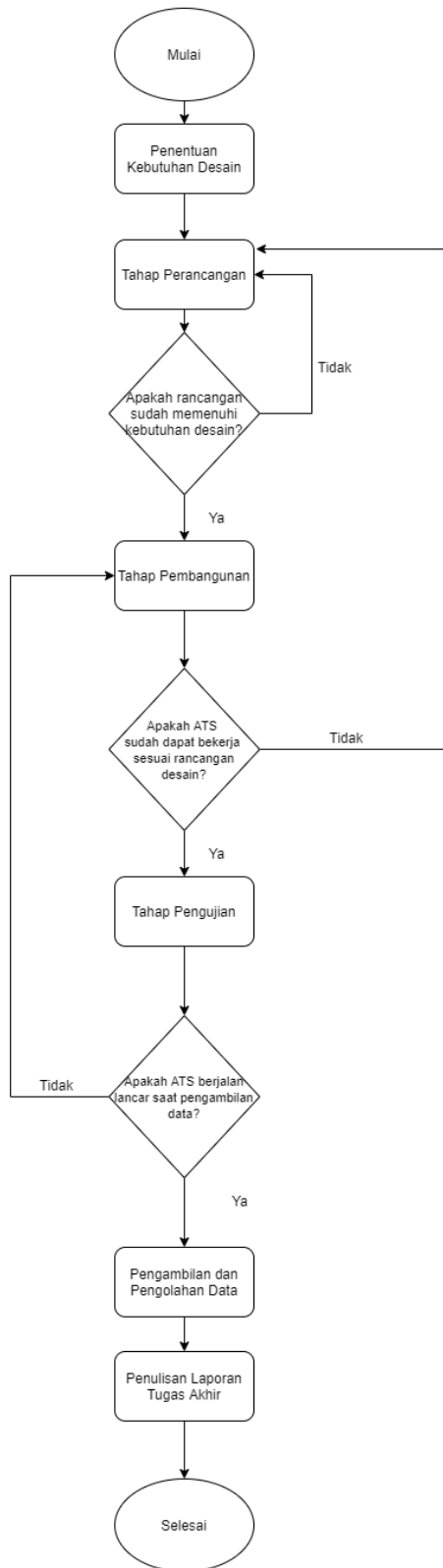


BAB III

METODOLOGI PERANCANGAN

3.1. Alur penelitian

Proses pengerjaan dibagi menjadi tiga bagian yaitu perancangan, pembuatan, dan pengujian. Pada tahap perancangan, perangkat ATS dan *Solar Home System* dirancang agar sesuai dengan kebutuhan desain. Tahap pembuatan adalah tahap dimana rancangan dari ATS diwujudkan. Tahap terakhir yaitu tahap pengujian, dimana ATS diuji di dalam lab dan juga di lapangan. Untuk penjelasan mengenai tahap penelitian yang mendetail akan dijelaskan di sub bab berikutnya. Gambar 3.1. menunjukkan alur penelitian dari perangkat ATS.



Gambar 3.1. Alur Penelitian

3.2. Kebutuhan Desain

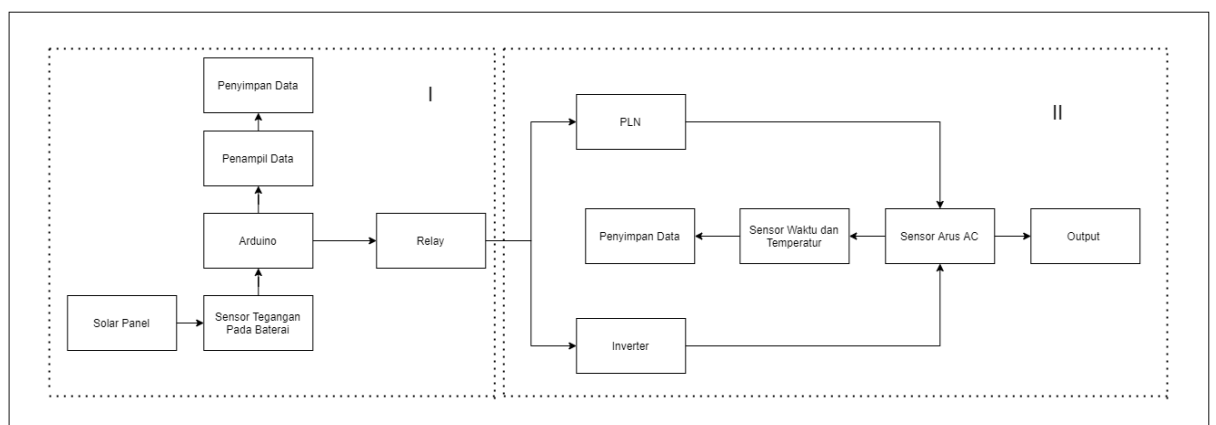
ATS yang dirancang harus dapat memenuhi kebutuhan sebagai berikut :

1. Dapat mengalihkan tegangan dari sistem panel surya dengan grid.
2. Maksimal waktu perpindahan dari satu sumber ke sumber lainnya adalah 100ms.
3. ATS harus dapat memantau data tegangan dari baterai.
4. ATS harus aman dan mudah untuk digunakan.
5. ATS dapat menampilkan data di LCD.
6. ATS harus dapat mengutamakan grid ketika baterai sedang melakukan *charging*.
7. ATS memiliki data logger untuk mencatat voltase baterai dan juga sumber apa yang sedang digunakan.
8. Panel surya dan *Solar Home system* harus berdekatan untuk mengurangi *power losses* yang diakibatkan resistansi pada kabel.
9. *Solar Home System* harus dapat bertahan dari air hujan .

3.3. Cara Kerja Perangkat ATS

Perangkat ATS dipasang pada *Solar Home System*. Pada relai pertama disambungkan pada live PLN, live beban dan juga live *inverter*. Yang nantinya live beban dihubungkan dengan sensor ACS712. Pada relai kedua disambungkan pada neutral PLN, neutral beban, dan neutral dari *Inverter*. ATS ini ditenagai dari USB port yang terletak pada *charge controller*. Pertama kali ATS menyala membaca tegangan dari aki yang tersambung secara seri dengan *inverter* dan juga sensor INA

219 pada *Charge controller*. Sensor mengirimkan sinyal digital melalui pin SDA dan SCL Arduino. Kemudian Arduino mengolah data tersebut untuk menentukan baterai berada 100% atau belum. Jika belum maka Arduino memerintahkan relai untuk menggunakan PLN terlebih dahulu hingga baterai mencapai 100%. Ketika baterai sudah mencapai 100% maka Arduino memerintahkan untuk menggunakan baterai hingga baterai mencapai 12%. Ketika baterai sudah mencapai 12% maka Arduino memerintahkan relai untuk menggunakan listrik PLN hingga baterai menyentuh 100% lagi. Selama ATS dinyalakan maka ada penyimpanan pada SD card dan tampilan pada LCD berupa tegangan baterai dan juga sumber mana yang sedang digunakan. Blok diagram dibagi menjadi dua bagian, blok diagram I untuk proses *monitoring* dan blok diagram II untuk proses *switching*. Gambar 3.2 menunjukkan diagram blok dari sistem modul ATS. Blok diagram I untuk proses *monitoring* dan blok diagram II untuk proses *switching*.

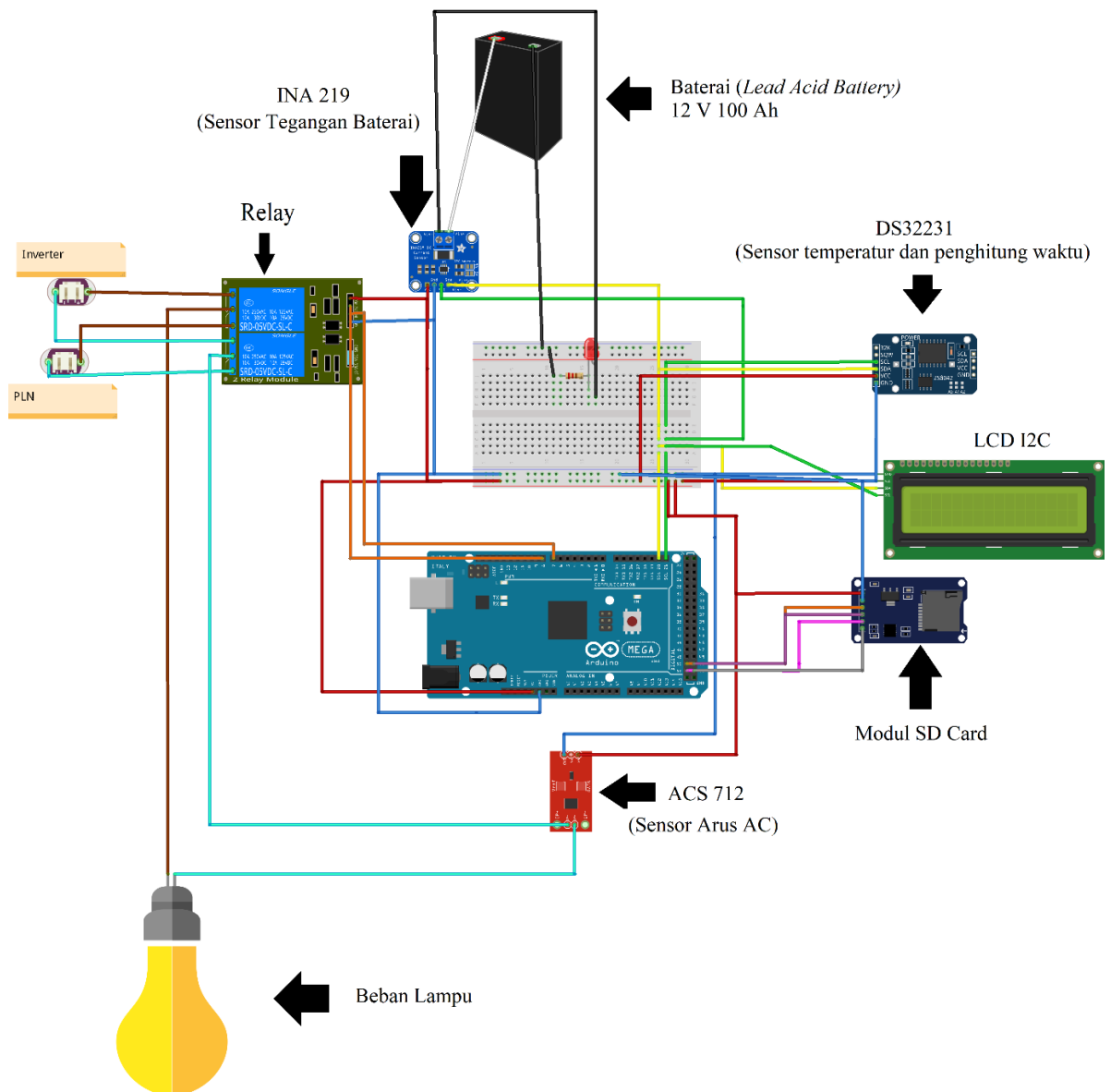


Gambar 3.2. Blok Diagram Dari Modul ATS Blok Diagram I Untuk Proses *Monitoring* Dan Blok Diagram II Untuk Proses *Switching*

3.4. Alur Perancangan

Perangkat ATS terdiri dari 5 komponen utama yaitu Arduino, sensor tegangan, relai, LCD I2C dan *data logger*. Sensor tegangan digunakan untuk membaca arus dan tegangan dari aki yang diterukan menuju Arduino, lalu Arduino mengolah data yang didapat dan juga untuk pengambilan keputusan menyalakan relai *live* atau *ground*. Modul *data logger* sendiri terdiri dari RTC untuk mencatat waktu pengambilan data dan juga ada modul *SD card* yang digunakan untuk menyimpan data. Untuk bagian *Solar Home System* terdiri dari panel surya, aki (*Lead Acid Battery*), *Charge controller*, dan *Inverter*.

Jeda 100 ms merupakan maksimal waktu perpindahan ATS ketika ATS di desain dengan "*Closed Transition Delay*". *Closed Transition Delay* merupakan jeda perpindahan ketika sumber lainya dan beban sudah terhubung, sehingga dimana ada dua sumber yang terhubung terhadap beban sehingga dapat mengurangi jeda [30]. Gambar 3.3 adalah proses perancangan dengan menggunakan software fritzing.



fritzing

Gambar 3.3. Skema Perancangan ATS

3.5. Pemilihan Komponen

Sebelum ATS dan *Solar Home System* dirangkai adanya pemilihan komponen yang sesuai justifikasi antara harga, fungsi dan kualitas agar ATS dan *Solar Home System* dapat berjalan secara maksimal.

3.5.1. Pemilihan Arduino

Pada ATS, Arduino digunakan untuk mengendalikan relai. Pemilihan Arduino dikarenakan mikrokontroler yang mudah didapatkan dengan harga yang terjangkau dan juga bersifat modular sehingga jika ada penambahan fitur dapat dilakukan dengan mudah.

Arduino menerima input tegangan dan arus dari baterai lalu digunakan untuk mengambil keputusan pada relai. Dipilihnya Arduino mega dikarenakan Arduino mega banyak memiliki *port I/O* dan juga memiliki memori yang lebih besar jika dibandingkan Arduino Uno. Pemilihan ini juga bertujuan jika kedepanya ada tambahan parameter yang diukur masih menyisakan *port I/O*, selain itu memudahkan juga untuk menambahkan *Internet of Things* dikarenakan Arduino mega ini memiliki memori 256 kB lebih besar bila dibandingkan dengan Arduino Uno sebesar 32 kB [31].

3.5.2. Pemilihan Panel Surya

Panel yang dipilih adalah 100 Wp. Panel ini dipilih dikarenakan beban yang digunakan adalah lampu tembak 50 W. Lampu 50 W digunakan untuk mensimulasikan beban penerangan dirumah. Panel 100 Wp diharapkan dapat menyalakan lampu dan mengisi baterai pada siang hari. Panel surya yang digunakan bermerk LUMINOUS dengan bobot 7.5 Kg dengan spesifikasi *Open circuit voltage (Voc)* sebesar 22.1 V dengan *Rated Voltage (Vmp)* 18.3 V dan *Short circuit current*

(*I_{sc}*) sebesar 6 A dengan *Rated current (I_{mp})* sebesar 5.49 A. Dengan spesifikasi tersebut dapat menghasilkan 100 W *Peak* dengan *power tolerance range* sebesar 3%.

3.5.3. Pemilihan Aki

Dipilih aki dengan merk INCOE N100T dengan kapasitas 12 Volt dan 100 Ah dikarenakan memiliki *deep cycle* yang memang didesain untuk secara regular digunakan semua kapasitasnya. Aki merk INCOE N100T juga didesain untuk penggunaan *Solar Home System* dan juga Penerangan Jalan Umum. Aki INCOE juga memiliki fitur langsung bertenaga setelah diisi accu zuur [32]. Kutub positif dan negatif aki disambungkan dengan menggunakan konektor dengan bahan kuningan yang menjepit kedua kutub nya, ini dilakukan untuk memudahkan pemasangan kabel. Dikarenakan peletakan aki diluar maka kedua kutub aki ditutup dengan menggunakan karet pelindung agar kutub aki tidak mengalami korosi dan kerusakan akibat rembesan air hujan. Gambar Aki ditunjukkan pada gambar 3.4

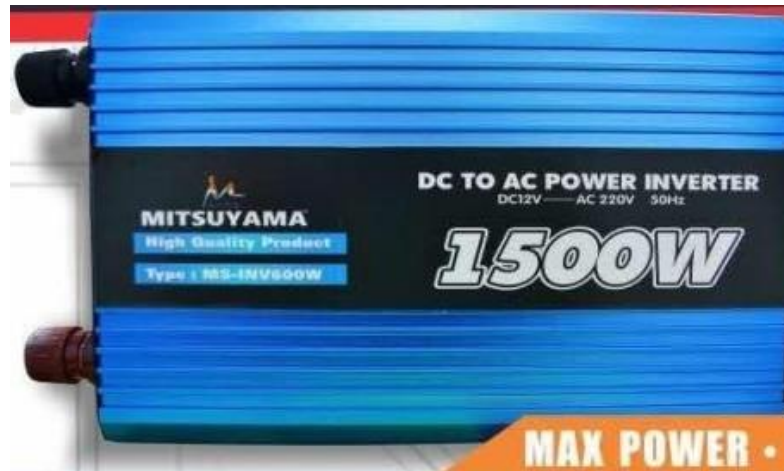


Gambar 3.4. Aki INCOE N100T [33]

3.5.4. Pemilihan *Inverter*

Dipilih *inverter* dengan adanya keluaran *pure sine wave* dikarenakan hasil yang dikeluarkan oleh *inverter* mirip dengan listrik yang ada di rumah. Dipilih *inverter* merk mistuyama dengan kapasitas 1500 Watt, selain memiliki *pure sine wave inverter* ini memiliki 5 sistem perlindungan seperti beban listrik berlebih, temperatur tinggi, arus pendek pada output dan juga perlindungan dari tegangan tinggi dan juga tegangan rendah [34]. Selain itu *inverter* ini juga mudah untuk dibongkar. Awalnya dipilih ini karena ingin mengatur nyala dan mati *inverter*, tetapi karena banyaknya sistem perlindungan membuatnya sangat sulit dilakukan. Maka dari itu penulis memutuskan untuk tidak melakukan perubahan apa apa demi

keamanan dari *Solar Home System*. Gambar 3.5 merupakan gambar *inverter* yang digunakan.



Gambar 3.5. *Inverter* Mitsuyama 1500 W [34]

3.5.5. Pemilihan Relai

Relai yang dipilih adalah relai dual channel dalam 1 modul. Relai ini memiliki tegangan utama atau VCC 5 volt dengan kapasitas pemindahan di 10 A di 250 V AC. Relai ini dipilih dikarenakan dapat menghemat slot kabel dibandingkan harus membeli 2 modul relai. Relai ini juga memiliki 5 volt VCC yang sesuai dengan output dari Arduino yaitu 5 volt.

3.5.6. Pemilihan Sensor Tegangan dan Arus.

Dipilih INA 219 dikarenakan ingin melanjutkan dari perangkat pendahulu. Akan tetapi INA 219 harus memiliki beban untuk melihat arus dan juga tegangannya. Ditambahkannya sensor ACS712 pada output beban berfungsi sebagai sensor arus AC dengan cara mengonversi dari tegangan yang sudah ada.

3.5.7. Pemilihan kartu memori

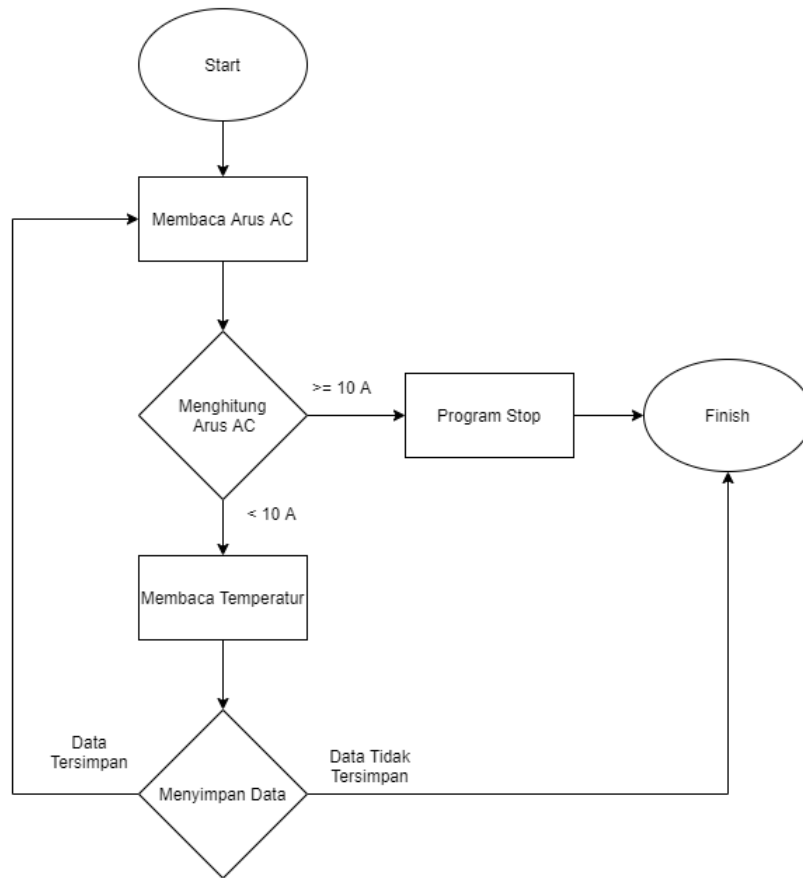
Kartu memori yang dipilih adalah merk V-Gen dengan kapasitas 4 GB dengan format FAT 32. Kartu memori ini dipilih karena kompatibilitasnya terhadap Arduino dan juga modulnya yang hanya dapat membaca dan menulis dengan format FAT 32 tanpa harus melakukan perubahan format yang dapat merusak kartu memori itu sendiri. Gambar 3.6. menunjukkan MicroSD yang digunakan .



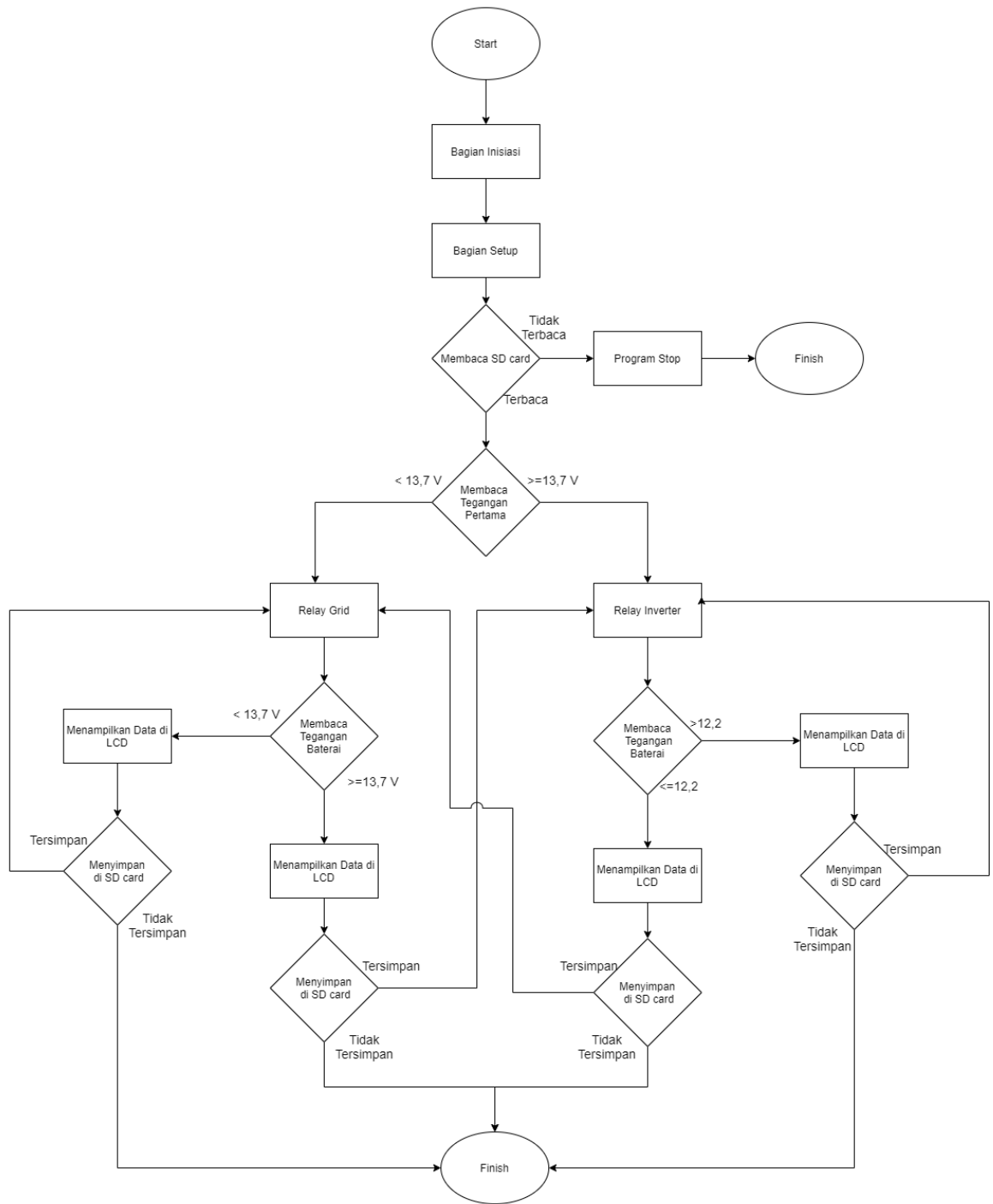
Gambar 3.6. MicroSD V-Gen 4 Gb [35]

3.6. Rancangan Pemograman.

Pemrograman ATS dilakukan dengan 5 bagian. Tahap inisiasi dilakukan dengan cara memasukkan library dan juga inisiasi peralatan yang digunakan. Bagian setup dilakukan pengaturan agar peralatan dapat mengambil data secara benar. Bagian perhitungan merupakan program perhitungan dan rumus yang dipanggil di *switching* program. Bagian *switching* program dan yang terakhir adalah bagian program untuk menampilkan data. Gambar 3.7. menunjukkan alur pemograman I untuk proses *monitoring*. Gambar 3.8. menunjukkan alur pemograman II untuk proses *switching* ATS.



Gambar 3.7. Diagram Alur Pemograman I Untuk Proses *Monitoring*



Gambar 3.8. Diagram Alur Pemograman II Untuk Proses *Switching* ATS

3.6.1. Bagian Inisiasi

Ditahapan ini dilakukan inisiasi dan memasukan library. Library ini bertujuan agar Arduino dapat mengenali peralatan yang ingin disambungkan. Library yang digunakan adalah Wire.h. Library ini digunakan sebagai penghubung Arduino dengan peralatan I2C. Library Adafruit_INA219.h digunakan untuk berkomunikasi dengan sensor INA 219. Library LiquidCrystal_I2C.h digunakan agar Arduino dapat berkomunikasi dengan LCD I2C sebagai tempat data ditampilkan. Lalu untuk inisiasi parameter seperti `const int sensorIn = A7`, `int relay_live = 7`, `int relay_neutral = 8` digunakan sebagai pin pengatur relai dan masuknya data dari sensor ACS712. Bagian ini dapat dilihat pada gambar 3.9.

```
//Tahap pemasukan library
#include <Wire.h>
#include <Adafruit_INA219.h>
#include <LiquidCrystal_I2C.h>
#include <SD.h>
#include <SPI.h>
#include <DS3231.h>

//inisiasi
LiquidCrystal_I2C lcd (0x27, 16, 2);
File myFile;
DS3231 rtc(SDA, SCL);

int pinCS = 53; // Pin untuk mendeteksi kartu sd
const int sensorIn = A7;
int mVperAmp = 185; // pembagi untuk modul ACS 5 ampere
double Voltage = 0;
double VRMS = 0;
double AmpsRMS = 0;
float max_voltage = 0;
float min_voltage = 0;
int relay_live = 7;
int relay_neutral = 8;
float bus_voltage = 0;
float current = 0; // Dalam mili ampere
float power = 0;
int batre = 0; //1 = aki
// State nyala = 1, mati = 0
int state_1;
int state_2;
// inverter = 1, grid = 2
int state_ATS;
```

Gambar 3.9. Koding bagian Inisiasi

3.6.2 Bagian Setup

Di bagian ini digunakan untuk menentukan minimum dan maksimum tegangan di baterai yang berguna untuk menentukan kapan switching dilakukan, selain itu juga dibagian ini digunakan untuk mendapatkan data mentah tegangan baterai yang nantinya menentukan mulainya ATS berada dibagian high atau low. Bagian ini dapat dilihat pada gambar 3.10.

```
void setup() {
  Serial.begin(9600);
  sensor_tegangan.begin();

  pinMode(relay_live, OUTPUT);
  pinMode(relay_neutral, OUTPUT);
  //Test auto read
  bus_voltage = sensor_tegangan.getBusVoltage_V();
  current = sensor_tegangan.getCurrent_mA(); //mili ampere
  batre = 1;
  if(batre == 1) {
    max_voltage = 13.7;
    min_voltage = 12.2;
  }

  digitalWrite(relay_live, HIGH);
  digitalWrite(relay_neutral, HIGH);
  state_ATS = 2;

  Serial.println("mulai");
  //inisiasi SD card
  pinMode(pinCS, OUTPUT);

  if (SD.begin())
  {
    Serial.println("SD card is ready to use.");
  } else
  {
    Serial.println("SD card initialization failed");
    return;
  }
  rtc.begin();
}
```

Gambar 3.10. Koding bagian Setup

3.6.3 Switching Program

Pada bagian ini berisi logika switching ATS. Logikanya berupa jika voltase baterai kurang dari 12% (12.2 V) maka ATS berpindah dari *Inverter* menuju PLN, selain itu jika tegangan baterai lebih dari atau sama dengan 100% (13.7 V) persen maka ATS memindahkan dari PLN menuju *Inverter*. Namun jika awalnya baterai kurang dari 100% maka ATS memindahkan sumber menggunakan PLN. Bagian ini dapat dilihat pada gambar 3.11.

```
//main program
void loop() {
  bus_voltage = 0;
  current = 0;
  bus_voltage = sensor_tegangan.getBusVoltage_V();
  current = sensor_tegangan.getCurrent_mA();
  if(bus_voltage < min_voltage) /
  {
    delay(50);
    digitalWrite(relay_live, LOW);
    digitalWrite(relay_neutral, LOW);
    state_ATS = 2;
  }else if(bus_voltage > max_voltage)
  {
    delay(50);
    digitalWrite(relay_live, HIGH);
    digitalWrite(relay_neutral, HIGH);
    state_ATS = 1;
  }

  //Looping arus AC

  Voltage = getVPP();
  VRMS = (Voltage/2.0) *0.707; //root 2 is 0.707
  AmpsRMS = ((VRMS * 1000)/mVperAmp)-0.1 ;
```

Gambar 3.11. Koding *switching* Program

3.6.4 Bagian Perhitungan Arus AC

Program ini berfungsi sebagai perhitungan awal dari Arus AC pada beban, dikarenakan adanya fluktuasi pada arus AC jadi harus dapat mengumpulkan data selama 10ms yang nantinya diubah menjadi arus pada *switching* program. Gambar 3.12. merupakan koding bagian perhitungan arus AC.


```

//bagian perhitungan arus AC
float getVPP()
{
    float result;
    int readValue;           //data masuk dari sensor
    int maxValue = 0;        // menyimpan data terbesar
    int minValue = 1024;     // menyimpan data terkecil

    uint32_t start_time = millis();
    while((millis()-start_time) < 100) //sample untuk 100 ms
    {
        readValue = analogRead(sensorIn);
        // Logika ketika mempunyai data baru terbesar
        if (readValue > maxValue)
        {
            maxValue = readValue;
        }
        if (readValue < minValue)
        {
            minValue = readValue;
        }
    }

    // perhitungan untuk menjadi arus AC
    result = ((maxValue - minValue) * 5.0)/1024.0;

    return result;
}

```

Gambar 3.12. Koding bagian Perhitungan Arus AC

3.6.5. Print

Pada bagian ini program berfungsi untuk menampilkan data berupa sumber tegangan, tegangan baterai, arus baterai, pada bagian serial monitor dan juga LCD dan juga pada bagian SD card. Pada bagian ini dapat dilihat pada gambar 3.13. untuk serial monitor, Gambar 3.14. untuk LCD dan gambar 3.14. untuk mencetak pada SD card

```

void printVolt(float bus_voltage, float current) {

    Serial.println("Bacaan");
    Serial.print("Bus Voltage: ");
    Serial.print(bus_voltage);
    Serial.println(" V");

    Serial.print("Current: ");
    Serial.print(current);
    Serial.println(" mA");

    delay(70);
}

```

Gambar 3.13. Koding Print Serial Monitor

```

}
delay (70);
lcd.begin();
lcd.backlight();

    lcd.setCursor(0,0);
    lcd.print("V= ");
    //if (bus_voltage < 10.0)
    lcd.print(bus_voltage);
    lcd.print("V");
    lcd.setCursor(0,1);
    lcd.print("I= ");

// if (current < 100.0) lcd.print(" ");
// if (current < 10.0) lcd.print(" ");
    lcd.print(current,1);
    lcd.print(" mA");

    printVolt(bus_voltage, current);
}

```

Gambar 3.14. Koding Print LCD

```

if (state_ATS == 1) {
  Serial.print("Inverter");
}
else if (state_ATS == 2) {
  Serial.print("PLN");
}
myFile = SD.open("test.txt", FILE_WRITE);
if (myFile) {
  myFile.print(rtc.getDateStr());
  myFile.print(",");
  myFile.print(rtc.getTimeStr());
  myFile.print(",");
  myFile.print(int(rtc.getTemp()));
  myFile.print(",");
  myFile.print(bus_voltage);
  myFile.print(",");
  myFile.print(current);
  myFile.print(",");
  myFile.print(AmpsRMS);
  myFile.print(",");
  if (state_ATS == 1) {
    myFile.print("Inverter");
  }
  else if (state_ATS == 2) {
    myFile.print("PLN");
  }
  myFile.println(" ");
  myFile.close(); // close the file
}
// if the file didn't open, print an error:
else {
  Serial.println("error opening test.txt");
}
}

```

Gambar 3.15. Koding Print SD Card

3.7. Pengujian Rancangan ATS

Setelah ATS dirancang akan diuji dengan cara sebagai berikut:

1. Ujicoba Lab pertama. Ujicoba ini dilakukan dengan cara switching sebanyak banyaknya dalam waktu 30 detik. Ini dilakukan untuk melihat keefektifan dari ATS.
2. Ujicoba Lab kedua. Ujicoba ini dengan cara switching sebanyak 20 kali. Ujicoba ini dilakukan untuk melihat *delay* dari ATS.
3. Ujicoba Lapangan. Ujicoba ini dilakukan dengan mengintegrasikan antara *Solar Home System* dengan ATS. Ujicoba ini dilakukan untuk melihat performa dari ATS secara keseluruhan.