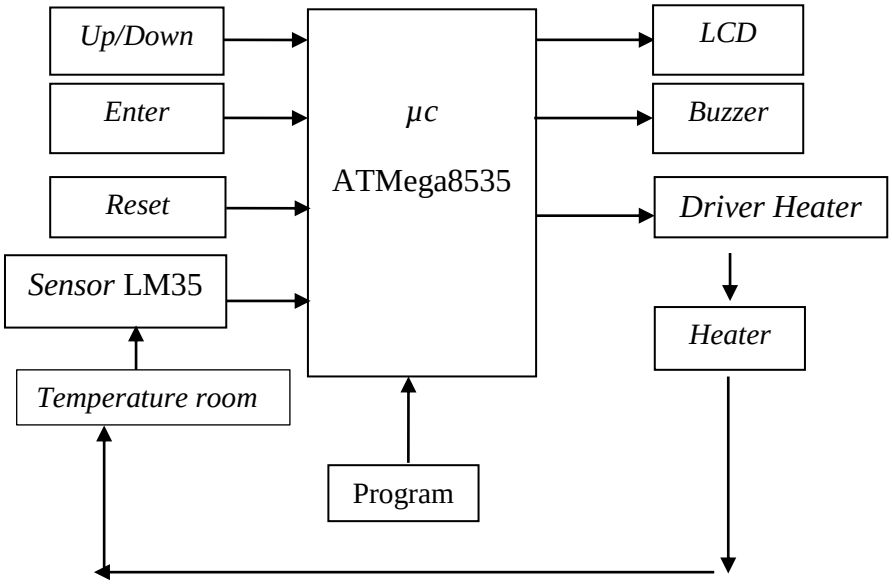
BAB III
METODOLOGI PENELITIAN

3.1 Perancangan Perangkat Keras

3.1.1 Diagram Blok Sistem

Adapun blok diagram sistem dari inkubator bakteri dilengkapi dengan suhu dan *timer* berbasis *mikrokontroler* ATMega8535, dapat dilihat pada Gambar 3.1.



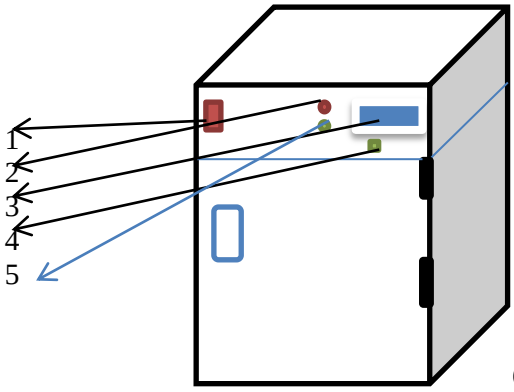
Gambar 3.1 diagram blok inkubator bakteri

Berdasarkan Gambar 3.1 IC ATMega8535 sebagai pusat pengendali data *input* dan *output*. IC ini akan bekerja berdasarkan dengan kode program yang telah di masukan kedalam IC *Mikrokontroler* ini. Kode program tersebut di masukan melalui *pin* SPI *bus master input/ slave output* (MISO), SPI *bus master output/ slave input* (MOSI), dan SPI *bus serial clock* (SCK). Data *input* yang masuk pada IC ATMega8535 yaitu sensor LM 35, Tombol *up, down,*

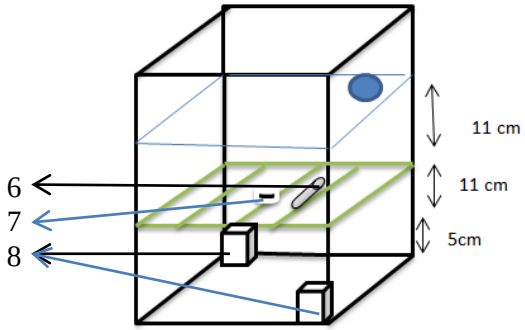
enter, *reset/silent.* Tombol tersebut digunakan untuk mengatur pemilihan menu pada layar *LCD.* Tombol *up* untuk menaikan data, tombol *down* untuk menurunkan data, dan tombol *enter* mengakses menu yang sudah dipilih. Data yang keluar akan ditampilan oleh LCD karakter 2x16

3.1.2 Diagram Mekanis Sistem

Berikut merupakan diagram mekanis sistem dari Inkubator bakteri dilengkapi dengan suhu dan *timer* berbasis *mikrokontroler* ATMega8535, dapat dilihat pada Gambar 3.2.



Gambar a



Gambar b

Gambar 3.2 Diagram Mekanis Inkubator Bakteri

a. Tampak luar

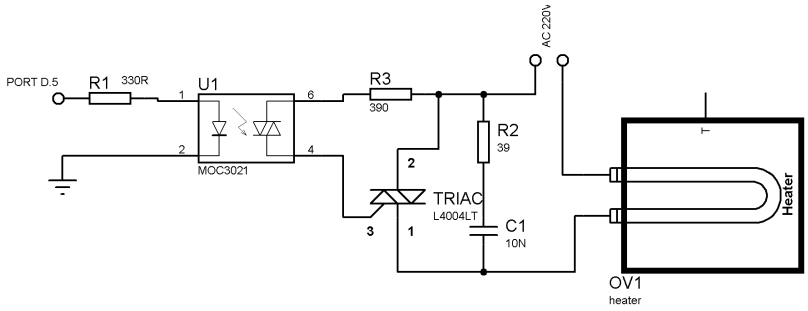
b. Tampak dalam

Pada gambar 3.2 modul inkubator bakteri memiliki volume 25cm x30cm x35cm. Di ruang inkubasi terdapat *plate* yang digunakan untuk peletakan *heater* dan meletakkan cawan petri yang digunakan untuk proses inkubasi bakteri. Berikut fungsi dari tombol-tombol yang terdapat pada inkubator bakteri, diantaranya:

1. Tombol *ON/OFF,* berfungsi untuk mematikan dan menyalakan inkubator bakteri.
2. Tombol *Up,* berfungsi untuk menambah waktu .
3. LCD karakter 2x16, sebagai tampilan dari inkubator bakteri.
4. Tombol *Reset,* untuk memulai ulang sistem dari inkubator bakteri.
5. Tombol *Down,* mengurangi waktu
6. *Heater*
7. *Sensor* LM35
8. *Fan*

3.1.3 Rangkaian *Driver Heater*

Rangkaian *driver heater* yang digunakan menggunakan MOC3041 dan Triac L4004LT yang ditunjukkan oleh Gambar 3.3.



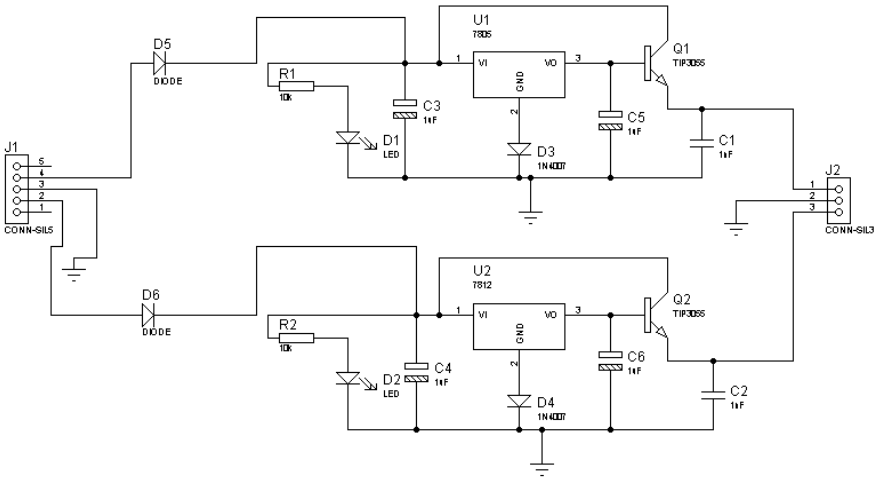
Gambar 3.3 Skematik Rangkaian *Driver Heater*

Rangkaian *driver* pada modul inkubator bakteri berfungsi sebagai pengkontak dari tegangan DC ke tegangan AC. Komponen MOC 3041 yang mendapatkan tegangan 5V dari mikrokontroler akan saturasi (*switch ON* sehingga berfungsi untuk mengidupkan *triac* 4004LT dengan kontak AC untuk menghidupkan pemanas. Ketika MOC3041 tidak mendapatkan tegangan maka *triac* akan mati karena *gate* tidak mendapatkan tegangan dari MOC dan pemanas mati *(Zigan, 2009)*.

Pada penggunan resistor untuk *driver heater* yaitu R1 sama dengan Vs dikurangi Vd yang hasilnya akan dibagi dengan arus, sehingga R1= (Vs-Vd)/I = (5-1,6)/0,01 = 340 Ohm *(Tunggal et al. 2016)*.

3.1.4 Modul Rangkaian *Power Supply*

Power supply yang digunakan inkubator bakteri ini menggunakan masukan tegangan sebesar 220V yang kemudian akan diturunkan oleh *transformator step down* sehingga keluaran tegangan 12 V dan 5 V. Gambar 3.4 menunjukkan *power supply* inkubator bakteri.

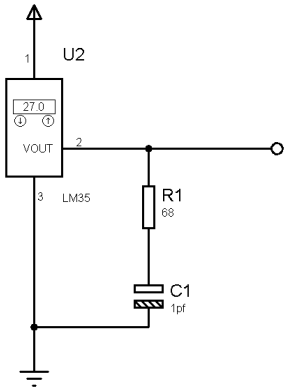


Gambar 3.4 *Power supply* Inkubator Bakteri

Berdasarkan Gambar 3.4 rangkaian *power supply* menggunakan 2 dioda IN5392 2A yang berfungsi sebagai penyearah tegangan AC menjadi tegangan DC. Tegangan yang masuk dari *transformator* sebesar 6VAC dan 12VAC akan disearahkan sehingga, tegangan yang akan berubah menjadi DC yang kemudian akan diturunkan oleh IC LM7805 dan 7812. Sehingga tegangan keluaran dari *power supply* adalah 5VDC dan 12VDC. Pada modul tegangan 5 VDC akan digunakan untuk mensuplai minimum sistem, serta tegangan 12 VDC digunakan untuk menyalakan *fan.*

3.1.5 Rangkaian *Sensor* LM35

Rangkaian *sensor* LM35 yang digunakan diambil dari *datasheet* LM35. Gambar 3.5 menunjukkan rangkaian *sensor* LM 35.



Gambar 3.5 Rangkaian LM35

Berdasarkan Gambar 3.5 rangkaian LM35 mendapatkan tegangan masukan 5VDC, sehingga membuat *sensor* aktif. Ketika *sensor* LM35 mendeteksi suhu sekitar, maka tegangan output di PIN.2 berubah menjadi 10mV setiap terjadi kenaikan suhu 1˚ C pada *chamber*. Fungsi rangkaian RC sebagai penyaring sinyal dengan memberikan tekanan/ blok. Berikut perhitungan rangkaian RC dengan mencari *frekuensi cut off:*

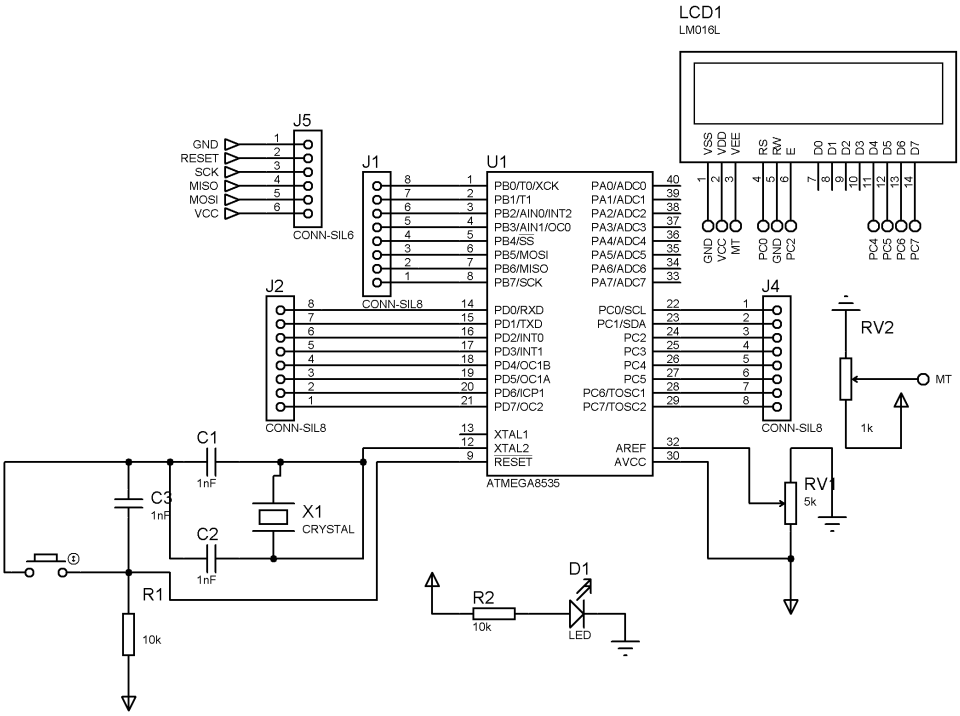
$$fc = \frac{1}{2\pi fc} \tag{3.1}$$

$$= 2341{,}7010\ Hz$$

Jadi, *frekuensi cutoff* nya adalah 2341, 7010 Hz.

3.1.6 Rangkaian *Minimum System*

Gambar 3.2 merupakan sistematik gambar minimum sistem ATMega8535 menggunakan aplikasi *proteus*.



Gambar 3.6 Rangkaian *minimum sistem*

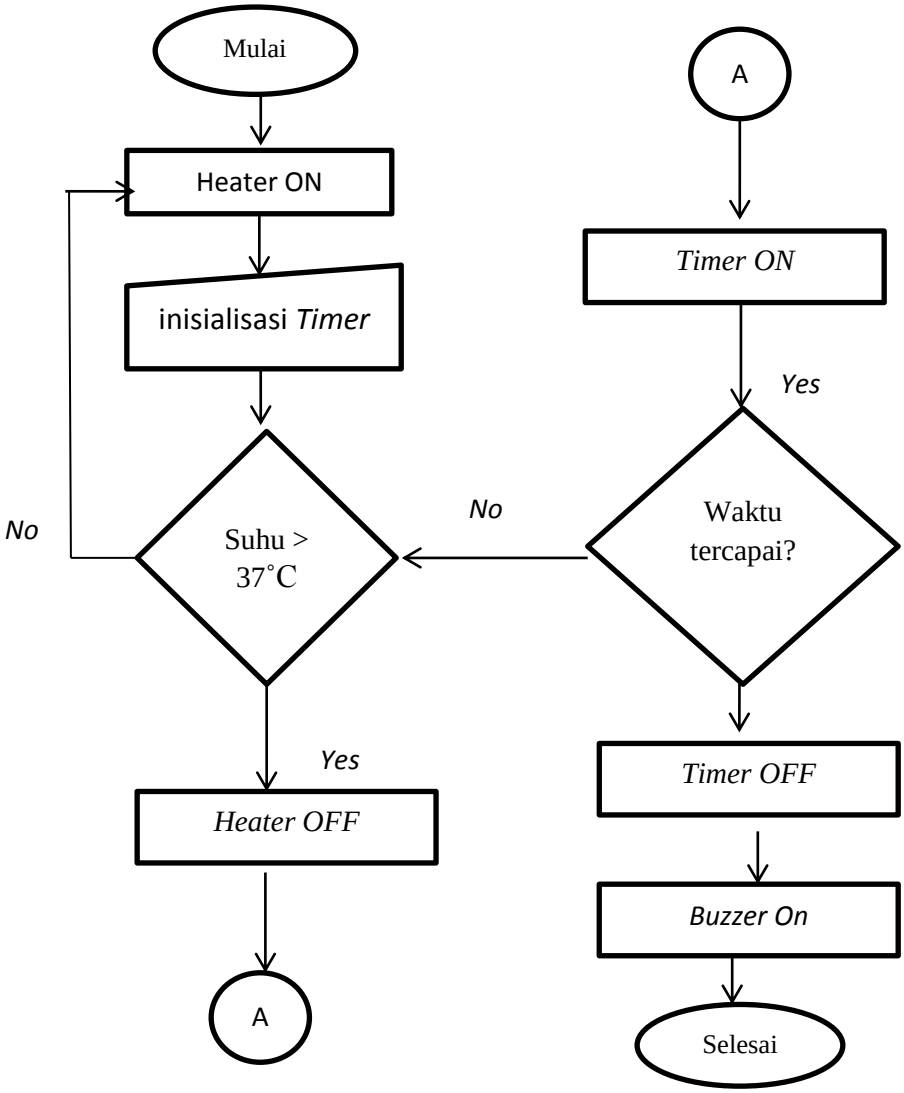
Berdasarkan Gambar 3.6 Spesifikasi rangkaian modul yang diperlukan adalah:

1. Tegangan kerja maksimum yang dibutuhkan adalah 5VDC dan *Ground* yang diambil dari *power supply* 5VDC.
2. IC *mikrokontorler* yang digunakan adalah ATMega8535dengan *fitur* ADC *internal* dan *timer internal.*
3. Menggunakan *push on* PORTB.1, PORTB.2, dan PORT B.3 untuk pemilihan sistem.
4. Menghubungkan PORT C dengan LCD Karakter2x16 sebagai tampilan.
5. Menggunakan PINA sebagai *input* ADC dari *sensor* suhu.

3.2 Perancangan Perangkat Lunak

3.2.1 Diagram Alir Proses/Program

Gambar 3.7 menunjukan Diagram Alir dari Inkubator Bakteri.



Gambar 3.7 Diagram Alir Inkubator Bakteri

Berdasarkan Gambar 3.7 saat modul dinyalakan, *mikrokontroler* yang mendapatkan tegangan 5VDC *power* supply akan melakukan inisialisasi LCD lalu program akan bekerja,

menyalakan *driver heater* dan menampilkan data. Kemudian *timer* diatur 12, 24 atau 48 jam, setelah suhu 37˚C *timer* mulai berjalan. Jika suhu lebih dari 37˚C maka *heater* mati. Jika waktunya belum tercapai maka *mikrokontroler* akan mengendalikan kembali sehingga suhu tetap 37˚C. Namun, jika waktu telah habis *timer OFF* dan *buzzer* akan menyala sebagai indikator bahwa waktu telah habis, dan proses berakhir.

3.2.2 *Listing Program*

Pembuatan program untuk modul ini menggunakan bahasa C dengan aplikasi CV. AVR. Program yang digunakan adalah ADC sebagai pengendali suhu, dan *timer* sebagai pengendali waktu. Berikut program yang digunakan:

1. Memanggil *library* yang akan digunakan

```
//Deklarasi Header
#include <mega8535.h>
#include <stdlib.h>
#include <delay.h>
```

Listing 3.1 Program memanggil *library*

2. Melakukan inisialisasi data

```
// Dekalarasi Variabel
unsigned char temp[6];
unsigned  char  detik=0,  a=1,temp2[6],  temp3[3],
temp4[4],menit=0,jam=0;
bit hitung_mundur=0;
float data,suhu;
bit timer_aktif=0,b=0;
```

Listing 3.2 Program *inisialisasi data*

3. Mengaktifkan ADC

```
#define ADC_VREF_TYPE 0x40
// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
//  Delay  needed  for  the  stabilization  of  the  ADC
input voltage
delay_us(10);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;
}
```

Listing 3.3 Program mengaktifkan ADC

Berdasarkan Listing 3.3 Proses inisialisasi ADC meliputi proses penentuan *clock*, tegangan referensi, format output data, dan mode pembacaan. Register yang perlu diset nilainya adalah ADMUX (*ADC multiplexer selection register*), ADCSRA (*ADC control and status register A*). ADMUX merupakan register 8 *bit* yang berfungsi untuk menentukan tegangan referensi ADC, format data *output*, dan saluran ADC yang digunakan (*Iswanto & Raharja, 2015)*.

4. Program fungsi untuk menyalakan *timer*

```
void mulai_timer()
 {
 if(suhu>=37){timer_aktif=1;}
 if(timer_aktif==1)
       {
       TCCR1B=0x04;
       }
else  if(timer_aktif==0)
       {
TCCR1B=0x00;
}
}
```

Listing 3.4 Program menyalakan *timer*

Berdasarkan *listing* 3.4 merupakan konfigurasi *timer* 1. Untuk *timer* menjadi *counter*, maka TCCR1B di isi dengan nilai 0x04. TCCR1B= 0x04 karena sumber *clock* eksternal pada pin T1 dan *clock* pada mencacah naik(*Iswanto & Raharja,2015).*

5. Program data ADC

```
data=read_adc(0);

   suhu=data*5/1024;

   suhu=suhu*100;

   ftoa(suhu,2,temp3);

   lcd_gotoxy(8,0);

   lcd_puts(temp3);
```

Listing 3.5 Program data ADC

6. Program fungsi untuk menampilkan waktu

```
void atur_tampilantimer()
{
      lcd_gotoxy(0,1);
      lcd_putsf("WAKTU");
      lcd_gotoxy(10,1); //menampilkan :
      lcd_putsf(":");
      lcd_gotoxy(13,1); //menampilkan :
      lcd_putsf(":");
```

Listing 3.6 Program menampilkan waktu

7. Program fungsi untuk menghentikan *timer.*

```
void stop_timer()
{
      if(jam==0&&menit==0&&detik==0)
            {
            timer_aktif=0;
            TCCR1B=0x00;
            }
}
```

Listing 3.7 Program menghentikan *timer*

8. Program mengaktifkan *driver heater*

```
void driver_set()
{
      if(timer_aktif==1)
      {
            if(suhu>=37)
            {
            PORTD.5=0;
            }
            Else
            {
            PORTD.5=1;
            }
      }
}
```

Listing 3.8 Program mengaktifkan *driver heater*

9. Program untuk memanggil setiap fungsi yang dieksekusi

```
         while (1)
{
      baca_suhu();
      driver_set();
      lcd_gotoxy(8,0);
      {
            b=1;
            timer_aktif=1;
      }
      setting_timer();
      mulai_timer();
      stop_timer();
      atur_tampilantimer();
  }
}
```

Listing 3.9 Program memanggil fungsi

10. Program untuk mengatur *timer*

```
void setting_timer() //pushon
{
      if(b==0)
            {
            if(PINB.0==0)
                  {
                  a++;delay_ms(500);lcd_clear();
                  }
```

Listing 3.10 Program mengatur *timer*

11. Program untuk mengatur *timer* (Lanjutan)

```
       if(a<1)
                     {
                     a=1;
                     }
       if(a>4)
                     {a=4;}
              else if(PINB.1==0)
                     {
                     a--;delay_ms(500);lcd_clear();
                     }
      if(a==1)
                     {
                     jam=6;
                     }
              else if(a==2)
                     {
                     jam=12;
                     }
              else if(a==3)
                     {
                     jam=18;
                     }
              else if(a==4)
                     {
                     jam=24;
                     }
              }
}
```

Listing 3.11 Program mengatur *timer*

3.3 Perancangan Pengujian

Pada perancangan pengujian ada beberapa parameter yang akan diujikan sehingga, mengetahui kondisi modul sesuai dengan diinginkan atau belum. Berikut merupakan parameter dari modul inkubator bakteri yang akan diujikan, diantaranya:

3.3.1 Jenis Pengujian

1. Pengukuran suhu menggunakan pembanding termometer

Sensor suhu LM35 berfungsi sebagai pengubah besaran fisis dari suhu menjadi besaran elektris tegangan. *Sensor* ini memiliki linieritas kenaikan tegangan sebesar 10mV setiap kenaikan suhu 1˚C. Pengukuran suhu bertujuan untuk mengetahui seberapa besar *error* dan *standard deviasi* yang didapat dari setiap perubahan suhu yang terjadi.

Pengukuran suhu dilakukan dengan cara membandingkan suhu tampilan LCD dengan suhu pada termometer dan dilakukan sebanyak 30 kali pengujian.

2. Pengukuran Tegangan *sensor* LM35 dengan pembanding perhitungan secara teori

   Pengukuran tegangan *sensor* LM35 bertujuan untuk mengetahui seberapa besar nilai *error* yang diperoleh jika dibandingkan dengan perhitungan teori.

   Pengujian tegangan keluaran *sensor* LM35 dilakukan dengan cara menghubungkan *output sensor* dengan positif AVO meter dan *ground sensor* dengan *ground* AVO meter. Nilai tegangan yang keluar pada pin2 LM35 menggunakan AVO meter akan dibandingkan dengan perhitungan secara teori. Karena setiap kenaikan suhu 1˚C sama dengan 10mV maka, jika nilai suhu di AVO meter tegangan yang keluar harus 370mV atau 0,37V.

3. Pengukuran waktu menggunakan pembanding pewaktu telepon genggam.

Timer merupakan fasilitas dari ATMega 8535 yang digunakan untuk perhitungan waktu. Pengujian *timer* yang dilakukan bertujuan untuk memastikan bahwa *timer* berfungsi dengan baik. Pada modul inkubator bakteri ini *timer* digunakan untuk mengatur lamanya proses inkubasi bakteri. Dalam pengujian ini *timer* yang diuji akan dibandingkan dengan *timer handphone.* Ada 2 pengujian dalam pengukuran waktu yaitu setiap 5 menit sekali dan 1 jam sekali yang masing-masing dilakukan sebanyak 30 kali pengujian.

3.3.2 Pengolahan Data Dalam Pengujian

1. Variabel Penelitian

Variabel yang diteliti dan diamati pada alatinkubator bakteri ini adalah menggunakan *heater* sebgai pemanas dan *sensor* LM35 sebagai pemantau suhu ruangan inkubator.

1) Variabel Bebas

Sebagai Variabel Bebas yaitu suhu dan waktu proses inkubasi.

2) Variabel Tergantung

Sebagai Variabel Tergantung yaitu sensor LM 35.

3) Variabel Terkendali

Variabel terkendali terdiri yaitu IC Mikrokontroler ATMega8535.

2. Sistematika Pengukuran

Setelah melakukan pengukuran modul inkubator bakteri membutuhkan rumus statistik untuk membuktikan bahwa modul yang

telah dibuat layak digunakan. Berikut rumus-rumus yang digunakan. Diantaranya:

1. Rata – rata

Rata–rata adalah bilangan yang di dapat dari hasil pembagian jumlah nilai data oleh banyaknya data dalam kumpulan tersebut.

Rumus rata – rata adalah **:**

$$\bar{X} \quad = \frac{\sum Xn}{n} \tag{3.2}$$

dengan :

$\bar{x}$ = rata-rata

$x_1 \cdots x_n$ = nilai data

n = banyaknya data(1,2,3 ...,n)

2. Simpangan *(Error)*

Adalah selisih dari rata-rata nilai dari harga yang dikehendaki dengan nilai yang diukur. Simpangn *(error)* dirumuskan sebagai berikut :

$$simpangan = x_n - \bar{x} \tag{3.3}$$

dengan :

$simpangan$ = Nilai error yang dihasilkan

x_n = Rata – rata data DPM

$\overline{x}$ = Rata – rata data modul

3. *Persentase Error*

Adalah nilai persen dari simpangan (*error)* terhadap nilai yang dikehendaki dirumuskan sebagai berikut:

$$Persentase\ error = \frac{Simpangan}{\bar{Y}} \text{ x } 100\% \quad (3.4)$$

dengan :

$\bar{Y}$ = rata-rata data kalibrator

$Persentase\ error$ = besarnya nilai simpangan atau *error* dalam %

4. *Standart Deviasi*

Standart Deviasi adalah suatu nilai yang menunjukan tingkat (derajat) variasi kelompok data atau ukuran standart penyimpangan dari *mean.*

Rumus *Standart Deviasi* adalah :

$$SD = \sqrt{\frac{(X_1 - \bar{X})^2 + (X_2 - \bar{X})^2 + \cdots + (X_5 - \bar{X})^2}{(n-1)}} \quad (3.5)$$

dengan:

SD = *standart deviasi*

$\bar{X}$ = rata-rata

$x_1 \cdots x_n$ =nilai data

n =banyak data (1,2,3 ….n)