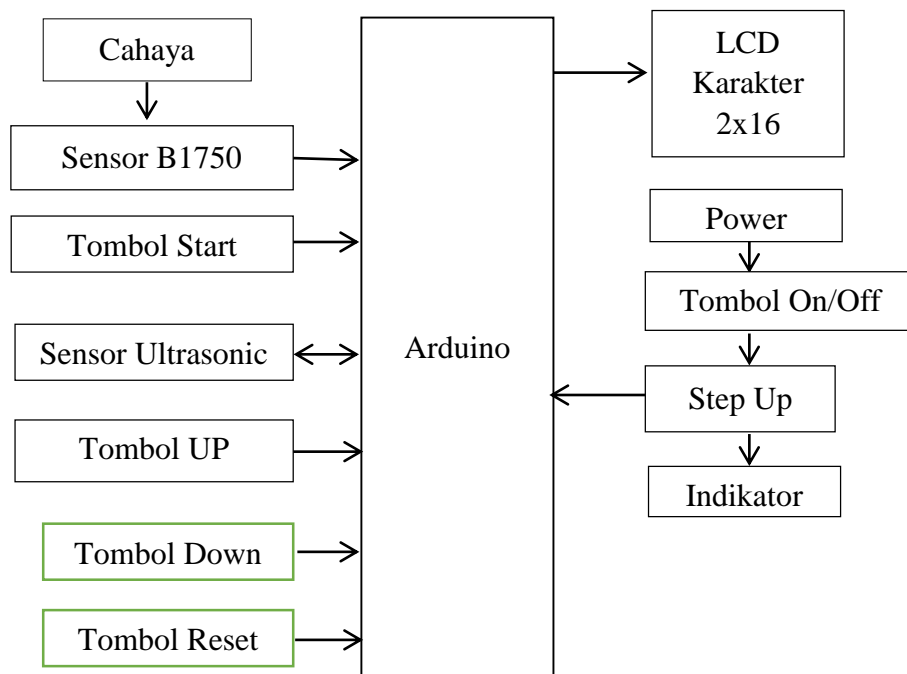


BAB III METODOLOGI PENELITIAN

3.1. Diagram Blok

Dibawah ini adalah digram blok yang penulis buat untuk menjelaskan secara singkat mengenai cara kerja alat penulis. Diagram blok dapat dilihat pada gambar 3.1. di bawah ini.



Gambar 3.1. Diagram Blok

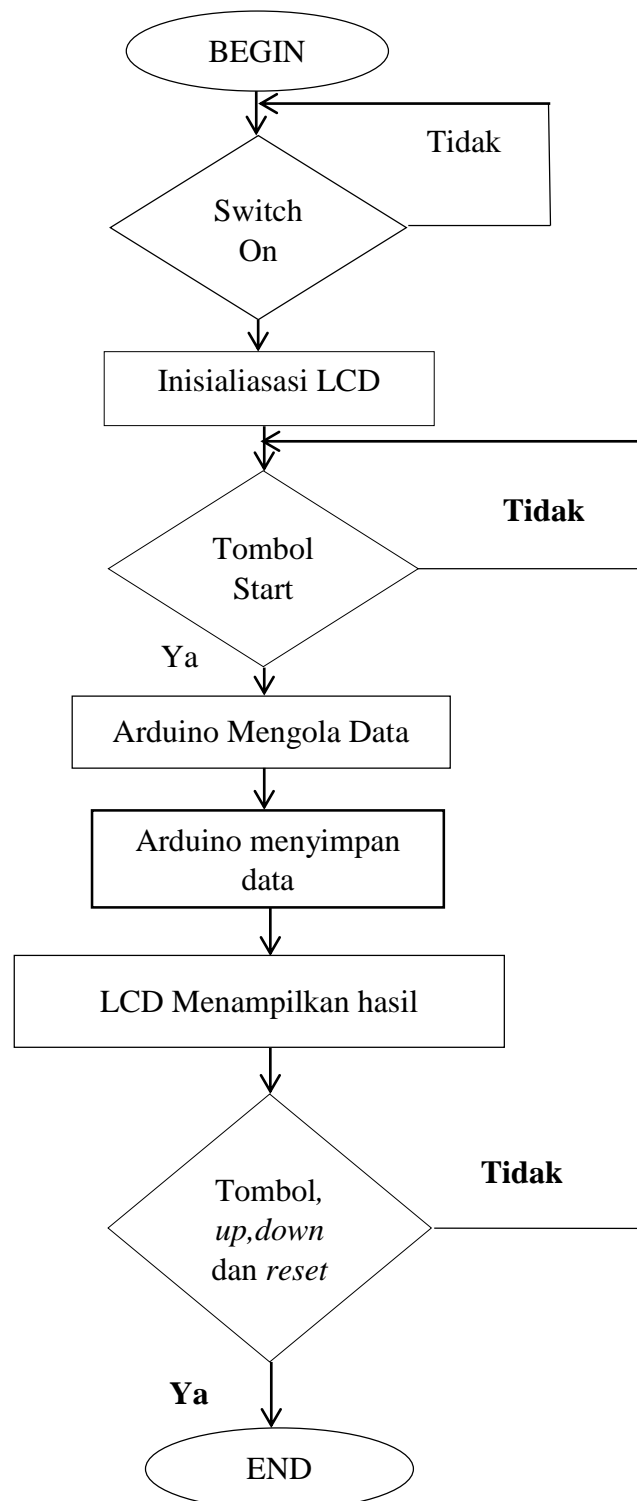
Cara Kerja Diagram Blok

- a. Pada saat tombol power di *ON* kan maka batrai akan mengalirkan arus untuk memberikan *supply* tegangan yang dibutuhkan oleh semua ragkaian dengan melewati *Step Up* tegangan terlebih dahulu bersamaan dengan nyalanya lampu indikator.

- b. Sinar lampu operasi menyinari sensor BH1750FVI sebagai inputan yang akan dikirim ke arduino untuk diolah.
- c. Sensor *ultrasonic* HC-SR04 menghitung panjang gelombang yang di pancarkan oleh *Trasnmiter* ke *casing* lampu operasi dan di pantulkan ke *reciver* dan di kirim ke arduino untuk di olah.
- d. Ketika tombol *start* ditekan, sensor BH1750FVI bekerja untuk membaca intensitas cahaya dan sensor *ultrasonic* HC-SR04 mengukur jarak, selanjutnya masuk ke *Anlog to Digital Comverter (ADC)* Arduino untuk diolah yang kemudian akan ditampilkan pada layar LCD.
- e. Tombol *up* dan *down* digunakan untuk melihat data tersimpan dan data rata-rata pengukuran.
- f. *reset* untuk *zero adjust* untuk mengambil ulang data.

3.2. Diagram Alir

Diagram alir merupakan sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran algoritma atau proses yang menampilkan langkah-langkah yang disimbolkan dalam bentuk kotak beserta urutannya dengan menghubungkan masing-masing langkah tersebut menggunakan tanda panah. Diagram ini bisa memberi selangkah demi selangkah untuk penyelesaian masalah yang ada didalam proses atau algoritma tersebut. Gambar 3.2 di bawah ini adalah gambar diagram alair yang telah penulis susun.



Gambar 3.2. Diagram alir

a. *Begin*

b. Mulai proses alat.

c. *Swicth On*

Alat menyala dan siap digunakan

d. inisialisai LCD

Pengenalan alat dan *ADC*

e. Tombol *Start*

Untuk memulai proses pembacaan. Jika tidak ulang ke tahap nomor 2.

f. Arduino Mengolah Data

Proses pengolahan *ADC* yang di kirim oleh sensor dan diolah menjadi sinyal digital.

g. tombol *Up*

Digunakan untuk melihat data yang tersimpan pertama dan data yang tersimpan selanjutnya

h. Tombol *Down*

Digunakan untuk melihat data yang tersimpan terakhir dan ke data yang tersimpan pertama

i. Arduino Menampilkan Hasil

Untuk menampilkan hasil dari pengolahan data oleh arduino.

j. Tombol *Reset*

Untuk mengulang pengambilan data.

k. Jika mengambil ulang data tekan tombol *reset* lalu tombol *start*

3.3. Perancangan Mekanis Alat

Sebelum meng-*implementasikan* ke bentuk nyata dari alat yang akan di buat terlebih dahulu penulis membuat perancangan bentuk mekanisnya. Rancang fisik dari alat yang dibuat dapat dilihat pada gambar 3.3. dan 3.4 di bawah ini.



Gambar 3.3. Alat tampak dari depan



Gambar 3.4. Alat tampak dari bawah

3.4. Rangkaian *hardware* (implementasi perangkat keras)

Berikut adalah rangkaian *hardware* yang telah penulis buat untuk menyusun alat.

a. Catu Daya

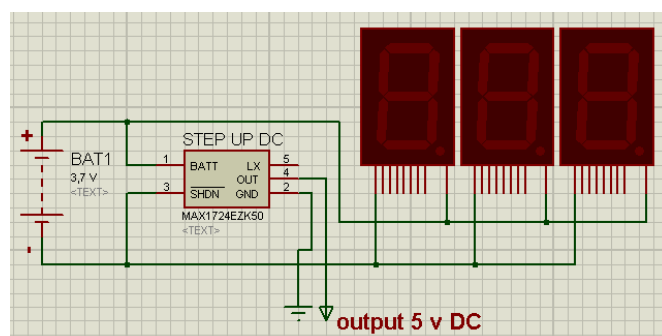
Catu daya yang akan digunakan pada perancangan alat ini adalah baterai *rechargeable* 3,7 V dan modul *step up*. Tanpa adanya masukan daya maka perangkat tidak dapat berfungsi. Begitu juga apabila pemilihan catu daya tidak tepat, maka perangkat tidak dapat bekerja dengan baik. Penentuan sistem catu daya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

1. Arus

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

2. Teknologi Baterai

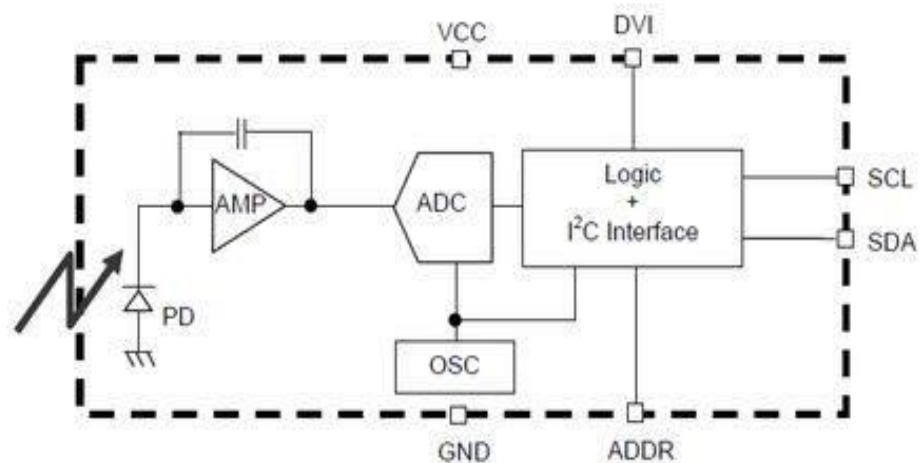
Baterai isi ulang ada yang dapat diisi hanya apabila benar-benar kosong dan ada pula yang dapat diisi ulang kapan saja tanpa harus menunggu baterai tersebut benar-benar kosong. Berikut adalah gambar rangkaian batrai dengan *step up* tegangan *DC* serta *volt* meter untuk indikasi daya yang ada pada batrai. Berikut ini adalah *interfae* (skematik hubungan) batrai dengan *volt* meter, dapat dilihat pada gambar 3.5. di bawah ini.



Gambar 3.5. Rangkaian batrai dengan modul step up tegangan dc

b. Sensor cahaya BH1750FVI

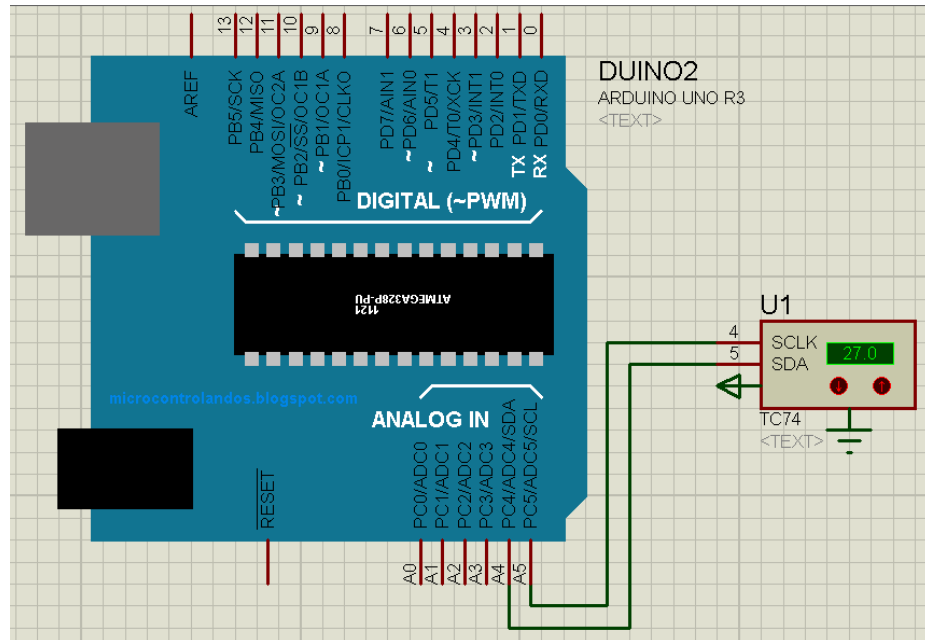
Sensor cahaya BH1750FVI merupakan modul sensor dengan 16 bit *Analog to Digital converter (ADC) built-in* yang dapat langsung output sinyal digital, tidak ada kebutuhan untuk perhitungan yang rumit. Sensor BH1750 ini lebih akurat dan lebih mudah untuk menggunakan, dari pada menggunakan versi *Photodiode*, atau LDR pada umumnya yang hanya *output* tegangan dan perlu dihitung untuk mendapatkan data intensitas. Dengan BH1750FVI intensitas *Light* Sensor dapat langsung diukur dengan keluaran luxmeter (lx), tanpa perlu untuk membuat untuk membuat perhitungan. Konfigurasi pin pada sensor dapat dilihat pada gambar 3.6. di bawah ini.



Gambar 3.6. rangkaian sensor BH 1750FVI

1. Pin VCC dihubungkan pada pin 5 V Arduino
2. Pin GND dihubungkan pada pin GND Arduino
3. Pin SCL dihubungkan pada pin A5 Arduino
4. Pin SDA dihubungkan pada pin A4 Arduino

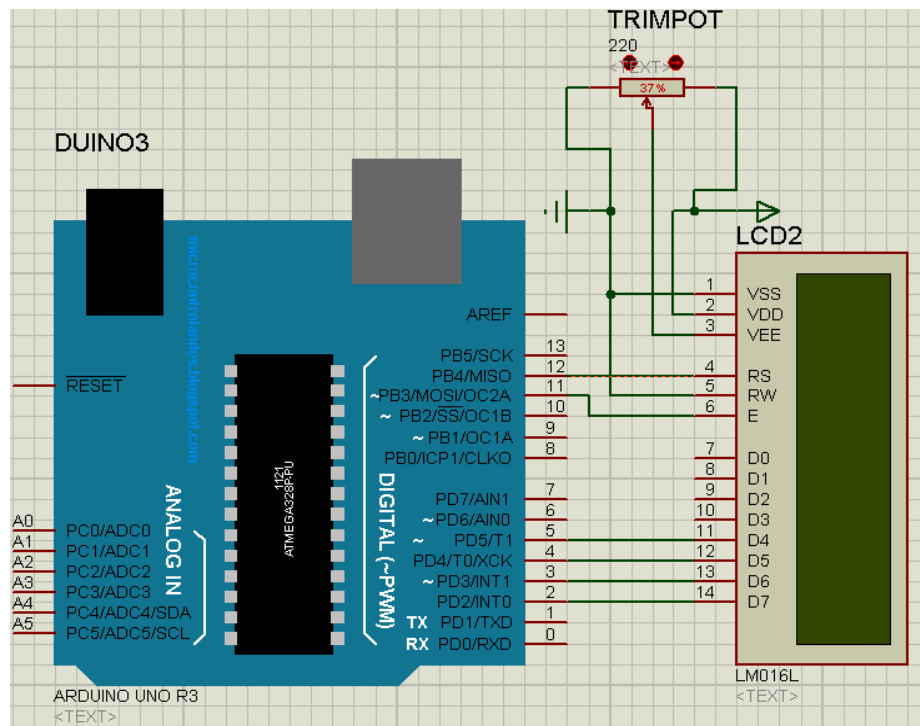
Untuk lebih jelasnya dapat dilihat pada gambar 3.7. di bawah ini,



Gambar 3.7. Rangkaian modul sensor bh1750fvi dengan arduino

c. Rangkaian *Liquid Crystal Display* (LCD)

Pada tugas akhir ini penulis menggunakan LCD *dot matrix* dengan karakter 2 x 16 , sehingga kaki – kakinya berjumlah 16 pin.. LCD ini hanya memerlukan daya yang cukup kecil, tegangan yang dibutuhkan juga rendah yaitu +5 Volt DC. Sebuah *display dot matrix* yang difungsikan untuk menampilkan tulisan berupa angka atau huruf sesuai dengan yang diinginkan. Berikut adalah skematik *implementasi LCD* dengan arduino, dapat dilihat pada gambar 3.8. di bawah ini.

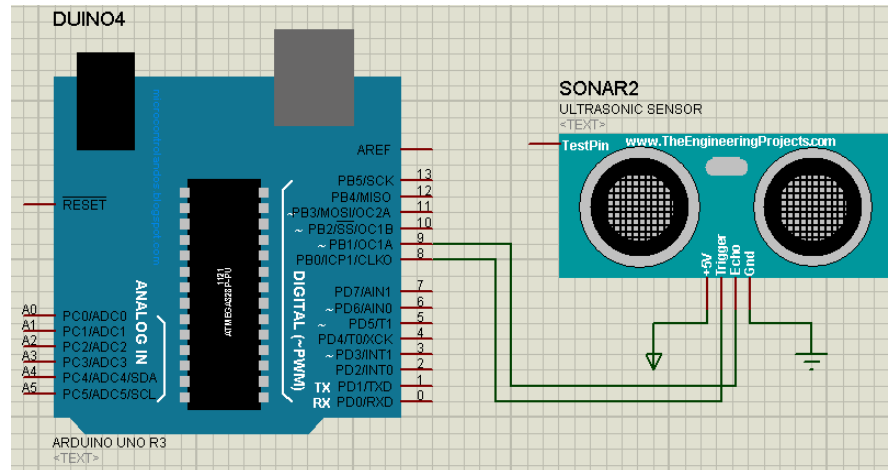


Gambar 3.8. Rangkaian lcd dengan arduino

d. Sensor *ultrasonic* HC-SR04

Sensor jarak yang penulis gunakan yaitu sensor *ultrasonic* HC-SR04 yang merupakan sensor yang digunakan dalam menentukan jarak objek dari modul, sensor ini memiliki rangkaian pengirim dan penerima yaitu *transmitter* dan *receiver*, dimana cara kerja sensor ini adalah, ketika *transmitter* mendapatkan tegangan maka akan mengeluarkan gelombang *ultrasonic* yang akan di kirim dan di pantulkan melalui bidang datar dan akan diterima oleh *reciver*, kemudian diubah menjadi tegangan, tegangan yang dikeluarkan oleh sensor akan masuk ke program *adc*, yaitu mengubah tegangan menjadi digital. Arduino digunakan sebagai pengolah data sensor dalam menentukan jarak. *Pin* yang digunakan adalah *pin* 8 sebagai *receiver* dan 9 sebagai *tranducer* dari sensor. Untuk

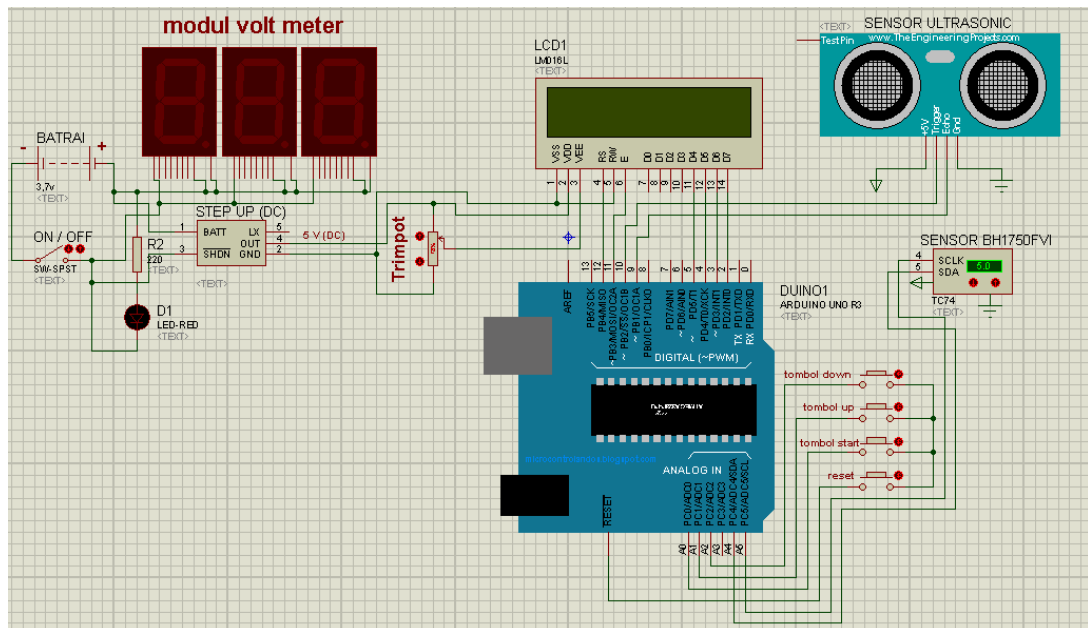
implementasi modul sensor *ultrasonic* HC-SR04 dapat dilihat pada gambar 3.9. dibawah ini.



Gambar 3.9. Rangkaian arduino dengan sensor *ultrasonic* HC-SR04

e. Rangkaian keseluruhan

Gambar skematik dibawah penulis menggunakan *software* proteus untuk merangkai komponen alat. Prinsip kerja pada alat ini cukup sederhana, ketika tombol *on/off* di tekan, led akan menyala dan baterai akan *supply* semua rangkaian yang telah di hubungkan. LCD akan memulai inisialisasi tentang penulis. Dengan menekan tombol start, alat akan mulai bekerja untuk mengambil data jarak dan intensitas cahaya selama 1 menit, ketika waktu tercapai 1 menit, alat akan menyimpan secara otomatis dan pengambilan data selanjutnya di menit yang kedua, begitu seterusnya sampai 6 data akan tersimpan secara otomatis. Untuk melihat data tersimpan digunakan tombol *up* dan *down*. Alat akan menampilkan data rata-rata setelah data ke 6 terlihat. Dan untuk *me-resset* alat, terdapat tombol *reset* untuk memulai ulang kerja alat. Berikut adalah skematik keseluruhan *implimentasi* alat yang telah penulis buat, dapat di lihat pada gambar 3.10. di bawah ini.



Gambar 3.10. rangkaian keseluruhan komponen.

3.5. Listing Program *Soft Whare* (Implementasi Program Perangkat Lunak)

Untuk pembuatan program pada alat penulis yang berjudul Lux Meter Dilengkapi Sensor Jarak Berbasis Arduino ini menggunakan aplikasi Arduino *Integrated Development Environment* (IDE) dengan bahasa program menggunakan bahasa C.

Program Arduino IDE adalah program utama yang digunakan untuk menampilkan nilai hasil pengukuran intensitas cahaya, jarak dan waktu di layar LCD, sensor BH1750FVI untuk membaca intensitas dari lampu operasi, sensor *ultrasonic* HC-SR04 digunakan untuk membaca jarak pengukuran. Penyusunan program dibagi menjadi beberapa bagian yaitu program utama (*declaration*), program inialisasi (*setup*), dan program pengulangan (*looping*).

a. *Listing Program Utama (declaration)*

Pada program utama atau deklarasi, dapat dijelaskan bahwa perintah *#include* digunakan untuk memanggil *library* sebuah komponen. Komponen yang dipanggil *library*-nya adalah sensor BH1750FVI, sensor *ultrasonic* HC-SR04, *LCD*, *i2c* dan *TimerOne* serta terdapat variabel-variabel sebagai pendukung untuk memudahkan dalam melakukan pemrograman. Berikut adalah *listing* program utama.

```
// library timer 1

#include <TimerOne.h>

// library lcd

#include <LiquidCrystal.h>

// konfigurasi lcd

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

// i2c library

#include <Wire.h>

// BH1750 library

#include <BH1750.h>

// BH1750 konfigurasi

BH1750 lightMeter;

// i/o konfigurasi

#define ledpin    13

#define echo      8

#define trigger   9

#define sw1       A0

#define sw2       A1
```

```

#define sw3      A2

int timer=6;

int menit=0, detik=0, last_menit;

unsigned int lux=0;

float distance;

bool kodesstart=0;

Float unsigned int lux_log[6]={0,0,0,0,0,0};

```

b. Program inisialisasi (*Void Setup*)

Program ini bertujuan untuk menginisialisasi program yang di eksekusi atau program yang akan dijalankan pertama kali dan hanya satu kali setelah alat diberikan catu daya dan setelah alat di *reset*, program inilah yang akan tertampil pada lcd pertama kali. Berikut adalah *listing* programnya.

```

void setup() {

// set BH1750 konfigurasi

  lightMeter.begin();

// set lcd konfigurasi

  lcd.begin(16,2);

// atur pin i/o

  pinMode(ledpin, OUTPUT);

  pinMode(triger, OUTPUT);

  pinMode(echo, INPUT);

// konsidi pullup (bernilai 1 saat tidak dapat input)

  pinMode(sw1, INPUT_PULLUP);

```

```

pinMode(sw2, INPUT_PULLUP);

pinMode(sw3, INPUT_PULLUP);

// aktifkan timer interrupt

Timer1.initialize(1000000); // atur ke mode 1000000 microseconds ( 1s )

// panggil program ISR

Timer1.attachInterrupt( timerIsr );

lcd.clear();

lcd.print("Bismillah");

delay(1000);

}

```

c. Program pengulangan (*Looping*)

Program *looping* atau yang lebih dikenal dengan program pengulangan secara terus menerus, hasil pembacaan sensor dan timer akan di tampilkan pada *LCD* setelah tombol start ditekan. Pada kondisi dimana tombol *start* tidak ditekan maka alat dalam keadaan *stand by* dan akan menjalankan perintah dari program berikut :

```

void loop() {

// panggil fungsi program

program();

}

// fungsi timer ISR

// fungsi ini dipanggil di attach interrupt (program dalam mode interupsi)

```

```
void timerIsr()
{
  // jika kode start bernilai 1
  if(kodestart==1){
    // kedipan led penanda timer
    digitalWrite( ledpin, digitalRead( ledpin ) ^ 1 );
    // timer mencacah
    if(menit<timer){
      detik++;
      if(detik>59){
        menit++;
        detik=0;
      }
    }
  }
}

// fungsi baca sensor ultrasonik
float read_ultra() {
  // lokal variable
  float mydistance;
  // suara ultrasonik on
  digitalWrite(triger, HIGH);
  // jeda
  delayMicroseconds(50);
  // suara ultrasonik off
```

```
digitalWrite(triger, LOW);  
  
// baca pulsa dan konvert ke cm  
  
mydistance=(float)pulseIn(echo,1)/58;  
  
// nilai balik  
  
return mydistance;  
  
}  
  
// fungsi program  
  
void program(){  
  
// baca sensor  
  
    if(kodestart==1)lux=lightMeter.readLightLevel();  
    distance=read_ultra()/100; // konversi ke meter  
    if(menit<timer){  
  
// tampilkan  
  
        lcd.clear();  
        lcd.setCursor(0,0);  
        lcd.print("Lux:");  
        lcd.print(lux);  
        lcd.setCursor(10,0);  
        lcd.print(menit);  
        lcd.print(":");  
        lcd.print(detik);  
        lcd.setCursor(0,1);  
        lcd.print("Jarak:");  
        lcd.print(distance,1);  
        lcd.print(" m");  
  
    }
```



```
else{  
    lcd.clear();  
    lcd.setCursor(0,0);  
    lcd.print("    Selesai    ");  
}  
  
// jika tombol di tekan mulai start timer  
if(digitalRead(sw1)==0){  
  
// pengaturan awal start timer  
    kodelistart=1;  
    detik=0;  
    menit=0;  
    last_menit=0;  
  
// set nol data loger  
    for(int i=0;i<6;i++){  
        lux_log[i]=0;  
        distance_log[i]=0;  
    }  
}  
  
if(kodelistart==1)  
{  
  
// menyimpan ke data loger saat menit mencacah  
    if(last_menit!=menit){  
        lux_log[menit-1]=lux;  
        distance_log[menit-1]=distance;  
        last_menit=menit;  
    }  
}
```

```

}

// jika tombol di tekan masuk untuk melihat data loger
if(digitalRead(sw2)==0) {
  int index=0; // lokal variabel

  // pengulangan
  long jumlahlux=0,rata2lux=0;
  float jumlahjarak=0,rata2jarak=0;
  for(int i=0;i<6;i++)jumlahlux=jumlahlux+lux_log[i];
  for(int i=0;i<6;i++)jumlahjarak=jumlahjarak+distance_log[i];
  rata2lux=jumlahlux/6;
  rata2jarak=jumlahjarak/6;
  while(1){

  // tampilkan
  if(index<6){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(index+1);
    lcd.setCursor(2,0);
    lcd.print("Lux:");
    lcd.print(lux_log[index]);
    lcd.setCursor(2,1);
    lcd.print("Jarak:");
    lcd.print(distance_log[index],1);
    lcd.print(" m");
  }
  else{

```

```
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print(index+1);  
lcd.setCursor(2,0);  
lcd.print("RLux:");  
lcd.print(rata2lux);  
lcd.setCursor(2,1);  
lcd.print("RJarak:");  
lcd.print(rata2jarak,1);  
lcd.print(" m");  
}  
  
// naik turunkan index dengan tombol  
if(digitalRead(sw2)==0)index++;  
if(digitalRead(sw3)==0)index--;  
  
// nilai batas index  
if(index>6)index=0;  
if(index<0)index=6;  
  
// keluar dari mode tampilan data loger  
if(digitalRead(sw1)==0)break;  
delay(150);  
}  
  
// set 0 kode start  
kodelistart=0;  
lcd.clear();  
delay(200);  
}
```


13	Perbaiki Eksperimen											
14	Ujicoba dan Evaluasi Metode											
15	Perbaiki bahan penelitian											
16	Tulis Bab IV dan V											
17	Perbaiki dan Rapikan KTI											
18	Presentasi Pra Ujian Pendadaran											
19	Persiapan dan Administrasi Ujian Pendadaran											
20	Ujian Kelayakan dan Pendadaran											

3.7. Bahan yang digunakan

Adapun komponen-komponen penting dalam pembuatan modul ini, penulis sudah memuatnya dalam tabel 3.2. di bawah ini.

Tabel 3.2. Bahan-bahan pembuatan alat

No	Nama Bahan	Jumlah
1	modul arduino	1
2	modul sensor BH1750FVI	1
3	Modul sensor ultrasonik HC-SR04	1
4	Modul <i>charger</i>	1
5	Batrai <i>li-Ion</i>	1
6	<i>Step Up tegangan DC to DC</i>	1
7	<i>LCD (Liquid Crystal Display</i>	1
8	kabel jumper	30
9	Box	1
10	Trimpot	1

11	Resistor	1
12	Kabel <i>USB</i>	1
13	Tenol	1

3.8. Peralatan Yang Digunakan

Tabel 3.3. di bawah ini adalah peralatan yang digunakan selama pembuatan tugas akhir ini antara lain :

Tabel 3.3. Alat-alat pembuatan alat

No	Nama Alat	Spesifikasi	Jumlah	Pemilik
1.	Tang potong	Blitz	1	Sendiri
2.	<i>Cutter</i>	Kenko K200	1	Sendiri
3.	Solder listrik	Deko DS40N 40W 220-240V	1	Sendiri
4.	Attractor	Deko	1	Sendiri
5.	Obeng 31 in 1	Celkit ck6030A	1	Sendiri
6.	Pasta solder	-	1	Sendiri
7.	Seperangkat laptop	Hp	1	Pinjam
8.	Meteran	Soligen	1	Pinjam

3.9. Analisa Data

Merujuk pada rumus analisa data pada dasar teori, dapat dihitung nilai rata-rata, *error*, simpangan, standar deviasi dan nilai ketidakpastian. Pada pengukuran intensitas cahaya lampu operasi selama 1 menit dengan jarak pengukuran 1 meter, pengukuran dilakukan sebanyak 20 kali.

Tabel 3.4. dibawah ini merupakan hasil pengukuran intensitas cahaya lampu operasi dengan alat penulis yang di bandingkan dengan alat *lux* meter pada jarak ukur 1 meter dari sumber cahaya lampu operasi.

Tabel 3.4. Tabel pengukuran cahaya selama 1 menit pada jarak 1 meter

No	Waktu (menit)	Jarak (meter)	Intensitas terbaca pada alat pembanding (<i>lux</i>)	Intensitas terbaca pada alat tugas akhir (<i>lux</i>)
1	01:00	1,0	14.590	14.621
2	01:00	1,0	14.590	14.620
3	01:00	1,0	14.590	14.618
4	01:00	1,0	14.620	14.618
5	01:00	1,0	14.620	14.618
6	01:00	1,0	14.640	14.609
7	01:00	1,0	14.640	14.620
8	01:00	1,0	14.640	14.630
9	01:00	1,0	14.640	14.645
10	01:00	1,0	14.660	14.652
11	01:00	1,0	14.680	14.652
12	01:00	1,0	14.680	14.653
13	01:00	1,0	14.680	14.653
14	01:00	1,0	14.690	14.660
15	01:00	1,0	14.690	14.665
16	01:00	1,0	14.690	14.702
17	01:00	1,0	14.690	14.704
18	01:00	1,0	14.690	14.706
19	01:00	1,0	14.690	14.706
20	01:00	1,0	14.690	14.706

Berikut ini adalah analisa perhitungan dari data yang sudah di dapatkan.

a. Rata-rata

Rata – rata adalah nilai atau hasil pembagian dari jumlah data yang diambil atau diukur dengan banyaknya pengambilan data atau banyaknya pengukuran.

$$\text{Rata – Rata } (\bar{X}) = \frac{\sum Xi}{n} \quad (1)$$

Dimana :

$$\bar{X} = \text{rata – rata}$$

$$\sum Xi = \text{Jumlah nilai data}$$

$$n = \text{Banyak data } (1,2,3,\dots,n)$$

Maka ;

$$\text{Rata – Rata pembanding } (\bar{X}) = \frac{293.100}{20} = 14.655$$

$$\text{Rata – Rata alat penulis } (\bar{X}) = \frac{293058}{20} = 14.653$$

b. Simpangan %

Simpangan adalah selisih dari rata–rata nilai harga yang dikehendaki dengan nilai yang diukur. Berikut rumus dari simpangan :

$$\text{Simpangan } \bar{X} = n - Y$$

Dimana :

$$Y = \text{rata-rata lux pembanding}$$

$$\bar{Xn} = \text{rerata data alat}$$

Maka ;

$$\text{Simpangan} = Xn - \bar{X}$$

$$\text{Simpangan} = 14.655 - 14.653$$

$$\text{Simpangan} = 2 \%$$

c. Error (%)

Error (kesalahan) adalah selisih antara *mean* terhadap masing-masing data. Rumus *error* adalah:

$$\text{Error \%} = \left(\frac{\text{DataSeting} - \text{Re rata}}{\text{Datasetting}} \right) \times 100\%$$

Maka ;

$$\text{Error \%} = \left(\frac{14.655 - 14.653}{14.655} \right) \times 100\% = 0,01 \%$$

d. Standart deviasi

Standart deviasi adalah suatu nilai yang menunjukkan tingkat (derajat) variasi kelompok data atau ukuran *standart* penyimpangan dari *meannya*.

Rumus *standart deviasi* (SD) adalah:

$$SD = \sqrt{\frac{\sum^n (X_i - \bar{X})^2}{(n-1)}}$$

Dimana :

SD = *standart Deviasi*

\bar{X} = nilai yang dikehendaki n = banyak data

maka ;

$$SD = \sqrt{\frac{(14.621 - 14.655)^2 + (14.620 - 14.655)^2 + (14.618 - 14.655)^2 + (14.618 - 14.655)^2 + (14.609 - 14.655)^2 + (14.620 - 14.655)^2 + (14.630 - 14.655)^2 + (14.645 - 14.655)^2 + (14.652 - 14.655)^2 + (14.652 - 14.655)^2 + (14.653 - 14.655)^2 + (14.653 - 14.655)^2 + (14.660 - 14.655)^2 + (14.665 - 14.655)^2 + (14.702 - 14.655)^2 + (14.704 - 14.655)^2 + (14.706 - 14.655)^2 + (14.706 - 14.655)^2}{(20 - 1)}}$$

$$SD = \sqrt{\frac{23029,8}{(19)}} = 34.82$$

e. Ketidakpastian (Ua)

Ketidakpastian adalah kesangsian yang muncul pada tiap hasil.

Atau pengukuran biasa disebut, sebagai kepresisian data satu dengan data yang lain.

Rumus dari ketidakpastian adalah sebagai berikut:

$$\text{Ketidakpastian} = \frac{stdv}{\sqrt{n}}$$

Dimana :

STDV = *Standar Deviasi*

n = banyaknya data

Maka ;

$$Ua = \frac{SD}{\sqrt{n}}$$

$$Ua = \frac{34,82}{\sqrt{20}} = 7,7$$

