 8-15, 2014. |
| [2] | PLN, "Statistik PLN 2023," PLN, Jakarta, 2023. |
| [3] | R. Ramadhan, "Rancang Bangun Automatic Transfer Switch (ATS)," Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang, 2016. |
| [4] | *Peraturan Menteri Energi dan Sumber Daya Mineral Nomor 53 Tahun 2018 tentang Perubahan Atas Peraturan Menteri Energi Dan Sumber Daya Mineral Nomor 50 Tahun 2017 Tentang Pemanfaatan Sumber Energi Terbarukan Untuk Penyediaan Tenaga Listrik,* Jakarta, 2018. |
| [5] | T. R. S. R. Nelly Safitri, Buku Teknologi PhotoVoltaic, Aceh: Yayasan Puga Aceh Riset, 2019. |
| [6] | D. N. R. S. H. Robinzon Pakpahan, "Rancang Bangun dan Implementasi Automatic Transfer Switch (ATS) Menggunakan Arduino UNO dan Relai," *Jurnal Elektro dan Telekomunikasi Terapan,* vol. 3, no. 2, 2017. |
| [7] | S. K. Yunus Tjandi, *KENDALI PERALATAN LISTRIK BERBASIS SMARTPHONE,* vol. I, no. Jurnal SCIENTIFIC PINISI, pp. 73-81, 2015. |
| [8] | "Riza Samsinar, Redi Rahman Fitria Mulyadi, Dwi Arief Prambudi," *Sistem Monitoring Besaran Listrik dan Energi Penerangan Jalan Umum Secara Realtime Berbasis Web,* vol. 1, no. RESISTOR, pp. 7-12. |
| [9] | I. M. W. I. B. S. Muhammad Rizky Nur Jayadi, "RANCANG BANGUN SISTEM MONITORING ENERGI LISTRIK," no. Prosiding Fintek, pp. 134-142, 2021. |
| [10] | K. T. M. M. F. N. P. C. Samsurizal, Pengenalan Pembangkit Listrik Tenaga Surya (PLTS), Jakarta Barat: Institut Teknologi PLN, 2021. |
| [11] | Jakartanotebook, "Jakartanotebook," [Online]. Available: https://www.jakartanotebook.com/p/taffware-solar-charger-controller-regulator-lcd-dual-usb-10a-12v-24v-w88-a-black. |
| [12] | MaxMaxPlaza, "Monqiqi Aki Kering Original 12V 12AH," [Online]. Available: https://www.lazada.co.id/products/monqiqi-aki-kering-original-12v-12ah-baterai-accu-tangki-alat-semprot-sprayer-elektrik-asli-12-volt-12-ampere-hour-battery-for-knapsack-sprayer-i7592256264.html. |
| [13] | panelsinarsurya, "Penjelasan Tentang Baterai (ACCU/AKI)," [Online]. Available: https://panelsinarsurya.wordpress.com/2016/09/20/penjelasan-tentang-baterai-accuaki/. |
| [14] | R. Sianipar, "Jurusan Teknik Elektro Universitas Trisakti," *DASAR PERENCANAAN PEMBANGKIT LISTRIK TENAGA SURYA,* vol. 11, pp. 61-78, 2014. |
| [15] | I. H. M. F. S. M. F. H. A. L. W. Rohmanita Duanaputri, "ELPOSYS: Jurnal Sistem Kelistrikan," *Sistem Monitoring Online Dan Analisis Perfomansi Plts Panel Surya Monocrystalline 100 Wp Berbasis Web,* vol. Vol. 10 No.1, 2023. |
| [16] | JakartaNotebook, "Taffware Power Inverter Mobil Pure Sine Wave DC 12V to AC 220V 4000W - NBQ4000W - Black," [Online]. Available: https://www.jakartanotebook.com/p/taffware-power-inverter-mobil-pure-sine-wave-dc-12v-to-ac-220v-4000w-nbq4000w-black. |
| [17] | E. Susanto, "Automatic Transfer Switch," *Jurnal Teknik Elektro Unnes Vol. 5 No. 1,* p. 18, 2013. |
| [18] | ABB, "Miniature circuit breakers (MCBs)," [Online]. Available: https://search.abb.com/library/Download.aspx?DocumentID=9AKK107046A0454&LanguageCode=en&DocumentPartId=&Action=Launch. |
| [19] | Espressif, "ESP32 WROOM Datasheet," 02 02 2023. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d\_esp32-wroom-32u\_datasheet\_en.pdf. |
| [20] | Espressif, "ESP32-DevKitC V4 Documentation," [Online]. Available: https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32/esp32-devkitc/user\_guide.html. |
| [21] | M. D. A. K. M Rizal Alfariski, "Automatic Transfer Switch (ATS) Menggunakan Arduino Uno, Relay dan Monitoring Berbasis IoT". |
| [22] | Swagatam, "How a Relay Works – How to Connect N/O, N/C Pins," [Online]. Available: https://www.homemade-circuits.com/how-a-relay-works-in-circuits-how-to-connect-it/. |
| [23] | Keithley, Switching Handbook, A Guide to Signal Switching in Automated Test Systems 6th Edition, Keithley Instruments, Inc., 2008. |
| [24] | "Qingxian Zeming Langxi Electronic," [Online]. Available: https://www.micro-transformer.com/2ma-2ma-voltage-transformer-ZMPT101B.html. |
| [25] | Q. INDIA, "AC VOLTAGE SENSOR MODULE ZMPT101B SINGLE PHASE Z101B," [Online]. Available: https://qbmindia.com/shop/sensors/other-sensors-modules/ac-voltage-sensor-module-zmpt101b-single-phase/. |
| [26] | A. MicroSystems, "ACS712 Datasheet," [Online]. Available: https://www.allegromicro.com/~/media/files/datasheets/acs712-datasheet.ashx. |
| [27] | C. N. E. ANUPI, "ACS712 30A Range Hall Current Sensor Arus Module ACS 712 For Arduino," [Online]. Available: https://www.tokopedia.com/cvnea/acs712-30a-range-hall-current-sensor-arus-module-acs-712-for-arduino. |
| [28] | T. A. B. S. Ahmad Imron, "Perancangan Akuisisi Data Pada Panel RTU PT.PLN (Persero) Berplatform Android," vol. 7, no. Program Studi Sarjana Departemen Teknik Elektro, Universitas Diponegoro, 2018. |
| [29] | G. W. Pambudi, "Cara Mengakses Sensor Tegangan DC menggunakan Arduino," [Online]. Available: https://www.cronyos.com/cara-mengakses-sensor-tegangan-dc-menggunakan-arduino/. |
| [30] | T. Instruments, "ADS111x Ultra-Small, Low-Power, I2C-Compatible," 2016. [Online]. Available: https://www.ti.com/lit/ds/symlink/ads1115.pdf. |
| [31] | components101, "ADS1115 Module with Programmable Gain Amplifier," [Online]. Available: https://components101.com/modules/ads1115-module-with-programmable-gain-amplifier. |
| [32] | Google, "Firebase Realtime Database," [Online]. Available: https://firebase.google.com/docs/database. |
| [33] | Google, "Firebase Realtime Database Console," [Online]. Available: https://console.firebase.google.com/u/0/project/tugas-akhir-c789/database/tugas-akhir-c789-default-rtdb/data?pli=1. [Accessed 15 12 2024]. |
| [34] | Nuxt, "The Intuitive Vue Framework," [Online]. Available: https://nuxt.com/. |
| [35] | Vue, "Introduction of Vue," [Online]. Available: https://vuejs.org/guide/introduction.html. |

# **LAMPIRAN**

**Hasil Rancangan Alat**



**Sensor Tegangan ZMPT101B PLN**

|  |  |  |
| --- | --- | --- |
| Sensor (V) | Alat Ukur (V) | *Error* (%) |
| 235 | 236 | 0,42% |
| 236 | 235 | 0,43% |
| 236 | 235 | 0,43% |
| 236 | 236 | 0,00% |
| 236 | 236 | 0,00% |
| 236 | 236 | 0,00% |
| 236 | 236 | 0,00% |
| 236 | 236 | 0,00% |
| 236 | 236 | 0,00% |
| 236 | 236 | 0,00% |
| Rata-rata persentase *error* | | 0,13% |

**Foto Hasil Pengujian Sensor Tegangan ZMPT101B PLN**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Sensor ZMPT101B PLTS**

|  |  |  |
| --- | --- | --- |
| Sensor (V) | Alat Ukur (V) | *Error* (%) |
| 237 | 237 | 0,00% |
| 222 | 237 | 6,33% |
| 237 | 242 | 2,07% |
| 237 | 242 | 2,07% |
| 237 | 240 | 1,25% |
| 242 | 240 | 0,83% |
| 236 | 239 | 1,26% |
| 231 | 239 | 3,35% |
| 237 | 240 | 1,25% |
| 238 | 240 | 0,83% |
| Rata-rata persentase *error* | | 1,92% |

**Foto Hasil Pengujian Sensor Tegangan ZMPT101B PLTS**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Sensor Tegangan DC Baterai**

|  |  |  |
| --- | --- | --- |
| Sensor (V) | Alat Ukur (V) | *Error* (%) |
| 12,66 | 12,65 | 0,08% |
| 12,66 | 12,65 | 0,08% |
| 12,66 | 12,65 | 0,08% |
| 12,66 | 12,65 | 0,08% |
| 12,66 | 12,65 | 0,08% |
| 12,66 | 12,65 | 0,08% |
| 12,66 | 12,65 | 0,08% |
| 12,66 | 12,65 | 0,08% |
| 12,66 | 12,65 | 0,08% |
| 12,66 | 12,65 | 0,08% |
| 12,66 | 12,65 | 0,08% |
| Rata-rata persentase *error* | | 0,08% |

**Foto Hasil Pengujian Sensor DC Baterai**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Sensor ACS712 I PLN**

|  |  |  |  |
| --- | --- | --- | --- |
| Beban (W) | PLN | | |
| Sensor (A) | Alat Ukur (A) | *Error* (%) |
| 10 | 0,091 | 0,08 | 13,75% |
| 25 | 0,112 | 0,1 | 12,00% |
| 40 | 0,173 | 0,17 | 1,76% |
| 60 | 0,254 | 0,26 | 2,31% |
| 75 | 0,318 | 0,33 | 3,64% |
| 100 | 0,423 | 0,45 | 6,00% |
| Rata-rata persentase *error* | | | 6,58% |

**Foto Hasil Pengujian Sensor Arus ACS712 I PLN**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

**Sensor Arus ACS712 I PLTS**

|  |  |  |  |
| --- | --- | --- | --- |
| Beban (W) | PLTS | | |
| Sensor (A) | Alat Ukur (A) | *Error* (%) |
| 10 | 0,093 | 0,08 | 16,25% |
| 25 | 0,134 | 0,11 | 21,82% |
| 40 | 0,178 | 0,19 | 6,32% |
| 60 | 0,264 | 0,27 | 2,22% |
| 75 | 0,336 | 0,34 | 1,18% |
| 100 | 0,429 | 0,49 | 12,45% |
| Rata-rata persentase *error* | | | 10,04% |

**Foto Hasil Pengujian Sensor Arus ACS712 I PLTS**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

**Sensor Arus ACS712 II PLN**

|  |  |  |  |
| --- | --- | --- | --- |
| Beban (W) | PLN | | |
| Sensor (A) | Alat Ukur (A) | *Error* (%) |
| 10 | 0,09 | 0,08 | 12,50% |
| 25 | 0,108 | 0,1 | 8,00% |
| 40 | 0,172 | 0,17 | 1,18% |
| 60 | 0,251 | 0,26 | 3,46% |
| 75 | 0,316 | 0,33 | 4,24% |
| 100 | 0,451 | 0,47 | 4,04% |
| Rata-rata persentase *error* | | | 5,57% |

**Foto Hasil Pengujian Sensor Arus ACS712 II PLN**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

**Sensor ACS712 II PLTS**

|  |  |  |  |
| --- | --- | --- | --- |
| Beban (W) | PLTS | | |
| Sensor (A) | Alat Ukur (A) | *Error* (%) |
| 10 | 0,096 | 0,08 | 20,00% |
| 25 | 0,13 | 0,11 | 18,18% |
| 40 | 0,196 | 0,18 | 8,89% |
| 60 | 0,274 | 0,27 | 1,48% |
| 75 | 0,333 | 0,34 | 2,06% |
| 100 | 0,477 | 0,5 | 4,60% |
| Rata-rata persentase *error* | | | 9,20% |

**Foto Hasil Pengujian Sensor Arus ACS712 II PLTS**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

**Sensor ACS712 III PLN**

|  |  |  |  |
| --- | --- | --- | --- |
| Beban (W) | PLN | | |
| Sensor (A) | Alat Ukur (A) | *Error* (%) |
| 10 | 0,087 | 0,08 | 8,75% |
| 25 | 0,114 | 0,1 | 14,00% |
| 40 | 0,178 | 0,17 | 4,71% |
| 60 | 0,254 | 0,26 | 2,31% |
| 75 | 0,318 | 0,33 | 3,64% |
| 100 | 0,453 | 0,48 | 5,62% |
| Rata-rata persentase *error* | | | 6,50% |

**Foto Hasil Pengujian Sensor Arus ACS712 III PLN**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

**Sensor ACS712 III PLTS**

|  |  |  |  |
| --- | --- | --- | --- |
| Beban (W) | PLTS | | |
| Sensor (A) | Alat Ukur (A) | *Error* (%) |
| 10 | 0,093 | 0,08 | 16,25% |
| 25 | 0,127 | 0,11 | 15,45% |
| 40 | 0,189 | 0,18 | 5,00% |
| 60 | 0,261 | 0,26 | 0,38% |
| 75 | 0,343 | 0,34 | 0,88% |
| 100 | 0,439 | 0,47 | 6,60% |
| Rata-rata persentase error | | | 7,43% |

**Foto Hasil Pengujian Sensor Arus ACS712 III PLTS**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

***Source Code* ESP32**

#include <Arduino.h>

#include <WiFi.h>

#include <HTTPClient.h>

#include <Preferences.h>

#include <ArduinoJson.h>

#include <SPI.h>

#include <Adafruit\_ADS1X15.h>

HTTPClient http;

Preferences preferences;

Adafruit\_ADS1115 ads1;

Adafruit\_ADS1115 ads2;

#define PIN\_RELAY\_AC1 19

#define PIN\_RELAY\_AC2 23

#define PIN\_RELAY\_N1 18

#define PIN\_RELAY\_N2 17

#define PIN\_RELAY\_N3 16

#define WIFI\_SSID "WiFi-Ku"

#define WIFI\_PASSWORD "123456789"

String API\_URL = "https://skripsi.irvan789.local/api/iot";

String API\_KEY = "UMf1V68wZf2qV1iqvKIQNf2wU";

String DEVICE\_ID = "47d7c573-ade7-421b-a415-be516a9885f9";

int ADS\_ERROR = 0;

int WIFI\_CLIENT\_STATUS = 0;

float PLN\_VOLTAGE\_MULTIPLER = 80.661;

float PLTS\_VOLTAGE\_MULTIPLER = 95.721;

float BATT\_VOLTAGE\_MULTIPLER = 4.96;

float CURRENT\_SENSOR\_FACTOR\_1 = 0.0697;

float CURRENT\_SENSOR\_FACTOR\_2 = 0.1258;

float CURRENT\_SENSOR\_FACTOR\_3 = 0.0687;

float CURRENT\_PLN\_MULTIPLER = 0.152;

float CURRENT\_PLTS\_MULTIPLER = 0.343;

int AUTOMATED = 1;

u\_int32\_t CONFIG\_TIME = 0;

int SOURCE\_PLN = 1;

int SOURCE\_PLTS = 1;

float SOURCE\_PLTS\_MIN = 200;

String RELAY\_N1\_SOURCE = "auto";

String RELAY\_N2\_SOURCE = "auto";

String RELAY\_N3\_SOURCE = "auto";

int CURRENT\_RELAY\_N1\_SOURCE = 1;

int CURRENT\_RELAY\_N2\_SOURCE = 1;

int CURRENT\_RELAY\_N3\_SOURCE = 1;

float RELAY\_N1\_MAX\_LOAD = 0.136;

float RELAY\_N2\_MAX\_LOAD = 0.227;

float RELAY\_N3\_MAX\_LOAD = 0.318;

float PLN\_VOLTAGE = 0;

float PLTS\_VOLTAGE = 0;

float BATT\_VOLTAGE = 0;

int CALIBRATED\_USED\_N1 = 1;

int CALIBRATED\_USED\_N2 = 1;

int CALIBRATED\_USED\_N3 = 1;

float ADS\_CURRENT\_AC1 = 0;

float ADS\_CURRENT\_AC2 = 0;

float ADS\_CURRENT\_AC3 = 0;

float RAW1\_CURRENT\_AC1 = 0;

float RAW1\_CURRENT\_AC2 = 0;

float RAW1\_CURRENT\_AC3 = 0;

float RAW2\_CURRENT\_AC1 = 0;

float RAW2\_CURRENT\_AC2 = 0;

float RAW2\_CURRENT\_AC3 = 0;

float CURRENT\_AC1 = 0;

float CURRENT\_AC2 = 0;

float CURRENT\_AC3 = 0;

int RELAY\_N1\_STATE = 0;

int RELAY\_N2\_STATE = 0;

int RELAY\_N3\_STATE = 0;

float POWER\_RELAY\_N1 = 0;

float POWER\_RELAY\_N2 = 0;

float POWER\_RELAY\_N3 = 0;

void getSavedDeviceConfig()

{

AUTOMATED = preferences.getInt("automated", AUTOMATED);

CONFIG\_TIME = preferences.getUInt("configTime", CONFIG\_TIME);

SOURCE\_PLN = preferences.getInt("pln", SOURCE\_PLN);

SOURCE\_PLTS = preferences.getInt("plts", SOURCE\_PLTS);

SOURCE\_PLTS\_MIN = preferences.getInt("plts\_min", SOURCE\_PLTS\_MIN);

RELAY\_N1\_SOURCE = preferences.getString("relay\_sn1", RELAY\_N1\_SOURCE);

RELAY\_N2\_SOURCE = preferences.getString("relay\_sn2", RELAY\_N2\_SOURCE);

RELAY\_N3\_SOURCE = preferences.getString("relay\_sn3", RELAY\_N3\_SOURCE);

RELAY\_N1\_MAX\_LOAD = preferences.getFloat("relay\_ln1", RELAY\_N1\_MAX\_LOAD);

RELAY\_N2\_MAX\_LOAD = preferences.getFloat("relay\_ln2", RELAY\_N2\_MAX\_LOAD);

RELAY\_N3\_MAX\_LOAD = preferences.getFloat("relay\_ln3", RELAY\_N3\_MAX\_LOAD);

}

void updateDeviceConfig()

{

if (WIFI\_CLIENT\_STATUS == 3)

{

http.begin(API\_URL + "/config?deviceId=" + DEVICE\_ID + "&key=" + API\_KEY);

int httpStatus = http.GET();

if (httpStatus == HTTP\_CODE\_OK)

{

DynamicJsonDocument doc(2048);

deserializeJson(doc, http.getString());

AUTOMATED = doc["data"]["automated"].as<boolean>() ? 1 : 0;

CONFIG\_TIME = doc["data"]["configTime"].as<u\_int32\_t>();

SOURCE\_PLN = doc["data"]["pln"]["active"].as<boolean>() ? 1 : 0;

SOURCE\_PLTS = doc["data"]["plts"]["active"].as<boolean>() ? 1 : 0;

SOURCE\_PLTS\_MIN = doc["data"]["plts"]["min\_voltage"].as<int>();

RELAY\_N1\_SOURCE = doc["data"]["relay"]["n1"]["source"].as<String>();

RELAY\_N2\_SOURCE = doc["data"]["relay"]["n2"]["source"].as<String>();

RELAY\_N3\_SOURCE = doc["data"]["relay"]["n3"]["source"].as<String>();

RELAY\_N1\_MAX\_LOAD = doc["data"]["relay"]["n1"]["max\_load"].as<float>();

RELAY\_N2\_MAX\_LOAD = doc["data"]["relay"]["n2"]["max\_load"].as<float>();

RELAY\_N3\_MAX\_LOAD = doc["data"]["relay"]["n3"]["max\_load"].as<float>();

preferences.putInt("automated", AUTOMATED);

preferences.putUInt("configTime", CONFIG\_TIME);

preferences.putInt("pln", SOURCE\_PLN);

preferences.putInt("plts", SOURCE\_PLTS);

preferences.putInt("plts\_min", SOURCE\_PLTS\_MIN);

preferences.putString("relay\_sn1", RELAY\_N1\_SOURCE);

preferences.putString("relay\_sn2", RELAY\_N2\_SOURCE);

preferences.putString("relay\_sn3", RELAY\_N3\_SOURCE);

preferences.putFloat("relay\_ln1", RELAY\_N1\_MAX\_LOAD);

preferences.putFloat("relay\_ln2", RELAY\_N2\_MAX\_LOAD);

preferences.putFloat("relay\_ln3", RELAY\_N3\_MAX\_LOAD);

doc.clear();

}

}

else

{

getSavedDeviceConfig();

}

http.end();

}

double computeACVolts(Adafruit\_ADS1115 ads, int channel, int sampling = 25)

{

float vmax = 0;

float vmin = 0;

float vnow = 0;

for (int i = 0; i < sampling; i++)

{

int16\_t adsChannel = ads.readADC\_SingleEnded(channel);

vnow = ads.computeVolts(adsChannel);

if (vnow > vmax)

{

vmax = vnow;

}

else

{

if (vmin == 0)

{

vmin = vnow;

}

else

{

if (vnow < vmin)

{

vmin = vnow;

}

}

}

delayMicroseconds(100);

}

return vmax - vmin;

}

double computeDCVolts(Adafruit\_ADS1115 ads, int channel, int sampling = 5)

{

float voltage = 0;

for (int i = 0; i < sampling; i++)

{

int16\_t adsChannel = ads.readADC\_SingleEnded(channel);

voltage += ads.computeVolts(adsChannel);

}

return voltage / sampling;

}

void setup()

{

delay(500);

preferences.begin("app", false);

delay(500);

WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD);

while (WiFi.status() != WL\_CONNECTED)

{

delay(5000);

WIFI\_CLIENT\_STATUS = WiFi.status();

if (WIFI\_CLIENT\_STATUS == WL\_NO\_SSID\_AVAIL || WIFI\_CLIENT\_STATUS == WL\_DISCONNECTED)

{

getSavedDeviceConfig();

break;

}

else

{

updateDeviceConfig();

}

}

delay(500);

ads1.setGain(GAIN\_TWOTHIRDS);

ads2.setGain(GAIN\_TWOTHIRDS);

delay(500);

if (!ads1.begin(0x48))

{

ADS\_ERROR += 1;

}

if (!ads2.begin(0x49))

{

ADS\_ERROR += 1;

}

delay(500);

pinMode(PIN\_RELAY\_N1, OUTPUT);

pinMode(PIN\_RELAY\_N2, OUTPUT);

pinMode(PIN\_RELAY\_N3, OUTPUT);

delay(500);

pinMode(PIN\_RELAY\_AC1, OUTPUT);

pinMode(PIN\_RELAY\_AC2, OUTPUT);

delay(500);

}

void loop()

{

WIFI\_CLIENT\_STATUS = WiFi.status();

if (WIFI\_CLIENT\_STATUS != 3)

{

WiFi.reconnect();

WIFI\_CLIENT\_STATUS = WiFi.status();

}

digitalWrite(PIN\_RELAY\_AC1, SOURCE\_PLN == 1 ? 0 : 1);

digitalWrite(PIN\_RELAY\_AC2, SOURCE\_PLTS == 1 ? 0 : 1);

int SOURCE\_PLTSN = 0;

if (ADS\_ERROR == 0)

{

float ADS\_PLN\_VOLTAGE = computeDCVolts(ads1, 0, 8);

float ADS\_PLTS\_VOLTAGE = computeDCVolts(ads1, 1, 8);

float ADS\_BATT\_VOLTAGE = computeDCVolts(ads1, 2, 5);

char CHAR\_ADS\_PLN\_VOLTAGE[10];

char CHAR\_ADS\_PLTS\_VOLTAGE[10];

char CHAR\_ADS\_BATT\_VOLTAGE[10];

sprintf(CHAR\_ADS\_PLN\_VOLTAGE, "%.2f", ADS\_PLN\_VOLTAGE);

sprintf(CHAR\_ADS\_PLTS\_VOLTAGE, "%.2f", ADS\_PLTS\_VOLTAGE);

sprintf(CHAR\_ADS\_BATT\_VOLTAGE, "%.2f", ADS\_BATT\_VOLTAGE);

float RAW\_PLN\_VOLTAGE = atof(CHAR\_ADS\_PLN\_VOLTAGE);

float RAW\_PLTS\_VOLTAGE = atof(CHAR\_ADS\_PLTS\_VOLTAGE);

float RAW\_BATT\_VOLTAGE = atof(CHAR\_ADS\_BATT\_VOLTAGE);

char CHAR\_RAW\_PLN\_VOLTAGE[10];

char CHAR\_RAW\_PLTS\_VOLTAGE[10];

char CHAR\_RAW\_BATT\_VOLTAGE[10];

sprintf(CHAR\_RAW\_PLN\_VOLTAGE, "%.0f", (SOURCE\_PLN && RAW\_PLN\_VOLTAGE > 0.65 ? RAW\_PLN\_VOLTAGE : 0) \* PLN\_VOLTAGE\_MULTIPLER);

sprintf(CHAR\_RAW\_PLTS\_VOLTAGE, "%.0f", (SOURCE\_PLTS && RAW\_PLTS\_VOLTAGE > 1.95 ? RAW\_PLTS\_VOLTAGE : 0) \* PLTS\_VOLTAGE\_MULTIPLER);

sprintf(CHAR\_RAW\_BATT\_VOLTAGE, "%.2f", (RAW\_BATT\_VOLTAGE > 0 ? RAW\_BATT\_VOLTAGE : 0) \* BATT\_VOLTAGE\_MULTIPLER);

PLN\_VOLTAGE = atof(CHAR\_RAW\_PLN\_VOLTAGE);

PLTS\_VOLTAGE = atof(CHAR\_RAW\_PLTS\_VOLTAGE);

BATT\_VOLTAGE = atof(CHAR\_RAW\_BATT\_VOLTAGE);

if (PLN\_VOLTAGE > 5)

{

SOURCE\_PLTSN += 1;

}

if (PLTS\_VOLTAGE > 5 && PLTS\_VOLTAGE > SOURCE\_PLTS\_MIN)

{

SOURCE\_PLTSN += 2;

}

if (SOURCE\_PLTSN != 0)

{

RELAY\_N1\_STATE = RELAY\_N1\_SOURCE == "auto" ? (SOURCE\_PLTSN == 1 ? 1 : (SOURCE\_PLTSN == 2 ? 0 : (POWER\_RELAY\_N1 > RELAY\_N1\_MAX\_LOAD ? 1 : 0))) : (RELAY\_N1\_SOURCE == "pln" ? 1 : 0);

RELAY\_N2\_STATE = RELAY\_N2\_SOURCE == "auto" ? (SOURCE\_PLTSN == 1 ? 1 : (SOURCE\_PLTSN == 2 ? 0 : (POWER\_RELAY\_N2 > RELAY\_N2\_MAX\_LOAD ? 1 : 0))) : (RELAY\_N2\_SOURCE == "pln" ? 1 : 0);

RELAY\_N3\_STATE = RELAY\_N3\_SOURCE == "auto" ? (SOURCE\_PLTSN == 1 ? 1 : (SOURCE\_PLTSN == 2 ? 0 : (POWER\_RELAY\_N3 > RELAY\_N3\_MAX\_LOAD ? 1 : 0))) : (RELAY\_N3\_SOURCE == "pln" ? 1 : 0);

digitalWrite(PIN\_RELAY\_N1, RELAY\_N1\_STATE);

digitalWrite(PIN\_RELAY\_N2, RELAY\_N2\_STATE);

digitalWrite(PIN\_RELAY\_N3, RELAY\_N3\_STATE);

CALIBRATED\_USED\_N1 = RELAY\_N1\_STATE == 1 ? 0 : 1;

CALIBRATED\_USED\_N2 = RELAY\_N2\_STATE == 1 ? 0 : 1;

CALIBRATED\_USED\_N3 = RELAY\_N3\_STATE == 1 ? 0 : 1;

ADS\_CURRENT\_AC1 = computeACVolts(ads2, 0, 20);

ADS\_CURRENT\_AC2 = computeACVolts(ads2, 1, 20);

ADS\_CURRENT\_AC3 = computeACVolts(ads2, 2, 20);

char CHAR\_ADS\_CURRENT\_AC1[15];

char CHAR\_ADS\_CURRENT\_AC2[15];

char CHAR\_ADS\_CURRENT\_AC3[15];

sprintf(CHAR\_ADS\_CURRENT\_AC1, "%.4f", ADS\_CURRENT\_AC1 \* 1000);

sprintf(CHAR\_ADS\_CURRENT\_AC2, "%.4f", ADS\_CURRENT\_AC2 \* 1000);

sprintf(CHAR\_ADS\_CURRENT\_AC3, "%.4f", ADS\_CURRENT\_AC3 \* 1000);

RAW1\_CURRENT\_AC1 = atof(CHAR\_ADS\_CURRENT\_AC1) \* CURRENT\_SENSOR\_FACTOR\_1;

RAW1\_CURRENT\_AC2 = atof(CHAR\_ADS\_CURRENT\_AC2) \* CURRENT\_SENSOR\_FACTOR\_2;

RAW1\_CURRENT\_AC3 = atof(CHAR\_ADS\_CURRENT\_AC3) \* CURRENT\_SENSOR\_FACTOR\_3;

char CHAR\_RAW1\_CURRENT\_AC1[15];

char CHAR\_RAW1\_CURRENT\_AC2[15];

char CHAR\_RAW1\_CURRENT\_AC3[15];

sprintf(CHAR\_RAW1\_CURRENT\_AC1, "%.2f", RAW1\_CURRENT\_AC1);

sprintf(CHAR\_RAW1\_CURRENT\_AC2, "%.2f", RAW1\_CURRENT\_AC2);

sprintf(CHAR\_RAW1\_CURRENT\_AC3, "%.2f", RAW1\_CURRENT\_AC3);

RAW2\_CURRENT\_AC1 = atof(CHAR\_RAW1\_CURRENT\_AC1);

RAW2\_CURRENT\_AC2 = atof(CHAR\_RAW1\_CURRENT\_AC2);

RAW2\_CURRENT\_AC3 = atof(CHAR\_RAW1\_CURRENT\_AC3);

char CHAR\_RAW2\_CURRENT\_AC1[15];

char CHAR\_RAW2\_CURRENT\_AC2[15];

char CHAR\_RAW2\_CURRENT\_AC3[15];

sprintf(CHAR\_RAW2\_CURRENT\_AC1, "%.3f", (RAW2\_CURRENT\_AC1 > 0.15 ? RAW2\_CURRENT\_AC1 : 0) \* (CALIBRATED\_USED\_N1 == 0 ? CURRENT\_PLN\_MULTIPLER : CURRENT\_PLTS\_MULTIPLER));

sprintf(CHAR\_RAW2\_CURRENT\_AC2, "%.3f", (RAW2\_CURRENT\_AC2 > 0.15 ? RAW2\_CURRENT\_AC2 : 0) \* (CALIBRATED\_USED\_N2 == 0 ? CURRENT\_PLN\_MULTIPLER : CURRENT\_PLTS\_MULTIPLER));

sprintf(CHAR\_RAW2\_CURRENT\_AC3, "%.3f", (RAW2\_CURRENT\_AC3 > 0.15 ? RAW2\_CURRENT\_AC3 : 0) \* (CALIBRATED\_USED\_N3 == 0 ? CURRENT\_PLN\_MULTIPLER : CURRENT\_PLTS\_MULTIPLER));

CURRENT\_AC1 = atof(CHAR\_RAW2\_CURRENT\_AC1);

CURRENT\_AC2 = atof(CHAR\_RAW2\_CURRENT\_AC2);

CURRENT\_AC3 = atof(CHAR\_RAW2\_CURRENT\_AC3);

POWER\_RELAY\_N1 = (CURRENT\_AC1 \* (CALIBRATED\_USED\_N1 == 0 ? PLN\_VOLTAGE : PLTS\_VOLTAGE));

POWER\_RELAY\_N2 = (CURRENT\_AC2 \* (CALIBRATED\_USED\_N2 == 0 ? PLN\_VOLTAGE : PLTS\_VOLTAGE));

POWER\_RELAY\_N3 = (CURRENT\_AC3 \* (CALIBRATED\_USED\_N3 == 0 ? PLN\_VOLTAGE : PLTS\_VOLTAGE));

RELAY\_N1\_STATE = RELAY\_N1\_SOURCE == "auto" ? (SOURCE\_PLTSN == 1 ? 1 : (SOURCE\_PLTSN == 2 ? 0 : (POWER\_RELAY\_N1 > RELAY\_N1\_MAX\_LOAD ? 1 : 0))) : (RELAY\_N1\_SOURCE == "pln" ? 1 : 0);

RELAY\_N2\_STATE = RELAY\_N2\_SOURCE == "auto" ? (SOURCE\_PLTSN == 1 ? 1 : (SOURCE\_PLTSN == 2 ? 0 : (POWER\_RELAY\_N2 > RELAY\_N2\_MAX\_LOAD ? 1 : 0))) : (RELAY\_N2\_SOURCE == "pln" ? 1 : 0);

RELAY\_N3\_STATE = RELAY\_N3\_SOURCE == "auto" ? (SOURCE\_PLTSN == 1 ? 1 : (SOURCE\_PLTSN == 2 ? 0 : (POWER\_RELAY\_N3 > RELAY\_N3\_MAX\_LOAD ? 1 : 0))) : (RELAY\_N3\_SOURCE == "pln" ? 1 : 0);

digitalWrite(PIN\_RELAY\_N1, RELAY\_N1\_STATE);

digitalWrite(PIN\_RELAY\_N2, RELAY\_N2\_STATE);

digitalWrite(PIN\_RELAY\_N3, RELAY\_N3\_STATE);

CALIBRATED\_USED\_N1 = RELAY\_N1\_STATE == 1 ? 0 : 1;

CALIBRATED\_USED\_N2 = RELAY\_N2\_STATE == 1 ? 0 : 1;

CALIBRATED\_USED\_N3 = RELAY\_N3\_STATE == 1 ? 0 : 1;

}

else

{

RELAY\_N1\_STATE = 1;

RELAY\_N2\_STATE = 1;

RELAY\_N3\_STATE = 1;

digitalWrite(PIN\_RELAY\_N1, RELAY\_N1\_STATE);

digitalWrite(PIN\_RELAY\_N2, RELAY\_N2\_STATE);

digitalWrite(PIN\_RELAY\_N3, RELAY\_N3\_STATE);

CALIBRATED\_USED\_N1 = 0;

CALIBRATED\_USED\_N2 = 0;

CALIBRATED\_USED\_N3 = 0;

POWER\_RELAY\_N1 = 0;

POWER\_RELAY\_N2 = 0;

POWER\_RELAY\_N3 = 0;

CURRENT\_AC1 = 0;

CURRENT\_AC2 = 0;

CURRENT\_AC3 = 0;

}

}

else

{

RELAY\_N1\_STATE = 1;

RELAY\_N2\_STATE = 1;

RELAY\_N3\_STATE = 1;

digitalWrite(PIN\_RELAY\_N1, RELAY\_N1\_STATE);

digitalWrite(PIN\_RELAY\_N2, RELAY\_N2\_STATE);

digitalWrite(PIN\_RELAY\_N3, RELAY\_N3\_STATE);

}

if (WIFI\_CLIENT\_STATUS == 3)

{

http.begin(API\_URL + "/sensor?deviceId=" + DEVICE\_ID + "&key=" + API\_KEY);

http.setTimeout(1000);

DynamicJsonDocument doc(2048);

doc["vpln"] = PLN\_VOLTAGE;

doc["vplts"] = PLTS\_VOLTAGE;

doc["vbatt"] = BATT\_VOLTAGE;

doc["lpln"] = (RELAY\_N1\_STATE == 1 ? POWER\_RELAY\_N1 : 0) + (RELAY\_N2\_STATE == 1 ? POWER\_RELAY\_N2 : 0) + (RELAY\_N3\_STATE == 1 ? POWER\_RELAY\_N3 : 0);

doc["lplts"] = (RELAY\_N1\_STATE == 0 ? POWER\_RELAY\_N1 : 0) + (RELAY\_N2\_STATE == 0 ? POWER\_RELAY\_N2 : 0) + (RELAY\_N3\_STATE == 0 ? POWER\_RELAY\_N3 : 0);

doc["relay"]["n1"]["power"] = POWER\_RELAY\_N1;

doc["relay"]["n1"]["current"] = CURRENT\_AC1;

doc["relay"]["n1"]["source"] = RELAY\_N1\_STATE == 0 ? "plts" : "pln";

doc["relay"]["n2"]["power"] = POWER\_RELAY\_N2;

doc["relay"]["n2"]["current"] = CURRENT\_AC2;

doc["relay"]["n2"]["source"] = RELAY\_N2\_STATE == 0 ? "plts" : "pln";

doc["relay"]["n3"]["power"] = POWER\_RELAY\_N3;

doc["relay"]["n3"]["current"] = CURRENT\_AC3;

doc["relay"]["n3"]["source"] = RELAY\_N3\_STATE == 0 ? "plts" : "pln";

String payload;

serializeJson(doc, payload);

int httpStatus = http.PUT(payload);

if (httpStatus == HTTP\_CODE\_OK)

{

DynamicJsonDocument doc2(512);

deserializeJson(doc2, http.getString());

u\_int32\_t NEW\_CONFIG\_TIME = doc2["data"]["configTime"].as<u\_int32\_t>();

if (NEW\_CONFIG\_TIME != CONFIG\_TIME)

{

updateDeviceConfig();

CONFIG\_TIME = NEW\_CONFIG\_TIME;

preferences.putUInt("configTime", NEW\_CONFIG\_TIME);

}

doc2.clear();

}

doc.clear();

http.end();

}

Serial.flush();

}