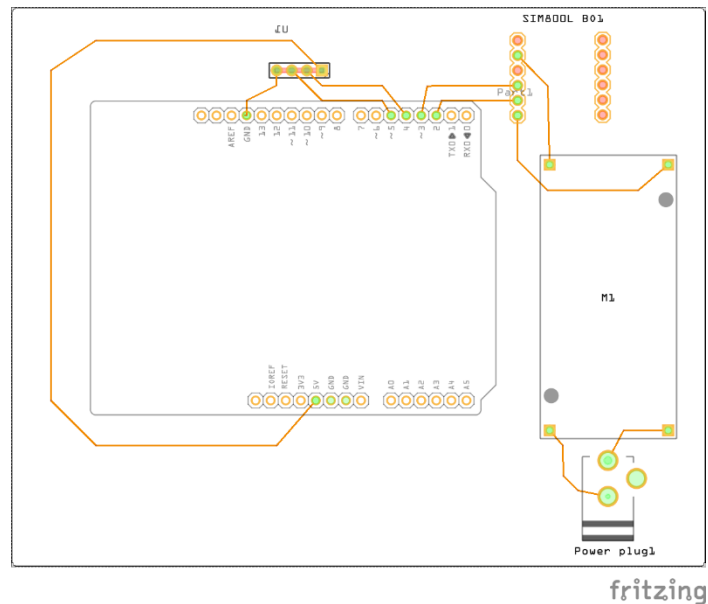


3.3 Realisasi

Sub sistem penerima ini akan direalisasikan menjadi dua bagian, yaitu realisasi perangkat keras dan realisasi perangkat lunak.

3.3.1. Realisasi Perangkat Keras

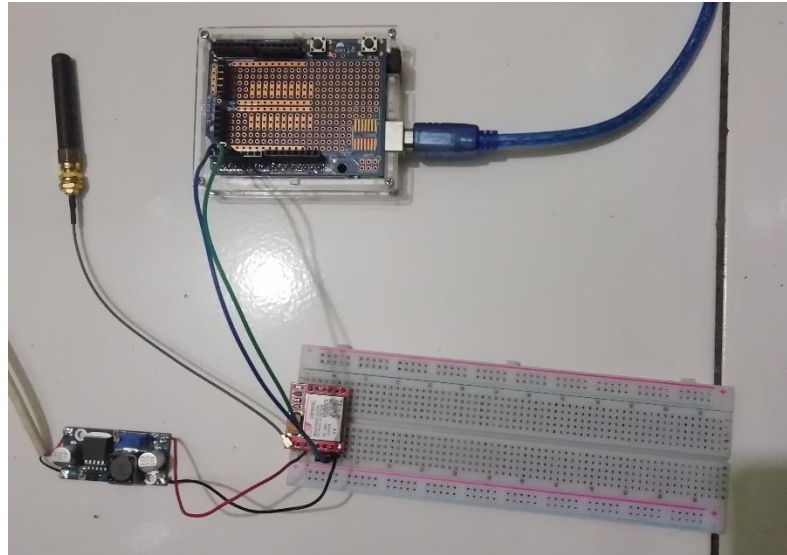
3.3.1.1. Realisasi PCB



Gambar 3. 1 Layout PCB

Gambar tersebut menunjukkan *layout* PCB yang direalisasikan dari skema elektronik yang telah dibuat sebelumnya yang ditunjukkan pada gambar 3.3. *Layout* PCB ini dibuat menggunakan aplikasi *fritzing*. Kemudian *layout* PCB ini akan dicetak agar menjadi sebuah papan PCB.

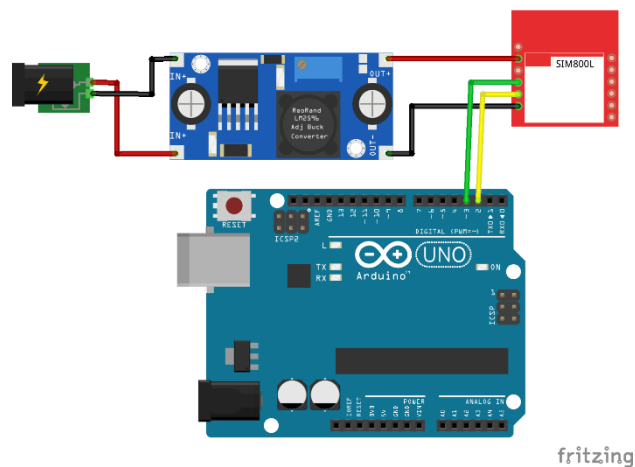
3. 3. 1. 2. Realisasi Perakitan



Gambar 3. 2 Realisasi Perakitan

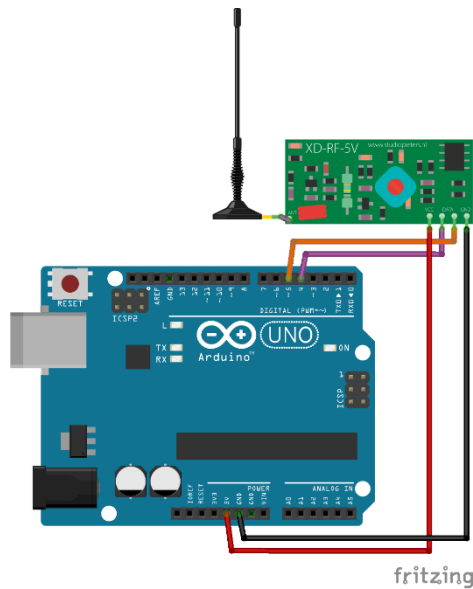
Gambar 3.9 menampilkan realisasi perakitan yang dilakukan pada papan percobaan. Modul SIM800L diletakan pada papan percobaan dan diberi sumber tegangan 3,8V yang berasal dari modul penurun tegangan DC dengan *input* 5V. SIM800L dihubungkan dengan antenna eksternal melalui konektor u.fl. Kemudian pin RX pada SIM800L dihubungkan menggunakan kabel *jumper* dengan pin D3 pada *Arduino Uno*, sedangkan pin TX pada SIM800L dihubungkan dengan pin D2 pada *Arduino Uno*.

3. 3. 1. 3. Realisasi Pengkabelan



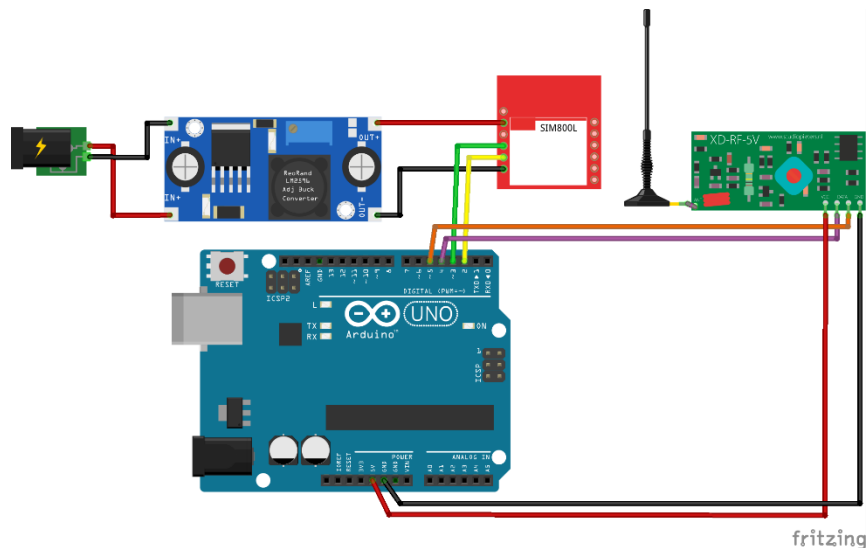
Gambar 3. 3 Sistem Pengkabelan SIM800L

Gambar tersebut menunjukkan sistem pengkabelan untuk modul SIM800L yang dibuat menggunakan aplikasi *fritzing*. Tegangan *input* untuk SIM800L dihasilkan dari modul penurun tegangan DC LM2596 yang diberi tegangan sumber sebesar 5V DC. Modul penurun tegangan ini diatur agar menghasilkan tegangan *output* sebesar 3,8V yang digunakan sebagai tegangan *input* modul SIM800L. Pin RX pada SIM800L terhubung dengan pin D3 pada *Arduino Uno* sedangkan pin TX pada SIM800L terhubung dengan pin D2 pada *Arduino Uno*.



Gambar 3. 4 Sistem Pengkabelan Penerima Radio 433 MHz

Gambar tersebut menunjukkan sistem pengkabelan untuk penerima radio 433MHz yang dibuat menggunakan aplikasi *fritzing*. Pin vcc pada penerima radio dihubungkan dengan *output* 5V pada *Arduino Uno* sedangkan pin *ground* pada penerima radio dihubungkan dengan pin *ground* pada *Arduino Uno*. Pin RX pada penerima radio dihubungkan dengan pin D4 pada *Arduino Uno* dan pin TX pada penerima radio dihubungkan dengan pin D5 pada *Arduino Uno*.



Gambar 3. 5 Sistem Pengkabelan Keseluruhan

Gambar tersebut merupakan sistem pengkabelan keseluruhan pada sub-sistem penerima yang telah dibuat dengan menggunakan aplikasi *fritzing*. Semua modul yang digunakan terhubung dengan mikrokontroler. Sistem pengkabelan ini kemudian akan direalisasikan dalam bentuk sistem pengkabelan fisik yang sesungguhnya.

3. 3. 2. Realisasi Perangkat Lunak

3. 3. 2. 1. Realisasi Program

Sub-sistem penerima ini menggunakan *Arduino Uno* sebagai mikroprosesor untuk pengolahan data. Mikrokontroler tersebut diisi oleh program yang dibuat dengan *software Arduino*.

```
#include <SoftwareSerial.h>
SoftwareSerial sim800(2,3); //RX,TX
float ketinggian=0.0;
float suhu=25.0;
```

Gambar 3. 6 Program Inisialisasi

Gambar 3.9 menunjukkan inisialisasi untuk pin RX TX dari SIM800L yaitu pin RX pada pin D2 dan pin TX pada pin D3. Selain itu juga diinisialisasi suatu tipe data *float* sebagai contoh data hasil pengamatan ketinggian dan suhu.

```

void setup() {
  // put your setup code here, to run once:
  sim800.begin(9600);
  Serial.begin(9600);
  delay(1000);
}

```

Gambar 3. 7 Program Deklarasi

Gambar 3.10 menunjukkan deklarasi untuk mengubah *baudrate* menjadi 9600. *Baudrate* yang digunakan untuk *Arduino* sama dengan *baudrate* yang digunakan untuk modul SIM800L untuk meminimalisasi adanya *error* saat proses pengiriman maupun penerimaan data.

```

void loop() {
  // put your main code here, to run repeatedly:
  ketinggian = ketinggian+1.0;
  suhu=suhu+0.1;
  senddata();
  delay(5000);
  if(sim800.available()){
    Serial.write(sim800.read());
  }
}

```

Gambar 3. 8 Program Utama

Gambar 3.11 menunjukkan program utama yang akan terus menerus diulang (*looping*). Program tersebut dimanipulasi untuk mengubah data suhu dan ketinggian agar bertambah secara konstan agar memudahkan melihat perbedaan data saat pengiriman ke *platform IoT*.

```

void senddata() {
  sim800.println("AT");
  delay(1000);
  serialdata();
  sim800.println("AT+CPIN=?");
  delay(1000);
  serialdata();
  sim800.println("AT+CREG?");
  delay(1000);
  serialdata();
  sim800.println("AT+CGATT?");
  delay(1000);
  serialdata();
  sim800.println("AT+CIPSHUT");
  delay(2000);
  serialdata();
  sim800.println("AT+CIPMUX=0");
  delay(2000);
  serialdata();
  sim800.println("AT+CSTT=\"internet\"");
  delay(1000);
  serialdata();
  sim800.println("AT+CIICR");
  delay(3000);
  serialdata();
  sim800.println("AT+CIFSR");
  delay(2000);
  serialdata();
}

```

Gambar 3. 9 Persiapan Pengiriman Data

Gambar 3.12 menunjukkan program untuk mempersiapkan SIM800L agar bisa menghasilkan koneksi GPRS sebelum dapat mengirimkan data ke *Platform IoT*.

```

sim800.println("AT+CIPSPRT=0");
delay(3000);
serialdata();
sim800.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"80\"");
delay(3000);
serialdata();
sim800.println("AT+CIPSEND");
delay(2000);
serialdata();
String str="GET http://api.thingspeak.com/update?api_key=6Q9XHRGVVV0FS50Bsfiedl=" + String(ketinggian) + String("&field3=") + String(suhu);
sim800.println(str);
delay(2000);
serialdata();
sim800.println((char)26);
delay(2500);
sim800.println();
serialdata();
sim800.println("AT+CIPSHUT");
delay(1000);
serialdata();
}

void serialdata() {
  while(sim800.available() != 0) {
    Serial.write(sim800.read());
  }
}

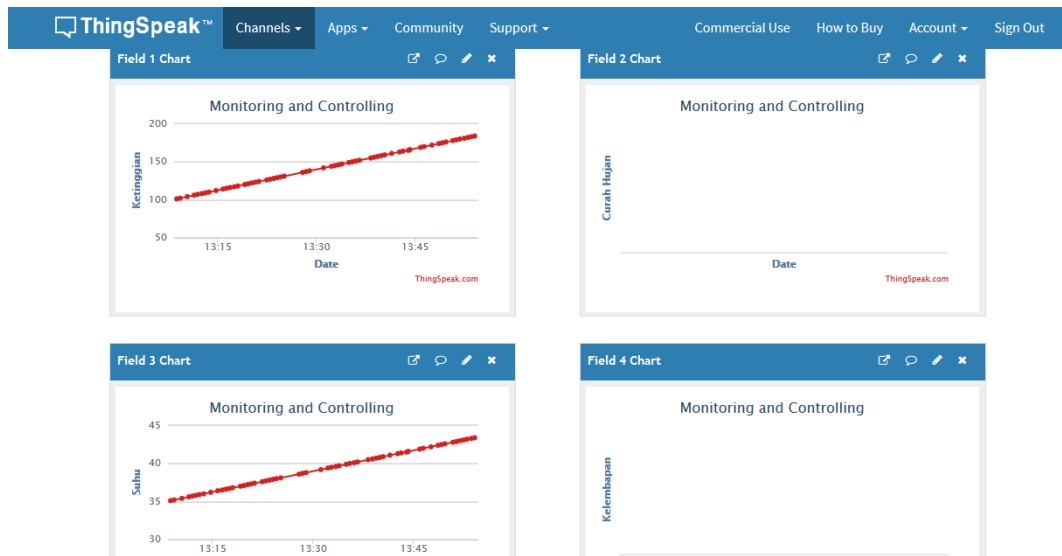
```

Gambar 3. 10 Program Pengiriman Data

Gambar 3.13 adalah program untuk mengirimkan data ke *Platform IoT* yaitu *Thingspeak* yang telah dipersiapkan sebelumnya. Program ini memanfaatkan *ATCommand* dan koneksi HTTP untuk mengirimkan data ketinggian dan suhu yang telah dideklarasikan sebelumnya.

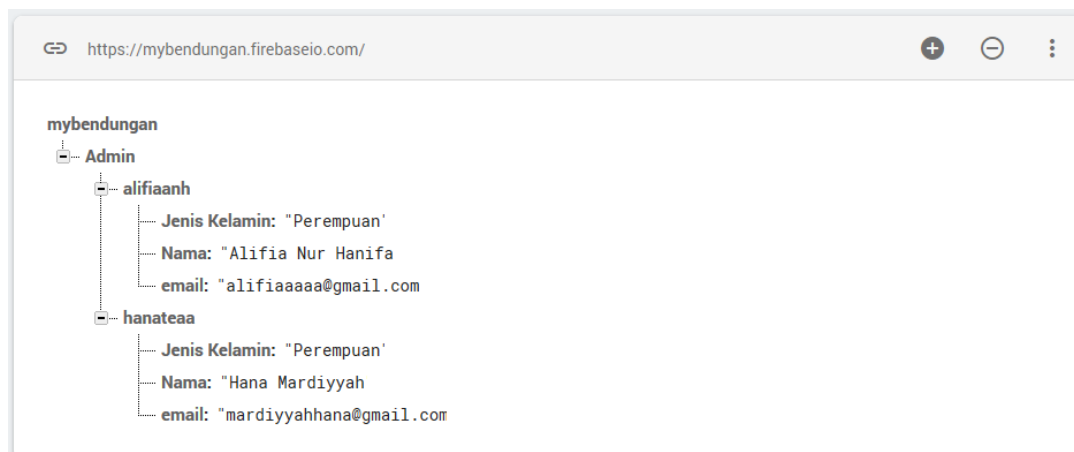
3. 3. 2. 2. Realisasi Database

Database yang digunakan pada aplikasi *monitoring* dan *controlling* bendungan ini ada dua, yaitu *Platform IoT* yang digunakan sebagai perantara pengiriman data dari mikrokontroler ke *internet* dan *Google Firebase* yang digunakan untuk menyimpan data admin petugas penjaga pintu air bendungan yang diberi akses untuk dapat mengatur buka/tutup pintu air bendungan dari aplikasi.



Gambar 3. 11 *Platform IoT*

Gambar 3.14 merupakan tangkapan layar *platform IoT* yang digunakan untuk menampung data dari mikrokontroler yang selanjutnya akan ditampilkan pada aplikasi *Android*.



Gambar 3. 12 Tampilan *Google Firebase*

Gambar 3.15 merupakan tampilan data pada *google firebase* yang menyimpan data identitas admin petugas pintu air bendungan. Admin yang terdaftar pada *database* tersebut dapat melakukan *login* pada aplikasi dan memiliki akses untuk control buka/tutup pintu air bendungan.