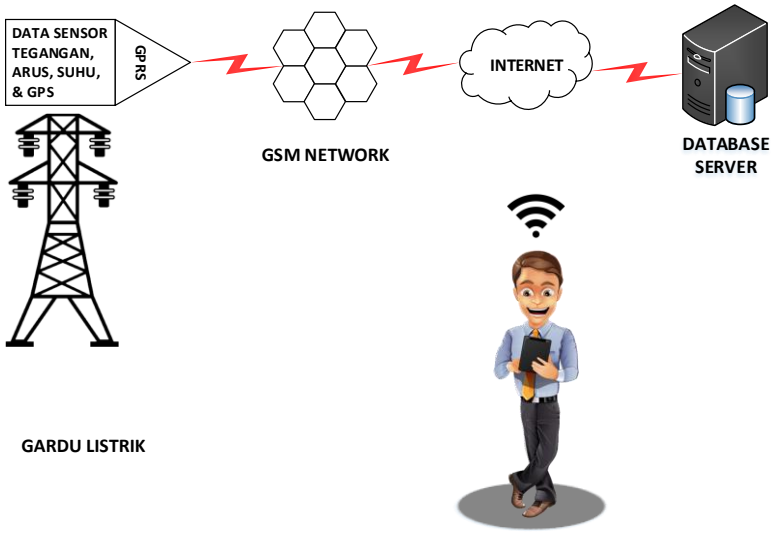
BAB III
PERANCANGAN SISTEM

Pada bab ini akan membahas tentang perancangan perangkat komunikasi *smart grid* pada gardu listrik yang akan mengirimkan data dari sebuah gardu listrik ke *server* basis data menggunakan media transmisi GPRS. Selain itu, pada bab ini juga akan membahas tentang perancangan *interfacing* sensor-sensor dengan Arduino Mega2560 sebagai komponen inti pada sistem yang akan dibangun. Pada gambar di bawah ini merupakan gambaran umum sistem secara keseluruhan.



Gambar 3.1 Gambaran Umum Sistem

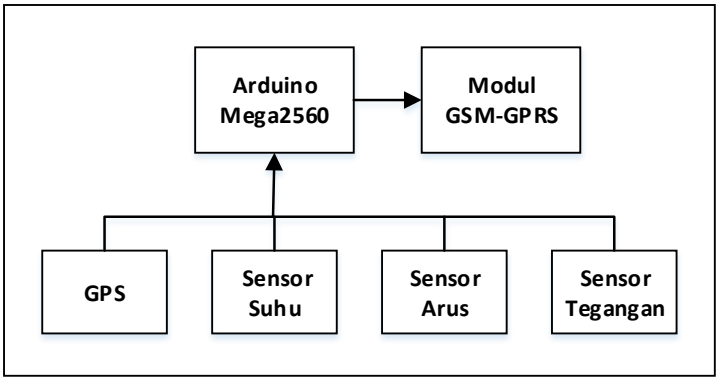
Pada Gambar 3.1 terdapat sebuah gardu listrik yang akan mengirimkan data yang diperoleh dari sensor-sensor melalui media transmisi GPRS menuju *server* basis data yang nantinya akan digunakan oleh pengguna untuk memantau keadaan gardu listrik tersebut melalui *smartphone* berbasis Android ataupun *website*.

Keterangan:

1. *Smartphone* Android dan *website* digunakan untuk menampilkan data hasil pemantauan.
2. Data dikirimkan dari gardu listrik menuju *server* basis data.
3. Sensor yang digunakan berupa sensor tegangan dan arus listrik, serta sensor suhu.

4. GPS digunakan sebagai penentu lokasi tempat gardu listrik berada.
5. *Server* berfungi sebagai tempat penyimpanan data lokasi dan sensor yang digunakan.

Sistem ini dibangun dengan menggunakan komunikasi satu arah yang terdiri dari perangkat pengirim pada gardu listrik dan basis data pada *server* sebagai tempat penyimpanan data gardu listrik. Perangkat pengirim bertugas untuk mengumpulkan data-data sensor kemudian mengirimkannya menggunakan transmisi GPRS. Berikut ini diagram blok dari sistem pengiriman data pada gardu listrik.



Gambar 3.2 Diagram Blok Sistem Pengirim

Keterangan:

1. Modul GSM-GPRS digunakan sebagai penghubung untuk mengirimkan data yang berasal dari sensor-sensor melalui Arduino Mega2560 menggunakan transmisi GPRS.
2. Arduino Mega2560 digunakan sebagai mikrokontroler yang bertugas untuk memproses data yang didapat dari sensor-sensor dan GPS kemudian mengatur pengirimannya menggunakan modul GSM-GPRS.
3. GPS digunakan untuk menentukan lokasi gardu listrik.
4. Sensor tegangan digunakan untuk mengukur tegangan listrik pada transformator gardu listrik.
5. Sensor arus digunakan untuk mengukur arus listrik pada transformator gardu listrik.
6. Sensor suhu digunakan untuk mengukur suhu transformator gardu listrik.

3.1 Komponen Sistem

Terdapat beberapa komponen yang digunakan dalam perancangan dan pembuatan sistem komunikasi *smart grid* dalam melakukan pemantauan terhadap gardu listrik. Komponen tersebut terbagi atas perangkat lunak dan perangkat keras.

3.1.1 Perangkat Lunak

Pada bagian ini menjelaskan perangkat lunak yang dibutuhkan untuk membangun sistem ini, yaitu:

1. Arduino IDE berfungsi sebagai *compiler* dan *editor coding* untuk *board* Arduino Mega2560.
2. SocketTest berfungsi untuk mensimulasikan protokol komunikasi HTTP dengan metode GET.
3. Parallax Serial Terminal digunakan sebagai serial monitor untuk mengkonfigurasi Modul GSM-GPRS SIM900A menggunakan AT Commands.

3.1.2 Perangkat Keras

Pada bagian ini menjelaskan perangkat keras yang dibutuhkan untuk membangun sistem ini yakni terlampir pada tabel berikut.

Tabel 3.1 Spesifikasi Perangkat Keras

Pferangkat Keras	Jumlah	Spesifikasi
Modul GSM-GPRS SIM900A	1	▪ *Dual-Band* 900/1900 MHz ▪ GPRS *multi-slot class* 10/8 GPRS *mobile station class* B ▪ Control via AT *Commands* ▪ PBCCH *support* ▪ CSD *up to* 14.4 kbps ▪ 0710 MUX *protocol* ▪ *embedded* TCP/UDP *protocol* ▪ FTP/HTTP ▪ *Support Text and* PDU *mode* ▪ *Two serial interfaces* ▪ *Analog audio interface* ▪ SPI *interface*

		▪ RTC *backup* ▪ *Coding schame* 1, 2, 3, 4 ▪ *Low power consumption*: 1.5 mA (*sleep mode*) ▪ *Operation temperature*: -40°C to +85°C
Modul *Receiver* GPS Holux GR-89	1	▪ SiRF GSC3f/LP *chipset with embedded* ARM7TDMI CPU ▪ 20-*Channel GPS Receiver for fast acquisition and reacquisition* ▪ 200,000 *effective correlators for fast Time To First Fix* (TTFF), *even at poor satellite signal* ▪ *Power* : *input voltage* 3.3V ~ 5.5V_{DC}, *current*: *less than* 65 mA (*without antenna*) ▪ *Support* NMEA-0183 v2.2 *data protocol and* SiRF *binary code* ▪ *Real time navigation for location based services* ▪ *Position Accuracy* DGPS 1-5 *meters*
Sensor Tegangan listrik ZMPT101B	1	▪ *Single-phase* AC *voltage transformer module* ▪ *Onboard precision* Op-Amp *circuit* ▪ *Can be measured within the* 250V_{AC} *voltage corresponding to the analog output can be adjusted* ▪ *Rated Input Current* 2mA ▪ *Phase angel error* ≤ 20° (*sampling resistance* 100Ω) ▪ *The linear range*: 0 ~ 1000V (0~10mA, *sampling* resistor 100Ω) ▪ *Isolation voltage* 4000V ▪ *Supply voltage* (VCC): 5-30V ▪ *Output voltage*: ½ VCC ▪ *Operating temperature*: -40°C - +70°C
Sensor Arus listrik ACS712	1	▪ *Low-noise analog signal path* ▪ 80 kHz *bandwidth* ▪ *Total output error* 1.5% *at* T_A = 25°C

		▪ 1.2 mΩ *internal conductor resistance* ▪ 2.1 kV_{RMS} *minimum isolation voltage* ▪ 66 *to* 185 mV/A *output sensitivity* ▪ *Output voltage proportional to* AC *or* DC *currents* (*up to* 30 A) ▪ *Supply Voltage* 4.5-5.5 V ▪ *Supply Current* 10mA ▪ *Operating Internal Leadframe Temperature* –40°C - 85°C
Sensor DHT22 (AM2302)	1	▪ *Power* supply: 3.3 – 6 V_{DC} ▪ *Current supply*: 1-1.5 mA ▪ *Output signal*: *digital signal* ▪ *Sensing element*: *Polymer capacitor* ▪ *Operating range*: *humidity* 0-100%RH; *temperature* -40~80°C ▪ *Accuracy*: *humidity* ±2%RH(*Max* ±5%RH); *temperature* < ±0.5°C ▪ *Repeatability*: *humidity* ±1%RH; *temperature* ±0.2°C ▪ *Humidity hysteresis*: ±0.3%RH ▪ *Long-term Stability*: ±0.5%RH/*year* ▪ *Sensing* period: *Average* 2s
Arduino Mega2560	1	▪ *Microcontroller ATmega2560* ▪ *Operating Voltage* 5 V_{DC} ▪ *Input Voltage (recommended)*: 7-12 V_{DC} ▪ *Input Voltage (limit)*: 6-20 V_{DC} ▪ *Digital I/O Pins*: 54 (*of which* 15 *provide* PWM *output*) ▪ 16 *Analog Input Pins* ▪ *DC Current per I/O Pin*: 20 mA ▪ *DC Current for* 3.3V *Pin*: 50 mA ▪ 256 *KB Flash Memory* (8 KB *used by bootloader*) ▪ 8 KB SRAM ▪ 4 KB EEPROM ▪ *Clock Speed* 16 MHz ▪ *Dimention*: 101.52 mm x 53.4 mm

3.2 Instalasi Sistem

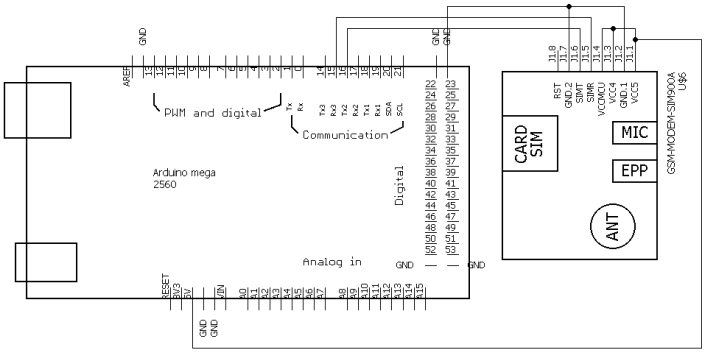
Dalam instalasi sistem diperlukan beberapa tahapan yang akan dilakukan mulai dari konfigurasi Arduino Mega2560, konfigurasi modul GSM-GPRS SIM900A, konfigurasi sensor-sensor, serta konfigurasi GPS *Receiver*.

3.2.1 Konfigurasi Arduino Mega2560

Konfigurasi Arduino Mega2560 dilakukan menggunakan *software* standar Arduino IDE. Pada sistem yang akan dibangun, Arduino Mega2560 berperan sebagai mikrokontroler yang yang akan membaca data dari sensor-sensor dan mengirimkannya ke *server* basis data melalui transmisi GPRS menggunakan modul GSM-GPRS SIM900A.

3.2.2 Konfigurasi Modul GSM-GPRS SIM900A sebagai Pengirim

Modul GSM-GPRS SIM900A dikonfigurasi menggunakan serial monitor pada *software* Arduino IDE. Namun sebelum dikonfigurasi, modul GPRS terlebih terlebih dahulu dirangkai dengan pin komunikasi serial pada Arduino Mega2560. Gambar berikut merupakan skematik rangkaian modul GSM-GPRS SIM900A dengan Arduino Mega2560.



Gambar 3.3 Rangkaian Modul GSM-GPRS SIM900A

Sebelum mengkonfigurasi modul GSM-GPRS SIM900A dengan Arduino Mega2560, ada beberapa langkah yang perlu diperhatikan yaitu:

1. Pastikan pin *transmitter* (TXD) dan *receiver* (RXD) pada modul GSM-GPRS SIM900A terhubung ke pin RX02 dan TX02 pada Arduino Mega2560.
2. Jalankan program Arduino IDE, lalu *upload* program berikut ini untuk menguji sambungan GPRS menggunakan komunikasi TCP.

```
char c = ' ';

void setup()
{
    Serial.begin(9600);
    while(!Serial);
    Serial.println("KONFIGURASI MODUL GSM-GPRS SIM900A");

    Serial2.begin(19200);
    delay(1000);
    Serial.println("SETUP MODUL GSM-GPRS SIM900A TELAH SIAP");
}

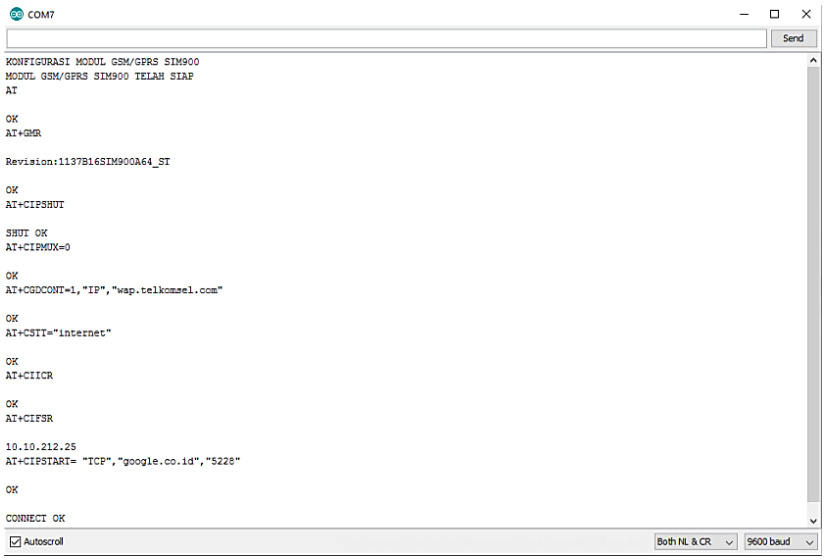
void loop()
{
    if (Serial2.available())
    {
      c = Serial2.read();
      Serial.write(c);
    }

    if (Serial.available())
    {
      c = Serial.read();
      Serial2.write(c);
    }

}
```

3. Setelah berhasil melakukan *upload* program, buka serial monitor dan ubah serial *baudrate* menjadi 19200 dengan mode *Both* NL & CR.
4. Lakukan konfigurasi modul GSM-GPRS SIM900A menggunakan AT *Command* SIM900A.

Pada Gambar 3.4 merupakan serial monitor dari Arduino IDE untuk konfigurasi modul GSM-GPRS SIM900A.

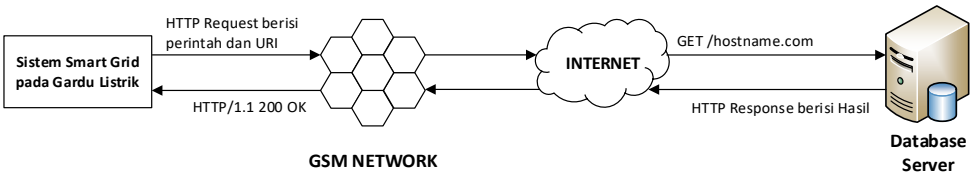


Gambar 3.4 Konfigurasi SIM900A pada Serial Monitor Arduino IDE

Penjelasan konfigurasi modul GSM-GPRS SIM900A menggunakan serial monitor Arduino IDE dengan AT Command dipaparkan sebagai berikut:

1. Perintah *AT* digunakan untuk memastikan modul GSM-GPRS SIM900A terhubung dengan baik.
2. Perintah *AT+GMR* digunakan untuk mengecek versi *firmware* dari modul GSM-GPRS SIM900A.
3. Perintah *AT+CIPSHUT* digunakan untuk menghentikan sambungan protokol IP pada sesi sebelummnya.
4. Perintah *AT+CIPMUX=0* digunakan untuk memilih mode komunikasi satu arah.
5. Perintah *AT+CGDCONT=1,"IP","wap.telkomsel.com"* digunakan untuk mengaktifkan (1) Internet *Protocol* (IP) dan menghubungkannya dengan accsess point jaringan seluler yang digunakan (wap.telkomsel.com).
6. Perintah *AT+CSTT="internet"* digunakan untuk memasukkan *Access Point Name* (APN) dari jaringan seluler yang digunakan.
7. Perintah *AT+CIICR* digunakan untuk mengaktifkan sambungan GPRS.
8. Perintah *AT+CIFSR* digunakan untuk menampilkan alamat IP DHCP yang digunakan.
9. Perintah *AT+CIPSTART="TCP","google.co.id","80"* digunakan untuk mencoba koneksi TCP/IP ke alamat **http://google.co.id** dengan port komunikasi 80.

Selain itu, pada sistem yang dibangun akan menggunakan protokol komunikasi HTTP agar dapat berkomunikasi dengan *server*. Berikut ini merupakan alur komunikasi dari sistem menuju *server* basis data.



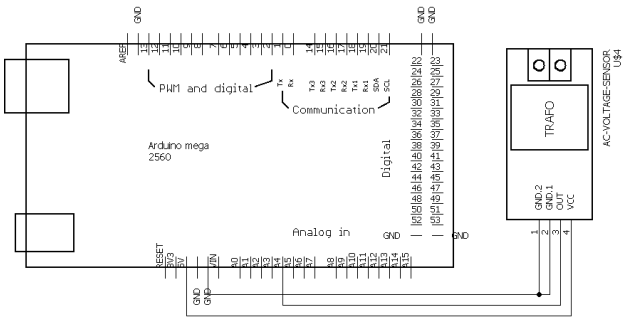
Gambar 3.5 Perancangan Alur Komunikasi

Untuk dapat berkomunikasi dengan *server* menggunakan protokol HTTP diperlukan tahapan yang harus dilalui yaitu:

1. *Client* membuat koneksi TCP dengan alamat IP dari hostname.com yang dituju.
   - *Client* mengirim paket SYN.
   - Web *server* mengirim paket SYN-ACK.
   - *Client* mengirim paket ACK ke web *server* serta membuat koneksi TCP tiga segmen.
2. *Client* mengirim HTTP *request* menuju web *server*.
3. *Client* menunggu respon permintaan dari *server*.
   - Web *server* memproses permintaan, menemukan *resource*, dan mengirimkan tanggapan ke *Client*. *Client* menerima segmen pertama dari web *server* yang berisi *header* dan konten HTTP *Response*.
4. *Client* memuat isi tanggapan dari *server*.
   - Web *server* mengirimkan paket TCP segmen kedua dengan PSH *flag*.
   - *Client* mengirimkan ACK. (*Client* mengirimkan ACK setiap dua segmen yang diterimanya dari web *server*)
   - Web *server* mengirimkan segmen TCP ketiga dengan HTTP_Continue.
5. *Client* mengirimkan sebuah paket FIN untuk menutup sambungan TCP.

3.2.3 Konfigurasi Sensor ZMPT101B

Sensor ZMPT101B merupakan modul sensor pengukur tegangan listrik arus bolak-balik (AC). Agar dapat melakukan pengukuran tegangan listrik AC, maka sensor ZMPT101B akan dirangkai dengan Arduino Mega2560. Berikut konfigurasi rangkaiannya.



Gambar 3.6 Rangkaian Sensor ZMPT101B

Selain itu, konfigurasi sensor ZMPT101B juga dilakukan pada *coding* pemrograman Arduino untuk membaca data tegangan listrik AC yang terukur. Sebelum mengukur tegangan listrik, sensor tegangan ZMPT101B dilakukan pengkalibrasian agar data terbaca dengan baik. Pada Tabel 4.1 merupakan teknik kalibrasi sensor tegangan tegangan pada perangkat. Sedangkan pada Tabel 4.3 merupakan hasil pengujian sensor tegangan pada perangkat berdasarkan hasil kalibrasi.

Tabel 3.2 Kalibrasi Sensor Tegangan ZMPT101B

No	Tegangan listrik terukur pada perangkat (Volt)	Data ADC yang terbaca pada Arduino Serial
1	0	515
2	6.7	525
3	9.5	530
4	12.1	536
5	14.7	541
6	222	696

Kalibrasi sensor tegangan ZMPT101B menggunakan persamaan (2.1), dengan menyatakan tegangan yang terbaca pada perangkat sebagai *x*, dan data ADC yang terbaca pada Arduino sebagai *y* ,maka didapatkan nilai pada tabel seperti berikut:

Tabel 4.3 Data Pengukuran Regresi Linear

No.	x	y	x^2	xy	y^2
1	0	511	0	0	261121
2	6.23	516	38.81	3214.68	266256
3	9.2	522	84.64	4802.4	272484
4	12.3	527	151.29	6482.1	277729
5	15.31	531	234.40	8129.61	281961

6	18.3	536	334.89	9808.8	287296
7	232.5	745	54056.25	173212.5	555025
∑	293.84	3888	54900.279	205650.09	2201872

1. Perhitungan untuk menentukan nilai konstanta a menggunakan persamaan (2.2):

$$a = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n\sum x^2 - (\sum x)^2}$$

$$a = \frac{(3888)(54900.279) - (293.84)(205650.09)}{7(54900.279) - (293.84)^2} = 513.57$$

2. Perhitungan untuk menentukan nilai konstanta b menggunakan persamaan (2.3):

$$b = \frac{n\sum xy - (\sum x)(\sum y)}{n\sum x^2 - (\sum x)^2}$$

$$b = \frac{7(205650.09) - (293.84)(3888)}{7(54900.279) - (293.84)^2}$$

$$b = \frac{297100.71}{297960.007} = 0.997116$$

Dari hasil perhitungan tersebut didapatkan persamaan sebagai berikut:

$$y = 513.57 + 0.997116x$$

Untuk mencari nilai tegangan, maka digunakan persamaan (2.4), sehingga menjadi:

$$x = (y - 513.57)/0.997116$$

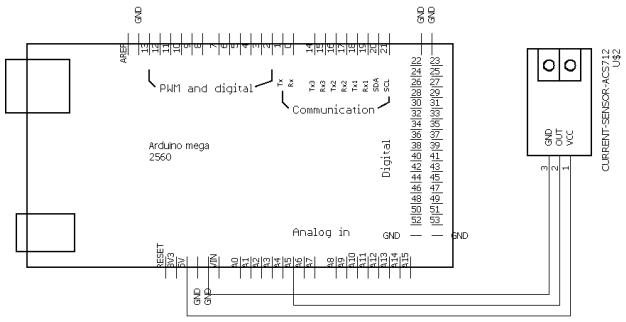
Setelah didapatkan persamaan untuk mendapatkan nilai tegangan yang terukur (persamaan mencari nilai *x*), persamaan tersebut langsung diimplementasikan ke *coding* Arduino pada Arduino IDE dan di-*upload* ke board Arduino Mega2560. Berikut ini merupakan potongan *coding* pada pemrograman Arduino untuk mengukur tegangan listrik menggunakan sensor tegangan listrik ZMPT101B.

```
double sensorValue=0;
double sensorValue1=0;
int climbhill=0;
double VmaxD=0;
double VeffD;
double Veff;
VmaxD=0;
  sensorValue1=sensorValue;
  delay(100);
  sensorValue = analogRead(A4);

  if (sensorValue>sensorValue1 && sensorValue>512){
    climbhill=1;
    VmaxD=sensorValue;
  }
  if (sensorValue<sensorValue1 && climbhill==1){
    climbhill=0;
    VmaxD=sensorValue1;
    Veff=(VmaxD - 513.57) / 0.997116;
  }
Serial.print("Tegangan = ");
Serial.println(Veff);
Delay(500);
```

3.2.4 Konfigurasi Sensor ACS712

Sensor ACS712 merupakan modul sensor pengukur arus listrik yang dapat digunakan untuk mengukur arus listrik bolak-balik (AC). Agar dapat melakukan pengukuran arus listrik, maka sensor ACS712 akan dikonfigurasikan dengan Arduino Mega2560. Berikut merupakan konfigurasi rangkaiannya dengan Arduino Mega2560.



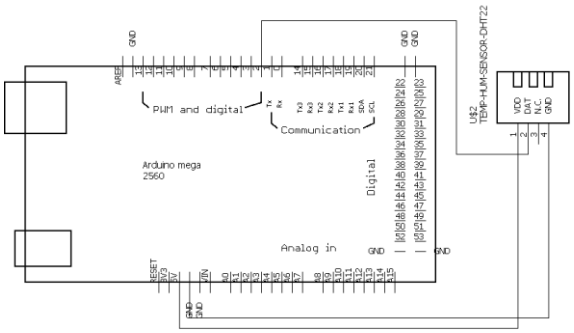
Gambar 3.7 Rangkaian Sensor ACS712

Dengan menggunakan persamaan (6) yang diimplementasikan pada *coding* Arduino, maka arus listrik akan dapat terukur. Selanjutnya adalah dilakukan pemrograman sensor pada Arduino IDE agar dapat membaca pengukuran arus listrik AC yang terukur. Berikut ini adalah potongan *coding* pada pemrograman Arduino untuk melakukan pengukuran arus listrik.

```
int mVperAmp = 185; // use 100 for 20A Module and 66 for 30A Module
double Voltage = 0;
double VRMS = 0;
double AmpsRMS = 0;
Voltage = getVPP();
VRMS = (Voltage/2.0) * 0.707106;
AmpsRMS = (VRMS * 1000)/mVperAmp;
float getVPP() {
  float result;
  int readValue;             //value read from the sensor
  int maxValue = 0;          // store max value here
  int minValue = 1023;       // 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;
       }
   }
   result = ((maxValue - minValue) * 5.0)/1023;
   return result;
 }
```

3.2.5 Konfigurasi Sensor DHT22

Sensor DHT22 adalah modul sensor yang yang dapat mengukur suhu dan kelembaban di sekitarnya. Pada perancangan sistem ini, hanya data suhu sajalah yang akan di ambil dari sensor DHT22. Untuk itu maka dirangkailah sensor DHT22 dengan Arduino Mega2560. Berikut konfigurasi rangkaiannya.



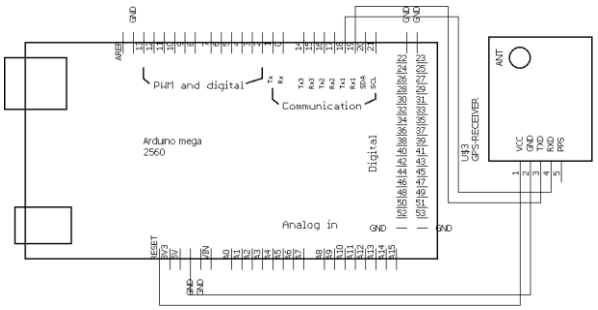
Gambar 3.8 Rangkaian Sensor DHT22

Selain itu, konfigurasi sensor DTH22 juga dilakukan pada *coding* pemrograman Arduino untuk membaca data suhu yang terukur. Berikut ini merupakan potongan *coding* pada pemrograman Arduino untuk mengukur suhu menggunakan DHT22.

```
#include "DHT.h"
#define DHTPIN 2         // what digital pin we're connected to
#define DHTTYPE DHT22   // DHT 22 (AM2302), AM2321
void setup() {
  Serial.begin(9600);
  Serial.println("DHTxx test!");
  dht.begin();
}
void loop() {
   delay(2000);
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.print(" *C ");
}
```

3.2.6 Konfigurasi Receiver GPS Holux GR-89

GPS *receiver* Holux GR-89 merupakan sebuah modul penerima data koordinat titik di bumi dari beberapa satelit. Agar dapat memerima data dari satelit, modul *receiver* GPS ini akan dikonfigurasi dengan Arduino Mega2560. Berikut konfigurasi rangkaiannya.



Gambar 3.9 Rangkaian *Receiver* GPS Holux GR-89

Setelah rangkaian selesai disusun, selanjutnya adalah dilakukan pemrograman pada Arduino IDE agar dapat membaca lokasi pada gardu listrik. Berikut ini adalah potongan *coding* pada pemrograman Arduino untuk membaca lokasi saat ini.

```
#include <TinyGPS.h>
TinyGPS gps;
void setup()
{
  Serial.begin(9600);
  Serial1.begin(9600);
}
void loop()
{
  bool newData = false;
  unsigned long chars;
  unsigned short sentences, failed;
  for (unsigned long start = millis(); millis() - start < 1000;)
  {
    while (Serial1.available())
    {
      char c = Serial1.read();
      if (gps.encode(c)) // Did a new valid sentence come in?
        newData = true;
    }
  }
  if (newData)
  {
    unsigned long age;
    gps.f_get_position(&flat, &flon, &age);
    Serial.print("LAT=");
    Serial.print(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6);
    Serial.print(" LON=");
    Serial.print(flon == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 6);
}
```