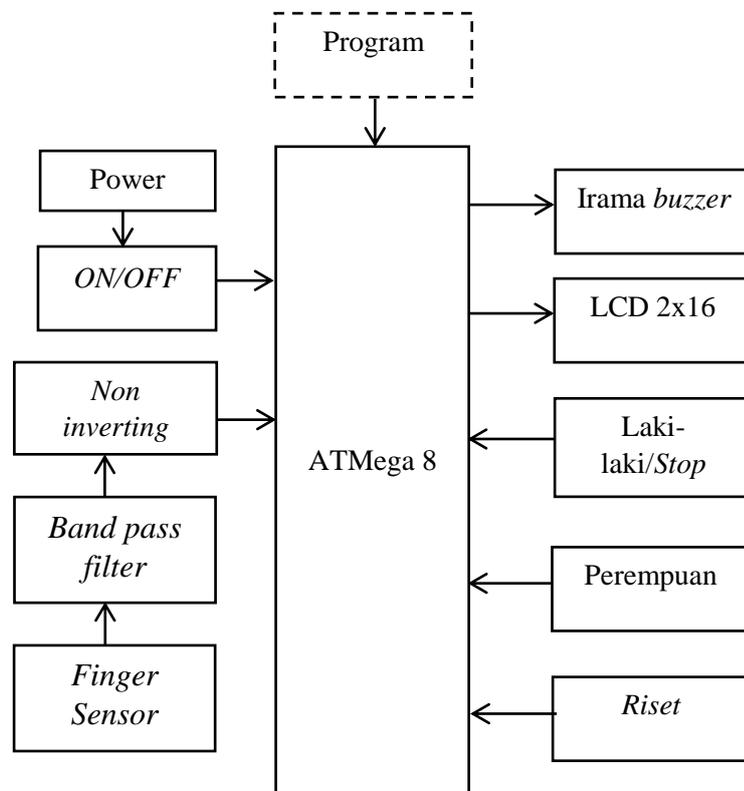


BAB III

METODOLOGI PENELITIAN

3.1 Blok Diagram

Pembuatan sistem dapat dijelaskan dengan lebih baik melalui blok diagram seperti yang terlihat pada Gambar 3.1.



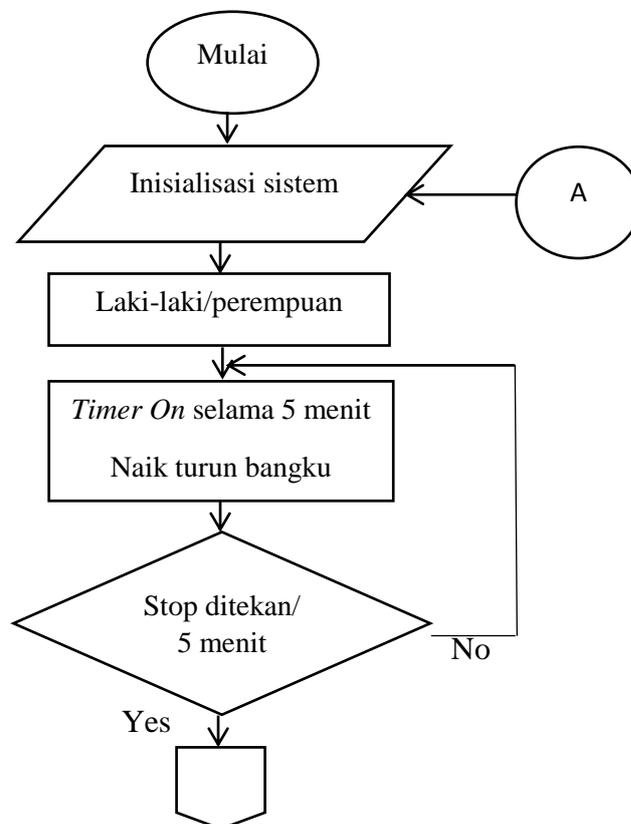
Gambar 3. 1 Blok diagram

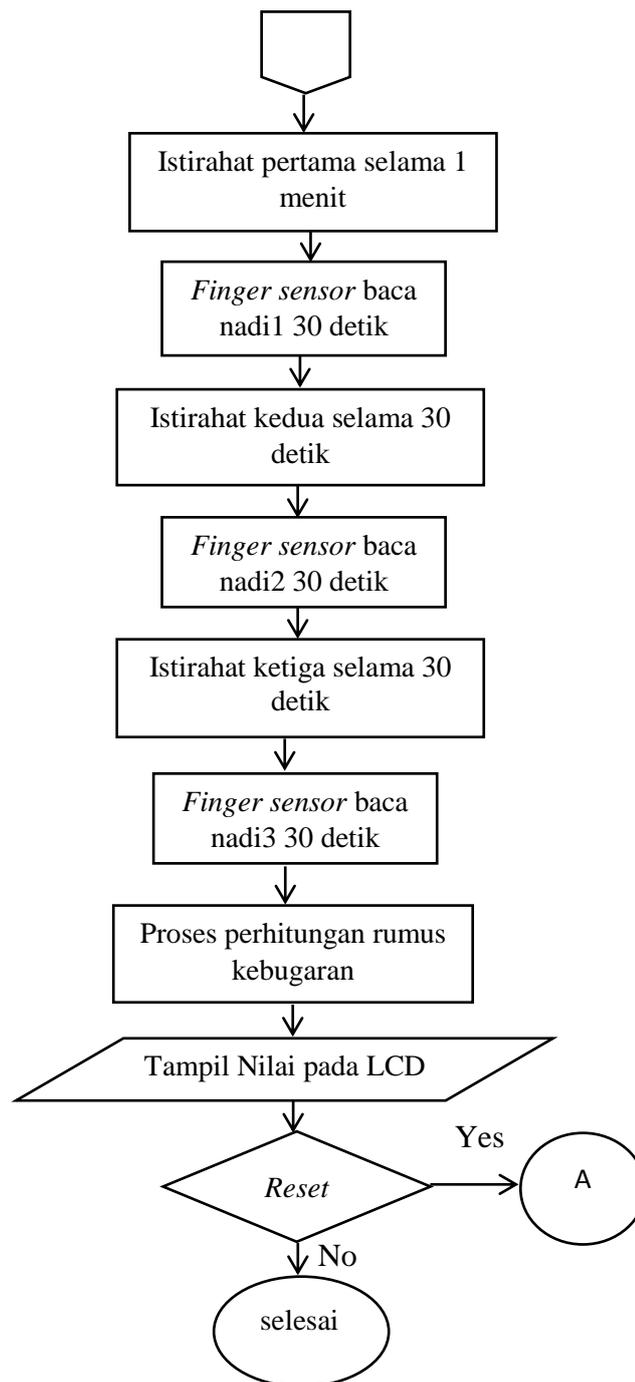
Tombol *On/Off* ditekan akan mengalirkan arus untuk menyuplai tegangan ke seluruh rangkaian. *Finger Sensor* berfungsi untuk mendeteksi jumlah detak jantung pada manusia yang diletakkan pada ujung jari telunjuk. Karena sinyal cahaya inframerah yang dipantulkan ke fotodiode masih kecil/*low* maka dikuatkan dengan rangkaian penguat OP-AMP *non inverting*. Rangkaian *band pass filter* berfungsi membatasi sinyal frekuensi di bawah frekuensi denyut nadi yaitu pada

frekuensi 1 – 3 Hz. Di dalam mikrokontroler akan mengolah hasil pembacaan data dari *pulse sensor* dengan 3 kali pembacaan nadi. Hasil akan ditampilkan pada LCD 2x16 dengan kategori baik sekali, baik, cukup dan buruk. Irama *buzzer* sebagai pengganti *metronome* berfungsi membantu menemukan ketukan dan tempo yang sama serta seimbang pada saat naik turun bangku.

3.2 Diagram Alir

Diagram alir merupakan gambar yang mewakili langkah-langkah sebuah proses terurut secara terpisah agar proses tersebut menjadi lebih sederhana sehingga mudah dipahamis. Diagram ini bisa memberi solusi selangkah demi selangkah untuk penyelesaian masalah yang ada di dalam proses. Adapun gambar dari diagram alir proses ditunjukkan oleh Gambar 3.2.





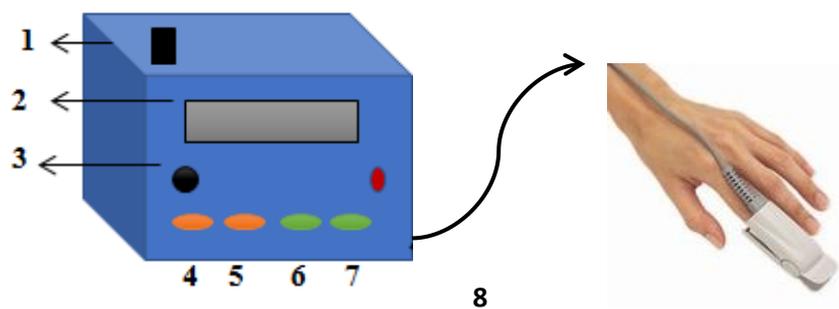
Gambar 3. 2 Diagram alir

Saat pertama alat dihidupkan atau dioperasikan, alat akan inialisasi sistem. Ketika tombol *start* ditekan maka *timer on* selama 5 menit waktu untuk melakukan naik turun bangku. Jika subjek mulai kelelahan maka tombol *stop*

dapat ditekan atau jika *timer* mencapai 5 menit maka subjek melakukan istirahat pertama dengan *timer* selama 1 menit. Setelah istirahat pertama maka *finger sensor* mulai menghitung nadi pertama selama 30detik. Selanjutnya *timer on* selama 30 detik untuk melakukan istirahat lagi yaitu istirahat kedua. Setelah istirahat kedua maka *finger sensor* mulai menghitung nadi kedua selama 30detik. *Timer on* lagi selama 30 detik untuk melakukan istirahat ketiga. Setelah istirahat ketiga maka *finger sensor* mulai menghitung nadi ketiga selama 30 detik. Terakhir, program akan melakukan proses perhitungan rumus indeks kebugaran yang hasil interpretasi akan ditampilkan pada *LCD*. Tombol *Reset* untuk mengulang program ke kondisi awal dan selesai untuk mengakhiri program.

3.3 Diagram Mekanis Sistem

Diagram mekanis alat digital harvard step test dapat dilihat pada Gambar 3.3.



Gambar 3. 3 Desain Modul *Digital Harvard Step Test*

Keterangan :

1. Tombol *ON/OFF*
2. LCD 2x16

3. Lubang Irama *Buzzer*
4. Tombol laki-laki/*STOP*
5. Tombol perempuan
6. Tombol menu
7. Tombol reset
8. Kabel *finger* sensor

3.4 Alat dan Bahan

3.4.1 Alat

Pada penelitian ini digunakan beberapa alat, dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Alat yang digunakan

No	Nama Alat	Jumlah
1	Solder	1
2	<i>Software Proteus</i>	1
3	<i>Tool Set</i>	1
4	<i>Mini Dril</i>	1
5	Spidor Permanen	1
6	Setrika	1
7	Laptop	1

3.4.2 Bahan

Pada penelitian ini digunakan beberapa alat dapat dilihat pada tabel 3.2.

Tabel 3. 2 Bahan yang digunakan

No	Nama Bahan	Jumlah
1	Regulator 7812	1
2	Regulator 7805	3
3	ATMega 8	1
4	<i>Crystal</i> 12MHz	1
5	<i>Buzzer</i>	1
6	LCD 2x16	1
7	Pin Deret	Secukupnya
8	Kabel <i>Male Female</i>	Secukupnya

Tabel 3.2 Bahan yang digunakan

(Lanjutan)

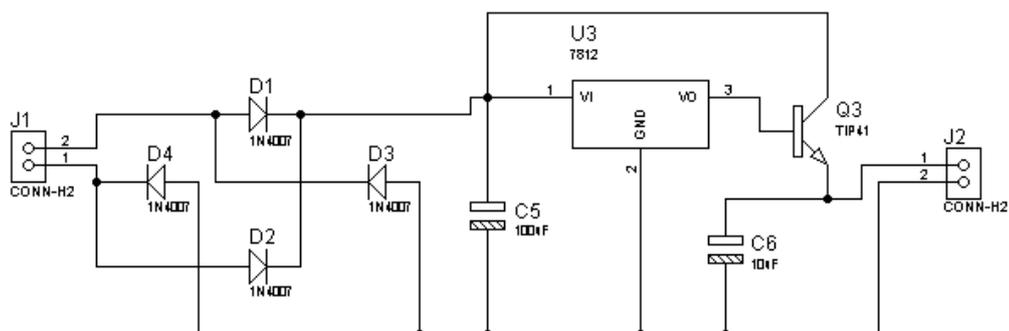
9	IC LM 358N	2
10	<i>Infrared</i>	1
11	Fotodiode	1
12	Dioda 1N4007	4
13	Trafo 1 A	1
14	Resistor 68K Ω	4
15	Resistor 680K Ω	2
16	Resistor 470 Ω	1
17	Resistor 10K Ω	1
18	Kapasitor Polar 1 μ F	2
18	Kapasitor Polar 22 μ F	2
19	Kapasitor Polar 100 μ F	1
20	Kapasitor non polar 1nF	2
21	Kapasitor non polar 100nF	2

3.5 Perancangan Perangkat Keras

3.5.1 Rangkaian *Power Supply*

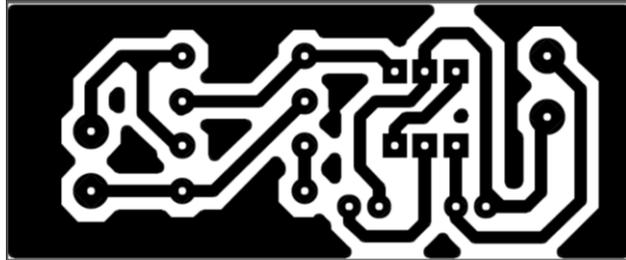
1. Langkah Perakitan

- a. Membuat skematik rangkaian dengan menggunakan aplikasi *proteus* pada laptop. Untuk gambar skematik rangkaian pada aplikasi dapat dilihat pada Gambar 3.4.



Gambar 3. 4 Skematik Minimum Sistem

- b. Membuat *layout power supply* dan disablon ke papan pcb. Untuk gambar *layout* pada papan pcb dapat dilihat pada Gambar 3.5.

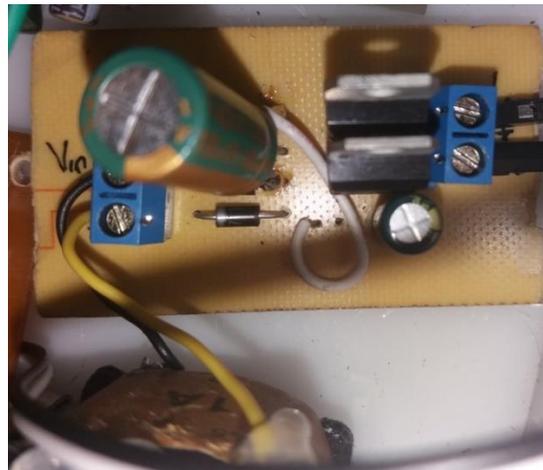


Gambar 3. 5 *Layout power supply*

- c. Rakit komponen yang dibutuhkan dengan menggunakan solder

2. Gambar rangkaian

Untuk gambar rangkaian *power supply* dapat dilihat pada Gambar 3.6.



Gambar 3. 6 Rangkaian *Power Supply*

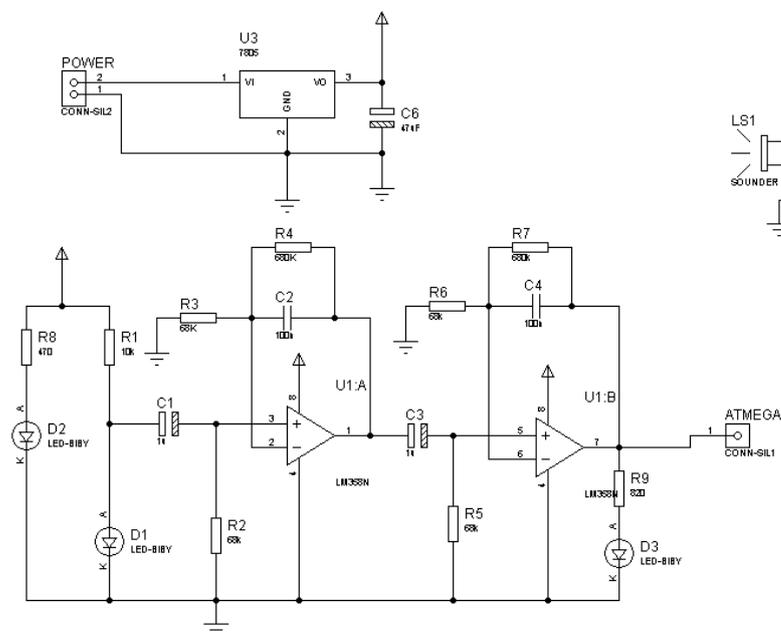
Rangkaian *power supply* pada alat ini berfungsi untuk mensupply tegangan ke seluruh rangkaian dengan menggunakan tegangan DC. Prinsip kerja power supply ini adalah mengubah tegangan 220 volt AC menjadi 15 volt AC dengan menggunakan transformator *step down* kemudian tegangan 15 volt AC disearahkan dengan menggunakan dioda menjadi tegangan 15 volt

DC. Pada alat ini menggunakan 12 volt DC sehingga menggunakan regulator 7812 untuk menurunkan tegangan. Sinyal DC yang dihasilkan oleh rangkaian dioda penyearah masih berbentuk *ripple* yang sangat besar, untuk mendapatkan sinyal tegangan DC rata (*low ripple*) maka dipasang kapasitor $1\mu\text{F}$ sebagai *filter* sehingga ripple tegangan yang dihasilkan akan sangat kecil sekali. Transistor TIP41 berfungsi sebagai penyangga beban pada rangkaian *power supply*.

3.5.2 Rangkaian Sensor Nadi, Rangkaian Penguat dan rangkaian *Band Pass Filter*

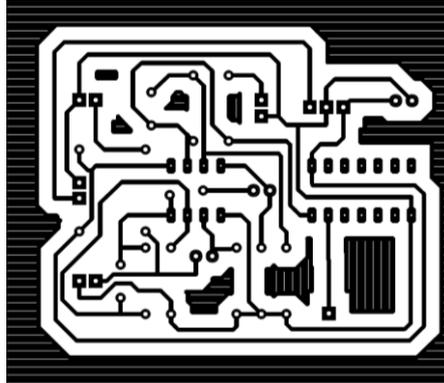
1. Langkah Perakitan

- a. Membuat skematik rangkaian dengan menggunakan aplikasi *proteus* pada laptop. Untuk gambar skematik rangkaian pada aplikasi dapat dilihat pada Gambar 3.7.



Gambar 3. 7 Skematik Sensor, penguat dan *band pass filter*

- b. Membuat *layout* dan disablon ke papan pcb. Untuk gambar *layout* pada papan pcb dapat dilihat pada Gambar 3.8.



Gambar 3. 8 *Layout* sensor nadi, penguat dan *band pass filter*

- c. Rakit komponen yang dibutuhkan dengan menggunakan solder.

2. Gambar rangkaian

Untuk gambar rangkaian sensor nadi, penguat dan *band pass filter* dapat dilihat pada Gambar 3.9.



Gambar 3. 9 Sensor nadi, penguat dan *band pass filter*

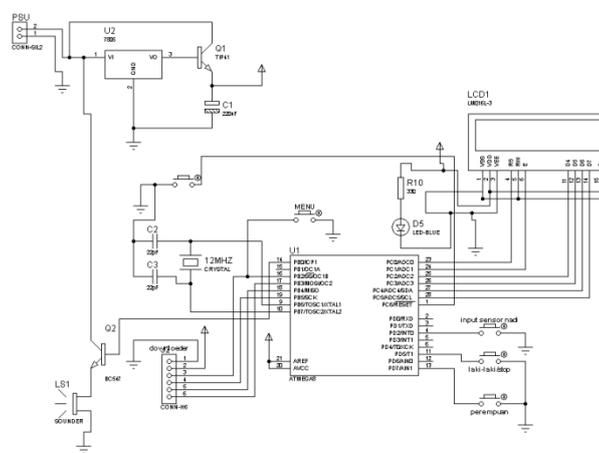
Sensor denyut nadi menggunakan *finger sensor* yang terdiri dari fhotodioda dan *infrared*. *Infrared* akan memancarkan cahaya infra merah yang kemudian diterima oleh fhotodioda. Sinyal yang diterima oleh

photodiode masih berupa resistansi maka dipasang resistor $10k\Omega$ berfungsi sebagai pembagi tegangan sehingga keluaran menghasilkan tegangan berupa *milivolt* yang akan dikuatkan oleh rangkaian penguat *non inverting* dengan IC 358. Jantung manusia berdetak dengan frekuensi 80 sampai 120 kali/menit. Jika diubah dalam satuan Hertz, detak jantung manusia dengan frekuensi sekitar 1 sampai 3 Hertz, untuk menghindari adanya gangguan frekuensi maka dibuat rangkaian *band pass filter* untuk membatasi *range* diatas dan dibawah frekuensi *cutoff*.

3.5.3 Perakitan Rangkaian Minimum Sistem ATmega8

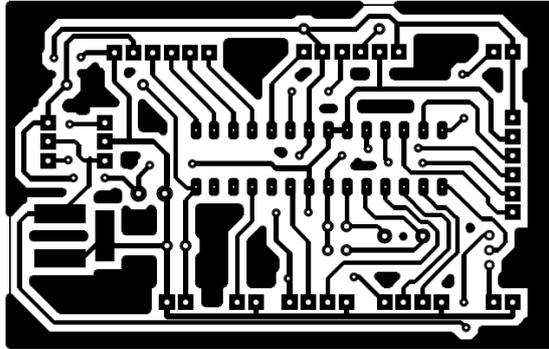
1. Langkah Perakitan

- a. Membuat skematik rangkaian minimum sistem dengan menggunakan aplikasi *proteus* pada laptop. Rangkaian minimum sistem ATmega8 adalah rangkaian yang berfungsi sebagai pengontrol, penerima, pengolah dan menghasilkan output guna mendukung bagian yang lain. Untuk gambar skematik rangkaian minimum sistem pada aplikasi dapat dilihat pada Gambar 3.10.



Gambar 3. 10 Skematik Minimum Sistem

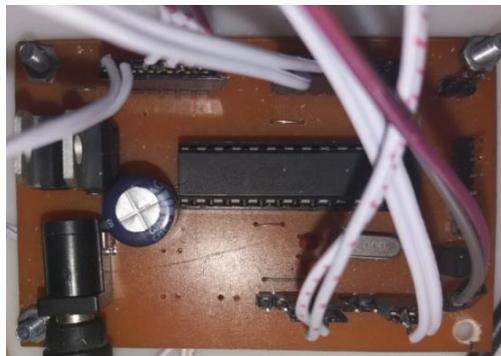
- b. Membuat *layout* dan disablon ke papan pcb. Untuk gambar *layout* minimum sistem pada papan pcb dapat dilihat pada Gambar 3.11.



Gambar 3. 11 *Layout* minimum sistem

- c. Rakit komponen yang dibutuhkan dengan menggunakan solder.
2. Gambar rangkaian minimum sistem

Untuk gambar rangkaian minimum sistem dapat dilihat pada Gambar 3.12.



Gambar 3. 12 Minimum Sistem ATmega8

Minimum sistem adalah sebuah rangkaian yang harus ada agar suatu *microcontroller* dapat bekerja. Rangkaian minimum sistem ini menggunakan IC ATmega8 dan akan dimasukkan program sebagai pengontrol sistem kerja alat secara keseluruhan. Minimum sistem membutuhkan tegangan sebesar 4,5-5,5 volt. Karena *ouput power supply* 12 volt maka minimum sistem dipasang regulator sebagai penurun tegangan

menjadi 5 volt dan transistor TIP41 berfungsi sebagai penyangga beban pada rangkaian minimum sistem. Pada ATmega8 port D.2 berfungsi sebagai input sensor, port D.5 sebagai input tombol *start/stop* laki-laki dan port D.7 sebagai input tombol *start/stop* perempuan. *Buzzer* berfungsi sebagai irama *buzzer* 120kali/permenit, dipasang di port B.0 dan transistor berfungsi sebagai *switch* pada *buzzer*.

3.6 Pembahasan *Software*

Software dalam alat ini dibuat dengan menggunakan bahasa C, dengan memanfaatkan compiler *CodeVision AVR 2.05.0 Evaluation*.

```
#include <mega8.h>
#include <stdio.h>
#include <delay.h>
#include <stdlib.h>
#include <alcd.h>
```

Listing 3. 1 Kode *File Header*

Kode file header berfungsi untuk menyertakan library dari program yang akan ditulis. `#include<mega8.h>` adalah pengarah untuk memberitahukan kepada program untuk menyertakan library mega8. `#include <stdio.h>` untuk pemrograman tampilan . `#include<delay.h>` adalah pengarah untuk memberitahukan kepada program untuk menyertakan library delay. `#include<alcd.h>` adalah pengarah untuk memberitahukan kepada program untuk menyertakan library lcd (memiliki bahasa sendiri).

```
lcd_init(16);
lcd_gotoxy(0,0);
lcd_putsf("Digital Harvard");
lcd_gotoxy(0,1);
lcd_putsf(" Step Test");
delay_ms(2000);
```

Lanjut

Lanjut

```

lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("1. Laki - Laki");
lcd_gotoxy(0,1);
lcd_putsf("2. Perempuan");

```

Listing 3. 2 Program tampilan awal

Program ini difungsikan sebagai tampilan awal ketika alat dihidupkan dengan delay 2s akan tereksekusi kemudian menampilkan pilihan laki-laki dan perempuan untuk pemilihan tombol.

```

interrupt [EXT_INT0] void ext_int0_isr(void)
{
// Place your code here
s=s+1;
}

```

Listing 3. 3 Program Pembacaan detak jantung

Interrupt Eksternal adalah salah satu fitur yang terdapat pada ATmega8. Ketika mode *Interrupt Eksternal* difungsikan maka ketika ada perubahan 1 logika pada pin tersebut maka program pada *interrupt* akan tereksekusi satu kali. $S = S + 1$ adalah program yang dibuat untuk menambah 1 pada variable S untuk 1 kali pulsa *high* yang masuk pada pin *External Interrupt 0*.

```

interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
TCNT1H=0xf8;
TCNT1L=0xad;
    { if (++milisecond >= 100)
    {
milisecond=0;
count=count+1;
if (++second >= 60) {
second=0;

```

Lanjut

Lanjut

```

if (++minutes >= 60)
    {
        minutes=0;
        if (++hour >= 24)
            {hour = 0;}
    } } } }

```

Listing 3. 4 Program *Stopwatch*

Program *timer/counter* ini difungsikan sebagai *stopwatch*, setiap *milisecond* tercapai 100kali maka tereksekusi sehingga menambah 1 untuk *second*, dan setiap 60 kali *second* menambah 1 untuk menit.

```

void hitung_mundur()
{
    lcd_clear();
    lcd_gotoxy(4,0);
    lcd_putsf("Bersiap..");
    lcd_gotoxy(7,1);
    lcd_putsf(" 5");
    delay_ms(1000);
    lcd_gotoxy(7,1);
    lcd_putsf(" 4");
    delay_ms(1000);
    lcd_gotoxy(7,1);
    lcd_putsf(" 3");
    delay_ms(1000);
    lcd_gotoxy(7,1);
    lcd_putsf(" 2");
    delay_ms(1000);
    lcd_gotoxy(7,1);
    lcd_putsf(" 1");
    delay_ms(1000);
    lcd_gotoxy(4,1);
    lcd_putsf("Mulai...");
    delay_ms(300);
        second=0;
        milisecond=0;
        minutes=0;
        hour=0;
        count=0;
    lcd_clear();
    indeks=1;
}

```

Listing 3. 5 Program hitung mundur

Program hitung mundur difungsikan untuk menampilkan waktu mundur tanda akan dimulainya tes. Ketika tombol *start* ditekan maka program akan tereksekusi satu kali dan akan lanjut ke *step* selanjutnya yaitu indeks1.

```

void buzzer_bunyi()
{  if (milisecond>=0 && milisecond<=30)
    {
        PORTB.0=1;
    } else if (milisecond>=0 && milisecond<=2)
    {
        lcd_clear();
    }
else if (milisecond>=31 && milisecond<=50)
    {
        PORTB.0=0;
    }
else if (milisecond>=51 && milisecond<=80)
    {
        PORTB.0=1
    }
else if (milisecond>=81 && milisecond<=100)
    {
        PORTB.0=0;
    } }

```

Listing 3. 6 Program *Buzzer*

Program ini difungsikan sebagai output suara buzzer, ketika PORTB.0 berlogika 1 maka buzzer berbunyi dan ketika PORTB.0 berlogika 0 maka buzzer mati.

```

void kategorimale()
{
if (ahir <91&&male==1)
    {
        lcd_gotoxy(0,1);
        lcd_putsf("      KURANG");
    }

else if (ahir >=91 && ahir <=102&&male==1)
    {
        lcd_gotoxy(0,1);
        lcd_putsf("      CUKUP");
    }
else if (ahir >=103 && ahir <=115&&male==1)
    {
        lcd_gotoxy(0,1);

```

Lanjut

Lanjut

```

    lcd_putsf("    BAIK");
  }
  else if (ahir > 115&&male==1)
  {
    lcd_gotoxy(0,1);
    lcd_putsf("    SANGAT BAIK");
  }
}
void kategorifemale()
{
  if (ahir <77&&female==1)
  {
    lcd_gotoxy(0,1);
    lcd_putsf("    KURANG");
  }
  else if (ahir >=77 && ahir <=83&&female==1)
  {
    lcd_gotoxy(0,1);
    lcd_putsf("    CUKUP");
  }
  else if (ahir >=84 && ahir <=91&&female==1)
  {
    lcd_gotoxy(0,1);
    lcd_putsf("    BAIK");
  }
  else if (ahir > 91&&female==1)
  {
    lcd_gotoxy(0,1);
    lcd_putsf("    SANGAT BAIK");
  }
}
}

```

Listing 3. 7 Program kategori indeks kebugaran laki-laki dan perempuan

Program ini difungsikan untuk menampilkan hasil hasil perhitungan rumus kategori kebugaran yang hasilnya akan ditampilkan pada LCD 2x16.

```

  if (PIND.7==0&&indeks==0)//tombol start ditekan
  {
    hitung_mundur();
    male=1;
  }
  if (PIND.5==0&&indeks==0)//tombol start ditekan
  {
    hitung_mundur();
    female=1;
  }

```

Listing 3. 8 Program pilihan tombol laki-laki dan perempuan

Program ini difungsikan untuk pemilihan tombol laki-laki dan perempuan, ketika tombol start ditekan pada PIND.7 maka tombol pada laki-laki aktif dan ketika tombol start ditekan pada PIND.5 maka tombol pada perempuan aktif.

```

if (indeks==1)//saat timer berjalan
{
    buzzer_bunyi();
    lcd();
    if (minutes==5)
    {
        PORTB.0=0;
        minutes=0;
        second=0;
        milisecond=0;
        a = count;
        delay_ms(200);
        lcd_gotoxy(0,0);
        sprintf(buf, "waktu = %d detik", a);
        lcd_puts(buf);
        delay_ms(1000); count=0;
        indeks=2;
        lcd_clear();
    }
    if (PIND.7==0&&indeks==1) //jika pada saat tes
berlangsung tombol stop ditekan
    {
        PORTB.0=0;
        minutes=0;
        second=0;
        milisecond=0;
        a=count;
        delay_ms(200);
        lcd_gotoxy(0,1);
        sprintf(buf, "waktu = %d detik", a);
        lcd_puts(buf);
        delay_ms(1000);
        count=0;
        indeks=2; //indeks diisi 2 untuk lanjut ke
program istirahat 1 menit
        delay_ms(500);
        lcd_clear();
    }
}

```

Listing 3. 9 Program *timer*

Program ini difungsikan saat naik turun bangku berlangsung, LCD 2x16 akan menampilkan timer selama 5 menit, ketika *timer* mengcounter selama 5 menit atau tombol *stop* ditekan pada tombol PIND.7 maka waktu akan terhenti dan akan disimpan pada variable “a”. Indeks diisi 2 untuk lanjut ke program selanjutnya.

```

if (indeks==2 && second>=1 && second <=29) //istirahat 1 menit
{ lcd_gotoxy(0,0);
  lcd_putsf("  Istirahat");
  if (second>=30 && second<=40)
  { lcd_gotoxy(0,0);
    lcd_putsf(" Letakkan Jari");
    lcd_gotoxy(0,1);
    lcd_putsf(" Jangan Dilepas"); }
  if (minutes==1)
  { minutes=0;
    second=0;
    indeks=3;//indeks diisi 3 untuk lanjut ke program
pembacaan nadi
    s=0;
    lcd_clear();
  }
}

```

Listing 4. 1 Program istirahat pertama

Program ini difungsikan saat istirahat setelah melakukan aktivitas naik turun bangku selesai. Selama *timer* mengcounter selama 1 menit akan LCD 2x16 akan menampilkan kalimat “Istirahat” dan “Letakkan jari jangan dilepas”. Indeks 3 diisi untuk lanjut ke program selanjutnya yaitu pembacaan nadi.

```

if (indeks==3) //pembacaan nadi selama 30 detik
{ lcd_gotoxy(0,0);
  lcd_putsf(" Membaca Nadi 1");
  lcd_gotoxy(0,1);
  sprintf(bufs," denyut = %d%",s);
  lcd_puts(bufs);
  if (second==30)
  {
    minutes=0;
    second=0;
    indeks=4; //indeks diisi 4 untuk lanjut ke program
istirahat
    detak1=s; //data nadi 1
    lcd_clear();
  }
}

```

Listing 3. 10 Program Pembacaan nadi pertama

Program ini difungsikan saat pembacaan nadi pertama selama 30 detik, dan hasil pembacaan nadi akan disimpan pada variable “detak1” dan indeks diisi 4 untuk lanjut ke program istirahat kedua.

```

if (indeks==4) // istirahat ke 2 30 detik
{
    lcd();
    lcd_gotoxy(0,1);
    lcd_putsf(" Letakkan Jari");
    if (second==30)
    {
        minutes=0;
        second=0;
        indeks=5; //indeks diisi 5 untuk lanjut ke baca
nadi ke 2
        s=0;
        lcd_clear();
    }
}

```

Listing 3. 11 Program Istirahat kedua

Program ini difungsikan saat istirahat kedua selama 30detik, selama timer mengcounter selama 30 detik LCD 2x16 akan menampilkan kalimat “Tetap Letakkan jari”. Indeks 5 diisi untuk lanjut ke program selanjutnya yaitu pembacaan nadi kedua.

```

if (indeks==5) //pembacaan nadi ke 2
{
    lcd_gotoxy(0,0);
    lcd_putsf(" Membaca Nadi 2");
    lcd_gotoxy(0,1);
    sprintf(bufs," denyut = %d",s);
    lcd_puts(bufs);
    if (second==30)
    {
        minutes=0;
        second=0;
        indeks=6; //indeks diisi 6 untuk lanjut ke
istirahat ke 3
        detak2=s; //data nadi ke 2
        lcd_clear();
    }
}

```

Listing 3. 12 Program pembacaan nadi ke dua

Program ini difungsikan saat pembacaan nadi kedua selama 30 detik, dan hasil pembacaan nadi akan disimpan pada variable “detak2” dan indeks diisi 6 untuk lanjut ke program istirahat ketiga.

```

if (indeks==6) // istirahat ke 3 30 detik
{
    lcd();
    lcd_gotoxy(0,1);
    lcd_putsf(" Letakkan Jari");
    if (second==30)
    {
        minutes=0;
        second=0;
        indeks=7; //indeks diisi 7 untuk lanjut ke baca
nadi ke 3
        s=0; //ambil
        lcd_clear();
    }
}

```

Listing 3. 13 Program istirahat ketiga

Program ini difungsikan saat istirahat ketiga selama 30detik, selama *timer mengcounter* selama 30 detik LCD 2x16 akan menampilkan kalimat “Tetap Letakkan jari”. Indeks 7 diisi untuk lanjut ke program selanjutnya yaitu pembacaan nadi ketiga.

```

if (indeks==7)//baca nadi ke 3
{
    lcd_gotoxy(0,0);
    lcd_putsf(" Membaca Nadi 3");
    lcd_gotoxy(0,1);
    sprintf(bufs," denyut = %d",s);
    lcd_puts(bufs);
    if (second==30)
    {
        minutes=0;
        second=0;
        indeks=8; // indeks diisi 8 untuk lanjut ke
program perhitungan
        TIMSK=0x00; //timer di matikan
        lcd_clear();
        detak3=s; //data nadi ke 3
    }
}

```

Listing 3. 14 Program pembacaan nadi ketiga

Program ini difungsikan saat pembacaan nadi ketiga selama 30 detik, dan hasil pembacaan nadi akan disimpan pada variable “detak3” dan indeks diisi 8 untuk lanjut ke program perhitungan rumus indeks kebugaran.

```

if (indeks==8) //perhitungan
    {
        calculate1=2*(detak1+detak2+detak3);
        calculate2=(float)100/calculate1;
        ahir=calculate2*a;
        lcd_gotoxy(1,0);
        sprintf(buf, " nilai = %2.1f  ", ahir);
        lcd_puts(buf);
        kategorimale();
        kategorifemale();
    }

```

Listing 3. 15 Program rumus kalkulasi indeks kebugaran

Program terakhir yaitu program rumus kalkulasi indeks kebugaran, hasil nilai dan kategori kebugaran ditampilkan pada LCD 2x16.

3.7 Langkah-langkah Pengujian Alat

Setelah pembuatan alat selesai, maka yang akan dilakukan adalah melakukan pendataan melalui beberapa tahap proses pengukuran dan pengujian. Tujuan pengukuran dan pengujian adalah untuk memastikan keakuratan alat dan memastikan seluruh rangkaian telah berkerja dan berfungsi sesuai dengan yang telah direncanakan.

Langkah-langkah pengukuran dan pengujian alat sebagai berikut.

1. Menyiapkan alat yang dibutuhkan terutama alat ukur dan alat pembanding.
2. Menyiapkan tabel untuk hasil pengukuran.
3. Melakukan pengujian sinyal *output* pada rangkaian sensor, rangkaian *high pass filter*, *low pass filter* dan rangkaian *non inverting* menggunakan *digital oscilloscope*.
4. Melakukan pengujian *timer* dengan cara memantau *timer* alat yang ditampilkan pada *display* dan membandingkannya dengan *stopwatch*.
5. Melakukan pengujian irama *buzzer* dengan cara memantau suara *buzzer* dan membandingkannya dengan suara *metronome*.

6. Melakukan pengujian kinerja alat dengan pengambilan data sebanyak 10 testi.
7. Mencatat hasil pengukuran dan perhitungan dalam tabel yang telah disediakan.
8. Melakukan perhitungan terhadap hasil pengukuran untuk mengetahui tingkat *error* dan rata-rata.

3.8 Teknik Analisis Data

Setelah dilakukan pengukuran maka akan dilakukan perhitungan data yang diperoleh sehingga dapat dianalisa menggunakan rumus, antara lain.

3.5.1 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 :

$$\bar{X} = \frac{\sum Xn}{n} \quad (3.1)$$

Dimana : \bar{X} = rata – rata

$X_1, \dots, X_n (\sum Xn)$ = jumlah nilai data

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

3.5.2 Error (Rata – rata Simpangan)

Error (Rata – rata Simpangan) adalah selisih antara mean terhadap masing-masing data. Rumus Error adalah :

$$Error \% = \frac{\bar{X} - \bar{Y}}{\bar{Y}} \times 100 \% \quad (3.2)$$

Dimana : \bar{Y} = Rata – rata data pembanding atau kalibrator.

\bar{X} = Rata – rata data modul.