

## **SPOTS ( SMART POWER TERMINAL SOCKET )**

### **PROYEK AKHIR**

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Diusulkan Oleh:

EDRY IRAWAN

NPM : 0031709

M. ANANDA AULIA

NPM : 0031746

**POLITEKNIK MANUFAKTUR NEGERI  
BANGKA BELITUNG 2020**

## LEMBAR PENGESAHAN

### SPOTS ( SMART POWER TERMINAL SOCKET )

Oleh:

Edry Irawan :0031709

M.Ananda Aulia :0031746

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan Diploma III Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui:

Pembimbing 1



Surojo, M.T

pembimbing 2



Zanu Saputra, M.Tr.T

Penguji 1



Yudhi, M.T

Penguji 2



Aan Febriansyah, M.T

penguji 3



Charlothia, M.Tr.T

## PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa 1: Edry Irawan

NIRM: 0031709

Nama Mahasiswa 2: MAnanda Aulia

NIRM: 0031746

Dengan Judul : SPOTS ( *Smart Power Terminal Socket* )

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 1 September 2020

Nama Mahasiswa

Tanda tangan

1. Edry Irawan



2. M.Ananda Aulia



## **ABSTRAK**

*Energi listrik merupakan salah satu kebutuhan yang sangat penting dan sebagai sumber daya ekonomis yang paling utama dalam berbagai kegiatan seluruh perangkat elektronik yang ada di dalam rumah yang menggunakan arus listrik dalam penggunaannya dalam sebuah elektronik terdapat alat-alat yang membutuhkan daya listrik untuk kebutuhan primer dari yang watt kecil hingga besar. Proyek akhir ini juga mampu menunda pengaktifan beban dari daya terkecil sampai dengan terbesar untuk meminimalisir adanya akumulasi inrush current yang berpotensi merusak komponen elektronik. Alat ini menggunakan sensor ACS-712 dan sensor ZMPT-101B untuk mengukur arus serta tegangan. Alat ini juga menggunakan driver RELAY SWITCH 4 CHANNEL untuk menjalankan fungsi sebagai monitoring on/off pada socket. Kemudian waktu ada dan tidaknya beban daya akan ditampilkan pada LCD 20x4. Alat ini juga memanfaatkan ESP8266-01 sebagai media komunikasi antara android dan socket dengan memanfaatkan jaringan WiFi. Monitoring ataupun transfer data dapat dilakukan melalui aplikasi android yang didownload di play store bernama BLYNK. Pada fungsi monitoring mengalami error maksimal 1.32% untuk pengukuran tegangan, error maksimal 0.6% untuk pengukuran arus pada beban ricecooker serta error maksimal 1.3% pengukuran arus pada beban setrika.*

*Kata Kunci: Android, inrush current, WIFI , socket, IOT*

## Abstract

*Electrical energy is one of the most important needs and as the most important economic resource in various activities of all electronic devices in the house that use electric current in its use in an electronics, there are devices that require electrical power for the primary needs of the wattage. small to large. This final project is also able to delay load activation from the smallest to the largest power to minimize the accumulation of inrush currents that could potentially damage electronic components. This tool uses the ACS-712 sensor and the ZMPT-101B sensor to measure current and voltage. This tool also uses the RELAY SWITCH 4 CHANNEL driver to function as a monitor on / off on the socket. Then when the power load is present and whether or not it will be displayed on the 20x4 LCD. This tool also utilizes the ESP8266-01 as a communication medium between android and socket by utilizing a WiFi network. Monitoring or data transfer can be done via an android application downloaded on the Play Store called BLYNK. The monitoring function has a maximum error of 1.32% for voltage measurements, a maximum error of 0.6% for current measurements at ricecooker loads and a maximum error of 1.3% for measuring currents at iron loads.*

*Keywords: Android, Inrush Current, WIFI, Socket, IOT*

## KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh. Segala puji bagi Allah Subhanahu wa ta'ala yang telah melimpahkan taufiq dan hidayah-Nya kepada penulis, sehingga penulis dapat menyusun laporan proyek akhir ini dengan Judul “SPOTS ( *Smart Power Terminal Socket* )” dan dapat menyelesaikan Program Studi Teknik Elektro dan Informatika di Politeknik Manufaktur Negeri Bangka Belitung. Sholawat serta salam semoga tercurahkan kepada Nabi Muhammad Shalallaahu 'Alayhi Wasallam beserta keluarga dan sahabat-sahabatnya, serta semoga semua umatnya senantiasa dapat menjalankan syari'at-syari'atnya, Aamiin.

Penulis menyadari sepenuhnya, bahwa dalam penyusunan laporan Proyek Akhir ini banyak kekurangan mengingat terbatasnya kemampuan penulis, namun berkat rahmat Allah Subhanahu wa ta'ala, serta pengarahan dari berbagai pihak, akhirnya laporan proyek akhir ini dapat diselesaikan. Harapan penulis semoga laporan proyek akhir ini dapat bermanfaat untuk kita bersama.

Sehubungan dengan itu, penulis menyampaikan ucapan terima kasih kepada:

1. Ibunda dan Ayahanda tercinta serta seluruh keluarga yang dengan penuh keikhlasan dan kesungguhan hati memberikan bantuan moral dan spiritual yang tak ternilai harganya.
2. I Made Setiawan, M Eng, Ph.D. Selaku Direktur di Politeknik Manufaktur Negeri Bangka Belitung yang telah banyak memberikan kemudahan dalam menyelesaikan pendidikan.
3. Surojo, M.T Selaku dosen pembimbing yang telah membimbing, mengarahkan dan memberi saran-saran dalam pembuatan dan penyusunan laporan proyek akhir ini.

4. Dosen dan Staf Pengajar di Politeknik Manufaktur Negeri Bangka Belitung yang telah mendidik, membina dan mengantarkan penulis untuk menempuh kematangan dalam berfikir dan berperilaku.
5. Teman–teman seperjuangan dan semua pihak yang telah memberikan bantuannya.
6. Sahabat-sahabat yang selalu memberi support selama ini.

Akhir kata penusun ucapkan terima kasih banyak kepada semua pihak yang telah membantu dan semoga Allah Subhanahu wa ta'ala melimpahkan karunianya dalam setiap amal kebaikan kita dan diberikan balasan. Aamiin.

Sungailiat, 1 September 2020



Penulis

## DAFTAR ISI

HALAMAN JUDUL .....	i
LEMBAR PENGESAHAN.....	ii
PERNYATAAN BUKAN PLAGIAT .....	iii
ABSTRAK .....	iv
KATA PENGANTAR .....	vi
DAFTAR ISI .....	viii
DAFTAR TABEL .....	x
DAFTAR GAMBAR .....	xi
DAFTAR LAMPIRAN.....	xiii
BAB 1 PENDAHULUAN .....	1
1.1 Latar belakang Masalah .....	1
1.2 Rumusan masalah .....	2
1.3 Batasan masalah .....	2
1.4 Tujuan Proyek Akhir .....	2
BAB 2 DASAR TEORI .....	1
2.1 Arus Listrik AC .....	1
2.2 Sensor .....	1
2.2.1 Sensor Arus ACS 712 .....	1
2.2.2 Sensor PZMPT10B .....	3
2.3 Arduino .....	4
2.4 Android .....	6
2.5 Modul WIFI ESP8266 01 .....	8
2.6 Relay <i>Switch</i> .....	10
2.7 LCD 20x4 <i>Blue SPI I2C Module</i> .....	11
2.8 Stop kontak .....	12
2.9 Catu daya ( <i>Power Supply</i> ) .....	13
2.10 Daya listrik .....	16
2.11 <i>Software</i> Blynk .....	17



BAB 3 METODE PELAKSANAAN .....	19
3.1 Metode Pembuatan Proyek Akhir .....	19
3.2 Rancangan Algoritma Kerja <i>Hardware</i> .....	21
3.3 Diagram Blok <i>Hardware</i> .....	23
BAB 4 PEMBAHASAN .....	26
4.1 Rancangan Alat .....	26
4.1.1 Memodifikasi Mekanik Stop kontak .....	26
4.1.2 <i>Hardware</i> Elektrik yang Telah Dirakit .....	27
4.2 Rangkaian catu daya .....	28
4.3 Modul WIFI ESP8266 01 .....	30
4.4 Pengujian Sensor arus ACS712.....	30
4.5 Sensor ZMPT10B.....	33
4.6 Prosedur Percobaan .....	34
4.6.1 Pengujian Sensor Dibanding Dengan Alat Ukur .....	36
4.7 <i>Software</i> Spots.....	44
4.7.1 Aplikasi “ BLYNK ” .....	44
4.7.2 Pengujian Aplikasi “ BLYNK ” .....	45
4.8 Pengujian EPROM .....	49
4.9 Rangkaian LED <i>aktif LOW</i> .....	51
BAB 5 KESIMPULAN DAN SARAN .....	52
5.1 Kesimpulan .....	52
5.2 Saran .....	52
DAFTAR PUSTAKA .....	53

## DAFTAR TABEL

Tabel 2.1 Spesifikasi Sensor acs 712 .....	2
Tabel 2.2 Spesifikasi Sensor ZMPT10B .....	3
Tabel 2.3 Spesifikasi Arduino mega 2560 .....	5
Tabel 2.4 Spesifikasi <i>Module</i> WIFI ESP8266-01 .....	9
Tabel 2.5 Spesifikasi Relay 4 <i>channel</i> .....	10
Tabel 2.6 Spesifikasi LCD 20x4 <i>Blue spi i2c module</i> .....	12
Tabel 2.7 Spesifikasi Stop Kontak 3 <i>socket</i> .....	13
Tabel 2.8 Spesifikasi Catu Daya <i>step down</i> .....	15
Tabel 4.1 Kriteria Catu Daya yang Digunakan .....	29
Tabel 4.3 Data pengujian beban <i>ricecooker</i> .....	37
Tabel 4.4 Perbandingan alat ukur dan sensor pada beban <i>ricecooker</i> .....	39
Tabel 4.5 Data pengujian beban Setrika .....	41
Tabel 4.6 Perbandingan alat ukur dan sensor pada beban Setrika .....	43

## DAFTAR GAMBAR

Gambar 2.1 Sensor Arus ACS712 .....	2
Gambar 2.2 Sensor Tegangan ZMPT10B .....	3
Gambar 2.3 Arduino mega 2560 .....	5
Gambar 2.4 Lambang-Lambang Android .....	7
Gambar 2.5 jenis-jenis ESP8266 .....	8
Gambar 2.6 Modul wifi ES8266-01 .....	9
Gambar 2.7 Relay <i>switch 4 channel</i> .....	10
Gambar 2.8 LCD 20x4 SPI <i>blue I2C module</i> .....	12
Gambar 2.9 Stop Kontak 3 <i>socket</i> .....	13
Gambar 2.10 Catu daya <i>step down</i> .....	15
Gambar 2.11 Logo <i>Software</i> Blynk .....	18
Gambar 3.1 Flowchart Pembuatan Proyek Akhir.....	19
Gambar 3.2 Flowchart Algoritma Kerja Alat.....	21
Gambar 3.3 Diagram Blok <i>Hardware</i> Elektrik .....	23
Gambar 3.4 Skematik rangkaian pada SPOTS.....	24
Gambar 4.1 Modifikasi terminal stop kontak 3 Socket .....	27
Gambar 4.2 <i>Hardware</i> elektrik yang telah dirakit.....	28
Gambar 4.3 Gambar catu daya .....	29
Gambar 4.4 Hasil pengujian Esp8266-01 .....	30
Gambar 4.5 <i>Connector Probe</i> .....	34
Gambar 4.6 <i>Clamp Ampere 5A</i> .....	35
Gambar 4.7 Menu <i>set up</i> Aat ukur.....	35
Gambar 4.8 pengujian tegangan pada alat <i>Power Quality Analyer</i> .....	36
Gambar 4.9 Pengujian pada beban <i>ricecooker</i> .....	37
Gambar 4.10 pengujian pada Setrika 320 watt.....	41
Gambar 4.11 Tampilan Utama pada Aplikasi BLYNK.....	45
Gambar 4.12 Menghidupkan Relay 1 .....	46

Gambar 4.13 Menghidupkan relay 2 .....	47
Gambar 4.14 Menghidupkan Relay 3 .....	47
Gambar 4.15 Memaatikan Relay 1 .....	48
Gambar 4.16 Mematikan Relay 2 .....	48
Gambar 4.17 Mematikan Relay 3 .....	49
Gambar 4.18 Pengujian Eprom .....	50
Gambar 4.19 Rangkaian LED <i>aktif low</i> .....	51

## **DAFTAR LAMPIRAN**

Lampiran 1: Daftar Riwayat Hidup

Lampiran 2: Program Utama

# **BAB 1 PENDAHULUAN**

## **1.1 Latar belakang Masalah**

Energi listrik merupakan salah satu kebutuhan yang sangat penting dan sebagai sumber daya ekonomis yang paling utama dalam berbagai kegiatan seluruh perangkat elektronik yang ada didalam rumah yang menggunakan arus listrik dalam penggunaannya dalam sebuah elektronik terdapat alat-alat yang membutuhkan daya listrik untuk kebutuhan primer dari yang watt kecil hingga besar dapat dilihat dari saat ini banyaknya peralatan tersebut, bermacam-macam, mulai dari peralatan memasak, peralatan kantor, hingga penerangan dan sebagainya dalam penggunaannya, masalah yang sering dihadapi adalah saat berpergian kadang kita lupa untuk mematikan alat elektronik yang tidak terlalu diperlukan walau sebagian sesuai dengan kebutuhan yang memang harus menyala 24 jam.

Ketika kondisi PLN dalam keadaan mati dan menyala kembali maka perangkat elektronik tersebut akan menyala secara bersamaan. Dimana sering terjadi akumulasi Inrush Current yang sering kali mengakibatkan trip pada MCB (*Miniature circuit Breaker*) diakibatkan banyaknya perangkat yang menyala secara bersamaan seperti yang biasa ditemukan setelah pemadaman listrik oleh PLN. Pada kondisi tersebut akan terjadi dimana arus akan terputus seketika yang dikhawatirkan dapat mengakibatkan rusaknya komponen-komponen pada alat-alat elektronik dan sebagainya. dari permasalahan tersebut munculah ide kami untuk membuat sebuah alat yang berfungsi untuk meminimalisir adanya arus yang melonjak sesaat yang bekerja dengan memanfaatkan besar daya setiap socket yang digunakan sebagai acuan dalam waktu tunda pengaktifan.

## 1.2 Rumusan masalah

Adapun rumusan masalah yang diangkat berdasarkan latar belakang proyek akhir ini antara lain:

1. Bagaimana menampilkan tegangan, arus dan daya setiap socket terminal pada LCD dan Android ?
2. Bagaimana mengatur penundaan pengaktifan setiap socket terminal ?
3. Bagaimana cara mengetahui nilai daya terakhir pemakaian setiap socket terminal sebelum mati lampu ?
4. Bagaimana merancang dan membuat alat yang menunda pengaktifan berdasarkan nilai daya?

## 1.3 Batasan masalah

Adapun batasan masalah pada proyek akhir ini adalah sebagai berikut:

1. Alat tidak bisa dikontrol jika mati lampu, karena *supply* utama menggunakan PLN.
2. Socket yang digunakan dalam pengujian alat sebanyak 3 socket.
3. Pengujian menggunakan alat-alat kebutuhan rumah tangga.

## 1.4 Tujuan Proyek Akhir

Adapun tujuan pembuatan proyek akhir yang berjudul “SPOTS ( *Smart Power Terminal Socket* )” ini adalah:

1. Membuat alat SPOTS yang bekerja dengan menunda waktu aktif berdasarkan daya terkecil sampai terbesar yang tersimpan pada EPROM.
2. Dapat meminimalisir *inrush current* yang berpotensi merusak komponen alat elektronik dan daya dapat dimonitoring di LCD dan Android.

## **BAB 2 DASAR TEORI**

### **2.1 Arus Listrik AC**

*Alternating Current* atau yang biasa disingkat AC merupakan tipe arus listrik bolak-balik. Ide mengenai arus AC dikembangkan oleh Nikola Tesla yang bekerjasama dengan perusahaan Westinghouse dan digunakan secara komersil pada pertengahan abad 20-an. *Alternating Current* atau yang biasa disebut dengan arus bolak balik, adalah arus listrik yang nilainya berubah-ubah terhadap satuan waktu. Sumber arus AC yang paling umum adalah berasal dari induksi elektromagnetik yaitu dari generator AC yang secara eksklusif dioperasikan oleh Perusahaan Listrik Negara (PLN) ataupun dari generator portabel (genset AC). Penggunaan arus AC yang paling umum adalah pada rumah tangga, di mana arus AC dimanfaatkan sebagai sumber energi untuk menyalakan perangkat-perangkat elektronik seperti televisi, lampu rumah dan lain sebagainya. [1]

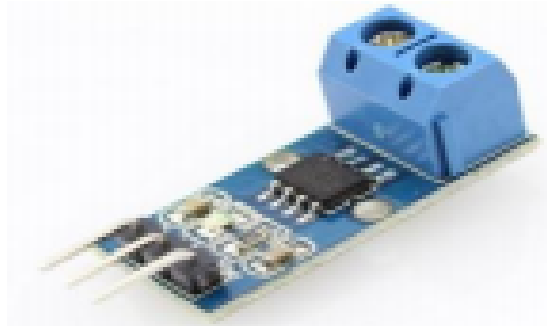
### **2.2 Sensor**

Sensor adalah suatu peralatan yang berfungsi untuk mendeteksi gejala-gejala atau sinyal-sinyal yang berasal dari perubahan suatu energi seperti energi listrik, energi fisika, energi kimia, energi biologi, energi mekanik dan sebagainya. [2]

#### **2.2.1 Sensor Arus ACS 712**

Sensor arus yang digunakan merupakan modul ACS712 untuk mendeteksi besar arus yang mengalir lewat blok terminal. Sensor ini dapat mengukur arus *positif* dan *negatif* dengan kisaran -30A sampai 30A. Sensor ini memerlukan suplai daya sebesar 5V. Untuk membaca nilai tengah (nol Ampere) tegangan sensor diset pada 2.5V yaitu setengah kali tegangan sumber daya  $VCC = 5V$ . Pada polaritas negatif pembacaan arus -30A terjadi pada tegangan 0,5V. Tingkat perubahan tegangan berkorelasi *linear* terhadap besar arus sebesar 400 mV/Ampere. [3] sensor acs712 pada gambar 2.1 sebagai berikut:





Gambar 2.1 Sensor Arus ACS712

Spesifikasi sensor acs712 pada tabel berikut [4]:

Tabel 2.1 Spesifikasi Sensor acs 712

NO	SPESIFIKASI
1.	Jalur sinyal analog dengan <i>noise</i> rendah
2.	<i>Bandwidth</i> perangkat diatur melalui pin <i>FILTER</i> baru
3.	5 $\mu$ s waktu naik keluaran sebagai <i>respons</i> terhadap arus masukan langkah
4.	<i>Bandwidth</i> 50 kHz
5.	Kesalahan total output 1,5% pada TA = 25 ° C, dan 4% pada -40 ° C hingga 85 ° C
6.	Jejak kecil, paket SOIC8 sederhana
7.	Resistensi konduktor <i>internal</i> 1,2 M $\omega$
8.	Tegangan isolasi <i>minimum</i> 2,1 kVRMS dari pin 1-4 ke pin 5-8
9.	5,0 V, operasi suplai tunggal
10.	<i>Sensitivitas</i> keluaran 66 hingga 185 mV / A
11.	Tegangan keluaran sebanding dengan arus AC atau DC
12.	Dipangkas dari pabrik untuk akurasi
13.	Tegangan <i>offset</i> keluaran yang sangat stabil
14.	Histeresis magnetis hampir nol
15.	Keluaran rasio dari tegangan suplai

### 2.2.2 Sensor PZMPT10B

Sensor Tegangan ZMPT101B adalah suatu modul sensor tegangan AC yang menggunakan trafo isolasi dengan rasio tegangan 1:1. Sensor Tegangan ZMPT101B memiliki kelebihan antara lain Transformer ukuran kecil, akurasi tinggi, konsistensi yang baik, untuk tegangan dan pengukuran daya. Modul sensor tegangan ZMPT101B mampu mengukur hingga 250 VAC. Sensor ini dilengkapi dengan trim multi-putar potensiometer untuk menyesuaikan output ADC. [5] ZMPT101B ditunjukkan pada gambar 2.2



Gambar 2.2 Sensor Tegangan ZMPT10B

Spesifikasi sensor ZMPT10B pada tabel berikut [6]:

Tabel 2.2 Spesifikasi Sensor ZMPT10B

NO	SPESIFIKASI
1.	ZMPT101B modul transformator tegangan AC fase tunggal keluaran aktif.
2.	Sirkuit op amp presisi onboard, sinyal untuk pengambilan sampel yang tepat dan fungsi kompensasi yang sesuai.
3.	Dapat diukur dalam tegangan AC 250V yang sesuai dengan output analog dapat disesuaikan.
4.	Sinyal keluaran adalah gelombang sinus, dapat digunakan untuk pengambilan sampel AD
5.	Ukuran papan PCB: 50 x 19mm

6.	Nilai <i>Input</i> Saat Ini: 2mA
7.	Retardasi ( <i>input</i> terukur): 20 ( <i>input</i> 2mA, resistansi sampling 100)
8.	Rentang linier: 0 ~ 1000V
9.	Tegangan isolasi: 4000V
10.	Tegangan suplai (VCC): 5 - 30VDC
11.	Tegangan keluaran: VCC / 2

### 2.3 Arduino

Arduino adalah sebuah *platform* komputasi fisik *open source* berbasiskan Rangkain *input/output* sederhana (I/O) dan lingkungan pengembangan yang mengimplementasikan bahasa *Processing*. Arduino dapat digunakan untuk mengembangkan obyek interaktif mandiri atau dapat dihubungkan ke perangkat lunak pada komputer anda (seperti Flash, Pengolahan, VVVV, atau Max / MSP). [3]

Arduino Mega2560 Revisi 3 memiliki fitur-fitur baru berikut:

- a. Ditambahkan pin SDA dan pin SCL yang dekat dengan pin AREF dan dua pin baru lainnya ditempatkan dekat dengan pin RESET, IOREF memungkinkan shield untuk beradaptasi dengan tegangan yang tersedia pada papan. Di masa depan, shield akan kompatibel baik dengan papan yang menggunakan AVR yang beroperasi dengan 5 Volt dan dengan Arduino Due yang beroperasi dengan tegangan 3.3 Volt. Dan ada dua pin yang tidak terhubung, yang disediakan untuk tujuan masa depan.
- b. Sirkuit RESET.
- c. Chip ATmega16U2 menggantikan chip ATmega8U2.

Papan Arduino mega2560 dapat beroperasi dengan pasokan daya eksternal 6 Volt sampai 20 volt. Jika diberi tegangan kurang dari 7 Volt, maka, pin 5 Volt mungkin akan menghasilkan tegangan kurang dari 5 Volt dan ini akan membuat papan menjadi tidak stabil. Jika sumber tegangan menggunakan lebih dari 12 Volt, regulator tegangan akan mengalami panas berlebihan dan bisa merusak papan. Rentang sumber tegangan yang dianjurkan adalah 7 Volt sampai 12 Volt. [7]

Arduino ATmega2560 memiliki 256 KB flash memory untuk menyimpan kode (yang 8 KB digunakan untuk bootloader), 8 KB SRAM dan 4 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM). Arduino mega 2560 ditunjukkan pada gambar 2.3 [7]



Gambar 2.3 Arduino mega 2560

Spesifikasi Arduino mega 2560 pada tabel sebagai berikut:

Tabel 2.3 Spesifikasi Arduino mega 2560

NO	SPESIFIKASI	KETERANGAN
1.	Digital I/O Pins	54 ( <i>of which 15 provide PWM output</i> )
2.	Analog Input Pins	16
3.	DC Current per I/O Pin	40 mA
4.	DC Current for 3.3V Pin	50 mA
5.	Flash Memory	256 KB <i>of which 8 KB used by bootloader</i>
6.	SRAM	8 KB

7.	EEPROM	4 KB
8.	<i>Clock Speed</i>	16 MHz

## 2.4 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux (Nasrudin Safaat H., 2011, h.1). Android menyediakan *platform* terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. [8] Secara garis besar, arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut [9]:

- a) *Applications* dan *Widgets Applications* dan *Widgets* ini adalah layer dimana berhubungan dengan aplikasi saja, di mana biasanya download aplikasi dijalankan kemudian dilakukan instalasi dan jalankan aplikasi tersebut.
- b) *Applications Frameworks Applications frameworks* ini adalah *layer* di mana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi Android, karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *contact providers* yang berupa sms dan panggilan telepon
- c) *Libraries Libraries* ini adalah *layer* di mana tur-tur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan di atas kernel, *Layer* ini meliputi berbagai *library* C/C++ inti seperti Libc dan SSL.
- d) *Android Run Time Layer* yang membuat aplikasi *Android* dapat dijalankan dimana dalam prosesnya menggunakan Implementasi Linux.

- e) Linux Kernel Linux Kernel adalah layer di mana inti dari *operating system* dari Android itu berada. Berisi *system* yang mengatur sistem *processing*, *memory*, *resource*, *drivers*, dan sistem sistem operasi android lainnya. Linux kernel yang digunakan *android* adalah linux kernel release 2.6. Eclipse Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (*platform independent*). Berikut ini adalah sifat dari Eclipse:
- f) *Multi platform*: target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
- g) *Multi language*: Eclipse dikembangkan dengan bahasa pemrograman Java. Akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya
- h) *Multi role*: selain sebagai IDE untuk pengembangan aplikasi, Eclipse bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, *test* perangkat lunak, pengembangan web dan lain sebagainya. Lambang-lambang Android
- Lambang android ditunjukkan pada gambar 2.4 berikut [10]:



Gambar 2.4 Lambang-Lambang Android

## 2.5 Modul WIFI ESP8266 01

Modul WiFi ESP8266 adalah modul mandiri dengan terintegrasi protokol TCP / IP yang dapat memberikan akses mikrokontroler ke jaringan WiFi. Setiap modul ESP8266 diprogram dengan firmware set perintah AT, yang dapat terhubung ke Arduino untuk mendapatkan atau menghubungkan ke WiFi dengan kemampuan sebagai WiFi *Shield*. [11]

ESP 8266 adalah sebuah *chip* yang sudah lengkap dimana didalamnya sudah termasuk *processor*, memori dan juga akses ke GPIO. Hal ini menyebabkan ESP8266 dapat secara langsung menggantikan Arduino dan ditambah lagi dengan kemampuannya untuk *men-support* koneksi wifi secara langsung. IoT (*Internet Of Things*) semakin berkembang seiring dengan perkembangan mikrokontroler, module yang berbasis Ethernet maupun wifi semakin banyak dan beragam dimulai dari *Wiznet*, Ethernet *shield* hingga yang terbaru adalah Wifi module yang dikenal dengan ESP8266. Ada beberapa jenis ESP8266 yang dapat ditemui dipasaran, namun yang paling mudah didapatkan di Indonesia adalah *type* ESP-01,07,dan 12 dengan fungsi yang sama perbedaannya terletak pada GPIO pin yang disediakan. [12] Berikut beberapa tipe ESP8266 pada gambar 2.5 berikut:



Gambar 2.5 jenis-jenis ESP8266

Tegangan kerja ESP-8266 adalah sebesar 3.3V, sehingga untuk penggunaan mikrokontroler tambahannya dapat menggunakan board arduino yang memiliki fasilitas tegangan sumber 3.3V, akan tetapi akan lebih baik jika membuat secara terpisah level shifter untuk komunikasi dan sumber tegangan untuk wifi module ini. [12] dan disini penulis menggunakan ESP-8266-01 pada gambar 2.6 berikut:



Gambar 2.6 Modul wifi ES8266-01

Spesifikasi *module* wifi ESP8266-01 pada tabel berikut [13]:

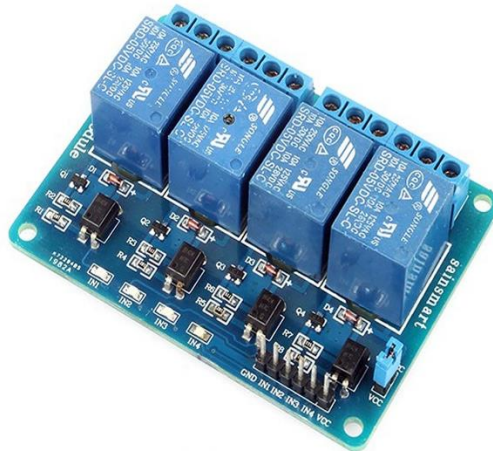
Tabel 2.4 Spesifikasi Module WIFI ESP8266-01

NO	SPESIFIKASI
1.	802.11 b / g / n
2.	MCU 32-bit daya rendah terintegrasi
3.	ADC 10-bit terintegrasi
4.	Tumpukan protokol TCP / IP terintegrasi
5.	WiFi 2,4 GHz, mendukung WPA / WPA2
6.	Mendukung Fungsi Tautan Cerdas untuk perangkat Android dan iOS
7.	SDIO 2.0, (H) SPI, UART, I2C, I2S, IR <i>Remote Control</i> , PWM, GPIO
8.	STBC, 1 × 1 MIMO, 2 × 1 MIMO
9.	Agregasi A-MPDU & A-MSDU & <i>interval</i> pelindung 0,4 detik
10.	Daya keluaran +20 dBm dalam mode 802.11b



## 2.6 Relay Switch

Relay adalah Saklar (*Switch*) yang dioperasikan secara listrik dan merupakan komponen Electromechanical (Elektromekanikal) yang terdiri dari 2 bagian utama yakni Elektromagnet (*Coil*) dan Mekanikal (seperangkat Kontak Saklar/*Switch*) Relay menggunakan Prinsip Elektromagnetik untuk menggerakkan Kontak Saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi. [14] disini penulis menggunakan jenis Relay *module 4 channel* gambar sebagai berikut [15]:



Gambar 2.7 Relay switch 4 channel

Spesifikasi relay 4 channel pada tabel berikut [15]:

Tabel 2.5 Spesifikasi Relay 4 channel

NO	SPESIFIKASI
1.	<i>Input</i> relay 5V DC
2.	Maksimum <i>load</i> 250VAC/10A 30VDC/10A
3.	Dilengkapi dengan optocoupler <i>isolation</i> untuk melindungi <i>board</i> microcontroller dari tegangan AC
4.	Memiliki LED <i>indicator</i>
5.	Menggunakan terminal <i>block</i> sehingga pemasangan kabel menjadi mudah

- 
6. *output* keluaran 4 channel maksimal 10A *Output* memiliki 3 pin terminal *block* yang ditandai dengan NO, COM dan NC. NO (*Normally Open*) = Tidak ada arus yang dialirkan (*OFF*), Jika ada signal *High / Low* dari microcontroller maka ON COM (*Common*) = Sumber tegangan yang akan dihubungkan (Bisa arus AC maupun DC 10A *max*) NC (*Normally Close*) = Arus dialirkan (*ON*), Jika ada *signal High / Low* dari microcontroller maka *OFF*
- 

## 2.7 LCD 20x4 Blue SPI I2C Module

LCD adalah lapisan dari campuran organik antara lapisan kaca bening dengan elektroda transparan indium oksida dalam bentuk tampilan *seven segment* dan lapisan elektroda pada kaca belakang. Ketika elektroda diaktifkan dengan medan listrik (tegangan). Lapisan *sandwich* memiliki polarizer cahaya *vertikal* depan dan polarizer cahaya *horizontal* belakang yang diikuti dengan lapisan reflektor. Cahaya yang dipantulkan tidak dapat melewati molekul-molekul yang telah menyesuaikan diri dan segmen yang diaktifkan terlihat menjadi gelap dan membentuk karakter data yang ingin ditampilkan. [16]

*Inter Integrated Circuit* atau sering disebut I2C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didesain khusus untuk mengirim maupun menerima data. Sistem I2C terdiri dari saluran SCL (*Serial Clock*) dan SDA (*Serial Data*) yang membawa informasi data antara I2C dengan pengontrolnya. Piranti yang dihubungkan dengan sistem I2C Bus dapat dioperasikan sebagai *Master* dan *Slave*. *Master* adalah piranti yang memulai *transfer* data pada I2C Bus dengan membentuk sinyal *Start*, mengakhiri *transfer* data dengan membentuk sinyal *Stop*, dan membangkitkan sinyal *clock*. *Slave* adalah piranti yang dialamati *master*. [16] Gambar LCD 20x4 Blue spi i2c module dapat dilihat pada gambar 2.8 berikut:



Gambar 2.8 LCD 20x4 SPI *blue I2C module*

Spesifikasi LCD 20x4 *Blue spi i2c module* dapat dilihat pada tabel berikut [17]:

Tabel 2.6 Spesifikasi LCD 20x4 *Blue spi i2c module*

NO	SPESIFIKASI
1.	<i>I2C</i> lampu latar biru
2.	Format Tampilan 20 Karakter x 4 baris
3.	Tegangan suplai 5V
4.	Biru menyala kembali dengan warna putih
5.	Tegangan suplai 5V
6.	Ukuran Pcb 60mm99mm
7.	Potensiometer Pengaturan <i>Kontras</i>
8.	Lampu Latar Atur <i>Jumper</i>
9.	Tegangan kerja VCC, GND, DO, AO

## 2.8 Stop kontak

Stop kontak adalah sebuah alat pemutus ketika terjadi kontak antara arus *positif*, arus *negatif* dan *grounding* pada instalasi listrik. Dan yang lebih penting lagi ELCB bisa memutuskan arus listrik ketika terjadi kontak antara listrik dan tubuh manusia. Umumnya bila peralatan listrik bekerja normal maka total arus yang mengalir pada kawat “*plus*” dan “*netral*” adalah sama sehingga tidak ada

perbedaan arus. Namun bila seseorang tersengat listrik, kawat “*plus*” akan mengalirkan arus tambahan melewati tubuh orang yang tersengat ke tanah. [18]  
Gambar 2.9 jenis stop kontak 3 socket sebagai berikut :



Gambar 2.9 Stop Kontak 3 *socket*

Tabel 2.7 Spesifikasi Stop Kontak 3 *socket*

NO	SPESIFIKASI
1.	<i>Safety - Quality – Durability</i>
2.	16A / 220 - 250V
3.	Pemasangan mudah dan lebih praktis
4.	Material berkualitas dan tahan panas

### 2.9 Catu daya (*Power Supply*)

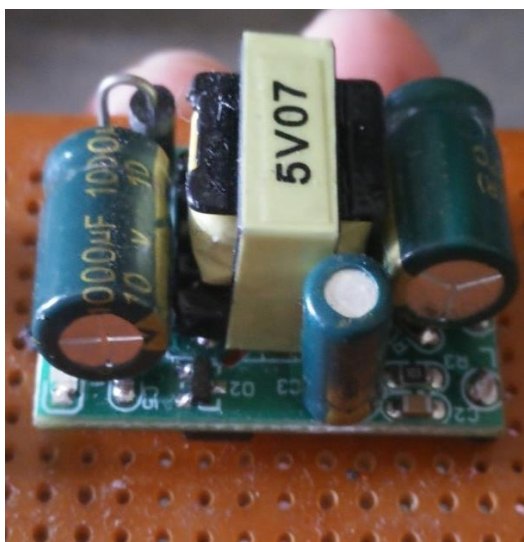
Catu daya atau sering disebut dengan *Power Supply* adalah perangkat elektronika yang berguna sebagai sumber daya untuk perangkat lain. Secara umum istilah catu daya berarti suatu sistem penyearah-*filter* yang mengubah ac menjadi dc murni. Sumber DC seringkali dapat menjalankan peralatan-peralatan elektronika secara langsung, meskipun mungkin diperlukan beberapa cara untuk meregulasi dan menjaga suatu ggl agar tetap meskipun beban berubah-ubah. Energi yang paling mudah tersedia adalah arus bolak-balik, harus diubah atau disearahkan menjadi dc berpulsa, yang selanjutnya harus diratakan atau disaring menjadi tegangan yang tidak berubah-ubah. [19]

Tegangan dc juga memerlukan regulasi tegangan agar dapat menjalankan rangkaian dengan sebaiknya. Secara garis besar, pencatu daya listrik dibagi menjadi dua macam, yaitu pencatu daya tak distabilkan dan pencatu daya distabilkan. Pencatu daya tak distabilkan merupakan jenis pencatu daya yang paling sederhana. Pada pencatu daya jenis ini, tegangan maupun arus keluaran dari pencatu daya tidak distabilkan, sehingga berubah-ubah sesuai keadaan tegangan masukan dan beban pada keluaran. Pencatu daya jenis ini biasanya digunakan pada peranti elektronika sederhana yang tidak sensitif akan perubahan tegangan. Pencatu jenis ini juga banyak digunakan pada penguat daya tinggi untuk mengkompensasi lonjakan tegangan keluaran pada penguat. [19]

Pencatu daya distabilkan pencatu jenis ini menggunakan suatu mekanisme lolos balik untuk menstabilkan tegangan keluarannya, bebas dari variasi tegangan masukan, beban keluaran, maupun dengung. Ada dua jenis yang digunakan untuk menstabilkan tegangan keluaran, antara lain [19]:

1. Pencatu daya linier, merupakan jenis pencatu daya yang umum digunakan. Cara kerja dari pencatu daya ini adalah mengubah tegangan AC menjadi tegangan AC lain yang lebih kecil dengan bantuan Transformator. Tegangan 7 ini kemudian disearahkan dengan menggunakan rangkaian penyearah tegangan, dan di bagian akhir ditambahkan kondensator sebagai penghalus tegangan sehingga tegangan DC yang dihasilkan oleh pencatu daya jenis ini tidak terlalu bergelombang. Selain menggunakan diode sebagai penyearah, rangkaian lain dari jenis ini dapat menggunakan regulator tegangan linier sehingga tegangan yang dihasilkan lebih baik daripada rangkaian yang menggunakan dioda.
2. Pencatu daya Sakelar, pencatu daya jenis ini menggunakan metode yang berbeda dengan pencatu daya linier. Pada jenis ini, tegangan AC yang masuk ke dalam rangkaian langsung disearahkan oleh rangkaian penyearah tanpa menggunakan bantuan transformer. Cara

menyearahkan tegangan tersebut adalah dengan menggunakan frekuensi tinggi antara 10KHz hingga 1MHz, dimana frekuensi ini jauh lebih tinggi daripada frekuensi AC yang sekitar 50Hz. Pada pencatu daya sakelar biasanya diberikan rangkaian umpan balik agar tegangan dan arus yang keluar dari rangkaian ini dapat dikontrol dengan baik. Disini penulis menggunakan catu daya *Step down* pada proyek akhir kami sebagai berikut:



Gambar 2.10 Catu daya *step down*

Spesifikasi Catu daya 5V07 pada tabel berikut:

Tabel 2.8 Spesifikasi Catu Daya *step down*

NO	SPESIFIKASI
1.	Modul <i>Step Down</i>
2.	Tegangan <i>input</i> AC 85 ~ 265 v 50/60 hz atau DC 100 ~ 370 v
3.	Arus masuk 0.0273A (AC110V) 0.014A (AC220V)
4.	<i>Input</i> Arus Masuk 20A
5.	Tegangan keluaran DC 5V (+ / - 0,2 V)
6.	Arus keluaran 700 mA
7.	Daya 3,5 W.

8. Suhu pengoperasian -20 ~ 60 °C

---

9. Kelembaban relatif 40-90% RH

---

10. *Output* 5V, arus min 0A, arus *min* 700mA, arus puncak: 800mA, kisaran keluaran: 4.8 ~ 5.2v, gelombang riak: 60mV  
Daya keluaran: 0-4W (arus DC)

---

11. Efisiensi keluaran: 80%

---

12. Waktu kenaikan tegangan keluaran: *MAX* 100MS

---

13. Ukuran: 3 x 2 x1.8cm

---

### 2.10 Daya listrik

Temy Nusa dalam jurnal Teknik Elektro dan Komputer Juli 2018, Universitas Sam Ratulangi yang berjudul Sistem *Monitoring* Konsumsi Energi Listrik Secara *Real Time* Berbasis Mikrokontroler mengatakan bahwa daya listrik merupakan laju hantaran energi listrik dalam rangkaian listrik. Daya listrik terbagi menjadi tiga jenis, yaitu daya aktif, daya reaktif dan daya semu [20].

Daya Aktif / Nyata (*Active / Real Power*) Daya dengan satuan Joule/detik atau watt disebut sebagai daya aktif. Simbolnya adalah P. Daya aktif adalah daya sebenarnya yang dihamburkan atau dipakai oleh beban. Daya aktif dihitung dengan persamaan 2.1 [20].

$$P = V \times I \times \cos \varphi \quad \dots\dots\dots (2.1)$$

Keterangan : V = Tegangan, *volt*

I = Arus, *ampere*

$\varphi$  = Sudut phi, derajat

Daya reaktif (Q) merupakan daya yang diperlukan untuk pembentukan medan magnet. Daya reaktif juga dipahami daya yang tidak dihamburkan oleh beban atau dengan kata lain merupakan daya yang diserap namun dikembalikan ke

sumbernya. Satuan daya reaktif adalah VAR (*Voltampere – reactive*). Daya reaktif dapat dihitung dengan persamaan 2.2 [20]

$$Q = V \times I \times \sin \phi \quad \dots\dots\dots (2.2)$$

Keterangan :  $V$  = Tegangan, *volt*

$I$  = Arus, *ampere*

$\phi$  = Sudut phi, derajat

Daya tampak atau daya semu ( $S$ ) merupakan hasil dari penjumlahan trigonometri daya aktif dan reaktif. Satuan daya nyata adalah VA (*Voltampere*). Daya semu dapat dihitung dengan persamaan 2.3 [20]

$$S = V \times I \quad \dots\dots\dots (2.3)$$

Keterangan :  $V$  = Tegangan, *volt*

$I$  = Arus, *ampere*

### **2.11 Software Blynk**

Blynk merupakan aplikasi yang berbasis IOS atau Android untuk mengontrol mikrokontroler berupa Arduino melalui internet. Aplikasi Blynk dapat membantu admin dalam memonitoring sesuatu dengan praktis. Blynk dirancang untuk Internet of Things. Dapat mengontrol perangkat keras dari jarak jauh, dapat menampilkan data sensor, dapat menyimpan data, memvisualisasikannya, dan melakukan banyak hal lainnya. *Software Blynk* pada gambar 2.11 berikut: [21] [22]





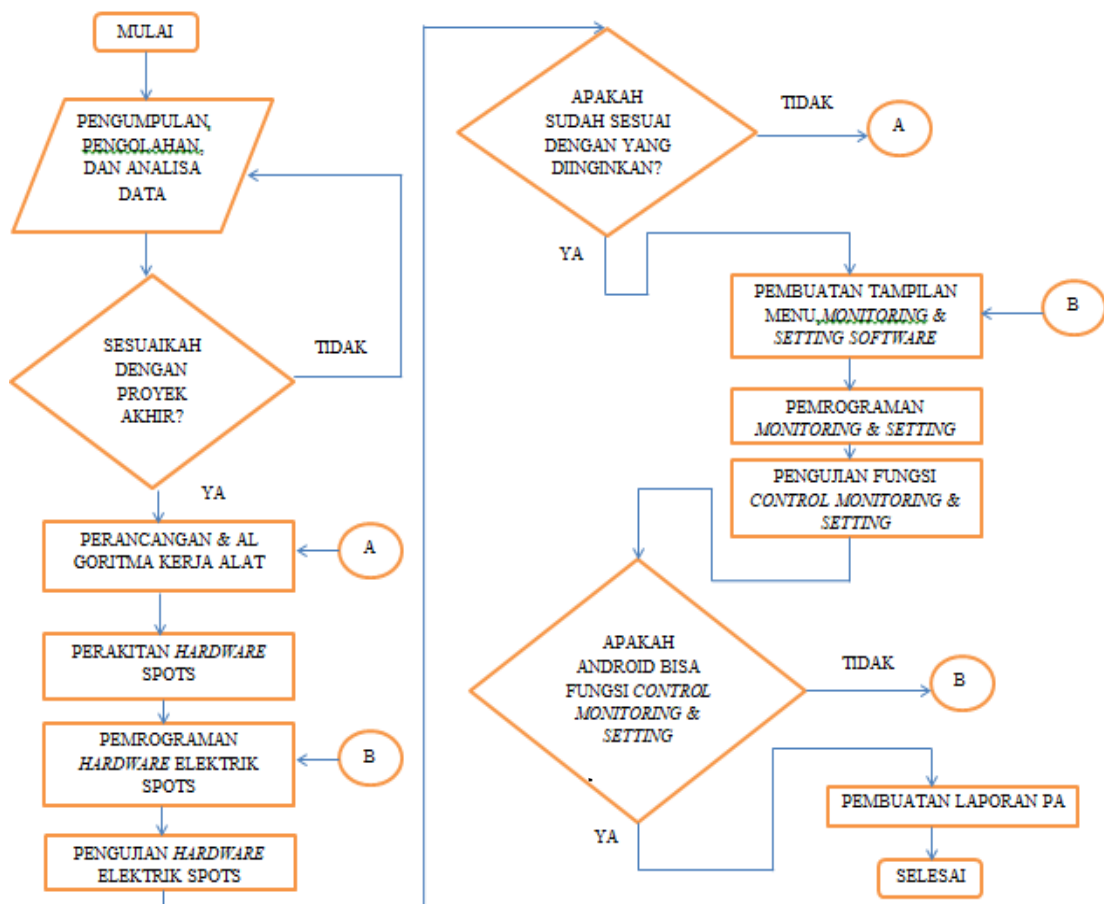
Gambar 2.11 Logo Software Blynk

### BAB 3 METODE PELAKSANAAN

Dalam pembuatan proyek akhir dengan judul “SPOTS (*Smart Power Terminal Socket*)”. ini dilakukan beberapa metode penelitian yang bertujuan untuk mempermudah dalam penyelesaian proyek akhir. Adapun metode penelitian yang diterapkan dalam pembuatan proyek akhir seperti berikut:

#### 3.1 Metode Pembuatan Proyek Akhir

Proses pembuatan proyek akhir dari mulai sampai selesainya proyek akhir dijelaskan pada *flowchart* pembuatan proyek akhir. *Flowchart* pembuatan Proyek ditunjukkan pada Gambar 3.1 sebagai berikut:



Gambar 3.1 Flowchart Pembuatan Proyek Akhir

Gambar 3.1 Merupakan *flowchart* pembuatan proyek akhir yang menjelaskan alur dari proses pengerjaan proyek akhir. Hal yang pertama dilakukan proses pembuatan proses akhir yaitu pengumpulan data yang diperlukan dalam menyelesaikan proyek akhir. Proses pengumpulan data dapat dilakukan secara langsung dan secara tidak langsung. Maksud dari pengumpulan data secara langsung merupakan pengumpulan data yang diperoleh dari konsultasi dengan dosen pembimbing sedangkan untuk pengumpulan data secara tidak langsung merupakan data yang terinspirasi dari Anak ITB yang berjudul “SPOS ( *Smart Power socket* )” [23], dan dari referensi – referensi dari penelitian yang masih berhubungan dengan proyek akhir yang ingin dibuat. Data - data yang didapatkan dari proses pengumpulan data selanjutnya diolah dan dianalisa. Data yang telah diolah dan dianalisa dikumpulkan lalu dipilih untuk menjadi referensi dan acuan dalam pembuatan proyek akhir.

Perancangan *hardware* SPOTS dan perancangan algoritma kerja alat dibuat agar *hardware* dan prinsip kerja alat sesuai dengan spesifikasi proyek akhir. Proses perakitan *hardware* elektrik SPOTS harus sesuai spesifikasi dan koneksi pin dari dari *hardware* elektrik yang kita gunakan. Pembuatan program *hardware* elektrik SPOTS dibuat pada *software* Arduino IDE. Pembuatan program *hardware* elektrik Spots harus sesuai dengan koneksi pin dan harus memenuhi spesifikasi proyek akhir yang ingin dibuat. Setelah pembuatan program *hardware* elektrik SPOTS selesai, maka dilakukan pengujian, yang dimana tujuannya untuk mengetahui apakah *hardware* elektrik Spots sudah sesuai dengan spesifikasi yang diinginkan.

Pembuatan *software* untuk memenuhi spesifikasi dari proyek akhir dibuat menggunakan BLYNK yang diakses menggunakan *browser* internet dan bisa didownload di *play store*. Pembuatan tampilan menu dilakukan pada bagian *designer* BLYNK. Pembuatan tampilan menu kontrol, *monitoring*, dan *setting* harus mudah dipahami serta digunakan dan tentunya tetap memenuhi nilai fungsi dari *software* yang akan dibuat. Pembuatan program menu

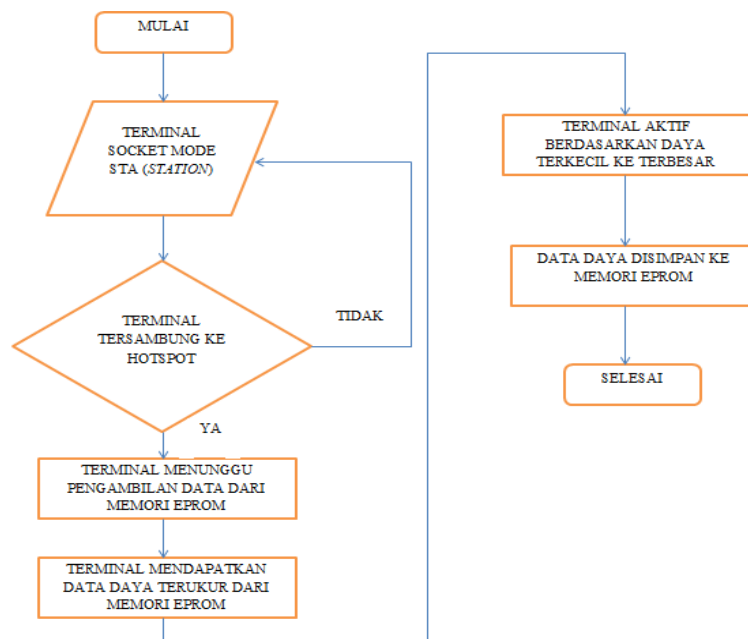
kontrol, *monitoring*, dan *setting software* SPOTS dilakukan pada bagian *bloks* blynk.

Pengujian fungsi kontrol, *monitoring* dan *setting software* SPOTS dilakukan dengan cara menginstall *software* yang telah dibuat ke Android. Pengujian fungsi dilakukan secara keseluruhan, baik dari segi kontrol, *monitoring* dan *setting software* harus dapat bekerja sesuai dengan fungsinya.

Proses pembuatan laporan merupakan proses terakhir dalam pembuatan proyek akhir. Pembuatan laporan bertujuan merangkum keseluruhan yang berhubungan dengan proyek akhir, seperti latar belakang, tujuan, rumusan masalah, batasan masalah, teori dasar, metode penelitian, pembahasan serta kesimpulan dan saran.

### 3.2 Rancangan Algoritma Kerja *Hardware*

Rancangan algoritma kerja *hardware* merupakan rancangan dari proses cara kerja *hardware* untuk memenuhi spesifikasi yang diinginkan. Berikut adalah *flowchart* rancangan algoritma kerja *hardware* ditunjukkan pada Gambar 3.2 berikut:



Gambar 3.2 Flowchart Algoritma Kerja Alat

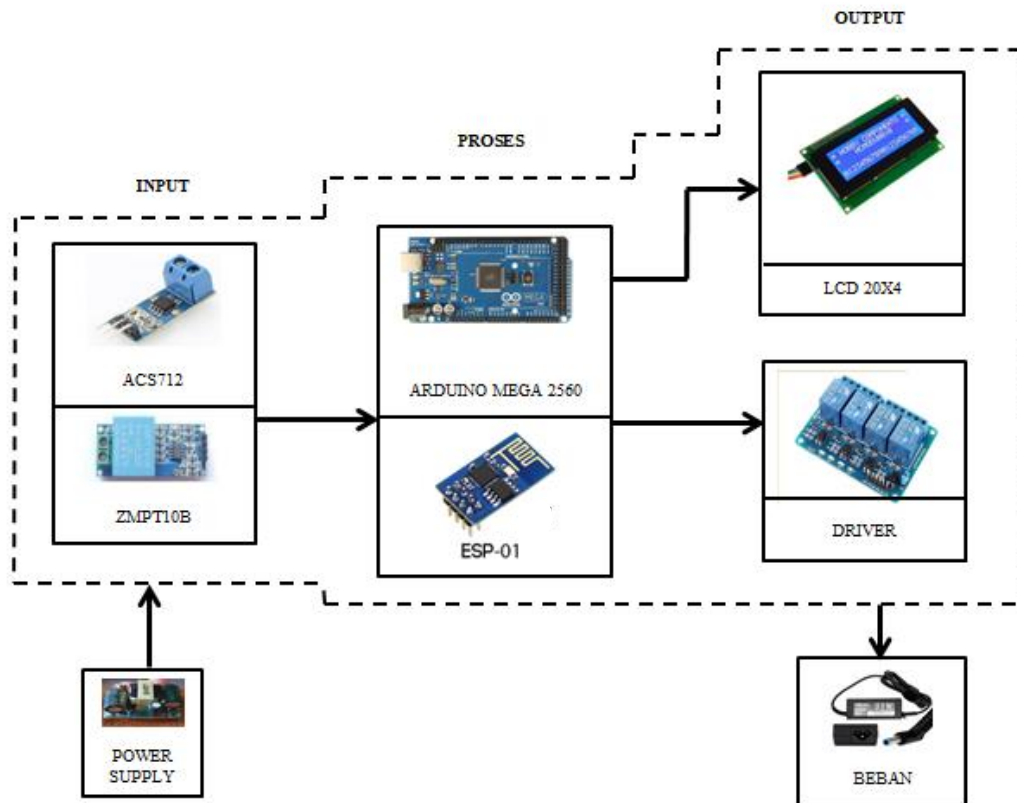
Pada Gambar 3.2 merupakan *flowchart* rancangan algoritma kerja alat yang digunakan dalam mencapai target penyelesaian sesuai dengan spesifikasi yang diinginkan. Dalam keadaan normal, maka sensor akan berfungsi sebagai membaca daya Saat dalam *mode* terhubung ke PLN. Arduino bertugas mengolah data daya yang dibaca oleh sensor dan menyimpan data yang terbaca dimemori aduino ( EPROM ) yang akan menjadi data acuan untuk pengaktifan *socket* dari daya terkecil hingga terbesar. Modul wifi menghubungkan ke *hotspot* yang sesuai dengan SSID (*Service Set Identifier*) dan program yang telah dibuat. *driver* bisa diaktifkan berdasarkan daya yang diolah dan bisa juga akses melalui *hotspot* yang SSID dan kata sandinya sesuai dengan program, namun jika hubungan antara *driver* dan *hotspot* terputus, maka *driver* tidak bisa diakses melalui wifi dan akan melakukan pencarian *hotspot* sampai berhasil terhubung kembali ke *hotspot* yang sesuai. Ketika *driver* terhubung ke *hotspot*, maka *driver* akan menunggu instruksi yang akan dikirim dari *hotspot* yang telah terhubung.

Saat *driver* mendapatkan instruksi maka *socket* akan aktif akan melakukan pengerjaan sesuai dengan instruksi yang didapat. Jika *socket* menerima instruksi dan telah *stand by* untuk menunggu beban dengan daya yang terkecil untuk diaktifkan terlebih dahulu hingga terbesar. maka *socket* akan mengaktifkan beban yang memiliki daya yang terkecil – terbesar dan menampilkannya ke lcd. Saat *socket* telah mendapatkan *beban* yang dimana beban tersebut memiliki *watt* terendah hingga terbesar, maka lampu indikator akan menjadi acuan aktifnya sebuah *socket* dan ketika disebuah *socket* beban tidak ada maka akan *me-reset* . *arduino* bertugas mengirimkan instruksi ke *driver* yang tersambung sesuai dengan instruksi dari Android.

Namun *driver* menerima instruksi dengan tidak melakukan instruksi ke *socket* maka *driver* akan melakukan tugas sesuai dengan instruksi yang diterima dan setelah selesai melakukan tugasnya maka *driver* kembali menunggu instruksi. Proses akan tetap terus berlanjut dan kembali ke proses mulai.

### 3.3 Diagram Blok *Hardware*

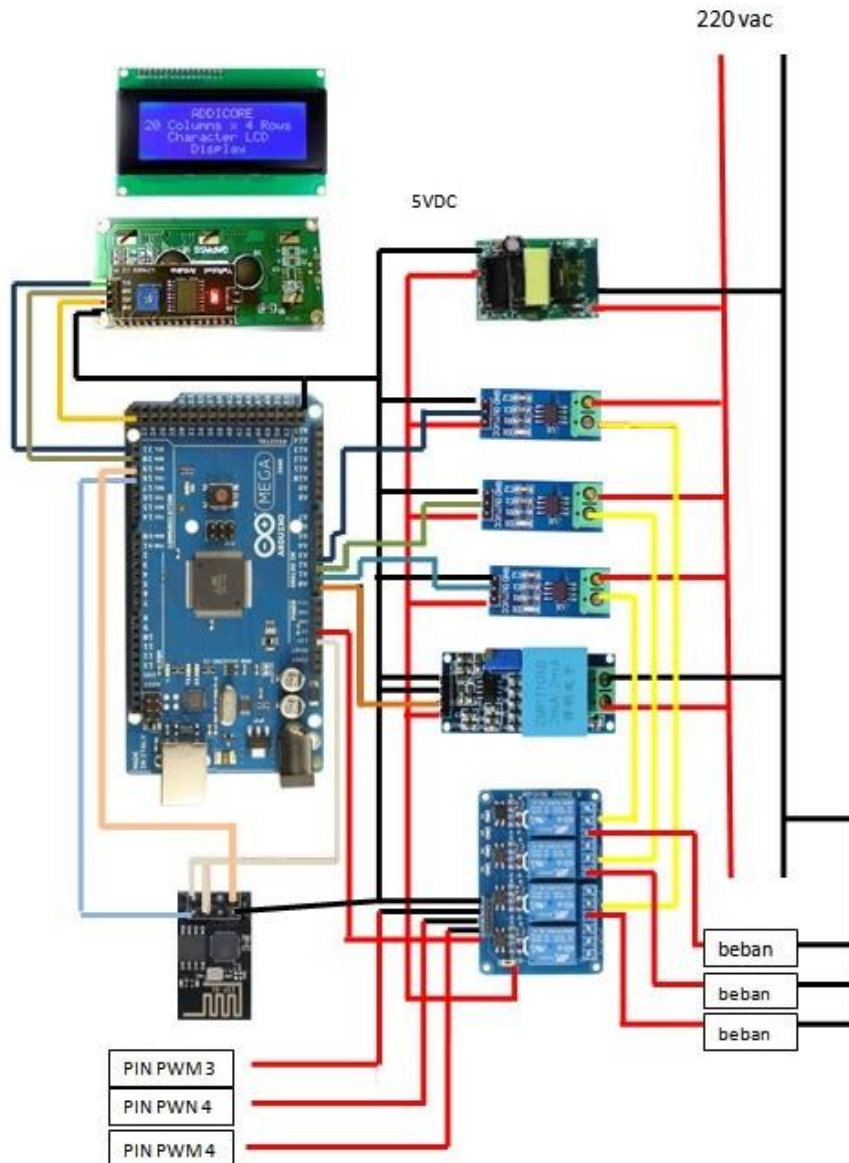
Setiap SPOTS yang dibuat memiliki *hardware* elektrik yang sama, sehingga SPOTS lampu mempunyai diagram blok *hardware* setiap yang sama. Berikut diagram blok *hardware* ditunjukkan pada Gambar 3.3 berikut:



Gambar 3.3 Diagram Blok *Hardware* Elektrik

Proses pengambilan data arus, tegangan dan daya dilakukan dengan menggunakan sensor ACS712 dan ZMPT10B. Data yang didapatkan pada kedua sensor tersebut lalu diolah oleh Arduino mega 2560 menjadi sebuah data *logger* sehingga arus, tegangan dikonversikan menjadi nilai daya yang menjadi data acuan untuk pengaktifan kembali beban berdasarkan daya terecil hingga terbesar. Dari hasil pengolahan data yang didapat selanjutnya ditampilkan pada Android serta LCD dan dikirimkan pada aplikasi pada Android yang telah dibuat dengan *software* blynk melalui jaringan WiFi

dengan menggunakan kontroler *Module* WIFI ESP8266-01. Android dapat menerima data serta mengirimkan perintah pada Arduino untuk men-*trigger* Relay sehingga dapat mematikan dan menghidupkan *socket* yang digunakan.berikut adalah skematik SPOTS ditunjukkan pada Gambar 3.4 berikut:



Gambar 3.4 Skematik rangkaian pada SPOTS

Pada dasarnya prinsip kerja Gambar 3.4 sama halnya dengan diagram blok Gambar 3.3 untuk mengetahui arus, tegangan serta daya pada listrik AC maka

digunakan sensor ACS712 dan ZMPT10B yang keluarannya melalui Analog A0-A3 pada Arduino mega 2560. selanjutnya untuk mengetahui aktifnya Socket digunakan Relay *switch* yang dikethui *supply* dari catu daya *step down* dan *output* dari Relay *switch* ke terminal stop kontak dengan komunikasi data yang terhubung dengan ESP-01. ini juga digunakan untuk *monitoring ON/OFF* pada relay dan akan *monitoring* daya pada LCD 20x4 Blue spi I2c. sensor yang membaca daya yang terukur akan menyimpan pada memori EPROM yang akan menjadi acuan untuk pengaktifan Socket pada proyek akhir ini.



## BAB 4 PEMBAHASAN

Pada bab ini akan dibahas mengenai proses pembuatan dan pengujian alat serta metode yang dipakai dalam pembuatan proyek akhir yang berjudul “SPOTS (*Smart Power Terminal Socket*)”, yaitu sebagai berikut:

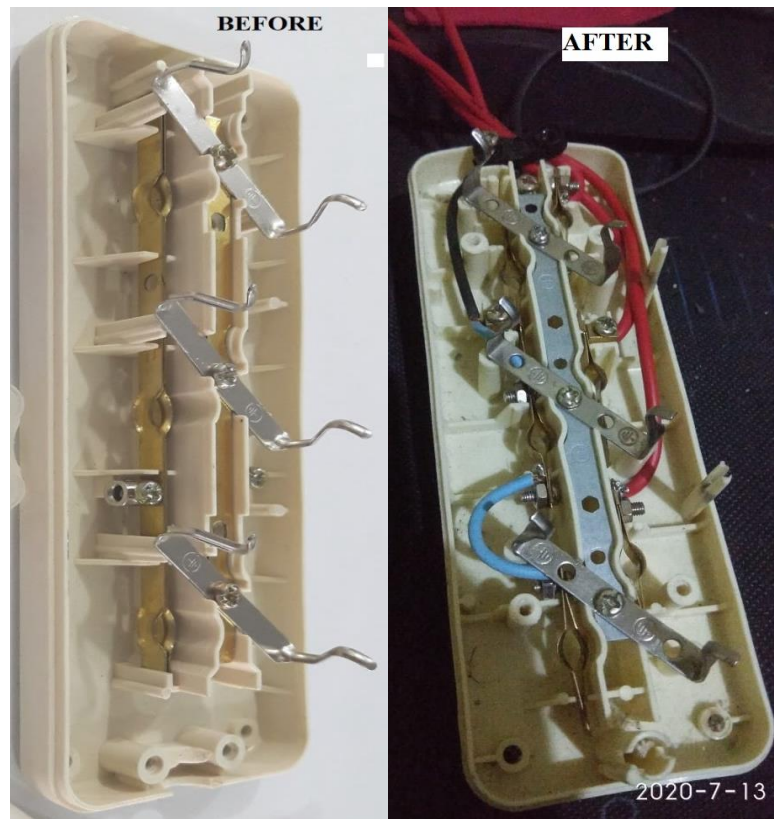
### 4.1 Rancangan Alat

Untuk memastikan alat yang dibuat sesuai dengan rumusan masalah dilakukan proses perancangan alat pada *hardware* mekanik maupun *hardware* elektrik. Hal ini sangat dibutuhkan karena selain memaksimalkan fungsi dari alat ini juga agar terdapat kesesuaian antara *hardware* mekanik dan *hardware* elektrik.

#### 4.1.1 Memodifikasi Mekanik Stop kontak

Dalam proses pembuatan *hardware* SPOTS dilakukan dengan pembelian stop kontak yang dijual dipasaran. Terdapat banyak jenis stop kontak yang dijual dipasaran. Banyak jenis dari stop kontak dari 1 *socket* sampai lebih dari 3 *socket*. Dari stop kontak yang didapat kemudian dimodifikasi dengan menambahkan 3 kabel fasa untuk ke *output* setiap Relay, bagian *output* Relay bisa mengaktifkan *socket* sesuai dengan nilai daya yang terbaca.

Dari terminal yang didapat kemudian di *modifikasi* dengan membuat atau memisah jalur fasa di setiap *socket*. Sehingga setiap socket memiliki jalur fasa tersendiri yang bisa di kontrol secara manual. *Hardware* mekanik versi 3 socket yang dijual di pasaran dan akan di gunakan pada proyek akhir ini.

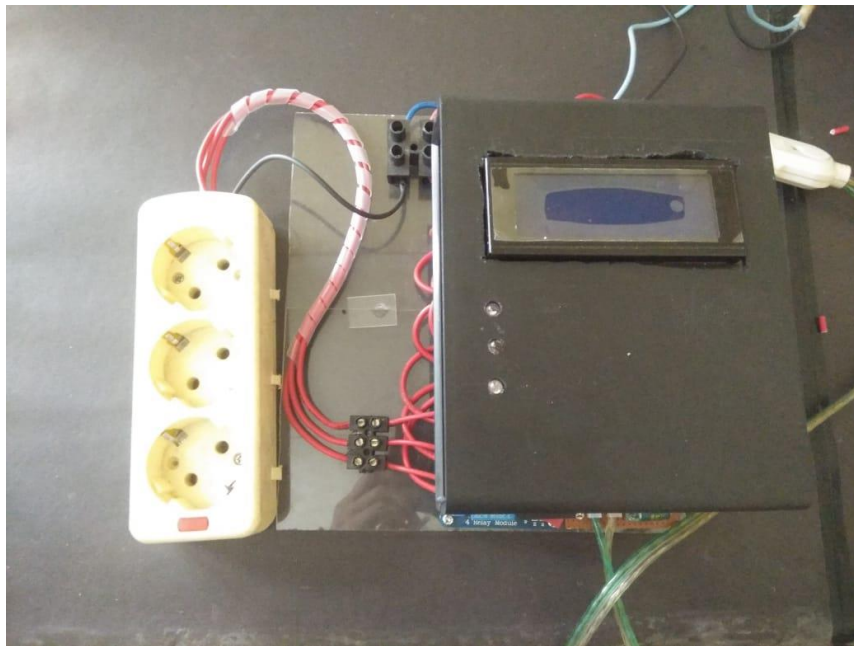


Gambar 4.1 Modifikasi terminal stop kontak 3 *Socket*

Pada Gambar 4.1 Berikut adalah terminal stop kontak 3 socket yang telah dimodifikasi menjadi sebuah rangkaian paralel yang akan digunakan pada proyek akhir ini.

#### **4.1.2 *Hardware* Elektrik yang Telah Dirakit**

Proses lain yang penting pada proyek akhir ini adalah perancangan, perakitan dan percobaan terhadap *hardware* elektrik. Pada sub bab ini akan dibahas *hardware* elektrik yang akan digunakan. Berikut gambar *hardware* elektrik yang telah dirakit pada Gambar 4.2 berikut:



Gambar 4.2 *Hardware* elektrik yang telah dirakit

Setelah selesai *hardware* elektrik yang telah penulis rakit, selanjutnya mengkoneksikan tiap jalur *ouput* relay ke tiap socket.

#### **4.2 Rangkaian catu daya**

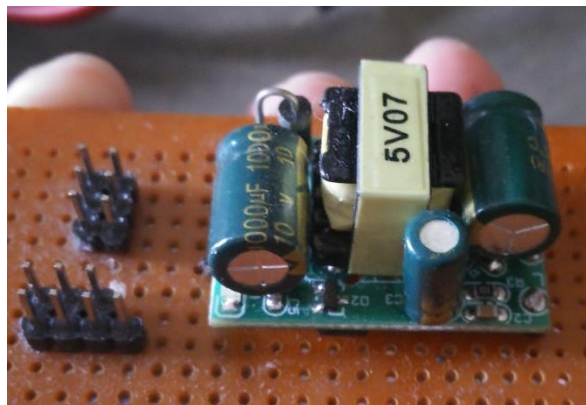
Rangkaian catu daya yang digunakan dalam pembuatan proyek akhir ini mempertimbangkan pemakaian arus total pada rangkaian kontrol yang digunakan. Rangkaian catu daya ini juga mempertimbangkan masalah dimensi karena harus menyesuaikan dengan besarnya dimensi alat yang digunakan.

Dalam pembuatan rangkaian catu daya pada proyek akhir ini diputuskan untuk membeli rangkaian catu daya yang sudah jadi dengan memperhitungkan dengan segala kebutuhan rangkaian kontrol yang digunakan. Catu daya yang digunakan sebanyak 2 buah. Dimana catu daya pertama dengan *output* 5v digunakan untuk *supply* modul relay dan *supply* vcc pada sensor ZMPT101B. selain itu juga digunakan adaptor dengan *output* 9V 2A yang digunakan sebagai *supply* arduino, ACS712, LCD 20x04 dan modul wifi esp8266-01. Setelah dilakukan pengujian didapat kriteria yang diinginkan dan ditunjukkan pada Tabel 4.1 berikut:

Tabel 4.1 Kriteria Catu Daya yang Digunakan

NO	KRITERIA	ALASAN
1.	<i>Input AC~220V</i>	Karena mendapatkan dari sumber PLN AC~220V
2.	<i>Output ±5V</i>	Agar sesuai dengan catu daya yang diizinkan pada rangkaian kontrol.
3.	Arus minimum 700 mA	Agar rangkaian kontrol tidak <i>me-reset</i>

Setelah melakukan pengamatan Tabel 4.1 maka didapatkan kriteria catu daya yang akan digunakan dengan input 220 VAC dan *output* dengan tegangan maksimal  $\pm 5V$  dan arus *output* minimal 700 mA. Rangkaian catu daya yang digunakan pada proyek akhir ini dapat ditunjukkan pada Gambar 4.3 berikut:



Gambar 4.3 Gambar catu daya

### 4.3 Modul WIFI ESP8266 01

Sebelum pengambilan data disini penulis ingin memastikan terlebih dulu modul Esp8266-01 dapat berfungsi baik dan sesuai dengan kinerja yang diinginkan maka dilakukan pengujian menghidupkan LED dengan menggunakan output Esp8266-01. Berikut adalah hasil pengujian terdapat pada gambar berikut:



Gambar 4.4 Hasil pengujian Esp8266-01

### 4.4 Pengujian Sensor arus ACS712

Sensor arus adalah perangkat yang mendeteksi arus listrik (AC atau DC) di kawat, dan menghasilkan sinyal sebanding dengan itu. Sinyal yang dihasilkan bisa tegangan analog atau arus atau bahkan digital. Hal ini dapat kemudian digunakan untuk menampilkan arus yang akan diukur dalam ammeter atau dapat disimpan untuk analisis lebih lanjut dalam sistem akuisisi data atau dapat dimanfaatkan untuk tujuan kontrol.[3].

Sensor ini digunakan untuk mengukur arus pada beban yang digunakan. Pada pengujian sensor ini yang dipakai adalah tipe sensor ACS712 dengan range 0 – 30 A. Pada pengujian sensor dengan cara melakukan

pengukuran tegangan, arus AC. Berikut adalah *list* program pengujian pada sensor ACS712:

```
double Voltage = 0;

double VRMS = 0;

double AmpsRMS = 0;

double I=0;

void setup(){
Serial.begin(9600);
}

//SENSOR ARUS

void loop(){
Voltage = getVPP();
VRMS = (Voltage/2.0) *0.707;
AmpsRMS = (VRMS * 1000)/mVperAmp;
I = AmpsRMS-0.17;
}

float getVPP()
{
float result;

int readValue; //value read from the sensor
int maxValue = 0; // store max value here
```

```

int minValue = 1024; // store min value here

uint32_t start_time = millis();

while((millis()-start_time) < 1000) //sample for 1 Sec
{
readValue = analogRead(sensorIn);

// see if you have a new maxValue
if (readValue > maxValue)
{
/*record the maximum sensor value*/
maxValue = readValue;
}

if (readValue < minValue)
{
/*record the maximum sensor value*/
minValue = readValue;
}

}

// Subtract min from max

result = ((maxValue - minValue) * 5.0)/1024.0;

return result;
}

```

*Coding program* yang dibuat akan mendapatkan nilai IRMS. Pada program tersebut digunakan teknik sampling sekurang-kurangnya selama 25 ms

yang mengacu pada periode 1 gelombang tegangan AC 220V sebesar 20 ms. Karena *output* dari sensor ACS712 berupa nilai analog berupa gelombang. Data yang telah didapat dibandingkan dengan alat ukur *Kew Kyoritsu Power Quality Analyzer*.

#### 4.5 Sensor ZMPT10B

*Single Phase AC Voltage sensor ZMPT101B* adalah modul untuk mengukur tegangan AC 220V 1 phase. Modul ini memiliki builtin transformer *step down AC to AC 2mA* tipe ZMPT101B dan rangkaian komparator untuk mengubah besar tegangan AC menjadi lebih rendah (*arduino/microcontroller compatible*) sehingga aman untuk dihubungkan ke Arduino atau *microcontroller* anda. *Output* modul ini adalah analog, bisa anda hubungkan ke *input* analog (ADC) di *arduino/microcontroller*. [6]

Pada pengujian sensor dengan cara melakukan pengukuran tegangan, arus AC. Berikut adalah *list* program pengujian pada sensor ZMPT10B:

```
#include "EmonLib.h" // Include Emon
Library

EnergyMonitor emon1; // Create an instance

void setup()

{

  Serial.begin(9600);

  emon1.voltage(2, 520, 1.1); // Voltage: input pin,
calibration, phase_shift

}

void loop()
```



```

{
    emon1.calcVI(25,1500); // Calculate all.
    No.of half wavelengths (crossings), time-out

    emon1.serialprint(); // Print out all
    variables (realpower, apparent power, Vrms, Irms, power
    factor)}

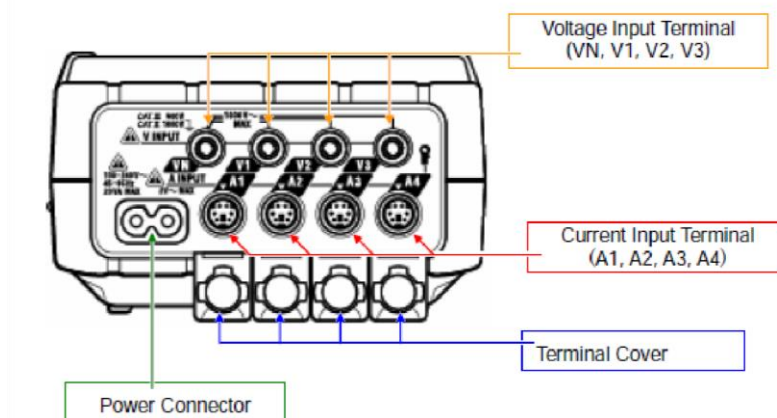
```

#### 4.6 Prosedur Percobaan

Untuk menguji kebenaran nilai arus, tegangan dan daya pada alat yang dibuat maka dilakukan perbandingan antara alat yang dibuat dengan alat ukur yang digunakan. Berikut langkah-langkah yang perlu diperhatikan dalam melakukan pengukuran dengan alat ukur.

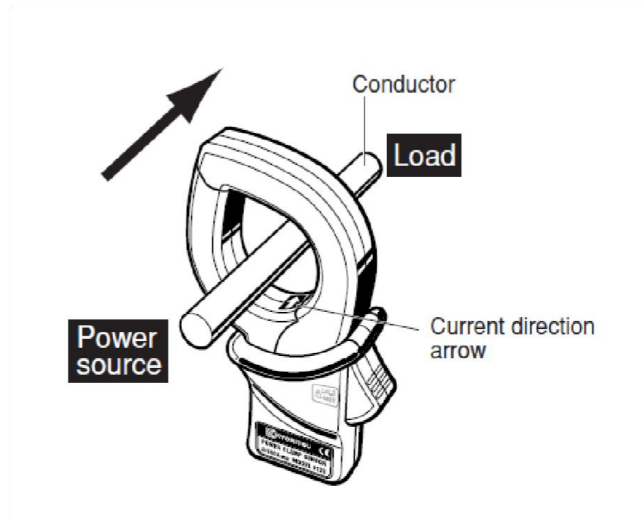
1. Pastikan alat yang dibuat telah siap untuk diuji.
2. Siapkan alat ukur *Kew Kyoritsu Power Quality Analyzer*
3. hubungkan kabel netral pada VN di alat ukur dan kabel fasa pada V1. Kemudian hubungkan pada beban yang akan diukur.

*Kew Kyoritsu Power Quality Analyzer* ditunjukkan pada Gambar 4.5



Gambar 4.5 *Connector Probe*

- Pasang klem meter pada beban. Cara memasang klem meter yang benar ditunjukkan pada Gambar 4.6



Gambar 4.6 Clamp Ampere 5A

Klem meter yang digunakan adalah klem meter yang mempunyai *range* arus dari 0-5A. Pada klem meter terdapat tanda panah yang digunakan sebagai acuan posisi klem meter tersebut. Klem meter dipasang pada kabel fasa dengan arah panah menuju beban.

#### 5. Setting alat ukur Kew Kyoritsu Power Quality Analyzer



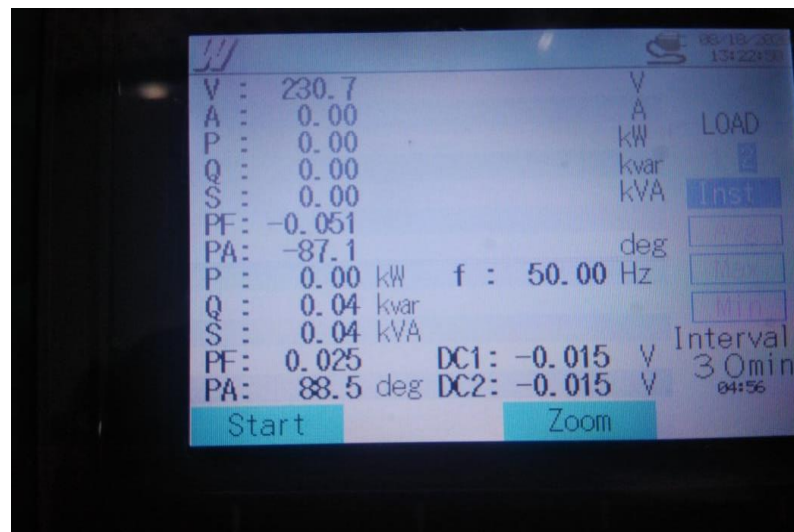
Gambar 4.7 Menu *set up* Alat ukur

Setelah selesai melakukan pengaturan pada alat ukur, tekan tombol *start* pada alat ukur. Maka tampilan pada alat ukur langsung dapat menampilkan arus, tegangan dan daya.

6. Bandingkan hasil pengukuran menggunakan sensor ACS712 & ZMPT10B dengan alat ukur *Kew Kyoritsu Power Quality Analyzer*.

#### 4.6.1 Pengujian Sensor Dibanding Dengan Alat Ukur

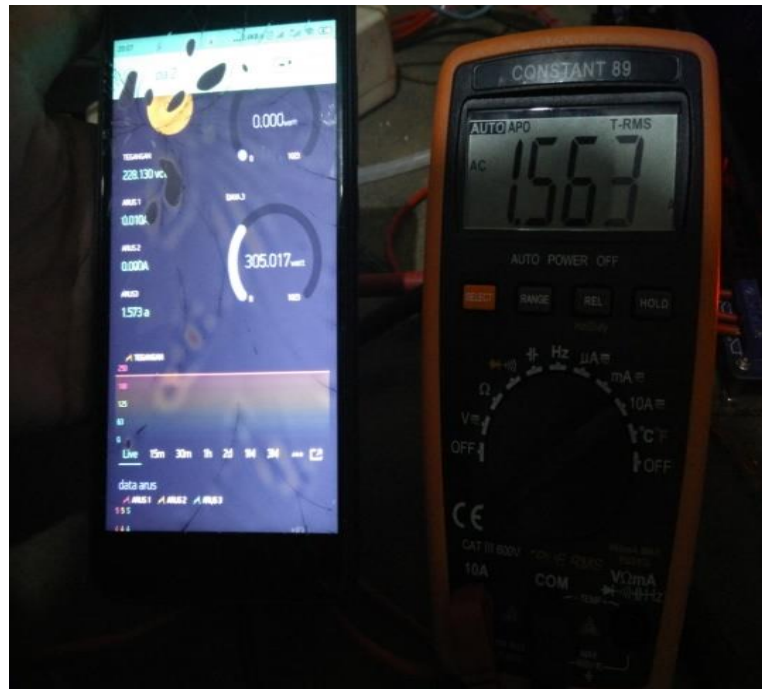
Untuk menguji kepresisian alat yang telah dibuat dalam mengukur nilai tegangan, arus dan daya yang digunakan, maka dibandingkan dengan alat ukur *Kew Kyoritsu Power Quality Analyzer*. Berikut hasil pengujian yang dilakukan pada beban yang berbeda.



Gambar 4.8 pengujian tegangan pada alat *Power Quality Analyer*

- a. Hasil Pengujian dengan beban *ricecooker* 300 watt

Pengujian dilakukan dengan menggunakan beban *ricecooker* hasil pengujian dapat dilihat pada gambar 4.9 berikut:



Gambar 4.9 Pengujian pada beban *ricecooker*

Dari pengukuran yang diperoleh terdapat *range* atau selisih pada alat ukur dan sensor. Untuk melihat besarnya selisih yang didapat pada sensor dan alat ukur dapat dilihat pada Tabel 4.2 berikut:

Tabel 4.2 Data pengujian beban *ricecooker*

PENGUJIAN KE	VUK	VA	IUK	IA	SA	SUK
1	229	228	1.563	1.57	357.96	357.92
2	229	225	1.564	1.52	342	358.15
3	229	228	1.561	1.57	357.96	357.46
RATA-RATA	229	227	1.563	1.55	352.32	357.84

Keterangan:

VUK = nilai tegangan yang terbaca pada alat ukur

VA = nilai tegangan yang terbaca pada alat yang dibuat

IUK = nilai arus yang terbaca pada alat ukur

IA = nilai arus yang terbaca pada alat yang dibuat

SA = nilai daya semu perhitungan pada alat yang dibuat

SUK = Nilai daya semu dialat ukur

PtUK = daya aktif dari alat ukur secara perhitungan

Pt = daya aktif dari alat yang dibuat secara perhitungan

PFt = nilai *power factor*

Berdasarkan data pada tabel 4.2 dapat dihitung Pft adalah:

$$\begin{aligned} \text{PFt} &= \frac{\text{Daya } \textit{ricecooker}}{\text{Tegangan rata – rata alat ukur} \times \text{ arus rata – rata alat ukur}} \\ &= \frac{300 \text{ watt}}{229 \text{ V} \times 1.563 \text{ A}} = 0.815 \end{aligned}$$

Dari Pft yang didapat maka dapat dihitung besarnya Pt (daya secara perhitungan) yang digunakan yaitu:

$$\begin{aligned} \text{Pt} &= \text{VA} \times \text{IA} \times \text{PFt} \\ &= 227 \times 1.55 \times 0.815 \\ &= 286.75 \text{ watt} \end{aligned}$$

$$\begin{aligned} \text{PtUK} &= \text{VUK} \times \text{IUK} \times \text{PFt} \\ &= 229 \times 1.563 \times 0.815 \end{aligned}$$

$$= 291.71 \text{ watt}$$

$$SA = VA \times IA$$

$$= 227 \times 1.55$$

$$= 351.85 \text{ VA}$$

$$SUK = VUK \times IUK$$

$$= 229 \times 1.563$$

$$= 357.92 \text{ VA}$$

Data yang didapat dari alat ukur dan sensor selanjutnya dibandingkan dan dapat dilihat pada Tabel 4.3 berikut:

Tabel 4.3 Perbandingan alat ukur dan sensor pada beban *ricecooker*

Ket	PFt	VUK	VA	IUK	IA	SA	SUK	Pt
SAT		(V)	(V)	(A)	(A)	(VA)	(VA)	(W)
RATA-RATA	0,831	229	227	1.563	1.55	352.32	357.84	286.75
% ERROR		0.8%		0.6%		1.5%		

Keterangan :

VUK = Tegangan rata-rata alat ukur

VA = Tegangan rata-rata alat yang dibuat

IUK = Arus rata-rata alat ukur

IA = Arus rata-rata alat yang dibuat

Pt = daya secara perhitungan

SUK = Daya semu pada alat ukur

SA = Daya semu pada alat yang dibuat

Perhitungan persentase *error* tegangan, arus dan daya yang didapat merupakan perbandingan pada alat ukur dan alat yang dibuat, sehingga rumusnya adalah:

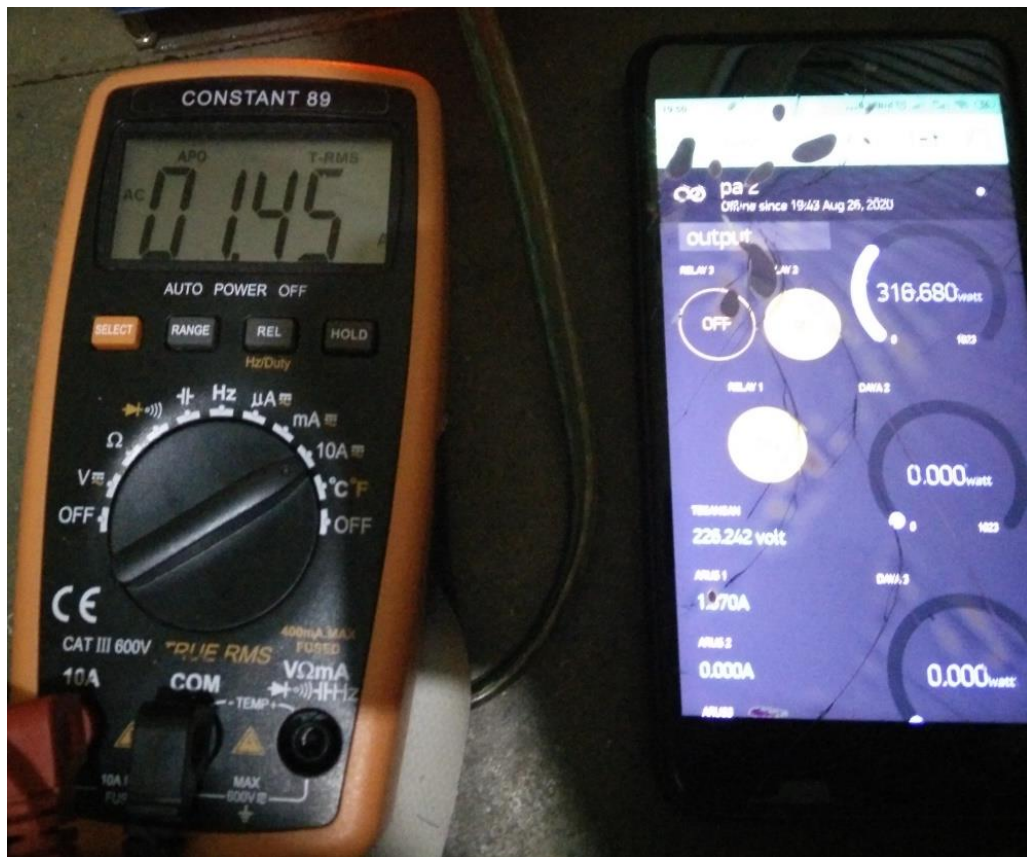
- Persentase *error* tegangan =  $\frac{229-227}{229} \times 100\%$   
=0.8%

- Persentase *error* arus =  $\frac{1.56-1.55}{1.56} \times 100\%$   
=0.6%

- Persentase *error* daya semu =  $\frac{357.84-352.32}{357.84} \times 100\%$   
=1.5%

b. Hasil pengujian dengan beban Setrika 320 watt

Pengujian dilakukan menggunakan beban Setrika berikut hasil dari pengujian ditunjukkan pada gambar 4.10 berikut:



Gambar 4.10 pengujian pada Setrika 320 watt

Dari pengukuran yang diperoleh terdapat *range* atau selisih pada alat ukur dan sensor. Untuk melihat besarnya selisih yang didapat pada sensor dan alat ukur dapat dilihat pada Tabel 4.4 berikut:

Tabel 4.4 Data pengujian beban Setrika

PENGUJIAN KE	VUK	VA	IUK	IA	SA	SUK
1	230	224.79	1.44	1.41	316.95	331.2
2	230	231.169	1.45	1.42	335.19	333.5
3	230	226.24	1.44	1.43	323.52	331.2
RATA-RATA	230	227.39	1.44	1.42	325.22	331.9



Keterangan :

- VUK = nilai tegangan yang terbaca pada alat ukur
- VA = nilai tegangan yang terbaca pada alat yang dibuat
- IUK = nilai arus yang terbaca pada alat ukur
- IA = nilai arus yang terbaca pada alat yang dibuat
- SA = nilai daya semu yang terbaca pada alat yang dibuat
- SUK = nilai daya semu pada alat ukur
- PtUK = daya aktif dari alat ukur secara perhitungan
- Pt = daya aktif dari alat yang dibuat secara perhitungan
- PFt = *Power factor*

Berdasarkan data pada tabel 4.4 dapat dihitung P<sub>Ft</sub> adalah:

$$\begin{aligned}
 P_{Ft} &= \frac{\text{Daya Beban Setrika}}{\text{Tegangan rata – rata alat ukur} \times \text{ arus rata – rata alat ukur}} \\
 &= \frac{320 \text{ watt}}{230 \text{ v} \times 1.44 \text{ A}} = 0.96
 \end{aligned}$$

dari P<sub>Ft</sub> yang didapat maka dapat dihitung besarnya P<sub>t</sub> (daya secara perhitungan) yang digunakan yaitu:

$$\begin{aligned}
 P_t &= VA \times IA \times P_{Ft} \\
 &= 230 \times 1.42 \times 0.96 \\
 &= 313.53 \text{ watt}
 \end{aligned}$$

$$\begin{aligned}
 P_{tUK} &= VUK \times IUK \times P_{Ft} \\
 &= 230 \times 1.44 \times 0.96 \\
 &= 308.016 \text{ watt}
 \end{aligned}$$

$$\begin{aligned}
 SA &= VA \times IA \\
 &= 230 \times 1.42 \\
 &= 326.6 \text{ VA}
 \end{aligned}$$

$$\begin{aligned}
 SUK &= VUK \times IUK \\
 &= 230 \times 1.44 \\
 &= 331.2 \text{ VA}
 \end{aligned}$$

Data yang didapat dari alat ukur dan sensor selanjutnya dibandingkan dan dapat dilihat pada tabel 4.5 berikut:

Tabel 4.5 Perbandingan alat ukur dan sensor pada beban Setrika

KET SAT	PFt	VUK ( V )	VA ( V )	IUK ( A )	IA ( A )	SA (VA )	SUK (VA )	Pt ( W )
RATA- RATA	0.933	230	227.3	1.44	1.42	325.22	331.9	319.73
% ERROR		1.17 %		1.3%		2.1%		

Keterangan:

- VUK = Tegangan rata-rata alat ukur
- VA = Tegangan rata-rata alat yang dibuat
- IUK = Arus rata-rata alat ukur
- IA = Arus rata-rata alat yang dibuat
- Pt = daya secara perhitungan
- SUK = Daya semu pada alat ukur
- SA = Daya semu pada alat yang dibuat

Perhitungan persentase *error* tegangan, arus dan daya yang didapat merupakan perbandingan pada alat ukur dan alat yang dibuat, sehingga rumusnya adalah:

- Presentase *error* tegangan =  $\frac{230-227.3}{230} \times 100\%$   
=1.17 %

- Presentase *error* arus =  $\frac{1.44-1.42}{1.44} \times 100\%$   
=1.3%

- Presentase *error* daya =  $\frac{331.9-325.22}{331.9} \times 100\%$   
=2.1%

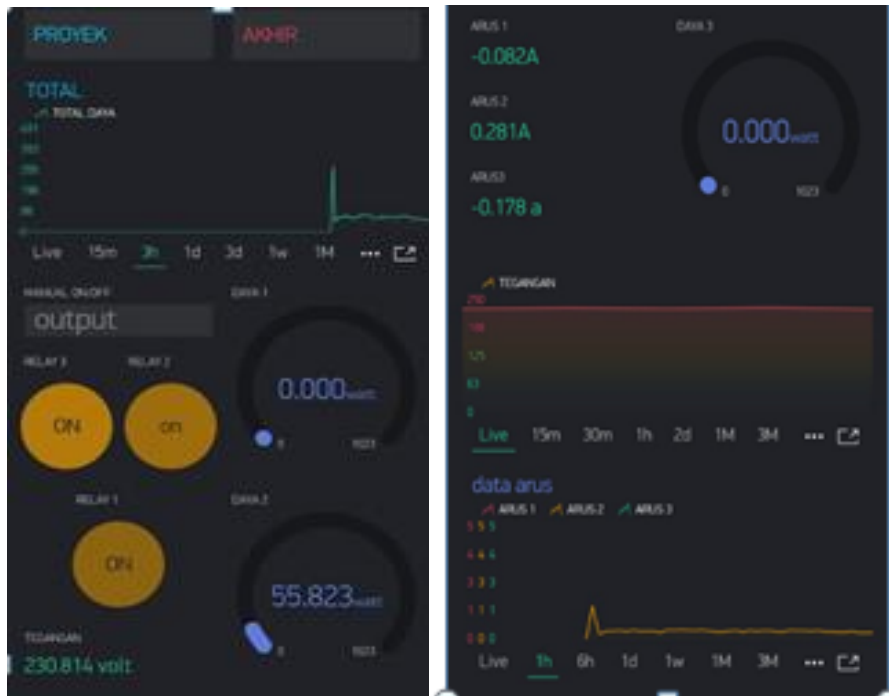
#### **4.7 Software Spots**

Proyek akhir “SPOTS”( *Smart Power Terminal Socket* ) menggunakan *Software* yang dapat digunakan di android. Aplikasi ini dapat digunakan untuk *monitoring* arus, tegangan, dan daya. Dalam hal ini penulis menggunakan *Software* Blynk yang didownload dari *play store*. Disamping mudah digunakan *Software* ini menyediakan fitur yang mendukung alat yang ingin dibuat.

##### **4.7.1 Aplikasi “ BLYNK ”**

Aplikasi “BLNYK” merupakan aplikasi yang di *download* dari *Play Store* pada android untuk *monitoring* SPOTS yang dibuat pada proyek akhir ini.

Berikut tampilan utama pada aplikasi BLYNK pada gambar 4.11 berikut:



Gambar 4.11 Tampilan Utama pada Aplikasi BLYNK

#### 4.7.2 Pengujian Aplikasi “ BLYNK ”

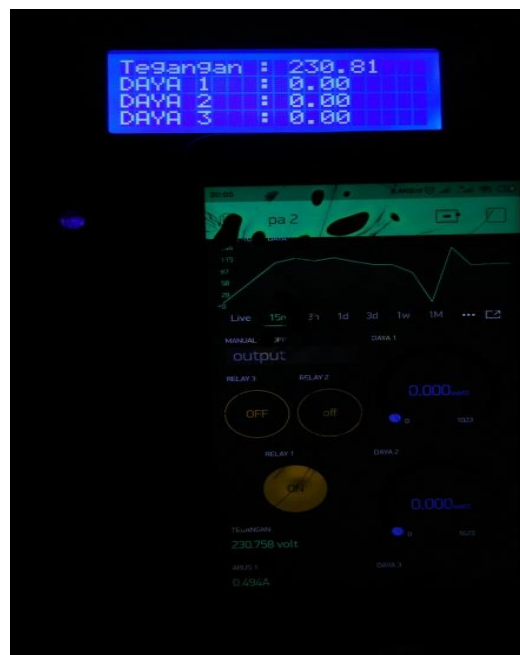
Pengujian aplikasi “BLYNK” dilakukan untuk mengetahui apakah alat yang dibuat sesuai dengan fungsi yang diinginkan. Pengujian dilakukan pada setiap menu yang dibuat, yaitu menghidupkan atau mematikan lampu, *monitoring* arus, tegangan, serta daya.

untuk melakukan pengujian pada aplikasi blynk dapat dilakukan dengan langkah-langkah sebagai berikut:

1. Menghidupkan *Hotspot* pada perangkat Android.
2. Tunggu hingga perangkat terhubung (*connected*) pada Android
3. Buka aplikasi “BLYNK”
4. Masuk ke menu setting lalu setting IP tekan oke lalu kembali ke menu utama
5. Masuk ke menu utama *ON/OFF*. Pilih Relay *ON* untuk menghidupkan *Socket* lalu pilih Relay *OFF* untuk mematikan *socket*
6. *monitoring* arus, tegangan dan daya pada menu utama BLYNK

Relay akan tetap menjalankan perintah yang telah diberikan oleh *user* melalui Android. Pada *Hardware* elektrik Spots terdapat lampu indikator untuk mengetahui *ON/OFF* suatu *socket* yang dimonitoring pada aplikasi BLYNK. Pada saat *monitoring* maka lampu indikator pada *hardware* elektrik akan menjadi acuan *ON/OFF* pada relay. Berikut hasil *monitoring* relay menggunakan aplikasi BLYNK sebagai berikut:

- a. Hasil pengujian pada Relay 1 terdapat pada Gambar 4.12 berikut:



Gambar 4.12 Menghidupkan Relay 1

- b. Hasil pengujian pada Relay 2 terdapat pada Gambar 4.13 berikut:



Gambar 4.13 Menghidupkan Relay 2

c. Hasil pengujian menghidupkan Relay 3 pada Gambar 4.14 berikut:



Gambar 4.14 Menghidupkan Relay 3

d. Hasil pengujian mematikan Relay 1 pada Gambar 4.15 berikut:



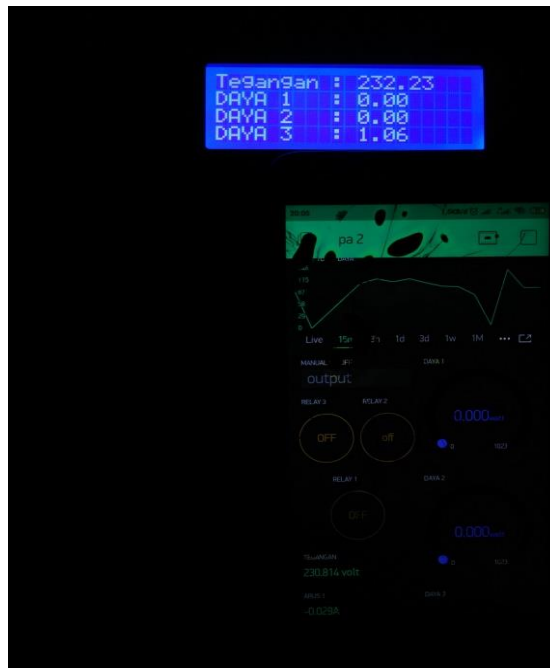
Gambar 4.15 Memaatikan Relay 1

e. Hasil pengujian mematikan Relay 2 pada Gambar 4.16 berikut:



Gambar 4.16 Mematikan Relay 2

f. Hasil pengujian mematikan Relay 3 pada Gambar 4.17 berikut:



Gambar 4.17 Mematikan Relay 3

Pada Gambar 4.12 – Gambar 4.14 merupakan proses untuk menghidupkan Relay dengan acuan lampu indikator sebagai *ON/OFF* Relay sedangkan pada Gambar 4.15 – Gambar 4.17 merupakan proses untuk mematikan Relay dengan acuan lampu indicator sebagai *ON/OFF*.

#### 4.8 Pengujian EPROM

Eprom merupakan sebuah chip/memori yang telah terpasang pada arduino mega 2560 yang dapat menyimpan data. ketika catu daya dimatikan dan ketika dihidupkan kembali maka Eprom akan mengambil data yang telah ditulis pada memori yang telah terbaca saat kondisi catu daya aktif. Jadi disini penulis menggunakan eprom untuk menjadi acuan pengaktifan *socket* dari daya yang terkecil hingga terbesar pada proyek akhir ini. Berikut adalah hasil pengujian Eprom pada Gambar 4.18 berikut:



```

EEPROM: 0.00, 0.00, 79.03
[1588]

  / _ ) / / _ _ _ _ / / _
 / _ / / / / _ \ / ' _ /
 / _ / _ \ _ , / / / _ \ \
    / _ / v0.6.1 on Arduino Meg.

[2208] Connecting to ryzen1234
[5259] AT version:0.60.0.0(Jan 29 2
SDK version:1.5.2(7eee54f4)
Ai-Thinker Technology Co. Ltd.
May 5 2016 17:30:30

```

Gambar 4.18 Pengujian Eprom

pada Gambar 4.18 merupakan hasil dari pengujian Eprom. data yang disimpan pada memori Eprom yang kapasitasnya 4 KB dapat dihapus/*reset* pada arduino mega 2560 bila data yang diambil dalam jumlah banyak.

Berikut list Program Eprom :

```

#include <EEPROM.h>

void setup() {
  EEPROM.get(address, p1);
  EEPROM.get(address + sizeof(double), p2);
  EEPROM.get(address + 2 * sizeof(double), p3);

  Serial.println("EEPROM: " + (String)p1 + ", " +
    (String)p2 + ", " + (String)p3);
}

```

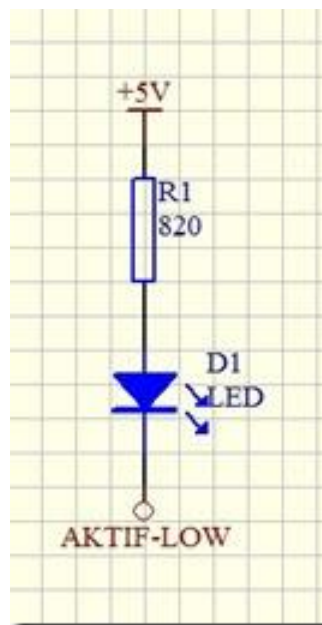
```

void loop() {
    EEPROM.put(address, p1);
    EEPROM.put(address + 2 * sizeof(double), p3);
    EEPROM.put(address + 2 * sizeof(double), p3);
}

```

#### 4.9 Rangkaian LED aktif LOW

Pada rangkaian LED *Aktif-Low*, LED akan menyala jika diberi logika 0 (*LOW*) dan akan mati jika diberi logika 1 (*HIGH*). Ketika diberi logika 0, maka tegangan pada kaki anoda LED lebih *positif* dibandingkan dengan kaki katoda. Selisih tegangan pada kaki anoda dan katoda LED ini melebihi tegangan jatuh LED, sehingga LED pun menghantar (dialiri arus listrik) dan menyala. Sebaliknya, ketika LED diberi logika 1, maka kaki anoda LED tidak lebih *positif* daripada kaki katoda, oleh karenanya LED tidak menghantar (tidak dialiri arus listrik) sehingga LED pun tidak menyala. Rangkaian *aktif low* pada Gambar 4.19 berikut:



Gambar 4.19 Rangkaian LED aktif low

## BAB 5 KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisa terhadap fungsi alat pada proyek akhir dengan judul “SPOTS ( *Smart Power Terminal Socket* )” ini maka dapat ditarik kesimpulan sebagai berikut:

1. Pengujian tegangan yang diukur oleh sensor ZMPT10B dan dibandingkan dengan alat ukur mempunyai hasil yang sangat baik dengan persentase *error* 1.32%
2. Pengujian arus yang diukur oleh sensor ACS712 dan dibandingkan dengan alat ukur mempunyai hasil yang baik dengan persentase *error* maksimal pengujian *ricecooker* 300 watt sebesar 0.6% pengujian Setrika 320 watt sebesar 1.3%
3. Penghitungan nilai *power factor* maksimal yang didapat adalah 0.96
4. Pengiriman data dari BLYNK ke Android tergantung pada baik atau tidaknya kualitas dari sinyal
5. Pengujian pada menu BLYNK *on/off* dapat berfungsi dengan baik, sehingga dapat mematikan setiap relay
6. Nilai tegangan, arus dan daya dapat ditampilkan pada menu *monitoring* sehingga dapat berfungsi sesuai keinginan
7. Pengukuran dibawah 1 A kurang baik untuk sensor 0-30 A

### 5.2 Saran

Apabila alat ini dikembangkan lebih lanjut, adapun fungsi yang perlu diperbaiki dan ditambahkan antara lain :

1. Dapat menggunakan sensor daya yang lebih stabil dan presisi pembacaannya
2. Dapat menambahkan jumlah pada *socket*
3. Dapat menggunakan *driver* yang lebih efisien

## DAFTAR PUSTAKA

- [1] Samuel Gideon, *Analisis Karakteristik Listrik Arus Searah dan Arus Bolak-Balik*. Medan: ready star-2.
- [2] M.Adrinta.A DKK., *Sensor*. Sumatra utara: article, 2017.
- [3] Ruri Ashari Dalimunthe, *PEMANTAU ARUS LISTRIK BERBASIS ALARM DENGAN SENSOR ARUS*. Sumut, 2018.
- [4] Allegro, *ACS712*.: Alldatasheet.
- [5] Nur Arifin, *Rancang Bangun Prototype Power Meter 1 Fasa*. Banda Aceh, Indonesia: KITEKTRO, 2019.
- [6] hwthinker, Ed., *ZMPT101B Sensor Tegangan Listrik PLN 220V 250V AC Single Phase Voltage*.: Shopee.
- [7] Leni Natalia Zulita, Hermawansyah JauhariArifin, *PERANCANGAN MUROTTAL OTOMATIS MENGGUNAKAN*. Bengkulu: Jurnal Media Infotama, 2016.
- [8] Gani Hamdi, *MEMBANGUN APLIKASI BERBASIS ANDROID "PEMBELAJARAN PSIKOTES"*. Yogyakarta: JURNAL DASI, 2011.
- [9] Murtiwiyati dan Glenn Lauren, *Rancang Bangun Aplikasi Pembelajaran Budaya*. Depok: Jurnal Ilmiah KOMPUTASI, 2013.
- [10] android62, *Urutan Nama Android dan Lambang Android Lengkap Terbaru*.: android62.
- [11] S. Samsugi dkk, *INTERNET OF THINGS (IOT): Sistem Kendali Jarak Jauh*. Bandar Lampung, Indonesia: Prosiding Seminar Nasional XII, 2017.
- [12] S.Kom, M.Kom Arafat, *SISTEM PENGAMANAN PINTU RUMAH BERBASIS Internet*.: "Technologia, 2016.
- [13] *Modul Wifi ESP8266*. Yogyakarta: sinauarduino , 2016.
- [14] Muhamad Saleh, *RANCANG BANGUN SISTEM KEAMANAN RUMAH MENGGUNAKAN RELAY*. Jakarta, 2017, vol. Vol. 8.
- [15] ELMECH PUBLISHER, *SONGLE - RELAY MODULE 5V 4 CHANNEL ARDUINO*.: ELMECH, 2016.

- [16] dkk. Yohanes C Saghoa, *Kotak Penyimpanan Uang Berbasis*. Manado: Jurnal Teknik Elektro dan Komputer, 2018, vol. Vol. 7.
- [17] zulfikarfa, *LCD 20x4 (Liquid Crystal Display)*.: unikom.
- [18] Emilia Hesti, *Rancang Bangun Kendali Terminal Stop Kontak Otomatis via SMS (Short*. Palembang: JURNAL TEKNIK ELEKTRO ITP, 2018, vol. Vol. 7.
- [19] *Catu daya*.: eprints.polsri.
- [20] Temy Nusa, *Sistem Monitoring Konsumsi Energi Listrik*. Manado: E-journal Teknik Elektro dan Komputer, 2015, vol. Vol.4.
- [21] Qurotul Aini dkk, *Rancang Bangun Alat Monitoring Pergerakan Objek*. Tangerang, Indonesia: Jurnal Teknik Elektro, 2018, vol. Vol. 10.
- [22] FUAD HASAN, *apa itu Blynk*.: Puak's Technology, 2019.
- [23] Maghfira Nursyabila, Ed., *Mahasiswa ITB Berhasil Temukan Stop Kontak Pintar saat Rumah Kosong*. JAKARTA, 2019.
- [24] *Prinsip Kerja DC Power Supply*.: eprints.polsri.

**LAMPIRAN 1**  
**DAFTAR RIWAYAT HIDUP**

## DAFTAR RIWAYAT HIDUP

### 1. Data Pribadi

Nama Lengkap : M.Ananda Aulia  
Tempat Tanggal Lahir : Sungailiat, 30 juli 1999  
Alamat Rumah : Jl.Sinar Raya Sempan RW 02  
No Handphone : 087891939230 (wa)  
087749557733  
Email : [anandaacaw@gmail.com](mailto:anandaacaw@gmail.com)  
Jenis kelamin : Laki-Laki  
Agama : Islam



### 2. Riwayat Pendidikan

SDN 05 Pemali Lulus 2011  
SMPN 3 Pemali Lulus 2014  
SMAN 1 Pemali Lulus 2017

### 3. Pendidikan Non Formal

Sungailiat, 19 Agustus 2020



M.Ananda Aulia

## DAFTAR RIWAYAT HIDUP

### 1. Data Pribadi

Nama Lengkap : Edry Irawan  
Tempat Tanggal Lahir : Tanjungpandan, 19 oktober 1999  
Alamat Rumah : Jl.Mualim, Aik Merbau RW 05  
RT 14  
No Handphone : 081992605386 (wa)  
Email : [edryir12@gmail.com](mailto:edryir12@gmail.com)  
Jenis kelamin : Laki-Laki  
Agama : Islam



### 2. Riwayat Pendidikan

SDN 17 Tanjungpandan Lulus 2011  
SMPN 2 Tanjungpandan Lulus 2014  
SMAN 1 Tanjungpandan Lulus 2017

### 3. Pendidikan Non Formal

Sungailiat, 19 Agustus 2020



Edry Irawan



**LAMPIRAN 2**  
**PROGRAM UTAMA**

```
#define BLYNK_PRINT Serial
#include <ESP8266_Lib.h>
#include <BlynkSimpleShieldEsp8266.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <EEPROM.h>
#include "EmonLib.h" // Include Emon
Library

EnergyMonitor emon1; // Create an instance

int ReCnctFlag; // Reconnection Flag
int ReCnctCount = 0; // Reconnection counter

int relay1 = 3;
int relay2 = 4;
int relay3 = 5;

double p1 = 0;
double p2 = 0;
double p3 = 0;;

bool status = 0;
char auth[] = "2c1b1929aac24645956cb090ce0ce3ab";
BlynkTimer timer;

//steARUs
const int sensorIn = A1;
```

```
const int sensorIn1 = A2;
const int sensorIn2 = A3;
int mVperAmp = 66;

double Voltage = 0; double Voltage2 = 0; double
Voltage3 = 0;
double VRMS = 0, VRMS2 = 0, VRMS3 = 0;
double AmpsRMS = 0, AmpsRMS2 = 0, AmpsRMS3 = 0;
double I = 0, I2 = 0, I3 = 0;
double DAYA1 = 0;
float mcuVoltage = 5.0;
float dayatotal = 0;
const uint8_t address = 0;
double volt = 0;

char ssid[] = "ryzen1234";
char pass[] = "ryzen321";

#define EspSerial Serial2
#define ESP8266_BAUD 115200

ESP8266 wifi(&EspSerial);
LiquidCrystal_I2C lcd(0x27, 20, 4);

void setup()
{

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

```
lcd.backlight();
lcd.clear();
delay(500);
EspSerial.begin(ESP8266_BAUD);
delay(10);
pinMode(relay1, OUTPUT);
pinMode(relay2, OUTPUT);
pinMode(relay3, OUTPUT);
emon1.voltage(0, 520, 1.1);

digitalWrite(relay1, HIGH);
digitalWrite(relay2, HIGH);
digitalWrite(relay3, HIGH);

EEPROM.get(address, p1);
EEPROM.get(address + sizeof(double), p2);
EEPROM.get(address + 2 * sizeof(double), p3);

Serial.println("EEPROM: " + (String)p1 + ", " +
(String)p2 + ", " + (String)p3);

checkRelay();

Blynk.begin(auth,          wifi,          ssid,
pass, "iwancilibur.my.id", 8080);
timer.setInterval(2000L, getACS712);
timer.setInterval(2000L, getACS712B);
timer.setInterval(2000L, getACS712C);
}
```

```

BLYNK_CONNECTED() {
    Serial.println("Cconnected");
    ReCnctCount = 0;
}

void loop()
{
    if (Blynk.connected()) { // If connected run as
normal
        delay(10000);
        // Blynk.syncAll();
        Blynk.run();

    } else if (ReCnctFlag == 0) { // If NOT connected
and not already trying to reconnect, set timer to try
to reconnect in 30 seconds
        ReCnctFlag = 1; // Set reconnection Flag
        Serial.println("Starting reconnection timer in 30
seconds...");
        timer.setTimeout(30000L, []() { // Lambda
Reconnection Timer Function
            ReCnctFlag = 0; // Reset reconnection Flag
            ReCnctCount++; // Increment reconnection Counter
            Serial.print("Attempting reconnection #");
            Serial.println(ReCnctCount);
            Blynk.connect(); // Try to reconnect to the
server
        }); // END Timer Function
    }

    timer.run();
}

```

```

lcd.setCursor(00, 00);
lcd.print("Tegangan :");
lcd.setCursor(00, 1);
lcd.print("DAYA 1   :");
lcd.setCursor(00, 2);
lcd.print("DAYA 2   :");
lcd.setCursor(00, 3);
lcd.print("DAYA 3   :");
lcd.setCursor(11, 00);
lcd.print(String(volt) + "  ");
lcd.setCursor(11, 01);
lcd.print(String(p1) + "  ");
lcd.setCursor(11, 02);
lcd.print(String(p2) + "  ");
lcd.setCursor(11, 03);
lcd.print(String(p3) + "  ");

//getACS712();
//getACS712B();
//getACS712B();
    emon1.calcVI(25, 1500);           // Calculate all.
No.of half wavelengths (crossings), time-out
    volt = emon1. Vrms;
    if (volt < 50) {
        volt = 0;
    }
    Blynk.virtualWrite(V4, float(volt));
    delay(500);

```

```

dayatotal = p1 + p2 + p3;
Blynk.virtualWrite(V10, float(dayatotal));

Serial.print(String("v1 = ") + volt + " V");
Serial.print(String("i1 = ") + I + " A");
Serial.print(String("p1 = ") + p1 + " Watts");
Serial.print(String("i2 = ") + I2 + " A");
Serial.print(String("p2 = ") + p2 + " Watts");
Serial.print(String("i3 = ") + I3 + " A");
Serial.println(String("p3 = ") + p3 + " Watts");
delay(50);

checkRelay();
delay(2000);
}

//SENSOR ARUS 1
float getVPP() {
  float result;
  int readValue;
  int maxValue = 0;
  int minValue = 1024;
  uint32_t start_time = millis();
  while ((millis() - start_time) < 1000) {
    readValue = analogRead(sensorIn);
    if (readValue > maxValue) {
      maxValue = readValue;

```

```

    }
    if (readValue < minValue) {
        minValue = readValue;
    }
}
result = ((maxValue - minValue) * 5.0) / 1024.0;
return result;
}
//SENSOR ARUS 2
float getVPP1()
{
    float result;
    int readValue; //value read from the sensor
    int maxValue = 0; // store max value here
    int minValue = 1024; // store min value here
    uint32_t start_time = millis();
    while ((millis() - start_time) < 1000) //sample for 1
    Sec
    {
        readValue = analogRead(sensorIn1);
        // see if you have a new maxValue
        if (readValue > maxValue)
        {
            /*record the maximum sensor value*/
            maxValue = readValue;
        }
        if (readValue < minValue)
        {
            /*record the maximum sensor value*/

```



```

        minValue = readValue;
    }
}
// Subtract min from max
result = ((maxValue - minValue) * 5.0) / 1024.0;
return result;
}
//SENSOR ARUS 3
float getVPP2()
{
    float result;
    int readValue; //value read from the sensor
    int maxValue = 0; // store max value here
    int minValue = 1024; // store min value here
    uint32_t start_time = millis();
    while ((millis() - start_time) < 1000) //sample for 1
    Sec
    {
        readValue = analogRead(sensorIn2);
        // see if you have a new maxValue
        if (readValue > maxValue)
        {
            /*record the maximum sensor value*/
            maxValue = readValue;
        }
        if (readValue < minValue)
        {
            /*record the maximum sensor value*/
            minValue = readValue;
        }
    }
}

```

```
    }  
  }  
  // Subtract min from max  
  result = ((maxValue - minValue) * 5.0) / 1024.0;  
  return result;  
}
```

```
// TARIK ARUS 1  
void getACS712()  
{  
  Voltage = getVPP();  
  VRMS = (Voltage / 2.0) * 0.707;  
  AmpsRMS = (VRMS * 1000) / mVperAmp;  
  I = AmpsRMS - 0.67;  
  if (I < 0.1) {  
    I = 0;  
  }  
  p1 = I * volt;  
  if (p1 < 0) {  
    p1 = 0;  
  }  
}
```

```
EEPROM.put(address, p1);  
Blynk.virtualWrite(V3, float(I));  
Blynk.virtualWrite(V9, float(p1));
```

```

}
// TARIK ARUS 2
void getACS712B()
{
  Voltage2 = getVPP1();
  VRMS2 = (Voltage2 / 2.0) * 0.707;
  AmpsRMS2 = (VRMS2 * 1000) / mVperAmp;
  I2 = AmpsRMS2 - 0.85;
  if (I2 < 0.1) {
    I2 = 0;
  }
  p2 = I2 * volt;
  if (p2 < 0) {
    p2 = 0;
  }

  EEPROM.put(address + sizeof(double), p2);
  Blynk.virtualWrite(V5, float(I2));
  Blynk.virtualWrite(V8, float(p2));

}
// TARIK ARUS 3
void getACS712C()
{
  Voltage3 = getVPP2();
  VRMS3 = (Voltage3 / 2.0) * 0.707;

```

```

AmpsRMS3 = (VRMS3 * 1000) / mVperAmp;
I3 = AmpsRMS3 - 0.68;
if (I3 < 0.1) {
    I3 = 0;
}
p3 = I3 * volt;
if (p3 < 0) {
    p3 = 0;
}

EEPROM.put(address + 2 * sizeof(double), p3);
Blynk.virtualWrite(V6, float(I3));
Blynk.virtualWrite(V7, float(p3));
}

bool isLargerThan(double first, double second) {
    return first > second;
}

void sortArray(double array[][2], size_t arraySize) {
    for (size_t i = 1; i < arraySize; i++) {
        for (size_t j = i; j > 0 && (isLargerThan(array[j - 1][0], array[j][0])); j--) {
            double tmp0 = array[j - 1][0];
            double tmp1 = array[j - 1][1];
            array[j - 1][0] = array[j][0];
            array[j - 1][1] = array[j][1];
            array[j][0] = tmp0;
            array[j][1] = tmp1;
        }
    }
}

```

```

    }
}
}

uint8_t* getRelayOrder() {
    static uint8_t relay[3];

    double arrayList[3][2] = {{p1, relay1}, {p2, relay2},
    {p3, relay3}};

    sortArray(arrayList, 3);

    for (int i = 0; i < 3; i++) relay[i] =
    arrayList[i][1];

    return relay;
}

void checkRelay() {
    if (volt <= 200) {
        if (status == 1) {
            Blynk.run();

            digitalWrite(relay1, HIGH);
            digitalWrite(relay2, HIGH);
            digitalWrite(relay3, HIGH);

            status = 0;
        }
    } else {
        if (status == 0) {
            uint8_t *relayPointer = getRelayOrder();

            Blynk.run();

            digitalWrite(*relayPointer, LOW);

            delay(5000);
        }
    }
}

```

```
        digitalWrite(*(relayPointer + sizeof(uint8_t)),
LOW);
        delay(5000);
        digitalWrite(*(relayPointer + 2 *
sizeof(uint8_t)), LOW);
        delay(5000);
        status = 1;
    }
}
}
```