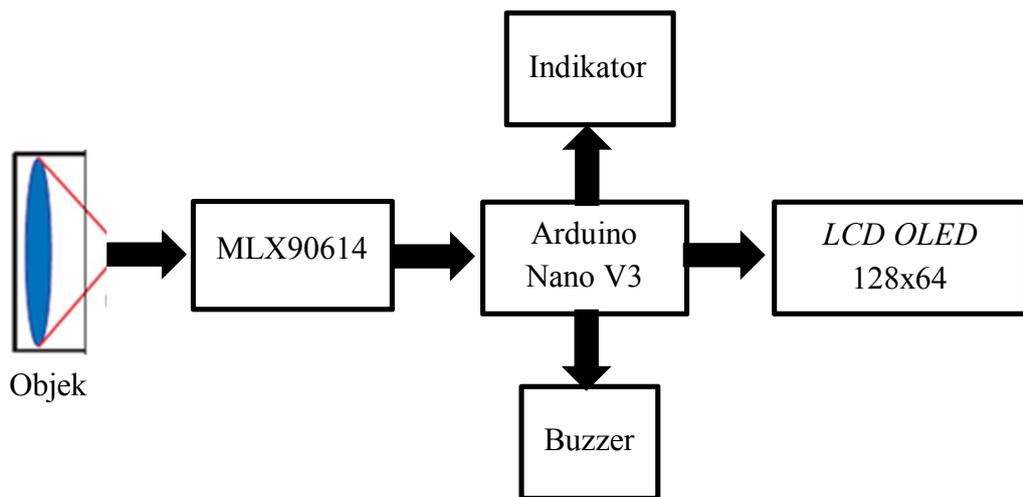


BAB III

METODE PENELITIAN

3.1 Diagram Blok Sistem

Diagram blok sistem pada penelitian ini diuraikan oleh Gambar 3.1:



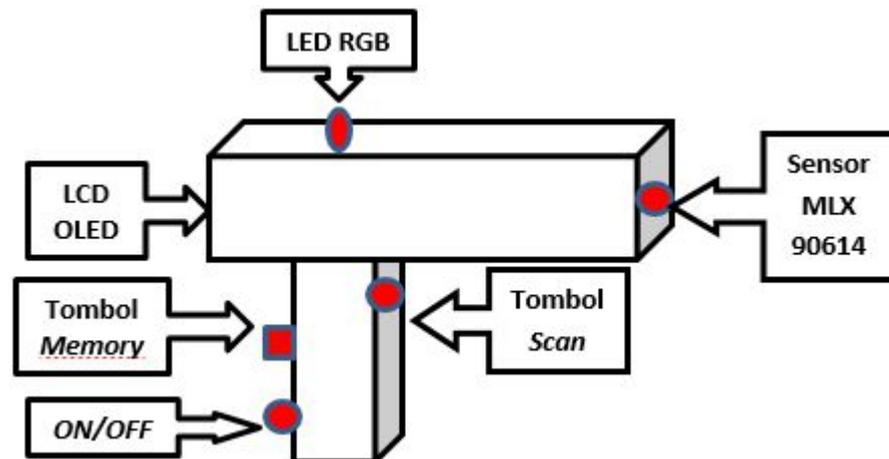
Gambar 3.1 Blok Diagram

Cara kerja pembacaan sensor MLX90614 yaitu adanya pancaran infra merah masuk melalui sensor MLX90614, karena sinar infra merah mengandung energi panas maka sensor MLX90614 akan menghasilkan arus listrik. Arus listrik inilah yang akan menimbulkan tegangan kemudian diubah menjadi sinyal digital oleh sensor. Sinyal akan diolah menjadi harga keluaran yang sesuai dengan temperatur target.

Board Arduino Nano V3 yang mengandung mikrokontroler sebagai tempat pengolahan data dari sensor, serta memberikan perintah pada rangkaian *buzzer* dan indikator. LCD OLED 128x64 sebagai penampil informasi suhu yang terbaca.

3.2 Diagram Mekanis Sistem

Diagram mekanis sistem dalam penelitian ini diuraikan oleh Gambar 3.2:



Gambar 3.2 Diagram Mekanis

Fungsi masing-masing bagian:

1. ON/OFF

Digunakan untuk menghidupkan/mematikan alat

2. Tombol *Scan*

Memulai proses *scanning* infra merah dari objek

3. Sensor MLX90614

Menerima infra merah dari objek dan meneruskannya ke Arduino Nano

4. Led RGB

Sebagai indikator selama maupun sesudah proses *scanning*

5. LCD OLED

Menampilkan keterangan berupa petunjuk penggunaan dan suhu objek

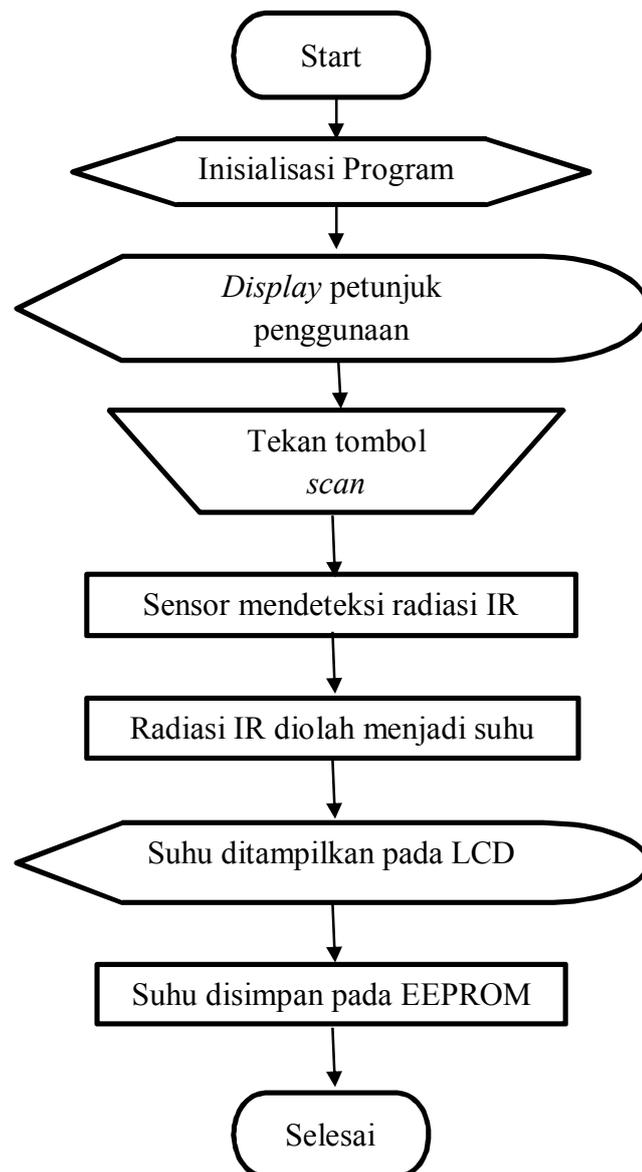
6. Tombol *Memory*

Melihat kembali data pengukuran suhu yang telah dilakukan sebelumnya sebanyak 5 buah suhu secara berurut

3.3 Diagram Alir Proses

1.3.1 Diagram Alir Proses *Scanning*

Diagram alir proses *scanning* dapat dilihat pada Gambar 3.3:

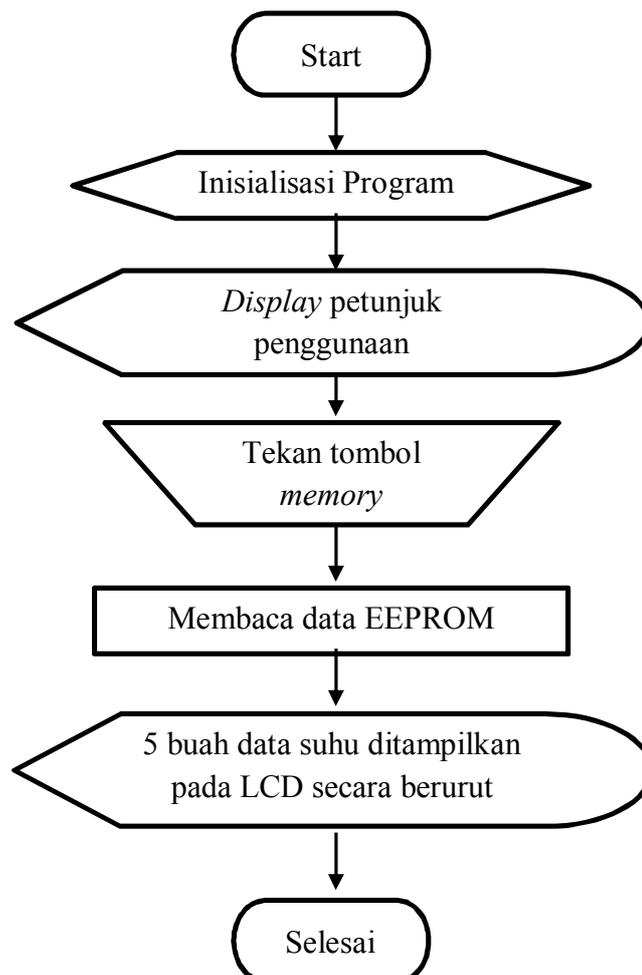


Gambar 3.3 Diagram Alir Proses *Scanning*

Saat tombol power ditekan, tegangan akan masuk ke semua rangkaian. Setelah itu Arduino Nano akan mulai melakukan inisialisasi program kemudian menampilkan petunjuk penggunaan alat pada *display* LCD OLED. Pada saat tombol *scan* ditekan maka sensor akan mendeteksi radiasi infra merah yang dipancarkan oleh objek. Data yang diperoleh dari sensor akan diolah menjadi besaran suhu oleh Arduino Nano kemudian ditampilkan pada *display* LCD OLED dan sekaligus akan disimpan pada memori internal (EEPROM) Arduino Nano.

1.3.2 Diagram Alir Proses *Memory*

Diagram alir proses *memory* dapat dilihat pada Gambar 3.4:



Gambar 3.4 Diagram Alir Proses *Memory*

Pengguna dapat melihat data sebelumnya yang telah diukur sebanyak 5 buah data suhu dengan menekan tombol *memory*. Pada saat tombol *memory* ditekan maka Arduino Nano akan membaca data yang telah tersimpan di EEPROM untuk ditampilkan kembali pada *display* LCD OLED.

3.4 Alat dan Bahan

1. Alat

Alat yang digunakan dalam penelitian ini diuraikan pada Tabel 3.1:

Tabel 3.1 Daftar Alat

No.	Nama	Jumlah
1.	Osiloskop	1
2.	Solder	1
3.	<i>Software</i> Proteus	1
4.	Adaptor	1
5.	Tool set	1
6.	<i>Multimeter</i>	1
7.	Bor	1
8.	<i>Attractor</i>	1
9.	Mata bor	3
10.	Pemanas Air	1
11.	Spidol permanen	1
12.	Setrika	1
13.	Laptop	1
14.	<i>Holder</i> Solder	1

2. Bahan

Bahan yang digunakan dalam penelitian ini pada Tabel 3.2:

Tabel 3.2 Daftar Bahan

No.	Nama	Jumlah
1.	Sensor MLX90614	1
2.	PCB	1
3.	Saklar ON/OFF	1
4.	Arduino Nano V3	1
5.	LCD OLED 128x64	1
6.	Baterai 3,7 V	1
7.	<i>Holder</i> Baterai	1
8.	Push Button	2
9.	Resistor	5
10.	Led RGB	1
11.	Transistor	1
12.	<i>Pin Header</i>	15
13.	Tenol	1
14.	FeCl	1
15.	<i>Amplas/Stillwool</i>	1
16.	Kabel	20

3.5 Alur Penelitian

Dalam alur penelitian alat ini, penulis akan menggambarkan secara garis besar bagaimana penelitian ini dilakukan. Urutan kegiatan dalam penelitian ini adalah sebagai berikut:

1. Mempelajari teori – teori dan mencari referensi yang berhubungan dengan masalah yang akan dibahas.

2. Membuat blok diagram dengan perencanaan secermat mungkin.
3. Mempelajari teknis pembuatan alat sistem *non-contact thermometer* serta menentukan parameter pada alat.
4. Membuat jadwal kegiatan untuk mengatur waktu pembuatan alat.
5. Menyiapkan bahan berupa komponen dan peralatan yang dibutuhkan dalam pembuatan alat.
6. Merancang dan menggabungkan seluruh sistem agar dapat berfungsi dengan baik.
7. Mengimplementasikan rancangan sistem pengendalian *non-contact thermometer* dan melakukan penyempurnaan rangkaian.
8. Perancangan sistem kendali pada *non-contact thermometer* telah selesai dan dilakukan pengujian terhadap sistem yang dirancang.
9. Menganalisis hasil pengujian untuk mendapatkan kesimpulan.

3.6 Implementasi Perangkat Keras

Penjelasan perancangan perangkat keras yang digunakan dalam sistem ini meliputi:

1. Catu Daya

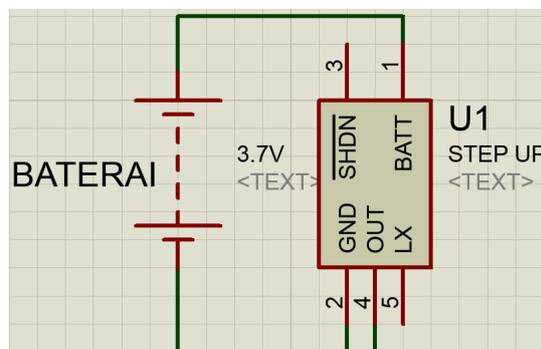
Catu daya yang akan digunakan pada perancangan alat ini adalah baterai *rechargeable* 3,7 V dan modul *step up*. Tanpa adanya masukan daya maka perangkat tidak dapat berfungsi. Begitu juga apabila pemilihan catu daya tidak tepat, maka perangkat tidak dapat bekerja dengan baik. Penentuan sistem catu daya yang akan digunakan ditentukan oleh beberapa faktor, diantaranya:

a. Arus

Arus memiliki satuan AH (*Ampere Hour*). Semakin besar AH, semakin lama daya tahan baterai bila digunakan pada beban yang sama.

b. Teknologi Baterai

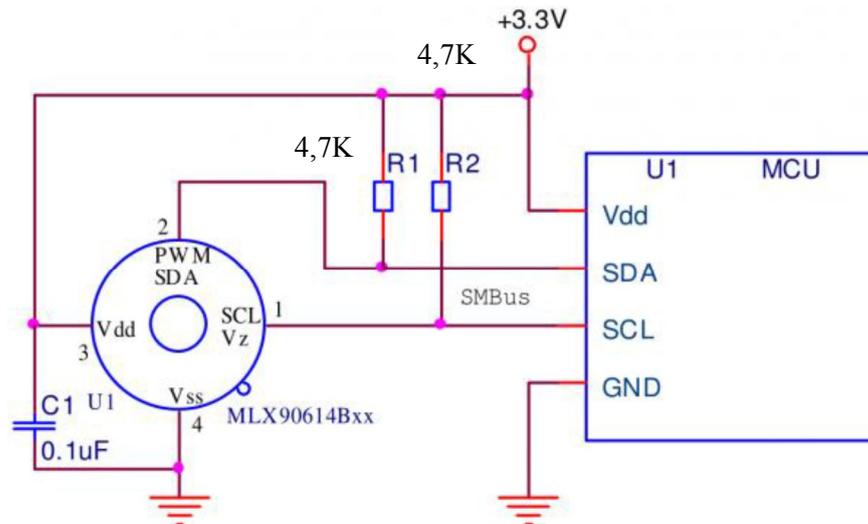
Baterai isi ulang ada yang dapat diisi hanya apabila benar-benar kosong dan ada pula yang dapat diisi ulang kapan saja tanpa harus menunggu baterai tersebut benar-benar kosong.



Gambar 3.5 Skematik Catu Daya

2. Sensor MLX90614

Sensor infra merah yang dibuat oleh Melexis dengan seri MLX90614 ini merupakan sensor yang digunakan pada sistem untuk menangkap radiasi infra merah yang dipancarkan benda dan kemudian diubah menjadi suhu. Prinsip kerja sensor ini menggunakan protokol komunikasi *Inter-Integrated Circuit* (I2C), yaitu komunikasi *synchronous* yang hanya membutuhkan 2 jalur komunikasi untuk mengirim/menerima data yang disebut dengan nama SCL dan SDA. Kabel data (SDA) dan kabel *clock* (SCL), harus di *Pull-UP* dengan resistor eksternal. Konfigurasi pin pada sensor diantaranya yaitu:



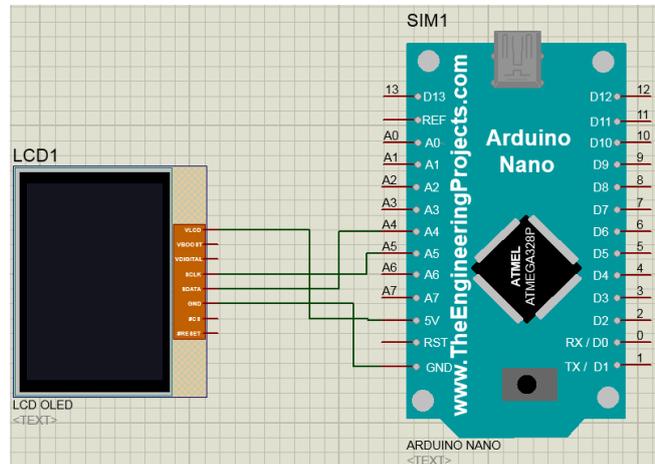
Gambar 3.6 Konfigurasi Pin Sensor MLX90614

- a. Pin VDD dihubungkan pada pin 3,3 V Arduino Nano
- b. Pin VSS dihubungkan pada pin GND Arduino Nano
- c. Pin SCL dihubungkan pada pin A5 Arduino Nano
- d. Pin SDA dihubungkan pada pin A4 Arduino Nano

3. LCD OLED 128x64

Implementasi dari rangkaian penampil pembacaan suhu tubuh manusia dengan menggunakan LCD OLED 128x64 yang bermaterial dasar *organic led* dan resolusi 128x64 piksel dengan dua warna (kuning + biru). Enam belas (16) piksel pada baris bagian atas berwarna kuning dan sisanya 48 piksel berwarna biru. Kelebihan dari display ini yaitu kontras pikselnya yang sangat tajam serta tidak memerlukan cahaya *backlight* tambahan yang membuat konsumsi dayanya menjadi hemat dalam rangkaian.

Berikut ini adalah rangkaian skematik untuk LCD OLED 128x64 di Arduino Nano:

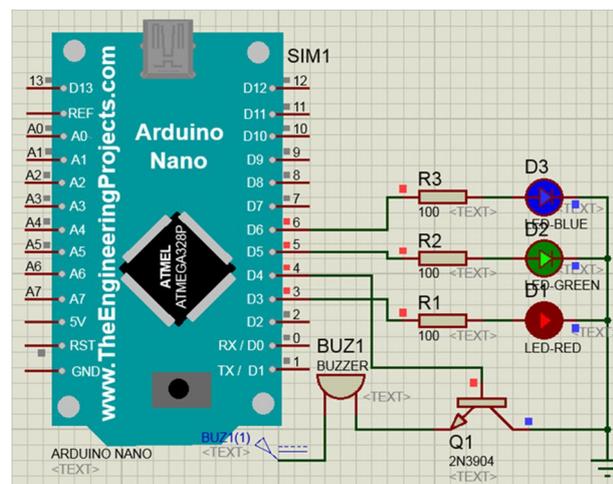


Gambar 3.7 Rangkaian LCD OLED

- Pin VDD dihubungkan pada pin 5 V Arduino Nano
- Pin VSS dihubungkan pada pin GND Arduino Nano
- Pin SCL dihubungkan pada pin A5 Arduino Nano
- Pin SDA dihubungkan pada pin A4 Arduino Nano

4. Buzzer dan Indikator

Berikut rangkaian skematik *buzzer* dan indikator yang telah dirancang oleh penulis:



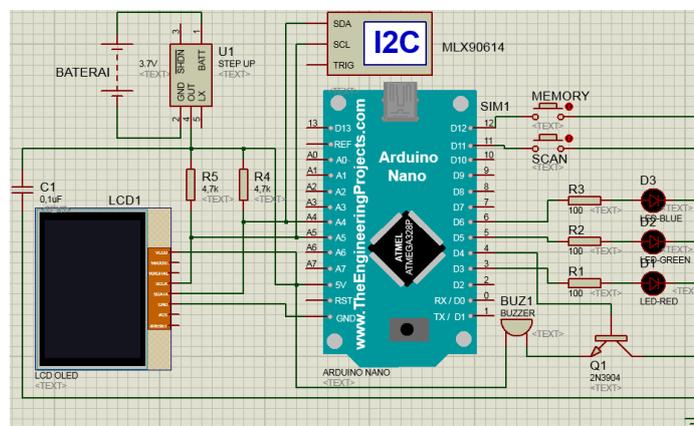
Gambar 3.8 Rangkaian Buzzer dan Indikator

Pada pin positif (+) *buzzer* sudah terhubung ke tegangan 5V dari awal. *Buzzer* akan aktif (berbunyi) yang menandakan sedang melakukan proses *scanning* suhu pada objek ketika tombol *scan* ditekan, yang mana mengakibatkan pin D4 akan berlogika *HIGH* (1). Pin D4 akan mengirimkan tegangan ke pin basis pada transistor NPN yang berperan sebagai saklar, sehingga ground yang awalnya hanya terhubung ke pin *collector* akan ikut tersambung ke pin *emitter* yang mana di pin *emitter* ini akan meneruskan ground ke pin negatif (-) pada komponen *buzzer* sehingga mengakibatkan *buzzer* menghasilkan suara.

Penulis menggunakan led RGB sebagai indikator dimana led ini mempunyai 3 warna sekaligus yaitu *red* (merah), *green* (hijau), dan *blue* (biru). Led akan berwarna biru bersamaan pada saat *buzzer* berbunyi yaitu ketika tombol *scan* ditekan. Led akan berwarna hijau setelah proses *scanning* selesai dengan syarat suhu yang ditampilkan $\leq 37,5^{\circ}\text{C}$ sedangkan jika suhu yang ditampilkan dengan rentang $> 37,5^{\circ}\text{C}$ dan $\leq 40^{\circ}\text{C}$ maka led akan berwarna merah yang menandakan bahwa manusia yang diukur suhunya sedang demam. Pada saat suhu yang ditampilkan lebih dari 40°C maka led akan menyala berkedip-kedip berwarna merah diiringi dengan suara alarm dari *buzzer* yang menandakan manusia yang diukur suhunya mengalami kondisi hipertermia (demam tinggi).

5. Rangkaian Keseluruhan

Simulasi alat dilakukan pada *software* proteus. Prinsip kerja dari alat ini yaitu ketika tombol *scan* ditekan maka sensor yang telah di program akan membaca suhu objek dan suhu tersebut ditampilkan pada tampilan LCD. Penulis menanamkan program, suhu dikatakan tidak normal atau berbahaya apabila suhu tubuh masuk ke dalam kategori demam dan hipertermia. Ketika suhu objek berkisar antara 37,5 - 40°C maka led merah akan menyala disertai dengan keterangan “demam” pada tampilan LCD. Ketika suhu objek lebih dari 40°C maka led merah akan menyala berkedip-kedip disertai dengan keterangan “hipertermia” pada tampilan LCD dan diiringi dengan suara alarm dari *buzzer*.



Gambar 3.9 Rangkaian Keseluruhan

3.7 Implementasi Perangkat Lunak

Alat pengukur suhu tubuh manusia menggunakan infra merah berbasis Arduino Nano V3 menggunakan bahasa pemrograman Arduino IDE. Karena Arduino Nano V3 sebagai rangkaian utama harus membuat program yang sesuai agar sistem dapat berjalan sesuai fungsi yang diinginkan.

Program Arduino IDE adalah program utama yang digunakan untuk menampilkan pemberitahuan nilai hasil pengukuran suhu tubuh manusia di layar LCD, sensor infra merah MLX90614 untuk membaca radiasi objek dan diubahnya menjadi suhu. Penyusunan program dibagi menjadi beberapa bagian yaitu program awalan, program *setup*, dan program *loop*.

1. *Listing* Program Awalan

Pada *Listing* 3.1 merupakan program awalan, dapat dijelaskan bahwa perintah *#include* digunakan untuk memanggil *library* sebuah komponen. Komponen yang dipanggil *library*-nya adalah sensor MLX90614, LCD OLED, dan memori internal (EEPROM) serta terdapat variabel-variabel sebagai pendukung untuk memudahkan dalam melakukan pemrograman.

```
// Include the library:
#include <Adafruit_MLX90614.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <EEPROMex.h>
#include <EEPROMVar.h>
#define OLED_RESET 4

Adafruit_SSD1306 display(OLED_RESET);
Adafruit_MLX90614 mlx = Adafruit_MLX90614(); //
Instantiate a sensor object:

    int save;          int memory;          int measurement;
    float celsius;    float suhu;
int history=6;   int scan=7;
    int red=8;        int buzzer=9;    int green=10;
int blue=11;
    int baca=5;      int batas_penyimpanan=20; //read
memory setiap dikurangi 20
    int last=25;          //memanggil data terakhir yang
tersimpan
    int reset_1;   int reset_2;    int reset_3;    int
reset_4;
```

***Listing* 3.1** Program Awalan

2. Listing Program Setup

Pada program *setup* digunakan untuk menampilkan karakter awalan yang tampil pada LCD ketika pertama kali alat diberi sumber daya (dinyalakan). LCD OLED yang bermaterial dasar *organic led* dan resolusi 128x64 piksel dengan dua warna (kuning + biru). Enam belas (16) piksel pada baris bagian atas berwarna kuning dan sisanya 48 piksel berwarna biru.

Pada saat dinyalakan pertama kali LCD akan memberi tampilan berupa kalimat yang akan menampilkan informasi tentang penulis. Pada kode “*mlx.begin()*,” digunakan untuk menginisialisasi *library* sensor MLX90614. Pada kode “*display.begin (SSD1306_SWITCHCAPVCC, 0x3C) ()*,” digunakan untuk menginisialisasi *library* LCD OLED 128x64.

```
void setup() {
  mlx.begin();
  save = EEPROM.read(last);    //read data from eeprom
  internal
  memory = save - baca;
  measurement = 1;

  // initialize with the I2C addr 0x3C (for the 128x64)
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.setTextColor(WHITE);
  display.clearDisplay();

  pinMode(history, INPUT_PULLUP);
  pinMode(scan, INPUT_PULLUP);

  pinMode(red, OUTPUT);        digitalWrite(red, HIGH);
  pinMode(buzzer, OUTPUT);
  digitalWrite(buzzer, LOW);
  pinMode(green, OUTPUT);
  digitalWrite(green, LOW);
  pinMode(blue, OUTPUT);      digitalWrite(blue, HIGH);

  display.setTextSize(1);    display.setCursor(14,0);
  display.print("Made By :");
  display.setCursor(6,8);
  display.print("Gusti Arya Dinata");
  display.setCursor(6,17);
  display.print("20143010028");
  .....
```

Listing 3.2 Program Setup

Kode “*display.clearDisplay();*” digunakan untuk me-reset tampilan LCD. Kode “*display.setTextSize(1);*” digunakan untuk mengatur ukuran karakter yang akan ditampilkan pada LCD OLED 128x64. Kode “*display.setCursor(x,y);*” digunakan untuk mengatur tata letak karakter yang akan ditampilkan pada LCD OLED 128x64. Kode “*display.print();*” digunakan untuk menampilkan karakter yang diinginkan. Kode “*display.display();*” digunakan untuk menjalankan perintah dari “*display.print();*” yang akan ditampilkan pada LCD OLED 128x64.

Kode “*EEPROM.read(address);*” digunakan untuk membaca/mengambil data dari EEPROM. Parameter “*address*” merupakan alamat EEPROM yang akan dibaca. Kode “*delay(3000);*” digunakan untuk memberikan jeda antar fungsi dalam satuan mili sekon (ms), dimana 1 detik setara dengan 1000 ms. Kode “*pinMode(pin, SET);*” digunakan untuk melakukan konfigurasi secara spesifik fungsi dari sebuah pin apakah sebagai *input* atau *output*. Parameter “SET” dapat diisi *output* atau *input*, tergantung dari kebutuhan, sedangkan parameter “pin” adalah nomor pin pada mikrokontroler yang akan diset sebagai *input* atau *output*.

3. Listing Program Loop

Pada program *loop* inilah program akan berulang terus menerus, hasil pembacaan suhu akan ditampilkan setiap tombol *scan* di tekan. Pada tombol *scan* tidak ditekan maka alat akan dalam keadaan *stand by* dengan menjalankan perintah sesuai program berikut:

```
void loop() {  
  
    if (digitalRead(scan) == 1)  
    {  
  
        // Read the temperature in degrees celsius  
        celsius = mlx.readObjectTempC();  
  
        digitalWrite(red, LOW);  
        digitalWrite(buzzer, LOW);  
        digitalWrite(blue, LOW);  
        digitalWrite(green, LOW);  
  
        display.clearDisplay();  
        display.setTextSize(1);  
        display.setCursor(3,0);  
        display.print("Jarak Objek 1 - 3 cm");  
  
        display.setTextSize(2);  
        display.setCursor(6,14);  
        display.print("Press Scan");  
  
        display.display();  
  
    }  
}
```

Listing 3.3 Program *Stand By*

Pada Listing 3.3 diketahui bahwa jika variabel *scan* pada pin 11 dalam kondisi HIGH(1) maka LCD akan menampilkan informasi berupa petunjuk penggunaan alat yaitu jarak objek 2-3 cm kemudian tekan tombol *scan*. Kode “*digitalWrite(pin, VAL);*” digunakan untuk menuliskan nilai secara digital pada suatu pin. Parameter “*VAL*” dapat berupa HIGH (ON) atau LOW (OFF) dan parameter “*pin*” adalah nomor pin pada Arduino yang akan diset.

Ketika tombol *scan* ditekan maka alat akan menjalankan program

scanning dengan perintah seperti berikut:

```

if (digitalRead(scan) == 0)
{
  // Read the temperature in degrees celsius
  celsius = mlx.readObjectTempC();
  suhu = (1.075 * celsius) - 0.8116;

  display.clearDisplay();
  display.setTextSize(2);
display.setCursor(0,8);
  display.print("Scanning..");
  display.setTextSize(1);
display.setCursor(33,0);
  display.print("Please Wait");
  display.display();
  digitalWrite(buzzer, HIGH);    digitalWrite(blue,
HIGH);
  delay(1000);                    // wait a second

  display.clearDisplay();
  display.setTextSize(3);
display.setCursor(0,12);
  display.print(suhu);
  display.print((char)247);      //simbol
degree/derajat (°)
  display.print("C");
  digitalWrite(blue, LOW);      digitalWrite(buzzer,
LOW);

```

Listing 3.4 Program *Scanning*

Pada Listing 3.4 diketahui bahwa jika variabel *scan* pada pin 11 dalam kondisi LOW(0) maka alat akan mulai memproses radiasi infra merah yang diterima dari objek diiringi dengan suara *buzzer* dan menyalanya led biru selama 1 detik, kemudian infra merah dari objek tersebut akan dikonversi menjadi informasi hasil baca suhu dengan satuan celsius yang ditampilkan pada LCD OLED 128x64.

Kode “*mlx.readObjectTempC()*,” digunakan untuk memulai proses *scanning* infra merah dari objek dan mengubahnya menjadi temperatur dalam satuan celsius. Data temperatur yang telah diperoleh akan diolah lebih lanjut dengan penghitungan matematika agar nilai temperatur lebih akurat.

Ketika proses pengolahan temperatur objek telah selesai, maka secara otomatis data temperatur tersebut akan disimpan pada memori internal (EEPROM) dengan program penulisan EEPROM yang berisi perintah seperti berikut:

```

        if (save > batas_penyimpanan) //value address
back to read from 0
    {
        save = 0;
    }
    reset_1=save + 1;           reset_2=save + 2;
    reset_3=save + 3;           reset_4=save + 4;
    EEPROM.write(save, 0);
EEPROM.write(reset_1, 0);
    EEPROM.write(reset_2, 0);
EEPROM.write(reset_3, 0);
    EEPROM.write(reset_4, 0);
    EEPROM.writeFloat(save, suhu); //write data to
eeprom internal
    save = save + baca;           //place address
read eeprom from 1 to 512
    EEPROM.writeFloat(last, save); //write data to
eeprom internal
    memory = save - baca;
    measurement = 1;

```

Listing 3.5 Program Penulisan EEPROM

Pada Listing 3.5 terdapat kondisi bahwa batas penyimpanan hanya sampai 5 data suhu. Jika sudah memenuhi batas penyimpanan maka akan secara otomatis dilakukan reset untuk diganti dengan data suhu yang baru. Kode “*EEPROM.writeFloat(addr, val)*,” digunakan untuk menulis data ke EEPROM. Parameter “*addr*” merupakan alamat EEPROM yang dimulai dari 0 s/d 1023, sedangkan parameter “*val*” merupakan nilai/data yang

tersimpan dalam alamat EEPROM tersebut. Dikarenakan lebar data hanya 8 bit maka *range* data yang dapat ditampung hanya bernilai 0 s/d 255.

Pada saat proses *scanning* telah selesai, maka alat akan menjalankan program indikator sesuai dengan kondisi yang sudah terpenuhi yang berisikan perintah seperti berikut:

```

display.setTextSize(1);

    if (suhu<=37.5)
    {
        display.setCursor(45,0);
        digitalWrite(green, HIGH);           // GREEN
    }

    if ((suhu>37.5) && (suhu<=40))
    {
        display.setCursor(45,0);
        display.print("DEAMAM");           // RED
        digitalWrite(red, HIGH);
    }

    if (suhu>40)
    {
        display.setCursor(32,0);
        display.print("HIPERTERMIA");       // RED
        digitalWrite(buzzer, LOW);delay(500);
        digitalWrite(buzzer, HIGH);digitalWrite(red,
HIGH);delay(500);

        digitalWrite(buzzer, LOW);digitalWrite(red,
LOW);delay(500);

        digitalWrite(buzzer, HIGH);digitalWrite(red,
HIGH);delay(500);

        digitalWrite(buzzer, LOW);digitalWrite(red,
LOW);delay(500);

        digitalWrite(buzzer, HIGH);digitalWrite(red,
HIGH);delay(500);
        digitalWrite(buzzer, LOW);
    }
}

```

Listing 3.6 Program Indikator

Pada Listing 3.6 diketahui terdapat beberapa kondisi yaitu jika suhu yang ditampilkan $\leq 37,5^{\circ}\text{C}$ maka led hijau akan menyala. Jika suhu yang ditampilkan $> 37,5^{\circ}\text{C}$ dan $\leq 40^{\circ}\text{C}$ maka led merah akan menyala dan akan muncul keterangan “DEMAM” pada tampilan LCD. Jika suhu yang ditampilkan $> 40^{\circ}\text{C}$ maka led merah akan menyala berkedap-kedip diiringi dengan suara alarm dari *buzzer* dan akan muncul keterangan “HIPERTERMIA” pada tampilan LCD.

Jika *user* perlu melihat kembali data-data pengukuran suhu sebelumnya, maka dapat dilihat dengan menekan tombol *memory* yang akan menjalankan program pembacaan EEPROM dengan perintah seperti berikut:

```

    if (digitalRead(history) == 0)
    {
        if (memory < 0)
        {
            memory = batas_penyimpanan;
//value addrees back to read from 20
        }

        if (measurement > 5)
        {
            measurement = 1;           //value
addresses back to read from 1
        }

        display.setTextSize(2);
        suhu = EEPROM.readFloat(memory); //read data
from eeprom internal
        display.clearDisplay();
        display.setCursor(5,10);
        display.print ( "T" );
        display.print (measurement);
        display.print ( "=" );
        display.print (suhu);
        display.print((char)247);      //simbol
degree/derajat (°)
        display.print("C");

```

Listing 3.7 Program Pembacaan EEPROM

Pada Listing 3.7 diketahui bahwa jika variabel *history* pada pin 12 dalam kondisi LOW(0) maka LCD OLED akan menampilkan data suhu sebelumnya sebanyak 5 data suhu terakhir secara berurutan dimulai dari data pengukuran terakhir yang telah dilakukan.

Kode “EEPROM.readFloat(*address*);” digunakan untuk membaca/mengambil data dari EEPROM. Parameter “*address*” merupakan alamat EEPROM yang akan dibaca. Fungsi dari “*measurement*” untuk menandakan nomor pengukuran suhu yang telah dilakukan dari nomor 1 - 5.

3.8 Pengujian Alat

Setelah mengimplementasikan perangkat keras dan lunak maka perlu dilakukan pengujian agar diketahui ketepatan dalam pembuatan alat yang penulis lakukan. Langkah-langkah pengujian alat ini dapat diuraikan sebagai berikut:

1. Menyiapkan objek yang akan di ukur
2. Menyiapkan alat pembanding yang masih dalam kondisi layak penggunaan
3. Menyiapkan tabel pengukuran
4. Melakukan pengujian tahap pertama yaitu pengukuran suhu pada tubuh manusia dengan jarak 1 sampai dengan 4 cm
5. Melakukan pengujian tahap kedua yaitu pengukuran suhu pada partisipan di bagian dahi yang berusia <20 tahun dan ≥ 20 tahun

6. Melakukan pengujian tahap ketiga yaitu pengukuran suhu pada partisipan di bagian lengan
7. Melakukan pengujian tahap keempat yaitu pengukuran suhu pada air yang dihangatkan dengan 5 titik suhu yang berbeda
8. Mencatat hasil pengujian dalam tabel yang telah disiapkan
9. Menganalisa data hasil pengujian untuk memperoleh kesimpulan.