

BAB III

PERANCANGAN ALAT

Pada Bab ini penulis akan menjelaskan mengenai perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*) pada rangkaian *Alarm Kebakaran Otomatis* ini.

3.1. Perencanaan Rangkaian Alarm Kebakaran Otomatis

3.1.1 Spesifikasi fungsi

Sebelum merealisasikan dari perencanaan-perencanaan yang dijabarkan nanti, maka penulis terlebih dahulu akan menjabarkan tentang spesifikasi dari alat ini. Rangkaian Alarm Kebakaran Otomatis ini berfungsi sebagai pendeteksi kebakaran secara otomatis, ketika sensor mendeteksi api maka alarm buzzer akan berbunyi, lampu LED akan meyalal dan LCD akan menampilkan status yang terjadi.

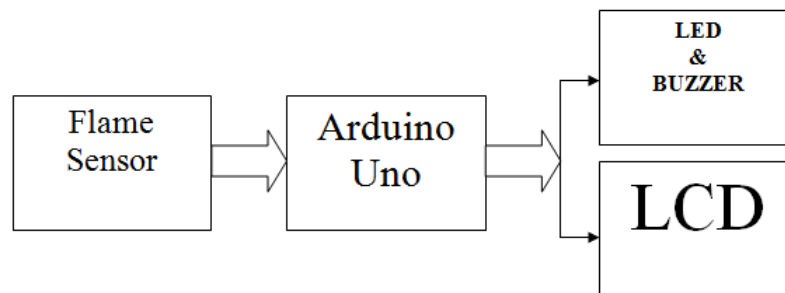
3.1.2 Spesifikasi alat

Untuk merealisasikan fungsi-fungsi di atas, maka perlu dirancang rangkaian-rangkaian yang sesuai dengan spesifikasinya.

- Tegangan supply Battery 9 VDC untuk arduino.
- Aplikasi Program Arduino IDE (Integrated Development Enviroment)
- LCD untuk status yang terjadi.
- Arduino Uno.
- Flame sensor untuk mendeteksi api.
- Buzzer dan LED untuk peringatan.

3.1.3 Perencanaan Blok Diagram

Secara keseluruhan rangkaian simulasi *Alarm Kebakaram* dapat ditunjukkan oleh gambar berikut :



Gambar 3.1 Blok diagram Alarm Kebakaran Otomatis

Fungsi blok rangkaian dari blok diagram diatas adalah sebagai berikut :

a. Rangkaian Arduino Uno

Rangkaian ini berfungsi untuk mengatur kerja keseluruhan blok rangkaian pada *Alarm Kebakaran Otomatis*. Input tegangan yang dimasukkan ke arduino uno menggunakan battery 9VDC. Arduino akan mengatur Flame Sensor yang akan mendeteksi api, jika api tersebut terdeteksi oleh Flame Sensor maka alarm akan menyala secara otomatis yaitu Buzzer dan LED akan menyala dan display LCD akan memberikan status yang terjadi.

b. Flame Sensor

Flame sensor ini dapat mendeteksi nyala api dengan panjang gelombang 760 nm ~ 1100 nm. Sensor nyala api ini mempunyai sudut pembacaan 60 derajat, dan beroperasi pada suhu -25 - 85 derajat Celcius.

c. LCD (*Liquid Crystal Display*)

LCD berfungsi untuk menampilkan informasi seperti inisialisasi atau status keadaan yang sedang terjadi.

d. Indikator Buzzer dan LED

Rangkaian Buzzer dan LED ini berfungsi untuk memberikan peringatan ketika Flame Sensor tersebut telah mendeteksi api secara audio dan visual.

3.2 Cara Kerja

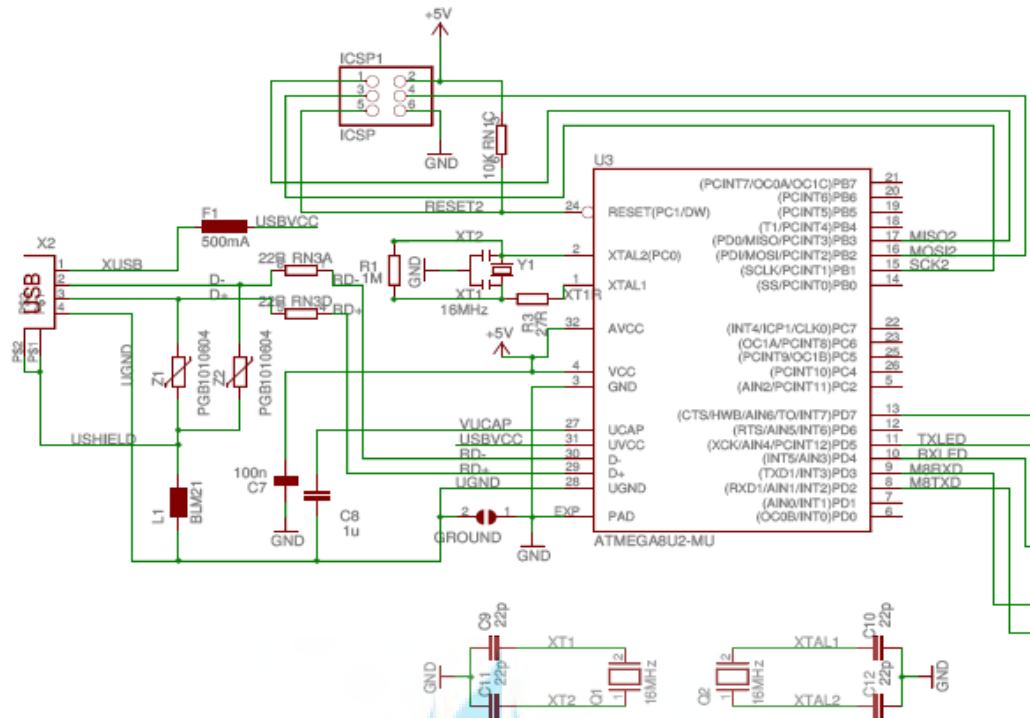
Ketika toggle switch di tekan maka supply battery akan mendistribusikan tegangan 9VDC ke rangkaian arduino pada alat. Display LCD karakter 2 x 16 terjadi inisialisasi. Alat ini bekerja sesuai settingan yang telah dipilih, yang sebelumnya sudah diproses atau deprogram di arduino. Bersamaan dengan itu arduino juga akan mengaktifkan Buzzer dan LED bergerak sesuai *settingan* yang telah dipilih. Ketika Arduino memonitor dan mendeteksi api maka arduino akan mengaktifkan indikator Buzzer, LED, dan LCD secara bersamaan.

3.3 Perancangan Rangkaian Arduino Uno

Rangkaian kontrol disini menggunakan Arduin uno yang akan mengolah data masukan dan keluaran pada setiap pinnya. Arduino yang digunakan dalam perancangan ini merupakan komponen utama, karena komponen ini adalah yang akan mengatur keseluruhan sistem agar dapat bekerja dengan baik dan optimal. Perencanaan rangkaian arduino uno dapat dilihat pada gambar 3.2.

Arduino uno akan memproses masukan dan keluran yang ada pada peralatan ini, pengontrolan tersebut dilakukan melalui pengaktifan masing-masing pin pada arduino. Untuk mengaktifkan pin-pin yang terdapat didalam arduino tersebut dilakukan dengan menggunakan perangkat lunak (Software) yang dimiliki oleh arduino.

Untuk mengaktifkan arduino uno maka perlu diberikan tegangan supply +5 Volt DC yang dapat langsung di ambil dari port USB pada komputer, atau dapat juga di ambil dari supply tegangan DC lain.



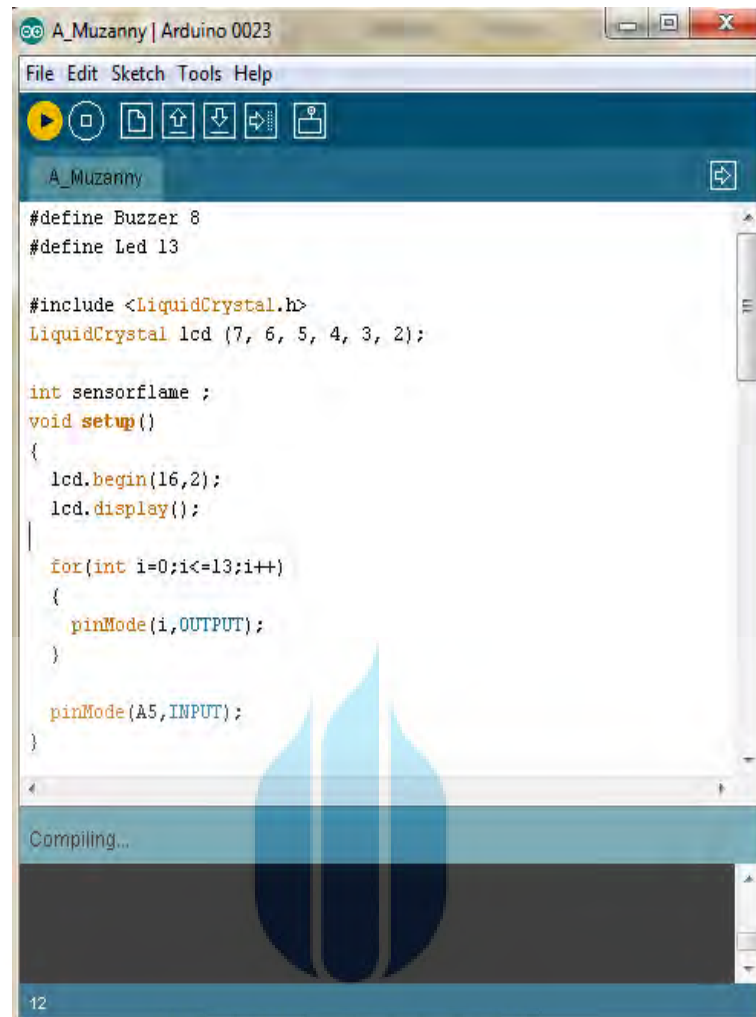
Gambar 3.2 Rangkaian arduino uno

Penggunaan pinnya sebagai berikut:

1. LCD di PIN 2, 3, 4, 5, 6, 7
2. Buzzer di PIN 8
3. Flame Sensor di PIN A5
4. LED di PIN13

3.4 Aplikasi Program Arduino IDE (Integrated Development Enviroment)

Setelah proses rangkaian selesai dibuat langkah selanjutnya adalah membuat program pada aplikasi Program Arduino IDE (Integrated Development Enviroment).



Gambar 3.3 Program Arduino IDE (Integrated Development Enviroment)

3.4.1. Bahasa Program Arduino IDE Yang Digunakan

- Float

Float-point numbers sering digunakan untuk memperkirakan nilai analog dan berkelanjutan karena nilai tersebut memiliki resolusi yang lebih besar daripada integers. Floating – point numbers dapat sebesar $3.4028235E + 38$ dan dapat serendah $3.4028235E - 38$. Angka tersebut disimpan sebanyak 32 bits (4bytes) dari informasi. Float hanya memiliki 6-7 digit decimal. Pada Arduino, double sama nilainya dengan Float.

Examples

```
float myfloat;
float sensorCalbrate = 1.117;
```

Syntax

```
float var = val;
```

- var - your float variable name
- val - the value you assign to that variable

Example Code

```
int x;
int y;
float z;

x = 1;
y = x / 2;           // y now contains 0, ints can't
hold fractions
z = (float)x / 2.0;   // z now contains .5 (you have to
use 2.0, not 2)
```

• Int

Integers adalah data type dasar untuk menyimpan angka dengan nilai 2 byte. Integers menyimpan angka – angka negative dengan teknik yang disebut 2s complement Math. Arduino dapat menjaga dan mengatur angka – angka negative, jadi operasi aritmatika dapat bekerja secara transparansi.

Example

```
int NYALA = 1;
int MATI = 0;
```

• Void (Setup)

Fungsi Setup muncul ketika memulai lembar kerja/Sketch. Kita dapat menggunakannya untuk mengartikan variables, pin modes, Start, using libraries, dan lain-lain. Fungsi Setup ini hanya akan bekerja sekali, setelah power up or reset pada Arduino.

Example:

```
// actions are performed in the functions "setup" and
// "loop"
// but no information is reported to the larger program

void setup()
{
  // ...
}

void loop()
{
  // ...
}
```

- **Serial begin()**

Untuk komunikasi dengan komputer, digunakan satu dari angka berikut ini : 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 atau 115200. Kita juga dapat menggunakan angka spesifikasi yang lain, contohnya untuk berkomunikasi melewati pin 0 dan pin 1 dapat digunakan particular boud rate yang memenuhi syarat.

Syntax

Serial.begin(speed)

Arduino Mega only:

Serial1.begin(speed)

Serial2.begin(speed)

Serial3.begin(speed)

Example:

```
void setup() {
  Serial.begin(9600); // opens serial port, sets data rate
  to 9600 bps
}
void loop() {}
```

Arduino Mega example:

```
// Arduino Mega using all four of its Serial ports
// (Serial, Serial1, Serial2, Serial3),
// with different baud rates:

void setup(){
  Serial.begin(9600);
  Serial1.begin(38400);
  Serial2.begin(19200);
  Serial3.begin(4800);

  Serial.println("Hello Computer");
  Serial1.println("Hello Serial 1");
  Serial2.println("Hello Serial 2");
  Serial3.println("Hello Serial 3");
}
void loop() {}
```

- **loop()**

Setelah mengaplikasikan fungsi Setup yang dapat menganalisa dan mengatur Values/nilai, fungsi Loop dapat merubah dan merespon program yang kita buat. Kita dapat menggunakan untuk mengaktifkan control pada papan Arduino.

Example

```
int buttonPin = 3;

// setup initializes serial and the button pin
void setup()
{
  beginSerial(9600);
  pinMode(buttonPin, INPUT);
}

// loop checks the button pin each time,
// and will send serial if it is pressed
void loop()
{
  if (digitalRead(buttonPin) == HIGH)
    serialWrite('H');
  else
    serialWrite('L');

  delay(1000);
}
```

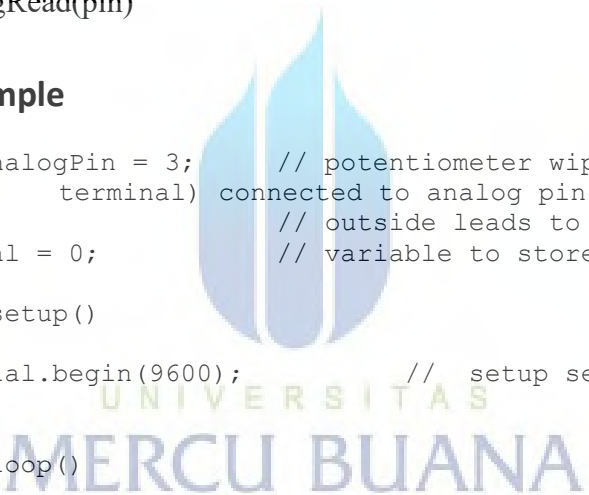

- **Analog Read()**

Papan Arduino memiliki 6 chanel, 10 – bit analog ke digital, artinya kita dapat memasukkan tegangan antara 0 dan 5 volts pada nilai integer antara 0 dan 1023. Kisaran input/masukan dan resolusi dapat dirubah menggunakan analog reference. Untuk membaca analog input, dibutuhkan sekitar 100 microsecond (0.0001 s). Jadi rata –rata membaca maksimumnya sekitar 10.000 kali dalam satu detik. Catatan : Jika pin analog input tidak dapat berkoneksi pada apapun, nilainya akan kembali dengan analogRead.

Syntax

analogRead(pin)

Example



```
int analogPin = 3;           // potentiometer wiper (middle
                             terminal) connected to analog pin 3
                             // outside leads to ground and +5V
int val = 0;                 // variable to store the value read

void setup()
{
  Serial.begin(9600);        // setup serial
}

void loop()
{
  val = analogRead(analogPin); // read the input pin
  Serial.println(val);         // debug value
}
```

- **Print()**

Angka –angka akan dicetak menggunakan karakter ASCII pada setiap digit. Floats juga tercetak dengan digit ASCII kedua angka decimal. Bytes akan terkirim sebagai karakter tunggal.

- Serial.print(78) gives “78”
- Serial.print(1.23456) gives “1.23”
- Serial.print(byte(78)) gives “N” (whose ASCII value is 78)
- Serial.print(„N”) gives “N”
- Serial.print(“Hello world.”) gives “Hello world.”

Opsi kedua parameter dispeksifikasi menggunakan BYTE, BIN, OCT, DEC, HEX.

- Serial.print (78, BYTE) gives “N”
- Serial.print (78, BIN) gives “1001110”
- Serial.print (78, OCT) gives “116”
- Serial.print (78, DEC) gives “4E”
- Serial.print (78, HEX) gives “4E”
- Serial.println(1.23456,0) gives “1”
- Serial.println(1.23456,2) gives “1.23”
- Serial.println(1.23456,4) gives “1.2346”

Syntax

```
Serial.print(val)
```

```
Serial.print(val, format)
```

Example:

```
/*
  Uses a FOR loop for data and prints a number in various formats.
*/
int x = 0; // variable
void setup() {
  Serial.begin(9600); // open the serial port at 9600
  bps:
}

void loop() {
  // print labels
  Serial.print("NO FORMAT"); // prints a label
  Serial.print("\t"); // prints a tab

  Serial.print("DEC");
  Serial.print("\t");

  Serial.print("HEX");
  Serial.print("\t");

  Serial.print("OCT");
  Serial.print("\t");

  Serial.print("BIN");
  Serial.print("\t");
```

```

    for(x=0; x< 64; x++){    // only part of the ASCII chart,
change to suit

    // print it out in many formats:
    Serial.print(x);        // print as an ASCII-encoded
decimal - same as "DEC"
    Serial.print("\t");     // prints a tab

    Serial.print(x, DEC);   // print as an ASCII-encoded
decimal
    Serial.print("\t");     // prints a tab

    Serial.print(x, HEX);   // print as an ASCII-encoded
hexadecimal
    Serial.print("\t");     // prints a tab

    Serial.print(x, OCT);   // print as an ASCII-encoded
octal
    Serial.print("\t");     // prints a tab

    Serial.println(x, BIN); // print as an ASCII-encoded
binary
    //                                then adds the carriage
return with "println"
    delay(200);              // delay 200 milliseconds
    }
    Serial.println("");      // prints another carriage
return
}

```

- **delay()**

Hentikan program sementara untuk mengukur waktu (dalam milliseconds) yang terspesifikasi pada parameter (ada 1000 milliseconds dalam setiap detik). Pada saat mudahnya membuat LED berkedip dengan fungsi Delay, beberapa Sketch/lembar kerja mengalami Delay. Membaca sensor, penghitungan matematika, atau memanipulasi pin dapat berkerja pada saat fungsi Delay berkerja. Beberapa program berbasis sepengetahuan biasanya menghindari menggunakan fungsi Delay untuk kegiatan yang membutuhkan waktu lebih dari 10 milliseconds.

Syntax

delay(ms)

Example

```
int ledPin = 13;           // LED connected to digital
                           pin 13

void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as
                           output
}

void loop()
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}
```

Selanjutnya hubungkan Arduino Uno ke komputer dengan menggunakan kabel USB (Universal Serial Bus). Lalu kita dapat mengetikkan program pada lembar kerja Sketch dan compile untuk mengecek atau memeriksa apakah kode sudah benar sebelum di kirim ke papan Arduino, program tersebut dapat diketik seperti dibawah ini :

// Program Ahmad Muzanny 41408010015 //

```
#define Buzzer 8
#define Led 13

#include <LiquidCrystal.h>
LiquidCrystal lcd (7, 6, 5, 4, 3, 2);

int sensorflame ;
void setup()
{
  lcd.begin(16,2);
  lcd.display();

  for(int i=0;i<=13;i++)
  {
    pinMode(i,OUTPUT);
```

```

    }

    pinMode(A5, INPUT);
}

void loop()
{
    sensorflame = analogRead(A5);

    lcd.setCursor(0,0);
    lcd.print("DATA SENSOR");
    lcd.setCursor(0,1);
    lcd.print(sensorflame);
    lcd.setCursor(1,1);
    lcd.print(" ");
    if(sensorflame>80)
    {
        for(;;)
        {
            int sensorflame = analogRead(A5);
            lcd.setCursor(0,0);
            lcd.print("MENDETEKSI API");
            lcd.setCursor(0,1);
            lcd.print(sensorflame);

            digitalWrite(Buzzer,HIGH);
            digitalWrite(Led,HIGH);
            delay(100);
            digitalWrite(Buzzer,LOW);
            digitalWrite(Led,LOW);
            delay(100);
            if(sensorflame<80)
            {
                break;
            }
        }
    }
    else
    {
        digitalWrite(Buzzer,LOW);
        digitalWrite(Led,LOW);
    }
}

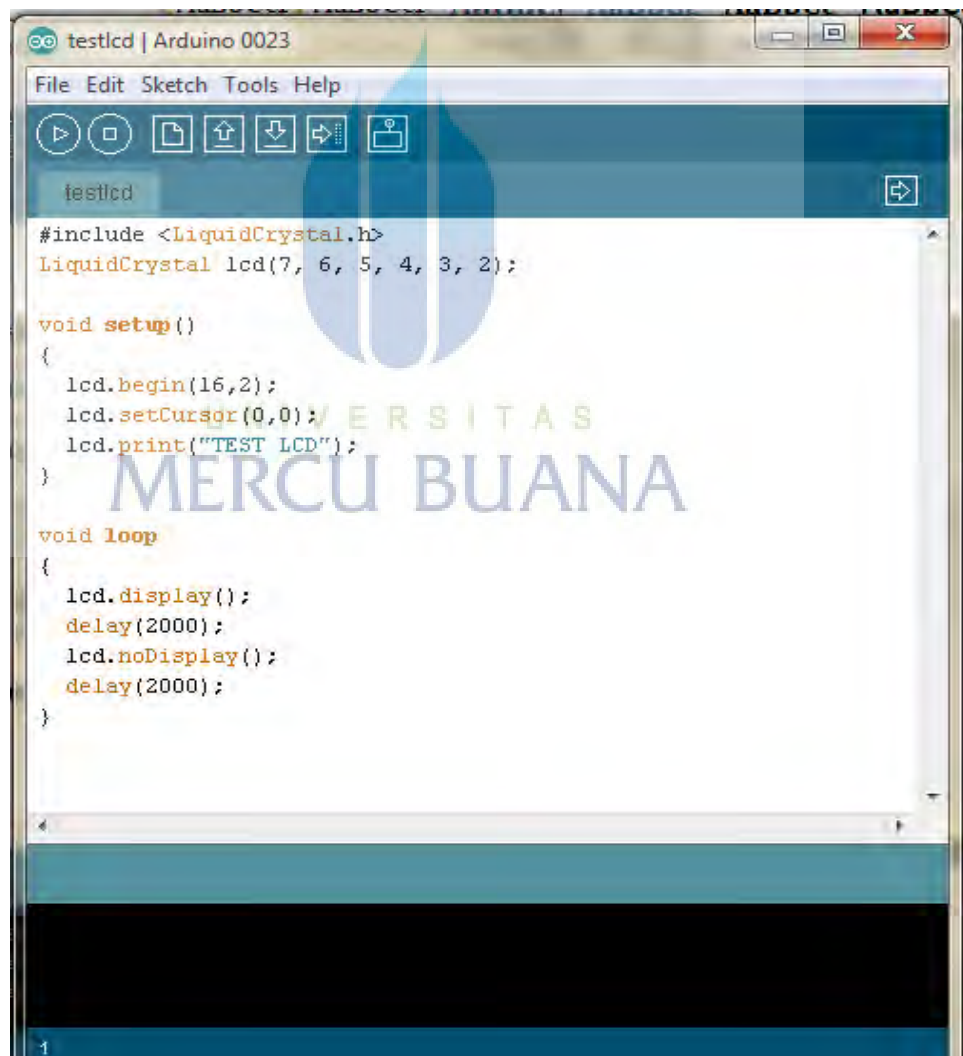
```

3.5 Perancangan LCD (*Liquid Crystal Display*)

Rangkaian display disini menggunakan *Liquid Crystal Display* (LCD) sebagai displaynya yang berfungsi untuk menampilkan settingan volume keluaran (ml/jam). LCD ini juga dapat menampilkan teks atau variable sesuai dengan kebutuhan dan sesuai dengan program yang digunakan untuk mengontrolnya. Input data rangkaian ini berasal dari PIN sistemnya langsung dikendalikan oleh arduino UNO yaitu pada PIN P.2, P.3, P.4, P.5, P.6, P.7.

Ketika power dinyalakan, display biasanya menampilkan sederet persegi gelap, mungkin hanya pada sebagian display. Sel-sel karakter ini sebenarnya merupakan bagian yang mati. Display ini mereset sendiri pada bagian awal ketika power dinyalakan, yang mana layar menjadi kosong sehingga karakter-karakter tidak dapat terlihat. Dengan demikian perlu diberikan perintah pada poin ini untuk menyalakan display.

LCD mempunyai display 2 baris 16 kolom atau biasa disebut sebagai LCD karakter 2x16, dengan 16 pin konektor. Display karakter pada LCD diatur oleh pin EN (P1.1), RS (P1.0) dan RW. Untuk pembuatan sebuah karakter dapat dilihat pada gambar dibawah ini :



Gambar 3.4 Program display karakter LCD

0			T	U	G	A	S			A	K	H	I	R		
1	A	L	A	R	M			K	E	B	A	K	A	R	A	N
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

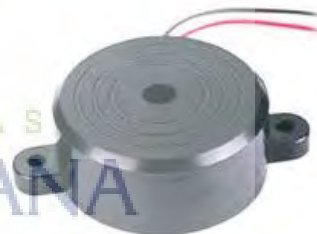
Gambar 3.5 Contoh posisi karakter pada LCD 2 baris x 16 karakter

3.6 LED dan Buzzer

Merupakan suatu komponen yang penting dalam alat alarm kebakaran otomatis, dimana LED sebagai tanda dalam bentuk visual, serta buzzer sebagai tanda bunyi / audio bahwa sedang terjadi kebakaran.. Berikut adalah contoh gambar LED dan Buzzer.



Gambar 3.6 LED



Gambar 3.7 Buzzer

Ada pun kedua alat tersebut merupakan keluaran dari arduino. Maka kedua alat tersebut memiliki program tersendiri untuk mengikuti perintah sesuai dengan yang kita kehendaki. Program pada LED dan Buzzer pada arduino, adalah sebagai berikut.

```
/*  
PROGRAM OUTPUT BUZZER DAN LED  
*/  
  
#define BUZZER 13  
#define LED 12  
void setup()  
{  
  pinMode(BUZZER, OUTPUT);  
  pinMode(LED, OUTPUT);  
}  
  
void loop()  
{  
  digitalWrite(LED, HIGH);  
  digitalWrite(BUZZER, HIGH);  
  delay(1000);  
  digitalWrite(LED, HIGH);  
  digitalWrite(BUZZER, LOW);  
  delay(1000);  
}
```

