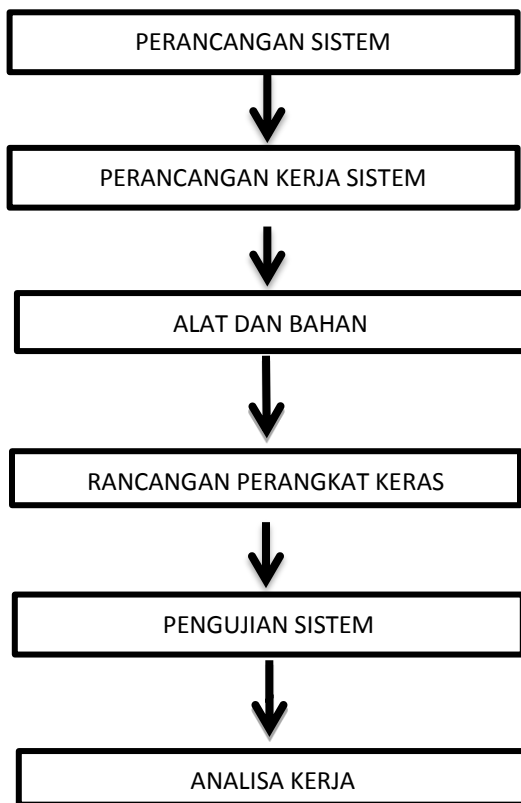


BAB III METODOLOGI PENELITIAN

3.1 Perancangan Sistem

Bab ini akan menjelaskan langkah - langkah penelitian yang akan dilakukan dalam membangun dan merancang Sistem Alarm Kebakaran Portable Pada Pos Pendakian Menggunakan SMS (*Short Message Service*) Berbasis Arduino. Alur penelitian yang digunakan seperti pada gambar 3.1.



Gambar 3.1. Alur Penelitian

3.1.1 Analisa Kebutuhan

Tahapan selanjutnya setelah membuat rancangan perangkat keras yaitu membuat analisa kebutuhan sistem. Analisa kebutuhan sistem dilakukan untuk mengetahui alat dan komponen serta perangkat lunak apa saja yang akan digunakan untuk mengimplementasikan sistem.

3.1.1.1 Alat

Sebelum membuat Rancang Bangun Sistem Alarm Kebakaran Portable Pada Pos Pendakian Menggunakan SMS (*Short Message Service*) Berbasis Arduino, ada beberapa peralatan yang harus disiapkan. Daftar peralatan yang digunakan dalam penelitian ini akan dituliskan pada Tabel 3.1.

Tabel 3.1. Alat Yang Dibutuhkan

No	Nama Alat	Spesifikasi	Fungsi	Jumlah
1	Komputer/ laptop	Window 7-10 32/64bit	Untuk membuat sebuah aplikasi yang akan di pakai di perangkat keras dan perangkat lunak	1 unit
2	Multitester	Analog/Digital	digunakan untuk mengukur tegangan (ACV-DCV), dan kuat arus (mA- μ A)	1 buah
3	Obeng	Obeng + dan -	Untuk merangkai alat	1 buah
4	Solder	-	Untuk menempelkan timah ke komponen	1 buah
5	Bor pcb	-	Untuk membuat lobang baut atau komponen	1 buah
6	Tang Potong	-	Untuk memotong kabel dan kaki komponen	1 buah
7	Kit Arduino	-	Komponen Komplit <i>Arduino Uno</i>	1 buah

3.1.1.2 Komponen

Sebelum membuat Rancang Bangun Sistem Alarm Kebakaran Portable Pada Pos Pendakian Menggunakan SMS (*Short Message Service*) Berbasis Arduino, ada beberapa komponen yang harus disiapkan. Daftar komponen yang digunakan dalam penelitian ini akan dituliskan pada Tabel 3.2.

Tabel 3.2. Komponen Yang Dibutuhkan

No	Nama Alat	Spesifikasi	Fungsi	Jumlah
1	Kit <i>Arduino Uno</i>	AT 328	Sebagai proses perintah yang akan di jalankan	1
2	Sensor MQ2	-	Digunakan sebagai inputan untuk membaca adanya asap	1
3	<i>Mini Sirine</i>	-	Sebagai alarm	1
4	<i>GSM Shiled</i>	-	Digunakan untuk mengirimkan informasi berupa SMS kepada user (petugas cagar alam)	1
5	Sensor Api	-	Digunakan sebagai inputan untuk membaca adanya api	1
6	<i>Relay</i>	5 volt	Digunakan untuk mengaktifkan <i>mini sirine</i>	1
7	LCD	16x2	Digunakan sebagai outputan berupa tulisan	1

3.1.1.3 Software

Sebelum membuat Rancang Bangun Sistem Alarm Kebakaran Portable Pada Pos Pendakian Menggunakan SMS (*Short Message Service*) Berbasis Arduino, ada beberapa *Software* yang harus disiapkan. Daftar *software* yang digunakan dalam penelitian ini akan dituliskan pada Tabel 3.3.

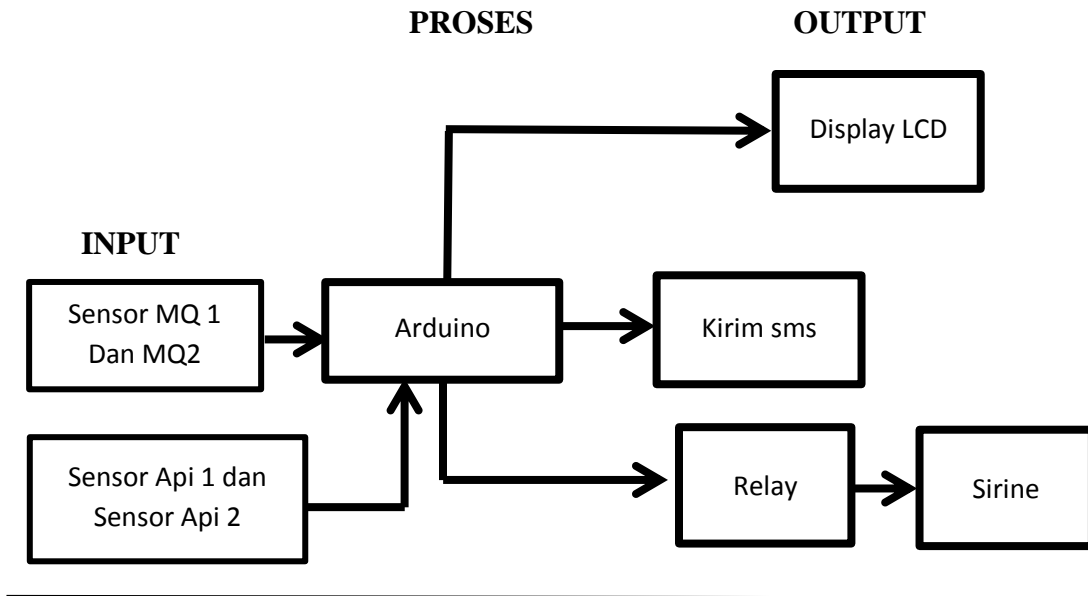
Tabel 3.3. Daftar *Software* yang digunakan

No	Nama	Spesifikasi	Fungsi
1	IDE Arduino	Arduino 1.6.3	Membuat program yang akan di-download perangkat arduino
2	Proteus	7.1 Profesional	Merancang rangkaian yang akan digunakan untuk membuat alat

3.1.2 Perancangan Kerja Sistem

Rancang Bangun Sistem Alarm Kebakaran Portable Pada Pos Pendakian Menggunakan SMS (*Short Message Service*) Berbasis Arduino ini meliputi perancangan perangkat keras (*hardware*) dan perangkat lunak (*software*). Sistem yang dirancang akan membentuk suatu sistem yang dapat menjadi sistem peringatan jika terjadi kebakaran seperti pada gambar 3.2.

Blok Diagram Sistem Pendeteksi Kebakaran Pada Pos Pendakian

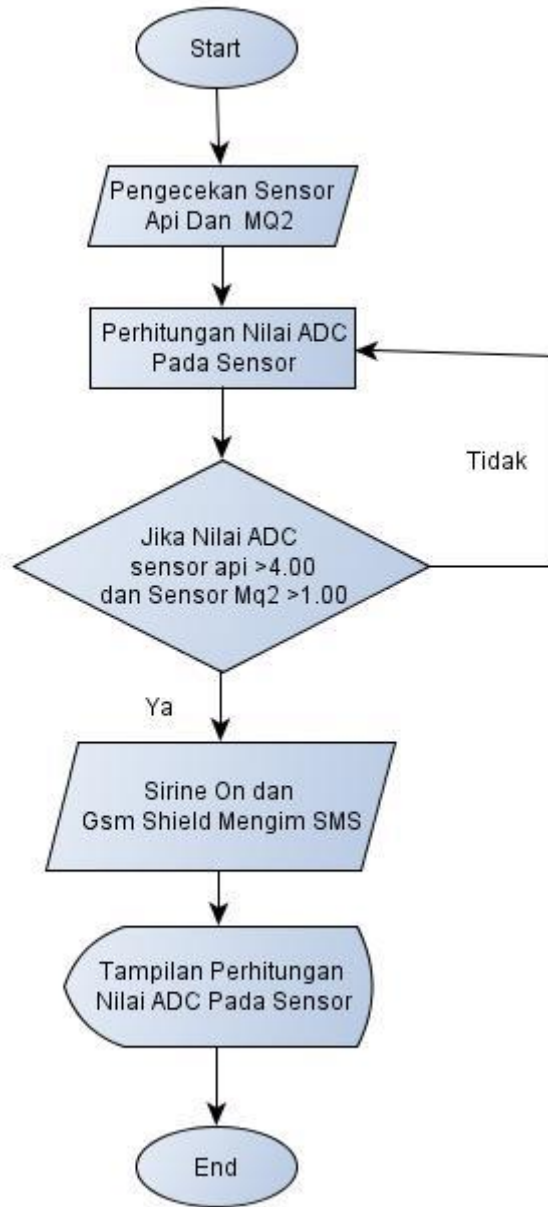


Gambar 3.2. Blok Diagram Sistem Pendeteksi Kebakaran Pada Pos Pendakian

Dari gambar blok diagram sistem pendeteksi kebakaran dapat diketahui yaitu jika salah satu sensor MQ dan Sensor Api mendeteksi adanya asap pada *basecamp* atau post pendakian maka arduino akan memproses agar *Global System for Mobile Communications (GSM) Shield* mengirimkan SMS (*Short Message Service*) kepada petugas cagar alam dan relay akan mengaktifkan *mini sirine* sebagai tanda telah terjadinya kebakaran. Hasil pembacaan dari sensor akan ditampilkan pada LCD 16x2.

3.1.3 Flowchart

Perancangan perangkat lunak dibuat dari pembuatan *flowchart* untuk pembuatan pada *hardware*. Pada gambar 3.3 akan ditampilkan *flowchart* sistem peringatan jika terjadi kebakaran yang akan dibuat dalam penelitian ini.



Gambar 3.3. Flowcart Kerja Sistem

Flow Chart kendali Sistem Alarm Kebakaran Portable Pada Pos Pendakian Menggunakan SMS (*Short Message Service*) Berbasis Arduino ini yaitu jika nilai ADC pada Sensor Api >4.00 desimal ADC dan Sensor MQ2 >1.00 desimal ADC maka *mini sirine* akan aktif yang artinya pos pendakian telah terjadi kebakaran dan *GSMS hield* akan mengirimkan SMS (*Short Message Service*) ke petugas pengelola kawasan cagar alam. Sedangkan jika nilai ADC pada sensor api <4.00 dan Sensor MQ2 <1.00 maka *mini sirine* tidak aktif yang artinya tidak terjadi kebakran

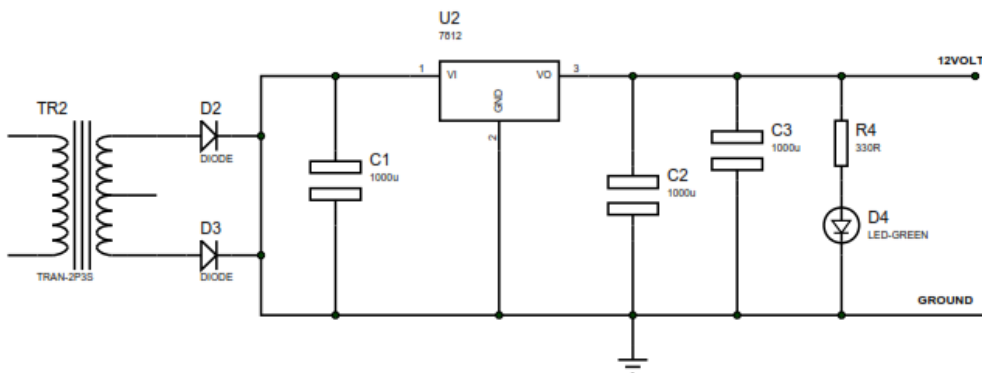
pada pos pendakian. Sedangkan hasil dari perhitungan kedua sensor akan ditampilkan pada LCD 16x2.

3.2 Rancangan Perangkat Keras

Perancangan perangkat keras menjadi bagian yang sangat penting dilakukan dalam pembuatan suatu alat karena dengan merancang terlebih dahulu dengan komponen yang tepat akan mengurangi berlebihnya pembelian komponen dan kerja alat sesuai dengan yang diinginkan. Untuk menghindari kerusakan komponen perlu dipahami juga akan karakteristik dari komponen - komponen tersebut.

3.2.1 Rancangan *Power Supplay*

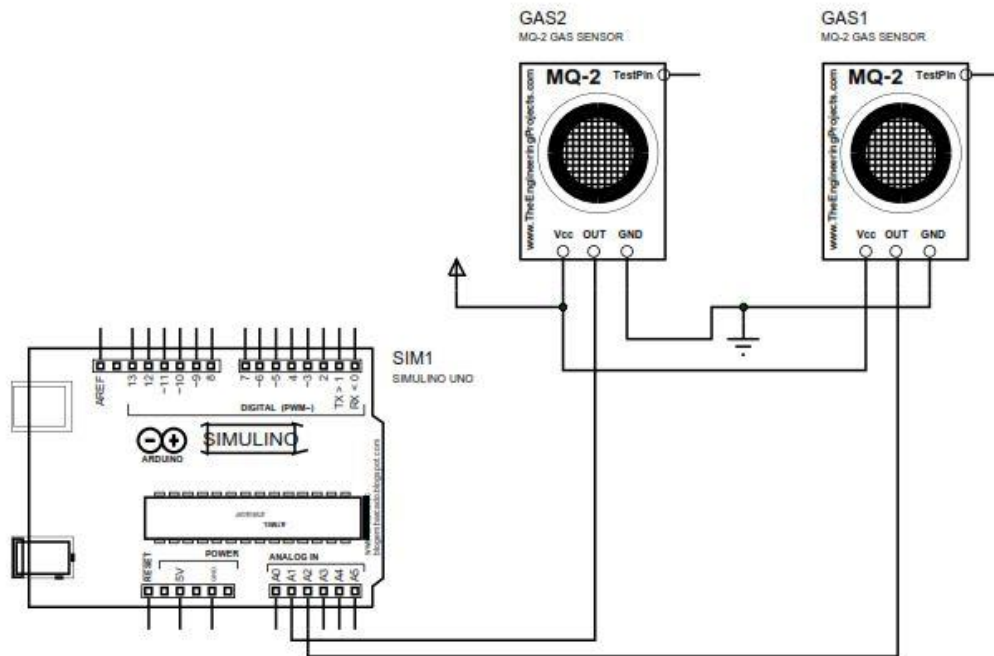
Rangkaian *power supply* digunakan untuk merubah tegangan AC 220V menjadi DC 9 V dalam pembuat *power supplay* 9 volt peneliti nmenggunakan IC LM7809 dan menyalurkan sumber tegangan ke semua komponen elektronika yang ada pada suatu rangkaian agar rangkaian tersebut dapat bekerja seperti pada gambar 3.4.



Gambar 3.4 Ragkaian *Power Supply*

3.2.2 Rancangan Rangkaian Sensor MQ2

Rangkaian MQ2 digunakan sebagai *inputan* dalam mendeteksi adanya asap yang akan diproses oleh arduino sehingga akan menghasilkan outputan untuk menyalakan relay yang akan mengaktifkan sirine kebakaran. Gambar rangkaian *RFID (Radio Frequency Identifikasion)*, layout dan tata letak dapat dilihat seperti pada gambar 3.5.



Gambar 3.5 Rangkaian Sensor MQ2

Pada rangkaian *RFID (Radio Frequency Identifikasion)* hanya beberapa kaki yang dihubungkan ke pin analog *Arduino Uno* agar hasil proses pada arduino dapat mengaktifkan *relay*. Penggunaan PIN *Arduino Uno* dan Sensor MQ2 ditampilkan pada tabel 3.4.

Tabel 3.4. Penggunaan Pin Sensor MQ 2 ke *Arduino Uno*

Pin Arduino	Keterangan	Pin Sensor MQ 2	Keterangan
A1	Pin Input Analog 1	OUT 1	Pin Output 1
A2	Pin Input Analog 2	OUT 2	Pin Output 2

Program arduino untuk menjalankan Sensor MQ 2:

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(2, 3, 4, 5, 6, 10);
```

```
float MQ2;
```

```
float MQ22;
```

```
int adc;
```

```
byte kedip;
```

```
byte keluar;
```

```
void setup(){
pinMode(11,OUTPUT);
digitalWrite(11,LOW);
lcd.begin(16, 2);
lcd.print("Alarm kebakaran ");
delay(2000);
lcd.clear();
lcd.print(" Tunggu sensor");
lcd.setCursor(0, 1);
lcd.print(" siap...");
delay(2000);
do{
adc=(analogRead(A0));
MQ2 = adc * (5.0 / 1023.0);
adc=(analogRead(A2));
MQ22 = adc * (5.0 / 1023.0);
}
while(MQ2 > 1.5);
while(MQ22 > 1.5);
lcd.clear();
}
void loop(){
adc=(analogRead(A0));
MQ2 = adc * (5.0 / 1023.0);
adc=(analogRead(A2));
MQ22 = adc * (5.0 / 1023.0);
lcd.setCursor(0,0);
lcd.print("Asap :");
lcd.print(MQ2);
lcd.print(" ");
lcd.setCursor(0,1);
```



```
lcd.print("Asap:");
lcd.print(MQ22);
lcd.print(" ");
// Sensor 1
if (MQ2 > 1.2 ){
lcd.setCursor(0,0);
lcd.print(" Ada kebakaran");
lcd.setCursor(0,1);
lcd.print("AlarmON ValveON");
digitalWrite(11,HIGH);
delay(1000);
keluar=1;
do{
adc=(analogRead(A0));
MQ2 = adc * (5.0 / 1023.0);
if(MQ2<1)keluar=0;
}
while(keluar);
digitalWrite(11,LOW);
lcd.clear();
}
delay(100);
// sensor 2
if (MQ22 > 1.2 ){
lcd.setCursor(0,0);
lcd.print(" Ada kebakaran");
lcd.setCursor(0,1);
lcd.print("AlarmON ValveON 1");
digitalWrite(11,HIGH);
delay(1000);
keluar=1;
```

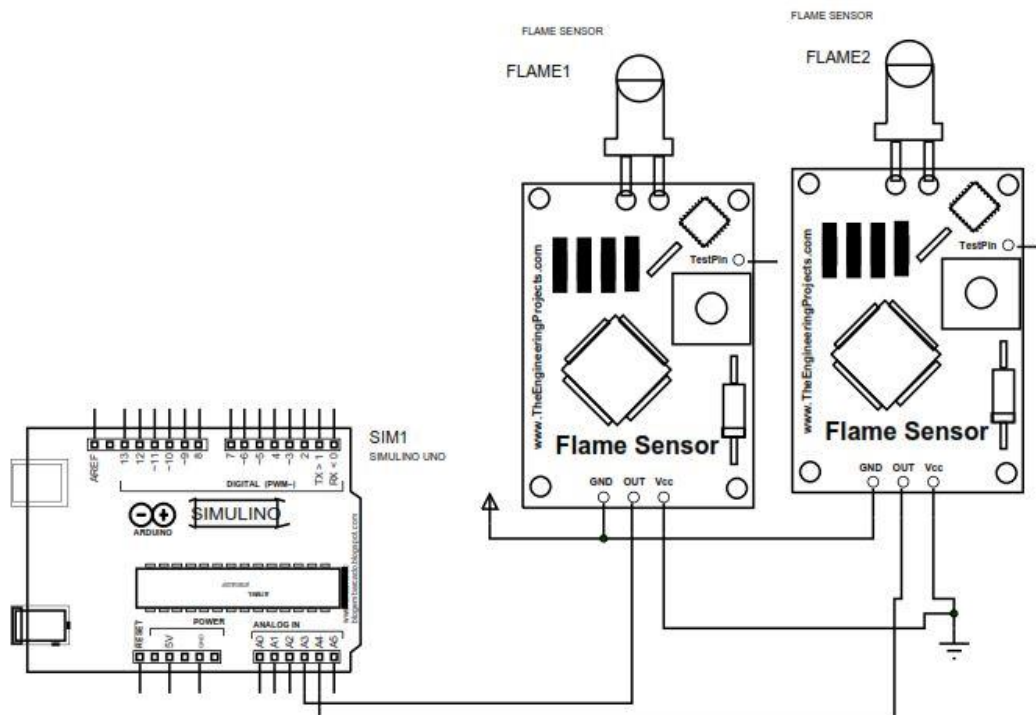
```

do{
adc=(analogRead(A2));
MQ22 = adc * (5.0 / 1023.0);
if( MQ22<1)keluar=0;
}
while(keluar);
digitalWrite(11,LOW);
lcd.clear();
}
delay(100);
}

```

3.2.3 Rancangan Rangkaian Api

Rangkaian Sensor Api digunakan sebagai *inputan* dalam mendeteksi adanya api atau kebakaran pada *basecamp* atau pos pendakian yang telah diolah oleh *Arduino Uno* sehingga akan menghasilkan outputan berupa pengaktifan *relay* dan pengiriman SMS (*Short Message Service*). Gambar rangkaian Sensor Api dan tata letak dapat dilihat seperti pada gambar 3.6.



Gambar 3.6. Perancangan Rangkaian Sensor Api

Pada rangkaian Sensor Api hanya beberapa kaki yang dihubungkan ke pin digital *Arduino Uno* agar hasil proses pada arduino dapat mengaktifkan *relay* dan mengirimkan SMS (*Short Message Service*). Penggunaan PIN *Arduino Uno* dan Sensor Api ditampilkan pada tabel 3.5.

Tabel 3.5. Penggunaan Pin Sensor Api ke *Arduino Uno*

Pin Arduino	Keterangan	PinSensor Api	Keterangan
A3	Pin Input Analog 3	OUT 1	Pin Output 1
A4	Pin Input Analog 4	OUT 2	Pin Output 1

Program arduino untuk menjalankan Sensor Api:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(2, 3, 4, 5, 6, 10);
float sensorApi;
float sensorApi1;
int adc;
byte kedip;
byte keluar;
void setup(){
pinMode(9,OUTPUT);
digitalWrite(9,LOW);
lcd.begin(16, 2);
lcd.print("Alarm kebakaran ");
delay(2000);
lcd.clear();
lcd.print(" Tunggu sensor");
lcd.setCursor(0, 1);
lcd.print(" siap...");
delay(2000);
lcd.clear();
}
void loop(){
```

```
adc=(analogRead(A1));
sensorApi = adc * (5.0 / 1023.0);
adc=(analogRead(A2));
sensorApi1 = adc * (5.0 / 1023.0);
lcd.setCursor(0,0);
lcd.print("Api :");
lcd.print(sensorApi);
lcd.print(" ");
lcd.print(" ");
```

```
lcd.setCursor(0,0);
lcd.print("Api :");
lcd.print(sensorApi1);
lcd.print(" ");
lcd.print(" ");
```

```
if (sensorApi<3){
lcd.setCursor(0,0);
lcd.print(" Ada kebakaran");
lcd.setCursor(0,1);
lcd.print("Sensor 1 ON");
digitalWrite(9,HIGH);
delay(1000);
keluar=1;
```

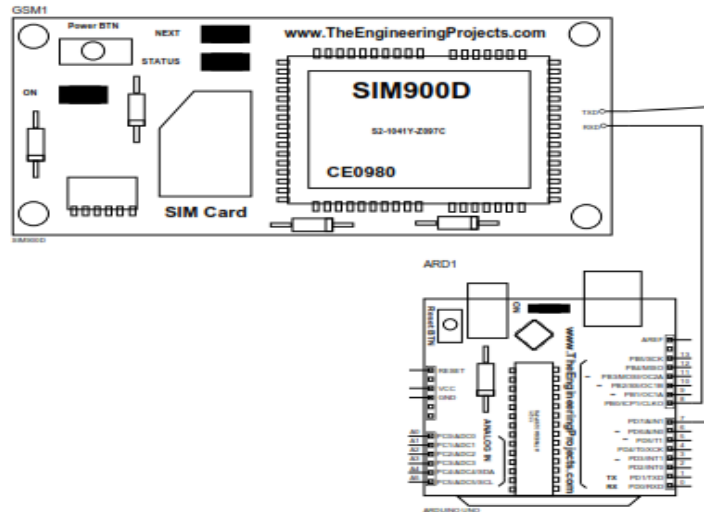
```
if (sensorApi1<3){
lcd.setCursor(0,0);
lcd.print(" Ada kebakaran");
lcd.setCursor(0,1);
lcd.print("Sensor 2 ON");
digitalWrite(9,HIGH);
```

```
delay(1000);
keluar=1;

do
{
sensorApi = adc * (5.0 / 1023.0);
if(sensorApi>4)keluar=0;
sensorApi1 = adc * (5.0 / 1023.0);
if(sensorApi1>4)keluar=0;
}
while(keluar);
digitalWrite(9,LOW);
lcd.clear();
}
delay(100);
}}
```

3.2.4 Rancangan Skema Rangkaian *GSM Shield SIM900*

Rangkaian *Global System for Mobile Communications (GSM) Shield Sim900* digunakan sebagai *output* untuk memberi informasi jika terjadinya kebakaran pada pos pendakian, yang telah diolah oleh *Arduino Uno* sehingga akan menghasilkan pengiriman SMS (*Short Message Service*). Gambar rangkaian *Global System for Mobile Communications (GSM) Shield* layout dan tata letak dapat dilihat seperti pada gambar 3.7.



Gambar 3.7. Perancangan Rangkaian GSM Shield

Pada rangkaian *Global System for Mobile Communications (GSM) Shield* hanya beberapa kaki yang dihubungkan ke pin digital *Arduino Uno* agar hasil proses pada arduino dapat mengirimkan SMS (*Short Message Service*) sehingga jika ada petugas dapat dengan segera memadamkan kebakaran pada pos pendakian. Penggunaan PIN *Arduino Uno* dan *Global System for Mobile Communications (GSM) Shield* ditampilkan pada tabel 3.6.

Tabel 3.6. Penggunaan Pin GSM Shield ke Arduino Uno

Pin Arduino	Keterangan	Pin GSM Shield	Keterangan
D8	Pin Input Digital 8	RXD	Pin Output RXD
D7	Pin Input Digital 7	TXD	Pin Output TXD

Program arduino untuk menjalankan *Global System for Mobile Communications (GSM) Shield*:

```
#include <SoftwareSerial.h>
SoftwareSerial SIM900A(8,7); // variabel gsm shield

void setup(){
  SIM900A.begin(9600); // Setting the baud rate of GSM Module
  Serial.begin(9600); // Setting the baud rate of Serial Monitor (Arduino)
  delay(1000);
  Serial.println ("SIM900A Ready");
}
```

```

}
void loop(){
  if (SIM900A.available(>0)
    Serial.write(SIM900A.read());
  }
void SendMessage1()
{
  Serial.println ("SIM900A Mengirim SMS");
  SIM900A.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode
  delay(1000); // Delay of 1000 milli seconds or 1 second
  Serial.println ("Set SMS Number");
  SIM900A.println("AT+CMGS="+6281295656440+"\r"); // Replace with your mobile number
  delay(1000);
  Serial.println ("Set SMS Content");
  SIM900A.println("ADA KEBAKARAN PADA POST 1 PENDAKIAN");// The SMS text
  you want to send
  delay(100);
  Serial.println ("Finish");
  SIM900A.println((char)26);// ASCII code of CTRL+Z
  delay(1000);
  Serial.println (" ->SMS Selesai dikirim");
}
void RecieveMessage()
{
  Serial.println ("SIM900A Membaca SMS");
  delay (1000);
  SIM900A.println("AT+CNMI=2,2,0,0,0"); // AT Command to receive a live SMS
  delay(1000);
  Serial.write (" ->Unread SMS Selesai dibaca");
}
void SendMessage2()

```

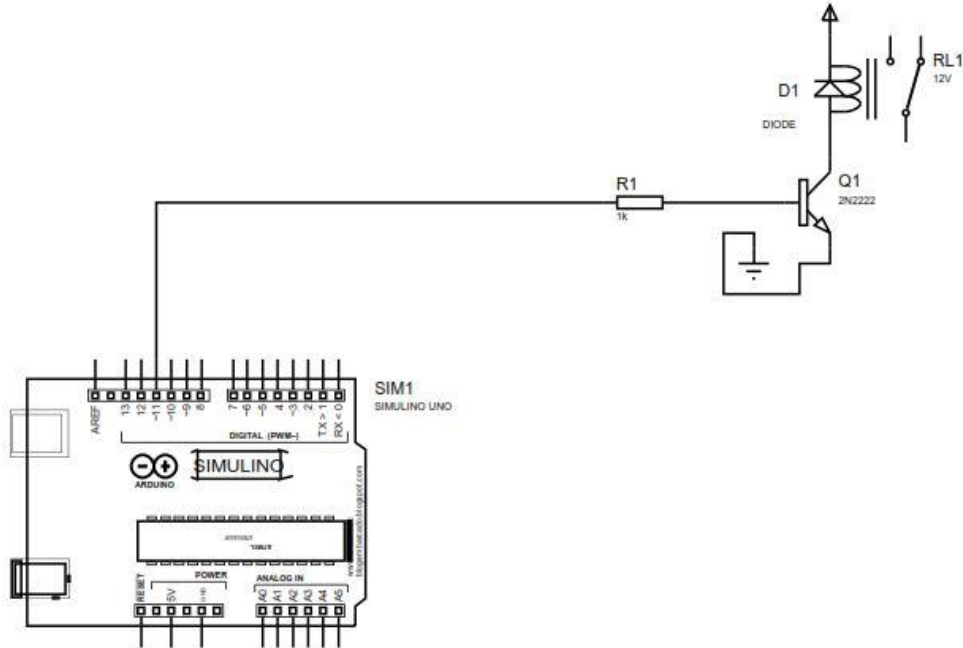
```

{
  Serial.println ("SIM900A Mengirim SMS");
  SIM900A.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode
  delay(1000); // Delay of 1000 milli seconds or 1 second
  Serial.println ("Set SMS Number");
  SIM900A.println("AT+CMGS="+6281295656440+"\r"); // Replace with your mobile number
  delay(1000);
  Serial.println ("Set SMS Content");
  SIM900A.println("ADA KEBAKARAN PADA POST 2 PENDAKIAN");// The SMS text
  you want to send
  delay(100);
  Serial.println ("Finish");
  SIM900A.println((char)26);// ASCII code of CTRL+Z
  delay(1000);
  Serial.println (" ->SMS Selesai dikirim");
}
void RecieveMessage1()
{
  Serial.println ("SIM900A Membaca SMS");
  delay (1000);
  SIM900A.println("AT+CNMI=2,2,0,0,0"); // AT Command to receive a live SMS
  delay(1000);
  Serial.write (" ->Unread SMS Selesai dibaca");
}

```

3.2.5 Rancangan Rangkaian *Relay*

Rangkaian *relay* digunakan sebagai *output* untuk menyalkan atau mematikan *mini sirine* peringatan kebakaran yang telah diolah oleh *Arduino Uno*. Gambar rangkaian *Relay* dan tata letak dapat dilihat seperti pada gambar 3.8.



Gambar 3.8. Perancangan Rangkaian Relay

Pada rangkaian *relay* hanya beberapa kaki yang dihubungkan ke pin digital *Arduino Uno* agar hasil proses pada arduino dapat memberi outputan hidup atau matinya *mini sirine*. Penggunaan PIN *Arduino Uno* dan *relay* ditampilkan pada tabel 3.7.

Tabel 3.7. Penggunaan Pin Relay Mini ke Arduino Uno

Pin Arduino	Keterangan	Pin Relay	Keterangan
D11	Pin Input Digital 11	OUT	Pin Output

Program arduino untuk menjalankan *Relay* :

```
void setup()
{
  Serial.begin(9600);
  // set pin 5 sebagai output
  pinMode(11,OUTPUT);
}
```

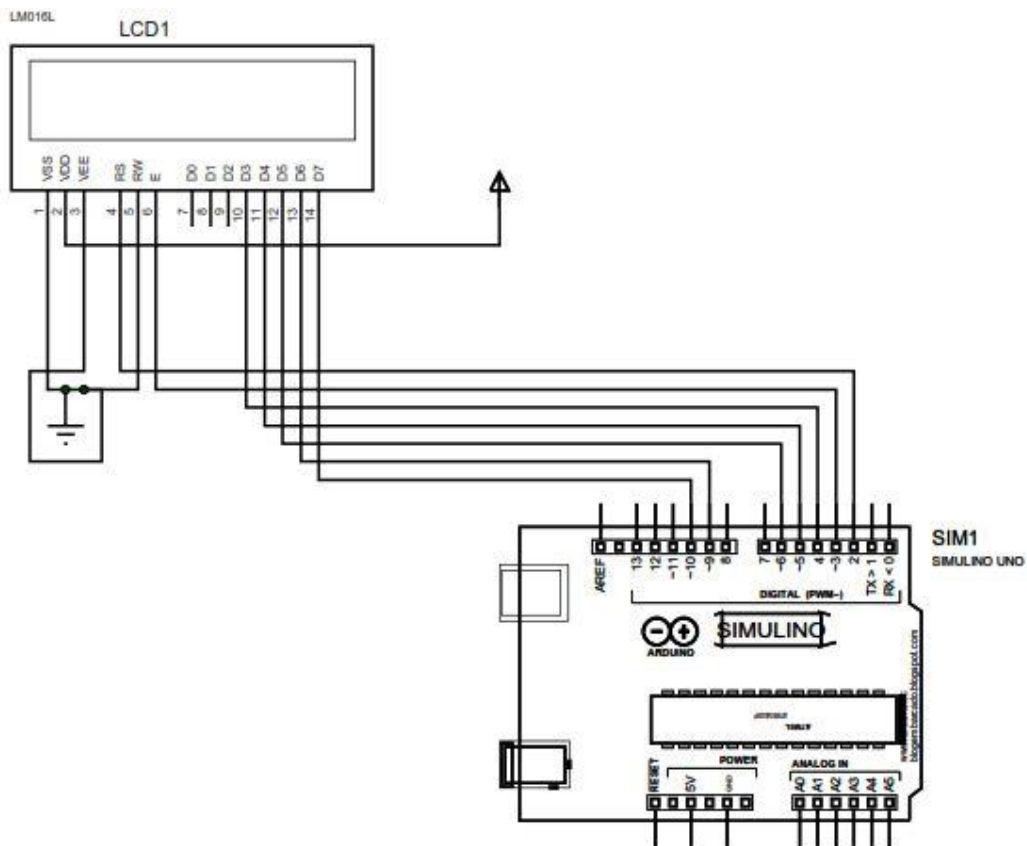
```

void loop()
{
digitalWrite(11 ,HIGH);
delay(1000);
digitalWrite(11 ,LOW);
delay (1000);
}

```

3.2.6 Rancangan Skema Rangkaian LCD (*Liquid Crystal Display*)

Rangkaian LCD (*Liquid Crystal Display*) Digunakan sebagai *output* untuk menampilkan informasi berbentuk data yang berasal dari sumber masukan dari *RFID* yang telah diolah oleh *Arduino Uno*. Gambar rangkaian LCD (*Liquid Crystal Display*), layout dan tata letak dapat dilihat seperti pada gambar 3.9.



Gambar 3.9. Rangkaian (*Liquid Crystal Display*) LCD 16 x 2

Pada rangkaian LCD (*Liquid Crystal Display*) hanya beberapa kaki yang dihubungkan ke pin digital *Arduino Uno* agar hasil proses pada arduino dapat ditampilkan kedalam LCD (*Liquid Crystal Display*). Penggunaan PIN *Arduino Uno* dan LCD (*Liquid Crystal Display*) ditampilkan pada tabel 3.8.

Tabel 3.8. Penggunaan Pin LCD (*Liquid Crystal Display*) ke *Arduino Uno*

Pin Arduino	Keterangan	Pin LCD	Keterangan
D2	Pin Output Digital 2	4	Kaki Input no 4 LCD
D3	Pin Output Digital 3	6	Kaki Input no 6 LCD
D4	Pin Output Digital 4	11	Kaki Input no 11 LCD
D5	Pin Output Digital 5	12	Kaki Input no 12 LCD
D6	Pin Output Digital 6	13	Kaki Input no 13 LCD
D7	Pin Output Digital 7	14	Kaki Input no 14 LCD

Program arduino untuk menjalankan LCD (*Liquid Crystal Display*) 16x2:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(2, 3, 4, 5, 6, 10);
void setup(){

lcd.begin(16, 2);
lcd.print("Alarm kebakaran ");
delay(2000);
lcd.clear();
lcd.print(" Tunggu sensor");
lcd.setCursor(0, 1);
lcd.print(" siap...");
delay(2000);
lcd.clear();
}
void loop(){

lcd.setCursor(0,0);
```

```
lcd.print("Ap :");  
lcd.print(sensorApi);  
lcd.print(" ");  
lcd.setCursor(0,1);  
lcd.print("As:");  
lcd.print(MQ2);  
lcd.print(" ");
```

```
lcd.setCursor(9,0);  
lcd.print("Ap :");  
lcd.print(sensorApi1);  
lcd.print(" ");  
lcd.setCursor(9,1);  
lcd.print("As:");  
lcd.print(MQ22);  
lcd.print(" ");
```

```
{  
lcd.setCursor(0,0);  
lcd.print(" Ada kebakaran");  
lcd.setCursor(0,1);  
lcd.print("Sensor 1 ON");  
}
```

```
lcd.clear();
```

```
}
```

```
delay(100);
```

```
{
```

```
lcd.setCursor(0,0);
```

```
lcd.print(" Ada kebakaran");
```

```
lcd.setCursor(0,1);
```

```
lcd.print("Sensor 2 ON");
```

```
}
```

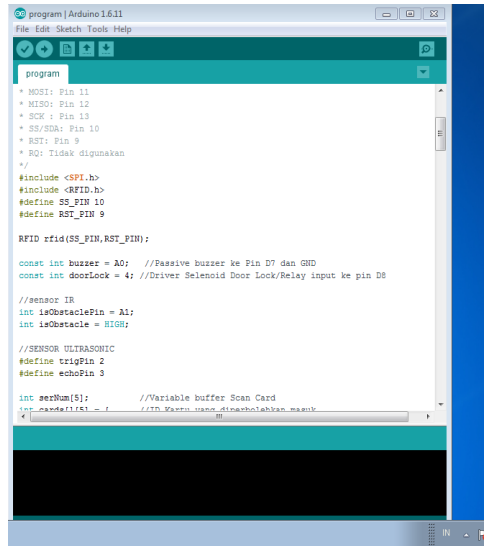
```
lcd.clear();  
}  
delay(100);  
}
```

3.3 Pengujian Sistem

Setelah perancangan *hardware* selesai, maka yang dilakukan adalah pengujian software yaitu *running* program dari masing-masing rangkaian agar mengetahui jika program yang sudah dibuat telah sesuai dalam menjalankan perintah dari masing-masing rangkaian sistem perangkat keras.

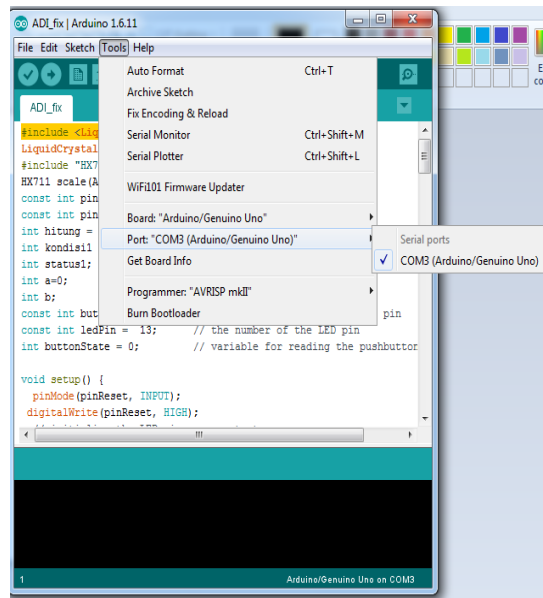
3.3.1 Pengujian Mikrokontroler

Penerapan perangkat lunak merupakan suatu tahap dimana program yang telah dirancang akan diupload kedalam modul *Arduino Uno* melalui *downloader* dan menggunakan *software* tertentu sesuai dengan bahasa pemrograman yang akan digunakan. Pada *Software Arduino* program ditulis kemudian di *compile*, tujuannya adalah untuk mengetahui apakah program yang dibuat sudah benar atau belum. Langkah terakhir yaitu meng-*upload* program kedalam modul *mikrokontroler*. Pada penelitian ini program yang dibuat, dirancang untuk dapat membuat suatu sistem keamanan kotak amal masjid dengan outputan SMS (*Short Message Service*) dan suara. Setelah perancangan *hardware* dan *software* selesai, maka yang dilakukan adalah *running* program, pengujian tiap - tiap rangkaian apakah sudah sesuai dengan yang diinginkan atau belum. Pengujian dilakukan pada bagian - bagian seperti pengujian *respon* dan rangkaian keseluruhan pada sistem ini. Berikut ini adalah tampilan *software* yang digunakan untuk menuliskan dan mengupload program kedalam *Arduino Uno* seperti pada gambar 3.10.



Gambar 3.10. Tampilan Software Arduino IDE

Untuk bisa meng-*upload* program ke *Arduino Uno* yang pertama harus mengatur port yang digunakan oleh Arduino. Pengaturan port Arduino dapat dilihat pada gambar 3.11.



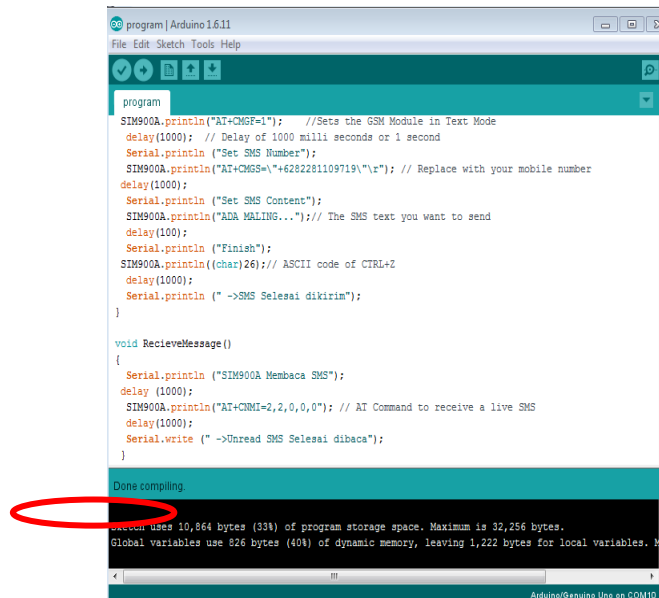
Gambar 3.11. Pengaturan Port Arduino Uno

Pengaturan port Arduino diatas menggunakan port COM3. Setelah pengaturan port langkah selanjutnya yaitu meng-*compile* program. Berikut adalah hasil *compile* program pada gambar 3.12.



Gambar 3.12. Hasil Compile Program

Setelah program berhasil di *compile* selanjutnya yaitu meng-*upload* file ke *Arduino Uno* seperti pada gambar 3.13.



Gambar 3.13. Hasil Upload Program.

3.3.2 Pengujian Rangkaian Sensor MQ2

Pengujian Sensor MQ2 bertujuan untuk mengetahui ketika adanya asap pada *basecamp* atau pos pendakian apakah sensor dapat dengan baik dalam mendeteksinya dan peneliti melakukan uji coba berapa lama delay yang dibutuhkan sensor MQ2 dalam membaca adanya asap dan jarak MQ2 dalam mendeteksi asap. Dalam mengukur respon peneliti menggunakan timer sedangkan dalam mengukur jarak peneliti menggunakan meteran.

3.3.3 Pengujian Rangkaian Sensor Api

Pengujian Sensor Api bertujuan untuk mengetahui ketika apakah sensor dapat dengan baik dalam membaca adanya api, peneliti akan melakukan uji coba yaitu respon dan jarak sensor dalam mendeteksi adanya api. Dalam mengukur respon peneliti menggunakan timer sedangkan dalam mengukur jarak peneliti menggunakan meteran.

3.3.4 Pengujian Rangkaian Relay

Rancangan pengujian *Relay* bertujuan untuk mengetahui ketika *relay* diberikan status aktif apakah *relay* dapat mematikan dan menghidupkan *mini sirine* tanda terjadinya kebakaran apakah telah sesuai dengan program pada arduino.

3.3.5 Pengujian Rangkaian GSM Shield SIM900

Pengujian *Global System for Mobile Communications (GSM) Shield sim900* bertujuan untuk mengetahui apakah *Global System for Mobile Communications (GSM) Shield* dapat dengan baik dalam mengirimkan SMS (*Short Message Service*) kepada petugas jika terjadinya kebakaran dan mengetahui berapa lama respon yang dibutuhkan dalam mengirimkan SMS (*Short Message Service*).

3.3.6 Pengujian Rangkaian Sistem Keseluruhan

Pengujian sistem secara keseluruhan bertujuan untuk memastikan semua komponen dapat berjalan dengan sempurna. Mulai dari power supply sensor MQ2, sensor Api, *Relay*, *Global System for Mobile Communications (GSM) Shield*, blok sistem *Arduino Uno* dan program yang mengatur jalannya sistem keseluruhan.

3.4 Analisis Kerja

Untuk analisa kerja, dilakukan bersama pada saat melakukan uji coba alat yang bertujuan untuk mengetahui kerja alat tersebut. Selain itu yang akan di analisa adalah keakuratan sensor Api dan Sensor MQ2 dalam mendeteksi api serta asap peneliti juga akan melakukan analisa jarak yang dapat di deteksi oleh ke 2 sensor. Berdasarkan hasil pengujian sistem yang telah di dapat akan dianalisis untuk memastikan bahwa sistem yang telah dibuat sesuai dengan harapan.