

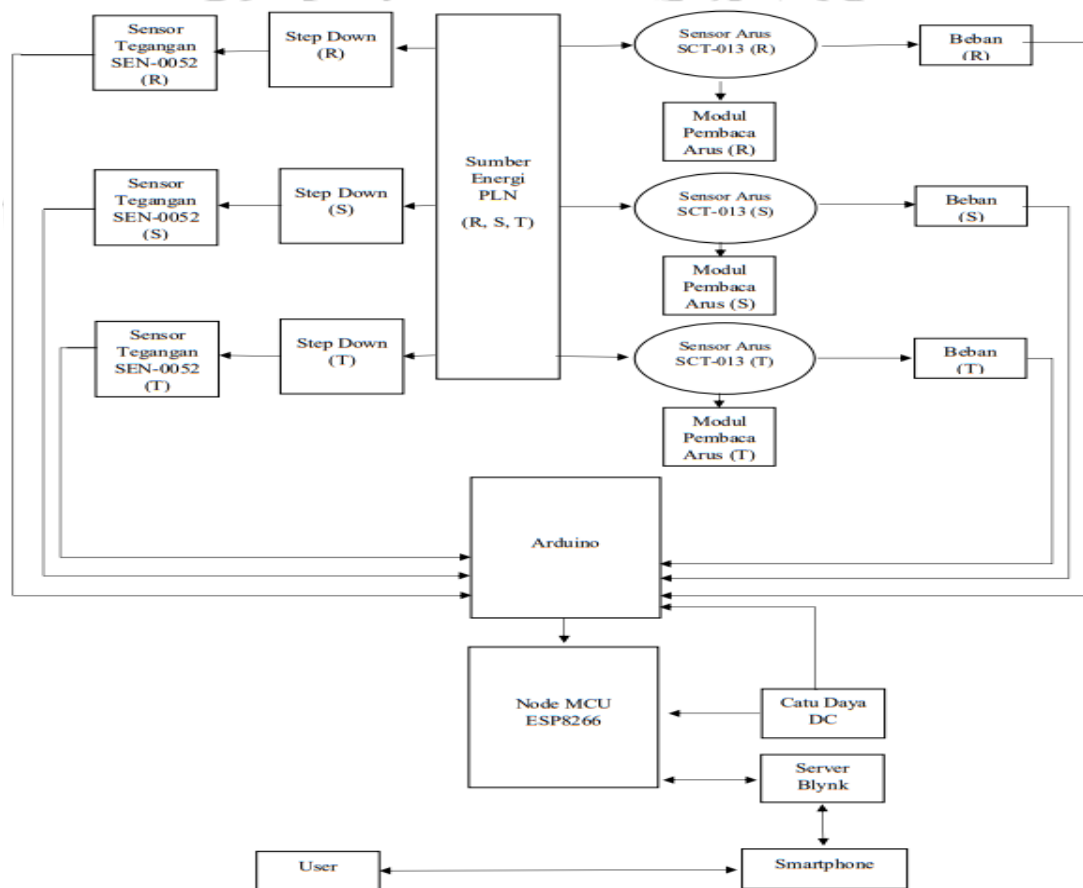
## BAB III

### METODOLOGI PENELITIAN

Metode yang digunakan dalam penelitian ini adalah metode kuantitatif, dimana penelitian ini akan menyajikan hasil dalam bentuk angka. Metode kuantitatif merupakan penelitian yang menggunakan angka mulai dari pengumpulan data, analisa data, dan menampulkan hasil data. [2]

#### 3.1 Diagram Blok dan Prinsip Kerja Sistem Monitoring

Sistem Secara garis besar penelitian ini ditunjukkan dengan diagram blok sistem

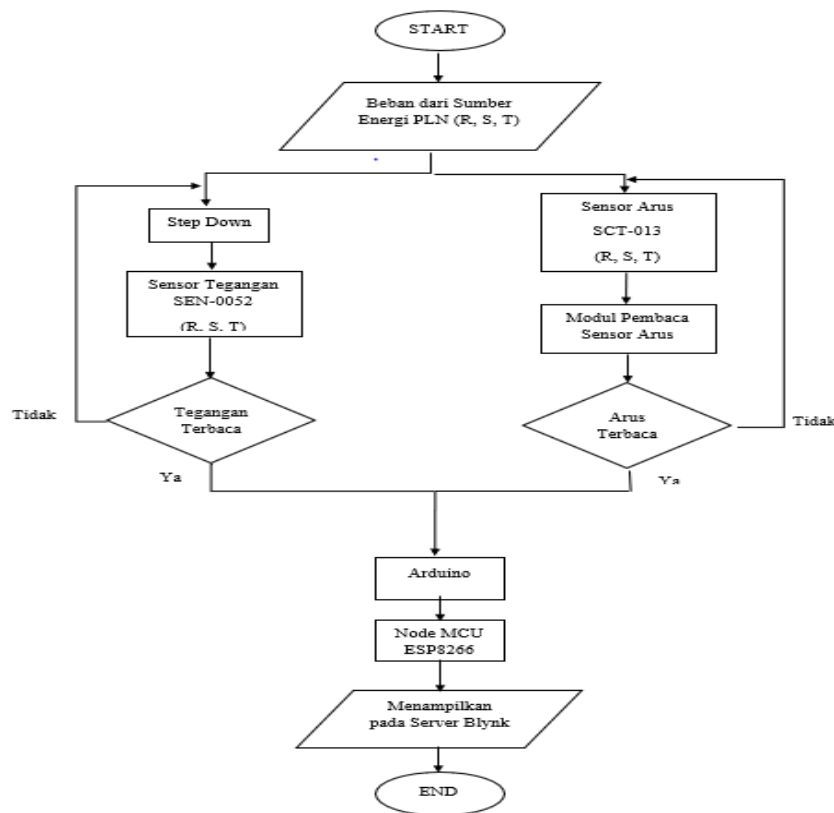


Gambar 3. 1 Blok Diagram Sistem

Pada gambar 3.1 diatas blok diagram sistem pada panel kontrol 3 phase dilakukan pengukuran nilai tegangan dan arus, saat melakukan pengukuran pada tegangan memerlukan rangkaian pembagi tegangan agar nilai tegangan yang

dihasilkan pada setiap phasanya dapat dibaca oleh mikrokontroller dan pada rangkaian pembagi tegangan terdapat penyearah gelombang penuh (*Full Wave Retifier*) agar penyearah tegangan AC menjadi tegangan DC dimana rangkaian pembagi tegangan sebagai sensor tegangan pada setiap phasanya. Pada pengukuran arus dimana menggunakan sensor SCT 103. Dimana *output* yang dihasilkan sensor SCT 103 akan langsung dapat dibaca oleh mikrokontroller karena sudah dalam kondisi DC.

*Server* Blynk merupakan layanan yang digunakan untuk mendukung *project* IoT [3]. Dimana Blynk pada penelitian ini berfungsi sebagai *server* untuk mengirimkan data tegangan dari setiap phasanya yang dibaca oleh sensor tegangan dan arus kemudian akan diolah oleh mikrokontroller, data tegangan dan arus pada setiap phasanya akan dikirimkan ke *server* Blynk yang menghasilkan *interface* pada *smartphone* yang dapat diakses setiap *user* Android maupun iOS yang ingin memonitoring pada panel kontrol 3 phase.

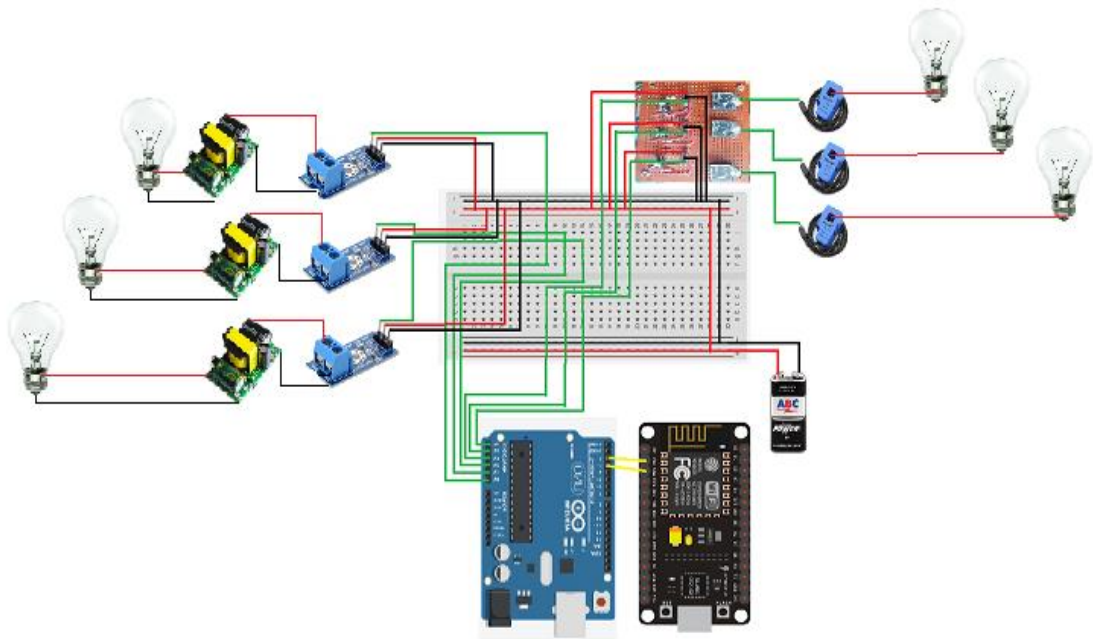


Gambar 3. 2 *Flowchart* Sistem Monitoring

Berdasarkan *flowchart* sistem Monitoring Tegangan dan Arus berbasis rangkaian pembagi tegangan maka data yang telah dibaca oleh rangkaian pembagi tegangan sebagai sensor tegangan dan sensor arus SCT – 013 akan dikirimkan pada mikrokontroller dan pada tegangan sebelum dikirimkan pada mikrokontroller terlebih dahulu melewati rangkaian pembagi tegangan dan akan diarahkan oleh penyearah gelombang penuh guna mendapatkan output DC agar dapat dibaca mikrokontroller. Setelah itu mikrokontroller akan mengirimkan pada *server* Blynk, *server* Blynk dapat diakses oleh setiap *user* untuk melakukan monitoring tegangan dan arus pada panel kontrol 3 phase.

### 3.2 Perancangan *Hardware*

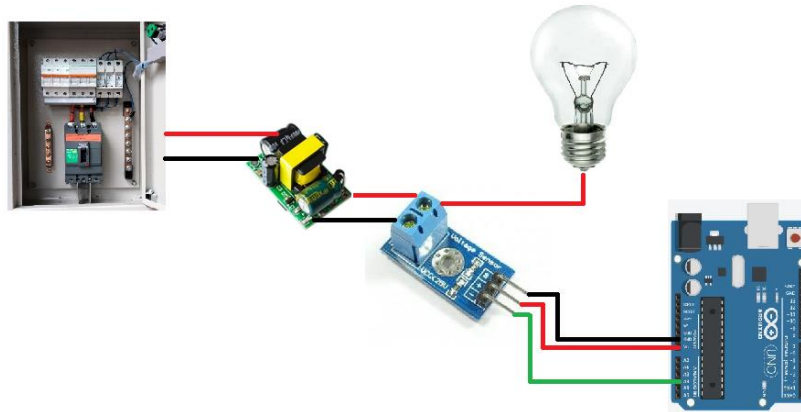
Pada gambar 3.2 dibawah menunjukkan isi keseluruhan sistem pada alat.



Gambar 3. 3 Perancangan *Hardware*

#### 3.2.1 Antarmuka Arduino ke Sensor Tegangan

Antarmuka Arduino ke rangkaian pembagi tegangan sebagai sensor tegangan berfungsi sebagai pusat pengendali untuk menerima *input* yang dihasilkan oleh sensor tegangan.



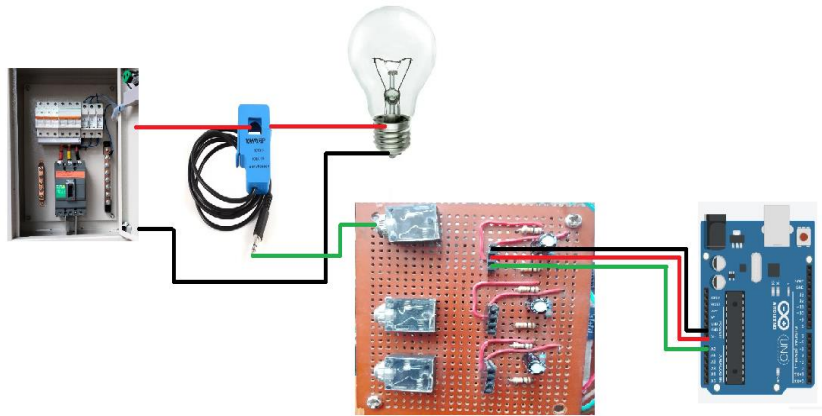
Gambar 3. 4 Antarmuka Arduino ke Sensor Tegangan SEN-0052

Tabel 3. 1 Konfigurasi Pin Antarmuka Arduino ke Sensor Tegangan SEN-0052

NO	Beban Dari Sumber Energi	Sensor Tegangan	Arduino
1	Positif Beban	L	-
2	Negatif Beban	N	-
3	-	S	A3, A4, A5
4	-	Plus (+)	Vin (5V)
5	-	Min (-)	GND

### 3.2.2 Antarmuka Arduino ke Sensor SCT – 013

Antarmuka Sensor arus disini menggunakan sensor SCT – 013, dimana fungsi dari sensor arus mengukur arus dari panel kontrol 3 phase dengan maksimum arus yang dapat terbaca 100A. *Output* dari sensor SCT – 013 akan menjadi *input* dari modul pembaca arus hasil pengukuran sensor SCT – 013.



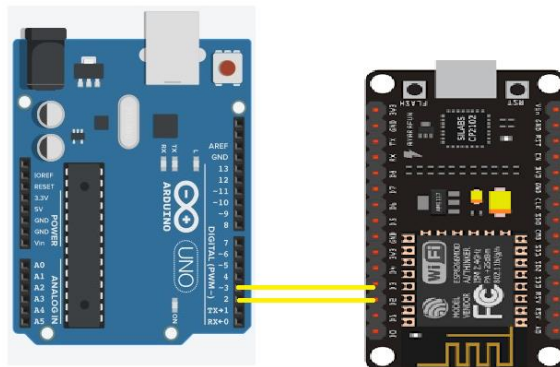
Gambar 3. 5 Antarmuka Arduino ke Sensor Arus SCT-013

Tabel 3. 2 Konfigurasi Pin Antarmuka Arduino ke Sensor Arus SCT-013

NO	Beban Dari Sumber Energi	Sensor Arus SCT-013	Arduino
1	Positif Beban	L	-
2	-	S	A0, A1, A2
3	-	Plus (+)	Vin (5V)
4	-	Min (-)	GND

### 3.2.3 Antarmuka Arduino ke Node MCU ESP8266

Node MCU ESP 8266 dalam antarmuka dengan Arduino disini berfungsi sebagai jembatan antara Arduino dengan *server* Blynk, dimana ESP 8266 bersifat *opensource* yang berupa USB to serial. [11]



Gambar 3. 6 Antarmuka Arduino ke Node MCU ESP8266

Tabel 3. 3 Konfigurasi Pin Antarmuka Arduino ke Node MCU ESP8266

No	Node MCU ESP 8266	Arduino
1	VCC	Vin (5V)
2	TX	RX
3	RX	TX
4	GND	GND

### 3.3 Perancangan *Software*

Perancangan *Software* atau perancangan perangkat lunak pada penelitian ini berfungsi sebagai menampilkan data dalam bentuk angka pada tegangan R,S,T dan arus R,S,T pada panel kontrol 3 Phase yang akan ditampilkan menggunakan aplikasi Blynk dengan cara komunikasi *Simplex* atau komunikasi satu arah dari Arduino ke Node MCU ESP 8266.

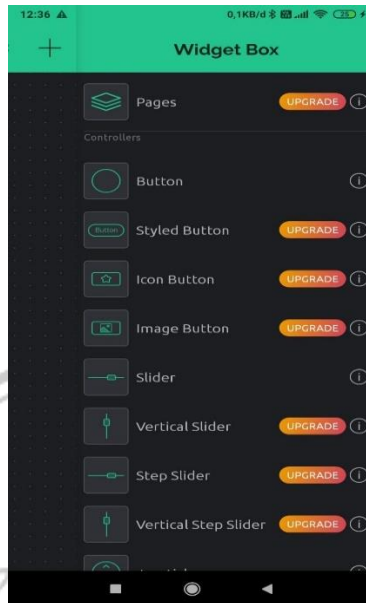
#### 3.3.1 Blynk ke Arduino IDE

Blynk dalam perancang *Software* ini berfungsi sebagai monitoring sistem dari jarak jauh dan juga dapat menyimpan data yang dihasilkan oleh sensor tegangan dan arus serta menampilkan hasil pengukuran dari data yang telah didapat. [3]

Dalam penelitian ini Arduino IDE sebagai *code* editor, aplikasi Blynk sebagai alat kendali dan Node MCU ESP 8266 sebagai mikrokontrollernya, dalam proses pemrograman Arduino IDE menambahkan *board* Node MCU ESP 8266 terlebih dahulu dan menambahkan *library* ke Arduino IDE.

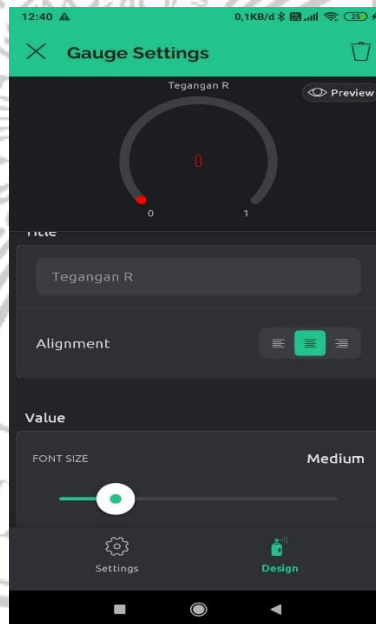
Proses perancangan Blynk sebagai berikut :

- Membuat *project* baru dan memilih perangkat yang akan digunakan yaitu, Node MCU ESP 8266
- Setelah itu akan masuk pada papan *project*, dan memilih *widget* yang akan digunakan.



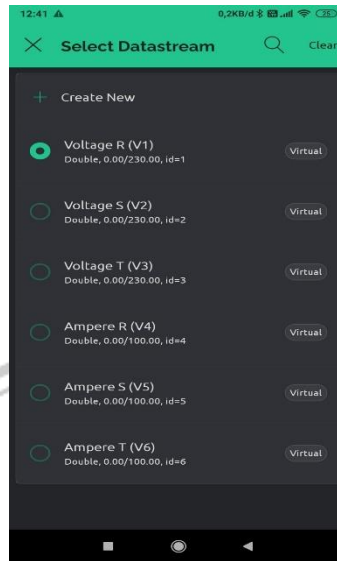
Gambar 3. 7 Widget Blynk

- Melakukan pelabelan nama pada *widget* yang digunakan.



Gambar 3. 8 Label Nama Blynk

- Melakukan *setting* pada *widget* untuk menunjukkan data mana yang akan ditampilkan.



Gambar 3. 9 Setting Blynk

- Langkah terakhir menghubungkan Blynk ke Node MCU ESP 8266 yang terdapat pada pemrograman.

### 3.3.2 Library Arduino IDE

Dalam penelitian ini untuk menunjang pembuatan alat diperlukan pemrograman pada Arduino IDE dan dimana *library* yang digunakan untuk program Arduino IDE dalam penelitian ini ada dua yaitu untuk program Arduino dan program untuk Node MCU ESP 8266.

#### 3.3.2.1 Arduino

```

Arduino_voltage | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Select Board
Arduino_voltage.ino
1 #include <Wire.h>
2 #include <EmonLib.h>
3 #include <SoftwareSerial.h>
4 SoftwareSerial Serial1(3, 2);
5 const int sensor1 = A5;
6 const int sensor2 = A4;
7 const int sensor3 = A3;
8 const int sensor4 = A2;
9 const int sensor5 = A1;
10 const int sensor6 = A0;
11
12 float raw1, raw2, raw3;
13 float tegangan1, tegangan2, tegangan3;
14 float t_beban1, t_beban2, t_beban3;
15 float P_T1, P_T2, P_T3;
16 double arus1, arus2, arus3;
17
18 EnergyMonitor emon1;
19 EnergyMonitor emon2;
20 EnergyMonitor emon3;
21
22 void setup() {

```

Gambar 3. 10 Program Arduino



*Library* dalam pemrograman Arduino lebih untuk program membaca hasil yang dihasilkan oleh sensor tegangan dan sensor arus, dimana *library Include* sebagai pemanggil yang digunakan sebagai berikut:

- <Wire.h> *Library* ini berfungsi untuk mengaktifkan komunikasi *serial I2C* pada arduino.
- <EmonLib.h> *Library* ini berfungsi untuk mengakses manajer perpustakaan dalam Arduino IDE.
- <SoftwareSerial.h> *Library* ini berfungsi sebagai komunikasi *serial* pada pin digital pada papan Arduino.

### 3.3.2.2 Node MCU ESP 8266

```

ESP_SEND_DATA | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Select Board
ESP_SEND_DATA.ino
1 #define BLYNK_TEMPLATE_ID "TMPLnhwZ8b6c"
2 #define BLYNK_DEVICE_NAME "SKRIPSI"
3 #define BLYNK_AUTH_TOKEN "B2T4CCGyHaaSNMwXyKB_1lw7L6H629Un"
4 #define BLYNK_PRINT Serial
5 #include <ESP8266WiFi.h>
6 #include <BlynkSimpleEsp8266.h>
7 #include <SoftwareSerial.h>
8
9 char auth[] = BLYNK_AUTH_TOKEN;
10 char ssid[] = "HC_UMUM";
11 char pass[] = "saleho123";
12 BlynkTimer timer;
13 SoftwareSerial Serial2(D1, D2);
14
15 void myTimerEvent(){
16   String dd;
17   int index1, index2, index3, index4, index5, index6;
18   double TB1, TB2, TB3, AB1, AB2, AB3;
19   while(Serial2.available()){
20     delay(10);
21     char c = Serial2.read();
22     dd += c;
  
```

Gambar 3. 11 Program Node MCU ESP8266

*Library* dalam pemrograman Node MCU ESP 8266 untuk dapat mengambil data yang telah diproses oleh Arduino berupa hasil dari pengukuran menggunakan rangkaian pembagi tegangan sebagai sensor tegangan dan sensor arus SCT-013.

Dalam pemrograman Node MCU ESP 8266 ini *library Include* sebagai pemanggil dan *Devine* sebagai Inialisasi, dimana untuk mengambil data dari Arduino menggunakan *Void Mytimerevent*, dan fungsi dari *Void* ada dua yaitu *Void Setup* sebagai *Setting* dan *Void Loop* sebagai perulangan, dimana *library define* dan *include* yang digunakan sebagai berikut ;

- BLYNK\_TEMPLATE\_ID sebagai pemanggil Blynk yang akan digunakan.
- BLYNK\_DEVICE\_NAME sebagai pemnggal nama *project* di Blynk yang akan digunakan.
- BLYNK\_AUTH\_TOKEN sebagai pengidentifikasi yang digunakan untuk menghubungkan *hardware* pada *smartphone*.
- BLYNK\_PRINT sebagai menampilkan seluruh layar yang ada pada Blynk.

