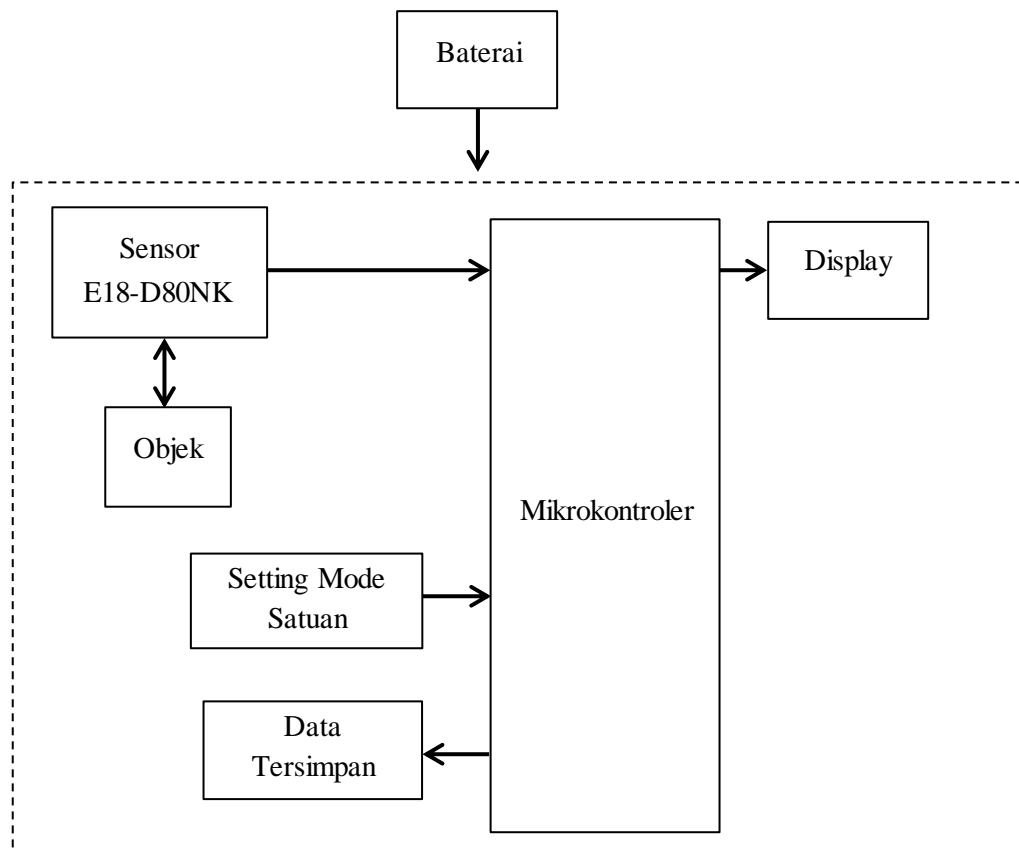


BAB III METODE PENELITIAN

3.1 Blok Diagram

Blok diagram dibuat untuk memetakan dari proses suatu kerja. Blok diagram berfungsi untuk memudahkan seseorang dalam memahami cara kerja itu sendiri.

Blok diagram alat ditunjukkan pada Gambar 3.1



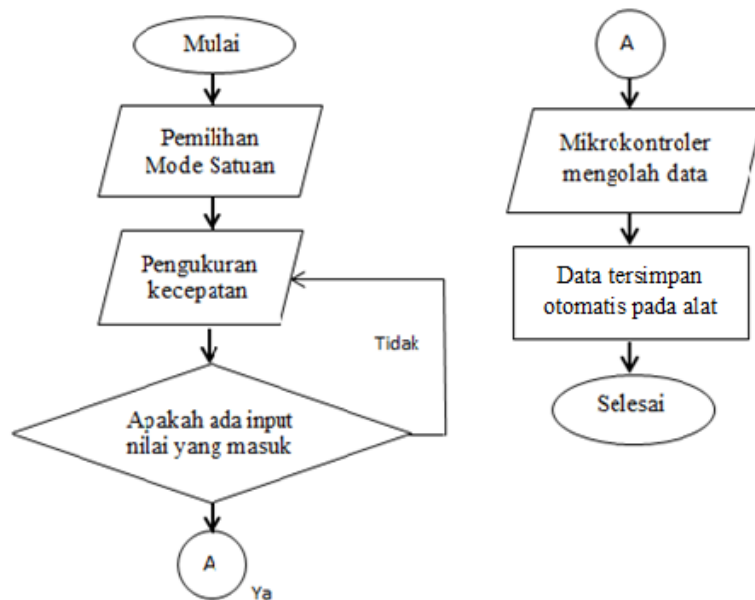
Gambar 3.1 Blok Diagram Alat

Transmitter inframerah memancarkan gelombang yang ditembakkan ke objek. Pada objek diberi garis putih sebagai bidang reflektif kemudian akan ditangkap oleh phototransistor sebagai *receiver*. *Output receiver* akan diteruskan dan dikonversi dalam bentuk RPM atau km/jam oleh

mikrokontroler. Hasil dari mikrokontroler akan ditampilkan ke *display* LCD 16x2. Sebelum dilakukan pengukuran dilakukan pemilihan mode pengukuran yang berupa RPM atau km/jam. Setelah itu tekan *start* untuk memulai pengukuran dan pengukuran akan berhenti otomatis dalam waktu 1 menit. Data tersimpan otomatis pada alat setiap 6 detik.

3.2 Diagram Alir Program

Diagram alir program ditunjukkan pada Gambar 3.2



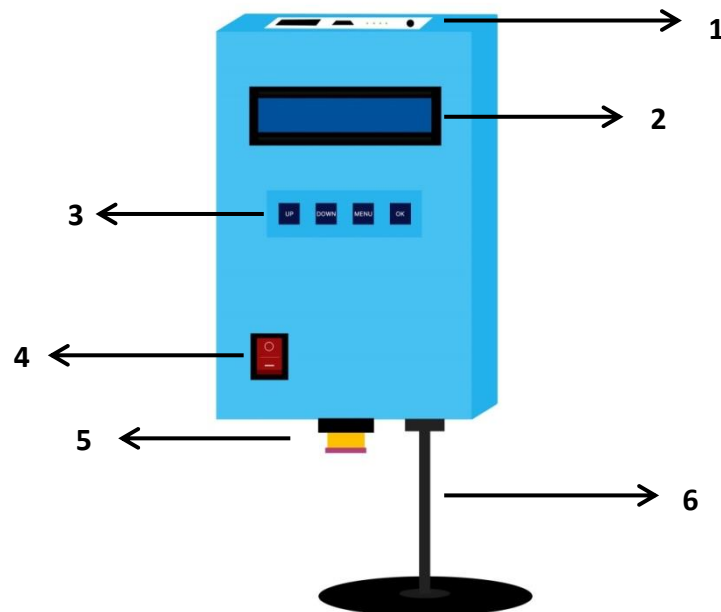
Gambar 3.2 Diagram Alir Program

Diagram alir atau *flowchart* 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.

Dilakukan pemilihan *setting* satuan pengukuran yang berupa *contact* (km/jam) atau *non contact* (RPM). Setelah itu tekan *start* untuk memulai pengukuran. *Transmitter* inframerah memancarkan gelombang yang ditembakkan ke objek. Pada objek diberi garis putih sebagai bidang reflektif kemudian akan ditangkap oleh phototransistor sebagai *receiver*. Setelah ada nilai pengukuran yang tertampil, *output receiver* akan diteruskan dan dikonversi dalam bentuk RPM atau km/jam oleh mikrokontroler. Pengukuran berhenti otomatis dalam 1 menit. Data tersimpan otomatis setiap 6 detik. Hasil dari mikrokontroler akan disimpan dan ditampilkan ke *display* LCD 16x2.

3.3 Diagram Mekanik

Bentuk dari rancangan alat dapat dilihat pada Gambar 3.3 di bawah ini



Gambar 3.3 Diagram Mekanik

Keterangan pada Gambar 3.3

1. Indikator Baterai
2. LCD
3. Tombol *up*, *down*, *menu*, dan *ok*
4. Saklar *ON/OFF*
5. Sensor E 18-D 80NK
6. Asessoris roda

3.4 Alat dan Bahan

3.4.1 Persiapan Alat

Untuk menunjang pembuatan alat, pengukuran, serta pengujian, penulis menggunakan beberapa peralatan. Peralatan-peralatan tersebut antara lain:

- a. Solder listrik
- b. *Soldering pump*
- c. *Tool set*
- d. Bor PCB
- e. Multimeter
- f. PCB

3.4.2 Persiapan Bahan

Dalam pembuatan alat, penulis membutuhkan bahan-bahan yang disesuaikan dengan kebutuhan alat tersebut. Bahan-bahan yang digunakan antara lain:

Tabel 3.1 Daftar Bahan

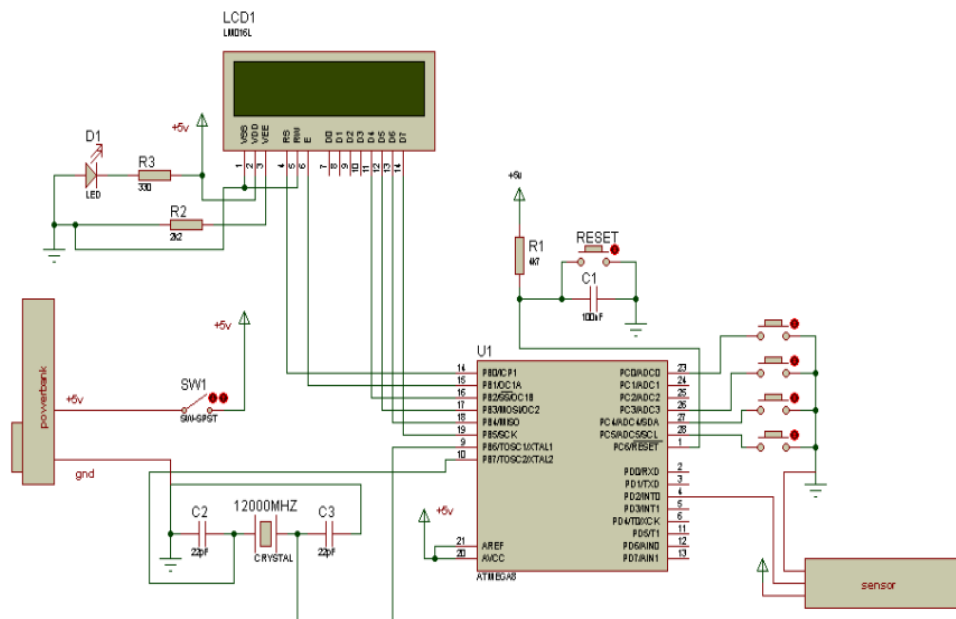
NO	NAMA BAHAN	JUMLAH
1	Resistor 330	2
2	Resistor 2200	1
3	LED	2
4	ATMega 8	1
5	<i>Pin Header</i>	20
6	Kapasitor 100 μ F	2
7	Kapasitor 100nF	2
8	Kapasitor 22pF	2
9	Saklar	1
10	Kabel <i>Jumper</i>	20
11	LCD 16x2	1
12	Sensor E18-D 80 NK	1
13	Akrilik	1
14	<i>Push Button</i>	5

3.5 Rancangan Perangkat Keras

3.5.1 Rangkaian Minimum Sistem

Blok rangkaian minimum sistem berfungsi sebagai otak dari alat yang berisi perintah dan kondisi dari sensor maupun blok lainnya terprogram di dalam blok rangkaian ini.

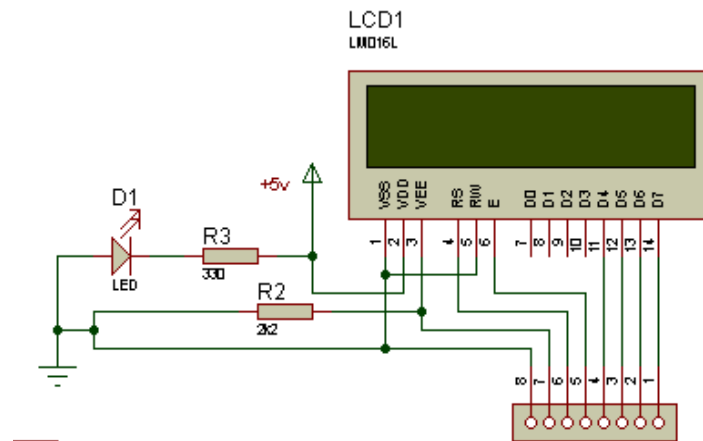
Rangkaian Minimum sistem ditunjukkan pada Gambar 3.4:



Gambar 3.4 Rangkaian Minimum Sistem

3.5.2 Rangkaian LCD

Rangkaian LCD ditunjukkan pada Gambar 3.5

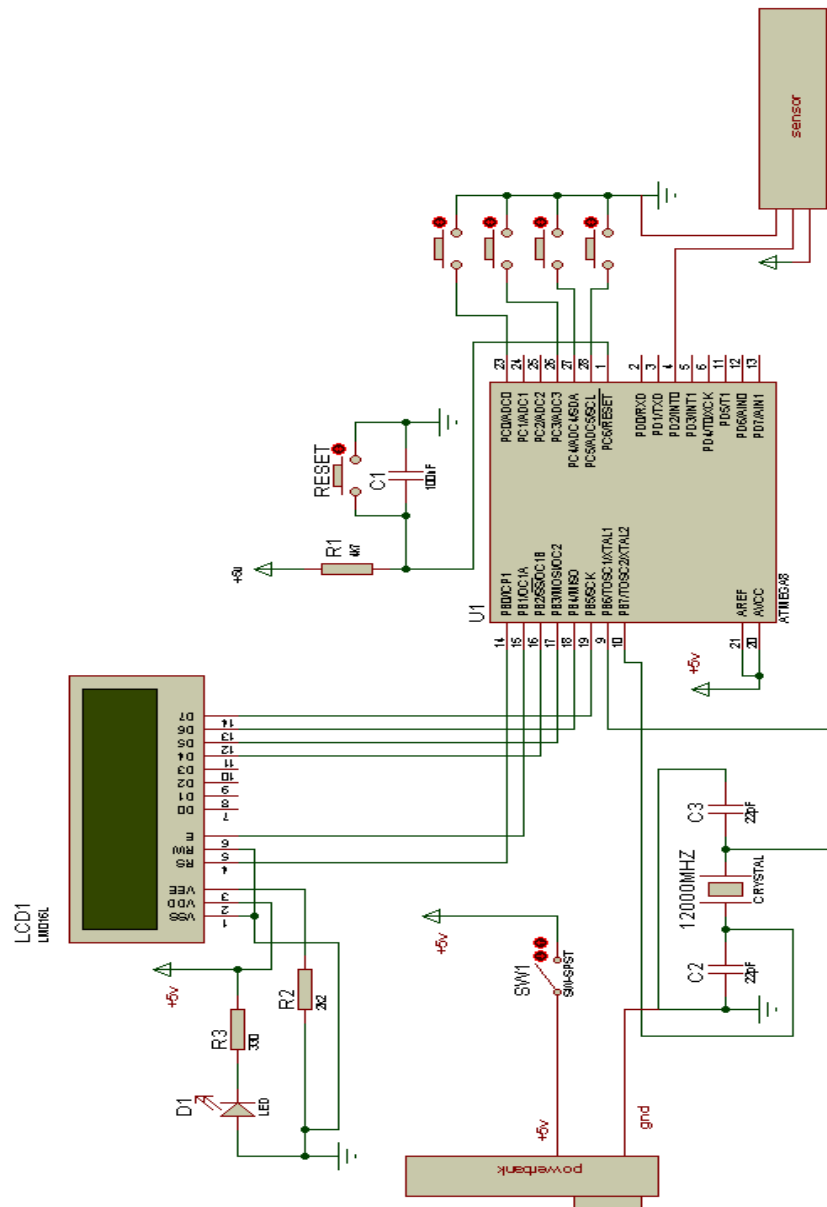


Gambar 3.5 Rangkaian LCD

Rangkaian LCD berfungsi untuk menghubungkan rangkaian minimum sistem dengan LCD, yang berguna untuk menampilkan perintah yang sedang dijalankan.

3.5.3 Rangkaian Keseluruhan

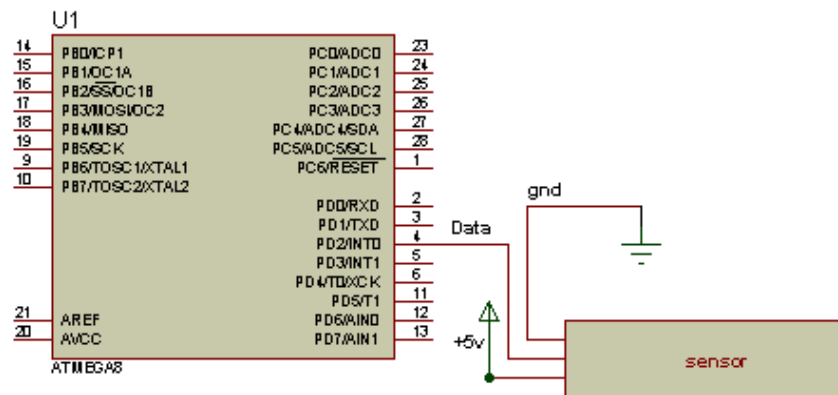
Rangkaian keseluruhan dari alat ini ditunjukkan pada Gambar 3.6



Gambar 3.6 Rangkaian Keseluruhan

Blok rangkaian keseluruhan merupakan gambaran semua rangkaian yang menjadi satu.

3.5.4 Rangkaian Sensor



Gambar 3.7 Rangkaian Sensor

Rangkaian *transmitter* pada sensor E 18-D 80NK yaitu ketika power ditekan maka *infrared* akan memancar ke reflektor objek yang sedang berputar. Kemudian masuk ke *rectifier regulator* untuk distabilkan tegangannya. Setelah tegangan stabil, masuk ke modulator. Modulator adalah suatu rangkaian yang berfungsi untuk menumpangkan data pada frekuensi gelombang pembawa ke sinyal agar bisa dikirim ke penerima melalui media tertentu misalnya kabel atau udara.

Pada rangkaian *receiver* terdapat *photo transistor* berfungsi menerima cahaya *infrared* yang dipancarkan/dipantulkan pada reflektor. Cahaya tersebut terkonversi menjadi tegangan, lalu dikuatkan oleh *amplifier*. Setelah dikuatkan, tegangan masuk ke demodulator. Demodulator yaitu proses membaca/memisahkan data dari sinyal yang diterima. Keluaran dari demodulator ditegaskan *clock logic* agar terbaca pada mikrokontroler. Setelah itu, masuk ke mikrokontroler melalui

PD2/INT0 yaitu *counter*. *Counter* berfungsi untuk menghitung *clock* yang berasal dari *external* mikrokontroler yaitu berasal dari sensor yang selanjutnya akan diproses ke mikrokontroler untuk dihitung nilai RPM atau km/jam.

3.6 Tahap Perancangan Perangkat Lunak

1. Listing Program Awal

```
#include <mega8.h>
#include <alcd.h>
#include <delay.h>
#include <stdlib.h>
#include <stdio.h>
#define mode PINC.5
#define up PINC.3
#define ok PINC.4
#define down PIND.0
```

Listing 3.1 Program Awal

Pada *Listing 3.1* merupakan program awal, dapat dijelaskan bahwa kode `#include <mega8.h>` digunakan untuk memasukkan *library* ATmega8. Kode `#include <alcd.h>` digunakan untuk memasukkan *library* LCD. Kode `#include <delay.h>` digunakan untuk memasukkan *library* waktu tunda. Kode `#include <stdlib.h>` digunakan untuk memasukkan *library* fungsi kontrol proses. Kode `#include <stdio.h>` digunakan untuk memasukkan *library* standarisasi *input output*. Kode `#define mode PINC.5` digunakan untuk mengalamatkan tombol *mode* pada PINC.5. Kode `#define up PINC.3` digunakan untuk mengalamatkan tombol *up* pada PINC.3. Kode `#define ok PINC.4` digunakan untuk mengalamatkan tombol *ok* pada PINC.4. Kode `#define down PINC.0` digunakan untuk mengalamatkan tombol *down* pada PINC.0.

2. Listing Program Menghitung RPM dan km/jam

```

TCNT1H=0xD23A >> 8;
TCNT1L=0xD23A & 0xff;
    speed=((float)frekuensi*19)*0.036;
    pulsa=((float)frekuensi*60);
    frekuensi=0;
    detik++;
    detiks--;
    if(detik==6){detik=0;memr=memr+1;mems=mems+1;}

```

Listing 3.2 Program Menghitung RPM dan km/jam

Pada program menghitung RPM dan km/jam, kode *TCNT1H=0xD23A >> 8;* dan *TCNT1L=0xD23A & 0xff;* digunakan untuk program *timer* untuk 1 detik. Kode *speed=((float)frekuensi*19)*0.036;* digunakan untuk menghitung kecepatan dalam satuan km/jam. Kode *pulsa=((float)frekuensi*60);* digunakan untuk menghitung kecepatan dalam satuan RPM. Kode *frekuensi=0* mempunyai arti bahwa frekuensi kembali ke 0. Kode *detik++;* mempunyai arti bahwa variabel detik akan menghitung waktu secara maju/bertambah. Kode *detiks--;* mempunyai arti bahwa variabel detiks akan menghitung waktu secara mundur/berkurang. Kode *if(detik==6){detik=0;memr=memr+1;mems=mems+1;}* mempunyai arti jika detik bernilai 6 maka detik akan kembali ke 0, serta akan menggeser penyimpanan ke slot berikutnya.

3. Listing Program Menampilkan Nilai RPM

```

void rpm() {
memr=0;
detik=0;
detiks=60;
while(1) {
    lcd_clear() ;
    lcd_gotoxy(0,0);
    lcd_putsf("RPM:");
    ftoa(pulsa,0,buf);
    lcd_puts(buf);
    lcd_gotoxy(0,1);

```

Listing 3.3 Program Menampilkan Nilai RPM

Pada program menampilkan nilai RPM, kode `void rpm()` mempunyai arti bahwa pendeklarasian fungsi RPM. Kode `memr=0;` mempunyai arti bahwa memori RPM datanya dimulai dari 0. Kode `detik=0;` mempunyai arti bahwa perhitungan detik dimulai dari 0. Kode `detiks=60;` mempunyai arti bahwa perhitungan detiks dimulai dari 60 dan berjalan mundur. Kode `lcd_clear();` mempunyai arti bahwa ketika pengaturan sudah terpenuhi semua, LCD akan menghapus tampilan dan kembali ke tampilan awal. Kode `lcd_gotoxy(0,0);` mempunyai arti bahwa letak koordinat yang akan ditampilkan pada LCD. Kode `lcd_putsf("RPM");` mempunyai arti bahwa LCD menampilkan karakter "RPM". Kode `ftoa(pulsa,0,buf)` mempunyai arti bahwa tampilan RPM terdiri dari bilangan tanpa koma. Kode `lcd_puts(buf);` mempunyai arti bahwa LCD mengambil karakter dengan variabel `buf` di atas. Kode `lcd_gotoxy(0,1);` mempunyai arti bahwa letak koordinat yang akan digunakan sebagai koordinat karakter.

4. Listing Program Melakukan Penyimpanan Data RPM

```
void memori_rpm()
{
menu_mem_rpm=0;
while(1){
lcd_clear();
if(!mem){delay_ms(200);menu_mem_rpm=menu_mem_rpm+1;}
if(menu_mem_rpm>9){menu_mem_rpm=0;break;}
if(menu_mem_rpm==0){
lcd_gotoxy(0,0);
lcd_putsf("Memori RPM 1:");
lcd_gotoxy(0,1);
ftoa(saver0,0,buf);
lcd_puts(buf);
```

Listing 3.4 Program Melakukan Penyimpanan Data RPM

Pada program melakukan penyimpanan data RPM, kode *void memori_rpm()* mempunyai arti pendeklarasian fungsi memori RPM. Kode *menu_mem_rpm=0;* mempunyai arti bahwa penyimpanan memori RPM yang pertama dimulai dari 0. Kode *while(1);* mempunyai arti bahwa program utama yang dibaca secara perulangan. Kode *lcd_clear();* mempunyai arti bahwa ketika pengaturan sudah terpenuhi semua, LCD akan menghapus tampilan dan kembali ke tampilan awal. Kode *if(!mem){delay_ms(200);menu_mem_rpm=menu_mem_rpm+1;}* mempunyai arti bahwa jika tombol memori ditekan, akan menuju ke memori penyimpanan selanjutnya. Kode *if(menu_mem_rpm>9){menu_mem_rpm=0;break;}* mempunyai arti bahwa jika memori sudah tersimpan sebanyak 10 data, maka akan keluar dari fungsi penyimpanan. Kode *lcd_gotoxy(0,0);* mempunyai arti bahwa letak koordinat yang akan ditampilkan pada LCD. Kode *lcd_putsf("Memori RPM 1:");* mempunyai arti bahwa LCD menampilkan karakter “Memori RPM 1”. Kode *lcd_gotoxy(0,1);* mempunyai arti bahwa letak koordinat yang akan digunakan sebagai koordinat karakter. Kode *ftoa(saver0,0,buf)* mempunyai arti bahwa tampilan data RPM yang tersimpan terdiri dari bilangan tanpa koma. Kode *lcd_puts(buf);* mempunyai arti bahwa LCD mengambil karakter dengan variabel *buf* di atas.

5. Listing Program Menampilkan Hasil km/jam

Pada program menampilkan hasil km/jam, kode *void_spedo()* mempunyai arti pendeklarasian fungsi km/jam. Kode *mems=0;*

mempunyai arti bahwa memori km/jam datanya dimulai dari 0. Kode *detik=0;* mempunyai arti bahwa perhitungan detik dimulai dari 0. Kode *detiks=60;* mempunyai arti bahwa perhitungan detiks dimulai dari 60 dan berjalan mundur. Kode *lcd_clear();* mempunyai arti bahwa ketika pengaturan sudah terpenuhi semua, LCD akan menghapus tampilan dan kembali ke tampilan awal. Kode *lcd_gotoxy(0,0);* mempunyai arti bahwa letak koordinat yang akan ditampilkan pada LCD. Kode *lcd_putsf("SPEED");* mempunyai arti bahwa LCD menampilkan karakter "speed". Kode *ftoa(speed,1,buf)* mempunyai arti bahwa tampilan km/jam terdiri dari bilangan dengan 1 *digit* angka dibelakang koma. Kode *lcd_puts(buf);* mempunyai arti bahwa LCD mengambil karakter dengan variabel *buf* di atas. Kode *lcd_gotoxy(0,1);* mempunyai arti bahwa letak koordinat yang akan digunakan sebagai koordinat karakter. Kode *lcd_putsf("Km/jam")* mempunyai arti bahwa LCD menampilkan karakter "speed". Kode *lcd_gotoxy(0,1)* mempunyai arti bahwa letak koordinat yang akan ditampilkan pada LCD.

```

void spedo() {
    mems=0;
    detik=0;
    detiks=60;
    while(1) {
        lcd_clear() ;
        lcd_gotoxy(0,0);
        lcd_putsf("SPEED:");
        ftoa(speed,1,buf);
        lcd_puts(buf);
        lcd_putsf("Km/jam");
        lcd_gotoxy(0,1);
    }
}

```

Listing 3.5 Program Menampilkan Hasil km/jam

6. Listing Program Melakukan Penyimpanan Data km/jamdx

```

void memori_speed()
menu_mem_speed=0;
while(1){
lcd_clear();
if(!mem){delay_ms(200);menu_mem_speed=menu_mem_speed+1;}
if(menu_mem_speed>9){menu_mem_speed=0;break;}
if(menu_mem_speed==0){
lcd_gotoxy(0,0);
lcd_putsf("Memori SPEED 1:");
lcd_gotoxy(0,1);
ftoa(saves0,0,buf);
lcd_puts(buf);
lcd_putsf(" Km/jam");
}
}

```

Listing 3.6 Program Melakukan Penyimpanan Data km/jam

Pada program melakukan penyimpanan data km/jam, kode *void memori_speed()* mempunyai arti pendeklarasian fungsi memori km/jam. Kode *menu_mem_speed=0;* mempunyai arti bahwa penyimpanan memori km/jam yang pertama dimulai dari 0. Kode *while(1);* mempunyai arti bahwa program utama yang dibaca secara perulangan. Kode *lcd_clear();* mempunyai arti bahwa ketika pengaturan sudah terpenuhi semua, LCD akan menghapus tampilan dan kembali ke tampilan awal. Kode *if(!mem){delay_ms(200);menu_mem_speed=menu_mem_speed+1;}* mempunyai arti bahwa jika tombol memori ditekan, akan menuju ke memori penyimpanan selanjutnya. Kode *if(menu_mem_speed>9){menu_mem_speed=0;break;}* mempunyai arti bahwa jika memori sudah tersimpan sebanyak 10 data, maka akan keluar dari fungsi penyimpanan. Kode *lcd_gotoxy(0,0);* mempunyai arti bahwa letak koordinat yang akan ditampilkan pada LCD. Kode *lcd_putsf("Memori Speed 1:");* mempunyai arti bahwa LCD menampilkan

karakter “Memori Speed 1”. Kode *lcd_gotoxy(0,1)*; mempunyai arti bahwa letak koordinat yang akan digunakan sebagai koordinat karakter. Kode *ftoa(saves0,0,buf)* mempunyai arti bahwa tampilan data km/jam yang tersimpan terdiri dari bilangan tanpa koma. Kode *lcd_puts(buf)*; mempunyai arti bahwa LCD mengambil karakter dengan variabel *buf* di atas.

3.7 Variabel Penelitian

a. Variabel Bebas

Variabel bebas pada penelitian ini adalah kecepatan putaran motor, karena putaran motor tidak terpengaruh oleh rangkaian pada alat kalibrasi yang akan dibuat.

b. Variabel Tergantung

Variabel tergantung pada penelitian ini adalah sensor inframerah E 18-D 80NK, karena penelitian ini menggunakan sensor tersebut maka penelitian ini tergantung pada sensor tersebut.

c. Variabel Terkendali

Variabel terkendali pada penelitian ini adalah IC mikrokontroler ATmega yang berfungsi sebagai pengolah data yang dapat dikendalikan.

3.8 Teknik Pengujian Alat

3.8.1 Teknik Pengujian dengan Satuan RPM

Proses pengujian alat dengan pembacaan kecepatan RPM pada *centrifuge* dilakukan di Laboratorium Teknik Elektromedik UMY.

Pengujian pembacaan kecepatan RPM pada *centrifuge* dilakukan pada beberapa kecepatan diantaranya pada kecepatan 360 RPM, 960 RPM, 1440 RPM, 1740 RPM, 2100 RPM, 3600 RPM, 5400 RPM, dan 7300 RPM.

Langkah – langkah dalam pengujian dan pengukuran alat yaitu:

- a. Mempersiapkan alat, alat pembanding, dan alat yang akan diukur kecepatannya. Penulis menggunakan *tachometer* sebagai alat pembandingnya dan alat yang diukur kecepatannya yaitu *centrifuge*.
 - b. Menyiapkan tabel pendataan sebelum dimulai pengukuran terhadap alat yang diukur.
 - c. Mengatur kecepatan alat yang diukur kecepatannya.
 - d. Mencatat semua hasil pengukuran dari alat dan alat pembanding.
- Masing – masing kecepatan dilakukan pengukuran sebanyak 20 kali.

3.8.2 Teknik Pengujian dengan Satuan km/jam

Proses pengujian alat dengan pembacaan kecepatan km/jam dilakukan di rumah saudara dari penulis. Pengujian pembacaan kecepatan km/jam pada *treadmill* dilakukan pada beberapa kecepatan diantaranya pada kecepatan 2,2 km/jam, 3,4 km/jam, 4,9 km/jam, 7 km/jam.

Langkah – langkah dalam pengujian dan pengukuran alat yaitu:

- a. Mempersiapkan alat, alat pembanding, dan alat yang akan diukur kecepatannya. Penulis menggunakan *setting* pada *treadmill* sebagai pembandingnya dan alat yang diukur kecepatannya yaitu *treadmill*.

- b. Menyiapkan tabel pendataan sebelum dimulai pengukuran terhadap alat yang diukur.
- c. Mengatur kecepatan alat yang diukur kecepatannya.
- d. Mencatat semua hasil pengukuran dari alat dan alat pembanding.
Masing – masing kecepatan dilakukan pengukuran sebanyak 20 kali.

3.9 Spesifikasi Alat

Gambar alat yang dirancang ditunjukkan pada Gambar 3.8



Gambar 3.8 Gambar Alat

Spesifikasi alat yang dirancang yaitu:

- a. Nama : *Tachometer Contact dan Non Contact* Berbasis Mikrokontroler
- b. Range Pengukuran : 360 RPM, 600 RPM, 960 RPM, 1.140 RPM, 1.260 RPM, 1.440 RPM, 1.560 RPM, 1.680 RPM, 1.740 RPM, 1.860

RPM, 2.160 RPM, 3500 RPM, 5.400 RPM, dan 7.300 RPM serta 2,2 km/jam, 3,4 km/jam, 4,9 km/jam, dan 7 km/jam.

- c. *Display* : LCD Karakter 16x2
- d. Tegangan : ± 5 Volt DC
- e. Dimensi : P = 10 cm, l = 5 cm, dan t = 15 cm
- f. Sensor : Sensor Inframerah E 18 – D 80NK

3.10 Langkah Penggunaan Alat

Langkah penggunaan alat sebagai berikut:

1. Menekan tombol *ON/OFF* pada posisi *ON* untuk menghidupkan alat
2. Memilih mode satuan kecepatan yang dikehendaki, yaitu memilih RPM untuk mengukur kecepatan dengan satuan RPM atau memilih *speed* untuk mengukur kecepatan dengan satuan km/jam.
3. Menghidupkan alat yang akan diukur kecepatannya, atur pada kecepatan tertentu.
4. Mengukur kecepatan alat yang akan dikalibrasi.
5. Nilai kecepatan yang terbaca oleh sensor akan tertampil pada LCD dalam satuan sesuai yang telah diatur. Kecepatan yang terbaca oleh sensor akan tersimpan otomatis setiap 6 detik dengan penyimpanan sebanyak 10 data dalam 1 menit.
6. Menekan tombol memori untuk melihat data yang tersimpan selama 1 menit.
7. Menekan tombol *OFF* untuk mematikan alat.