

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Alat dan Bahan**

##### **3.1.1 Alat**

Dalam melakukan penelitian ini digunakan beberapa peralatan diantaranya sebagai berikut :

- a. *Toolset*
- b. Multimeter
- c. PC (Komputer/Laptop)
- d. Alat Tulis
- e. *Digital Pressure Meter*
- f. *Stopwatch*

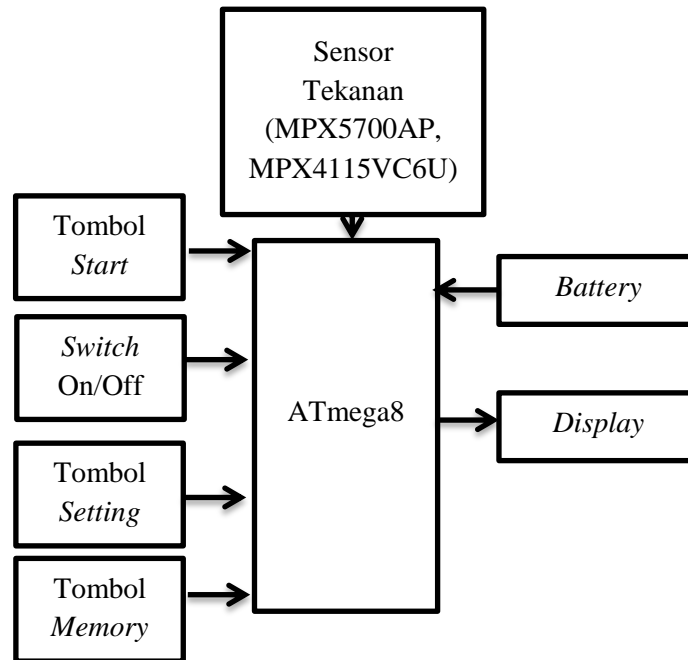
##### **3.1.2 Bahan**

Dalam melakukan penelitian digunakan beberapa bahan diantaranya sebagai berikut :

- a. Akrilik
- b. Kabel *female*
- c. Baut
- d. soket
- e. Sensor *MPX5700AP*
- f. Sensor *MPXV4115VC6U*
- g. Modul *Mikrokontroler ATmega8A*
- h. Modul sensor

- i. Modul *Charger Battery*
- j. *Liquid Crystal Display (LCD) 2x16*
- k. Selang

### 3.2 Diagram Blok Sistem

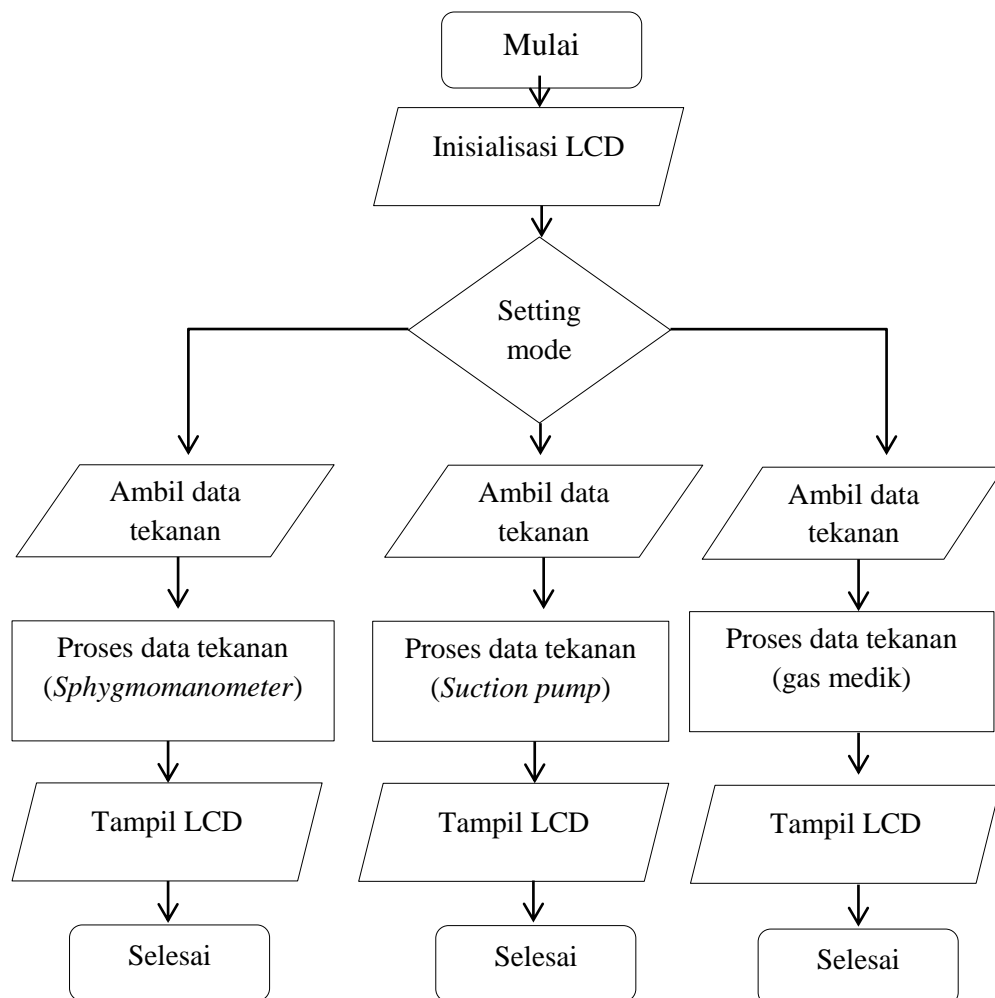


Gambar 3. 1 Diagram Blok

Saat alat tombol *power* ditekan, semua rangkaian mendapat tegangan termasuk sensor tekanan, sehingga sensor dalam keadaan *ready* dan siap untuk beroperasi. Sebelum melakukan pengukuran, pada menu tersedia mode pengukuran yaitu *air pressure*, *vacum pressure*, dan *oxygen pressure*. Setelah itu tekan *enter*. Sebelum ada tekanan, *display* menunjukkan nilai 0 mmHg. Pemompaan dilakukan secara manual. Tekanan yang masuk akan diterima oleh sensor tekanan, kemudian dikonversi menjadi tegangan *analog*. Tegangan tersebut kemudian akan diproses oleh rangkaian pengkondisi sinyal *analog* sebelum masuk ke ADC internal yang telah

tersedia dalam IC Mikrokontroler ATmega8. Setelah tegangan masuk ADC, tegangan analog akan dikonversikan menjadi tegangan digital untuk diproses oleh *microcontroller*. Data hasil dari pengolahan tersebut akan diproses melalui ATmega8 dan akan disimpan, kemudian ditampilkan melalui LCD.

### 3.3 Diagram Alir Proses/Program



Gambar 3. 2 Diagram Alir

Saat *start* alat dalam keadaan *ready*, *display* LCD akan menampilkan pemilihan mode, yaitu mode pengukuran atau pembacaan penyimpanan data. Ketika mode pengukuran dipilih, maka dapat dilakukan pengukuran.

Proses ini tanpa menggunakan delay atau waktu tertentu untuk mendapatkan hasil, saat tombol *enter* ditekan tekanan yang terdeteksi oleh sensor akan di proses oleh *microcontroller* dalam proses pengambilan data. Hasil perhitungan dari ADC akan ditampilkan ke *LCD*.

### 3.4 Diagram Mekanis Sistem



Gambar 3. 3 Diagram Mekanik Alat

Keterangan:

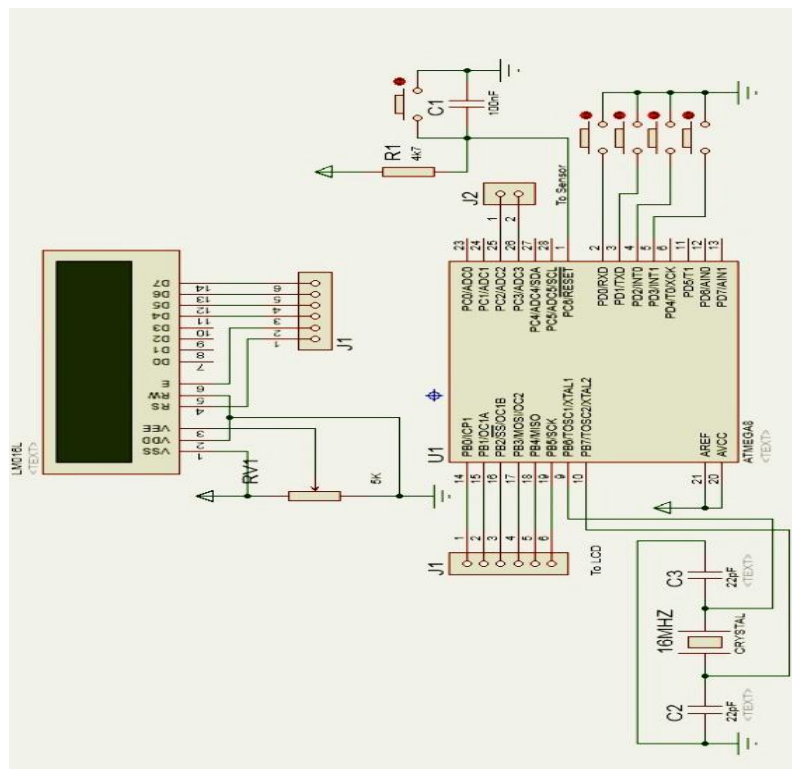
1. Selang *pressure*
2. Selang *vacum pressure*
3. *Display LCD*
4. *Enter Button*
5. *BAR/KPA Button*
6. *Mode Button*
7. *memory Button*

### 3.5 Perancangan Perangkat Keras

#### 1.5.1 Perakitan Rangkaian

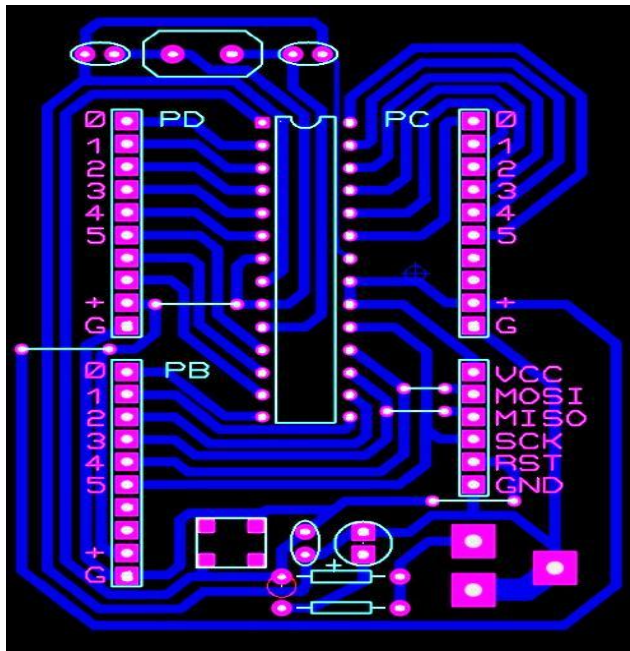
##### 1. Langkah Perakitan

- a. Membuat skematik rangkaian minimum sistem dengan menggunakan aplikasi pada laptop, aplikasi yang digunakan pada pembuatan alat ini adalah *proteus*. Untuk gambar skematik rangkaian minimum sistem pada aplikasi dapat dilihat pada gambar 3.4 berikut.



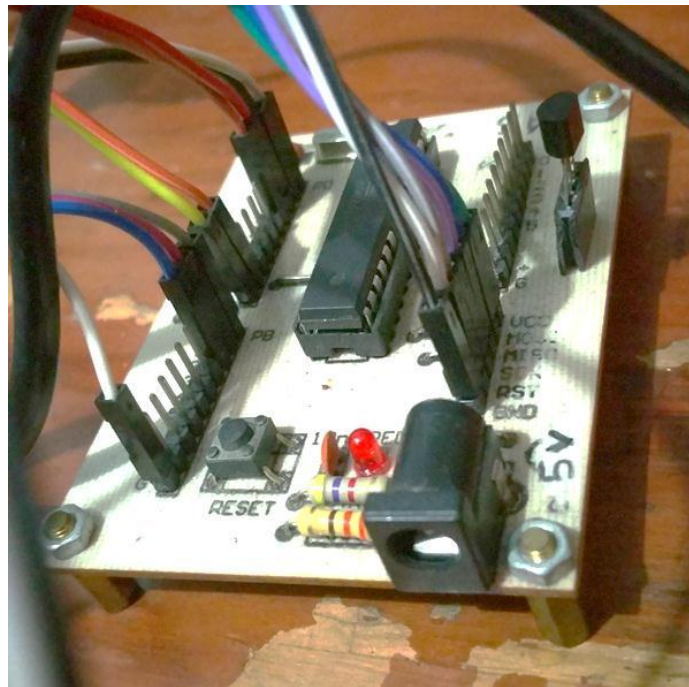
Gambar 3.4 Rangkaian Minimum Sistem

- b. Tahap selanjutnya membuat *layout* nya dan disablon ke papan PCB. Untuk *layout* minimum sistem pada papan PCB dapat dilihat pada gambar 3.5 berikut.



Gambar 3.5 Layout Keseluruhan

2. Gambar rangkaian Minimum Sistem

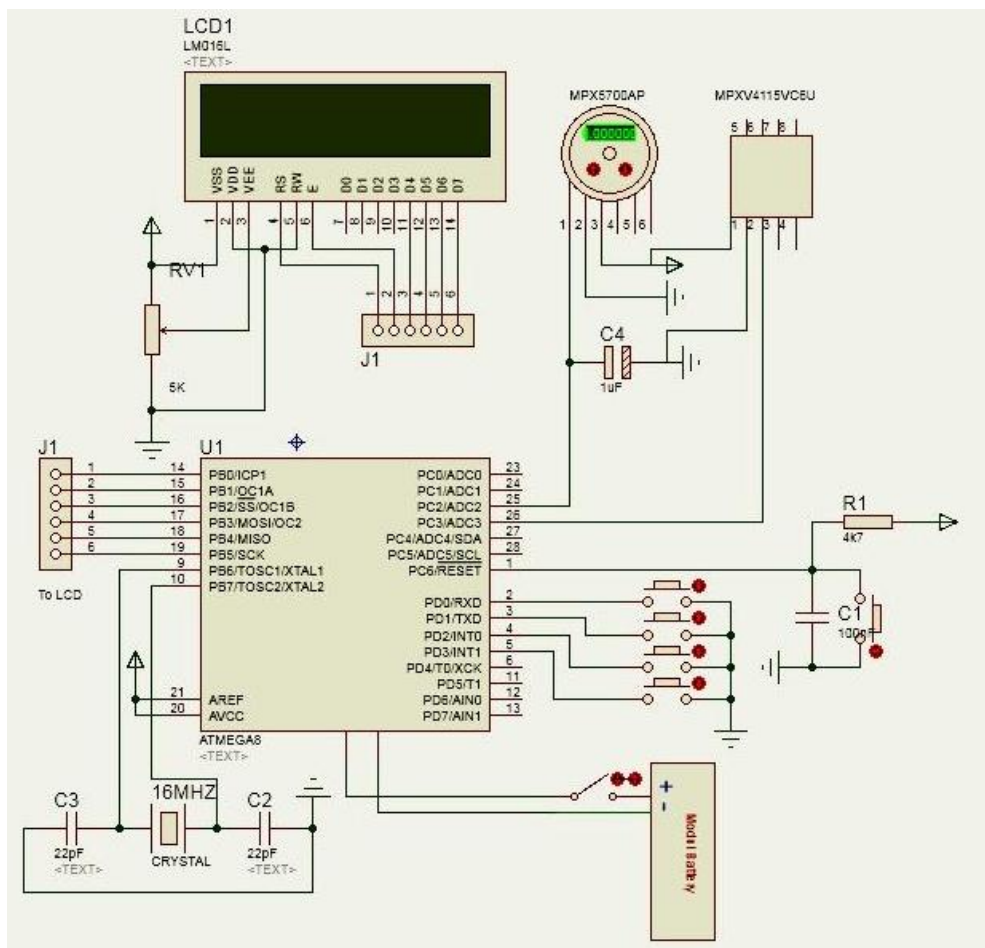


Gambar 3.6 Gambar Modul Minimum Sistem

### 3.6 Rangkaian Keseluruhan

Rangkaian ini tersusun oleh beberapa blok rangkaian yang telah terpasang komponen-komponen sesuai fungsinya, dan dijadikan satu secara elektrik agar menjadi sebuah sistem yang sesuai perancangan modul. Ada beberapa blok yang terpasang pada satu sistem ini, antara lain adalah:

1. Rangkaian Minimum Sistem
2. Rangkaian Sensor
3. Rangkaian LCD 16x2



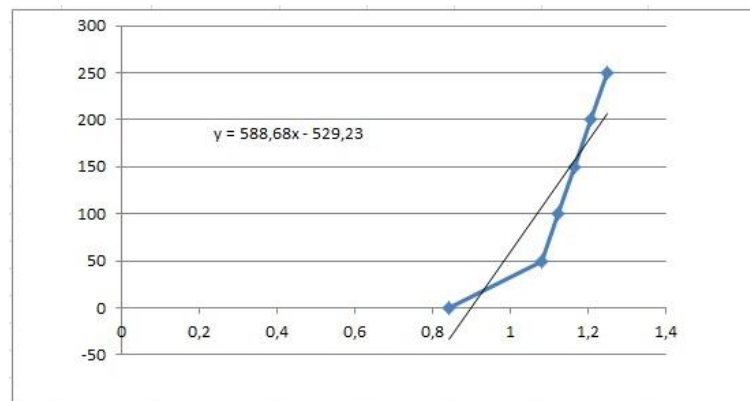
Gambar 3.7 Rangkaian Keseluruhan

### 3.7 Pembuatan Program Sistem

Sebelum penulis melakukan pembuatan program sistem, penulis melakukan pengolahan rumus sensor tekanan pada *excel*.

#### 3.7.1 Rumus Sensor Tekanan Udara

Pengolahan rumus sensor tekanan udara dengan menggunakan rumus persamaan garis lurus.



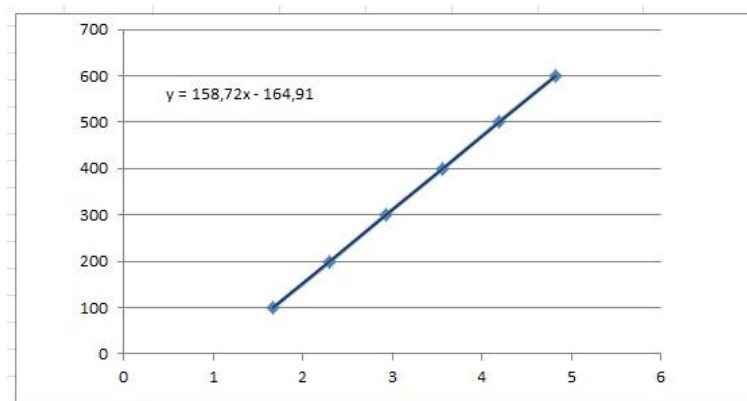
Gambar 3.8 Grafik Konversi dari Tegangan(Volt) ke Tekanan(mmHg)

0 mmHg sampai 300 mmHg.

#### 3.7.2 Rumus Sensor Tekanan Oksigen

Pengolahan rumus sensor tekanan vakum dengan menggunakan rumus persamaan garis lurus.





Gambar 3.9 Grafik Konversi dari Tegangan(Volt) ke Tekanan(kPa) 0 mmHg sampai 600 mmHg.

### 3.8 Perancangan Perangkat Lunak

#### 3.8.1 Program Void Setup

```

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

#define set PIND.0
#define mode PIND.1
#define ok PIND.2
#define timer PIND.3 //ganti push button

eeprom float saveGM;
eeprom float saveMM1;
eeprom float saveMM2;
eeprom float saveV, saveV1, saveV2;
eeprom int memGM=0,memMM1=0,memMM2=0,memV;
float adc1;
int data_adc1;
float Vpress, KPA, mmHg1;
unsigned char opsi=0, menu_run, menu_mem, i=0, timers=0;
int detiks;
char buf[33],adc[33],vout[33];

#define ADC_VREF_TYPE 0x00
interrupt [TIM1_OVF] void timer1_ovf_isr(void) //TIMER
{

```

```

// Reinitialize Timer1 value
//TCNT1H=0xD23A >> 8;
//TCNT1L=0xD23A & 0xff;
TCNT1H=0xC2F7 >> 8;
TCNT1L=0xC2F7 & 0xff;

    if(!timer)
    {
        timers=1;
    }
    if(timers==1){detiks++;}

if(detiks==60){detiks=0;memMM1=memMM1+1;memMM2=memM
M2+1;memV=memV+1;}
}

// 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;
}

```

### 3.8.2 Program Sensor *MPX5700AP*

```

void Vmmhg1(){          //Positif
detiks=0;
timers=0;
memMM1=0;
memV=0;
while(1){

    lcd_clear();
    adc1=0;
    for(i=0;i<100;i++) //sampling 100x
    {
        data_adc1 = read_adc(2);
        Vpress=(float)(data_adc1*4.9/1023);
    }
}
}

```

```

        adc1=adc1+Vpress;
    }
    adc1=adc1/100;

if(adc1>=0.825&&adc1<0.871)        //0-50
{mmHg1=(961.03*adc1)-786.79;}
if(adc1>=0.871 && adc1<0.912)    //50-100
{mmHg1=(1186.49*adc1)-981.4;}
if(adc1>=0.912 && adc1<0.956)    //100-150
{mmHg1=(1099.4*adc1)-900.87;}
if(adc1>=0.956 && adc1<0.998)    //150-200
{mmHg1=(1532.2*adc1)-1328.8;}
if(adc1>=0.998 && adc1<1.040)    //200-250
{mmHg1=(1058.93*adc1)-851.04;}
if(adc1>=1.040 && adc1<1.082)    //250-300
{mmHg1=(1238.5*adc1)-1039;}
for(o=0;o<100;o++)
{
    mmHg2=mmHg2+mmHg1;
}
mmHg2=mmHg2/100;

if(adc1<0.843){mmHg1=0;}

lcd_gotoxy(0,0);
lcd_putsf("Tekanan :");
ftoa(mmHg1,1,buf);
lcd_puts(buf);

lcd_gotoxy(0,1);
sprintf(buf,"0:%d ",detiks);
lcd_puts(buf);

lcd_gotoxy(10,1);
ftoa(adc1,3,buf);
lcd_puts(buf);
delay_ms(250);

if(memMM1==1){saveMM1=mmHg1;saveV1=adc1;}
if(memMM1==1){delay_ms(500);break;}
if(!ok){delay_ms(200);break;}
lcd_clear();

```

```

}

void GasMedis() { //
detiks=0;
memGM=0;
memV=0;
while(1){

    lcd_clear();
    adc1=0;
    for(i=0;i<100;i++) //sampling 100x
    {
        data_adc1 = read_adc(2);
        Vpress=(((float)data_adc1/1023)*4.75);
        adc1=adc1+Vpress;
    }
    adc1=adc1/100;

    if(adc1>=0.825&&adc1<5.00) //0-600
    {KPA=((1190.1*adc1)-984.93)/7.5;}

    if(adc1<0.843){KPA=0;}

    if(KPA<=0)
    {
        lcd_gotoxy(0,0);
        lcd_putsf("Tekanan : NONE");
    }
    if(KPA>0 && KPA<400)
    {
        lcd_gotoxy(0,0);
        lcd_putsf("Tekanan : LOW");
    }

    if(KPA>400 && KPA<500)
    {
        lcd_gotoxy(0,0);
        lcd_putsf("Tekanan : NORMAL");
    }
    if(KPA>500)
    {
        lcd_gotoxy(0,0);
        lcd_putsf("Tekanan : HIGH");
    }

    lcd_gotoxy(0,1);

```

```

ftoa(KPA,1,buf);
lcd_puts(buf);

lcd_gotoxy(10,1);
ftoa(adc1,3,buf);
lcd_puts(buf);

delay_ms(250);

if(!ok){ saveGM=KPA;saveV=adc1;delay_ms(200);break;}
lcd_clear();
}

```

### 3.8.3 Program Sensor *MPXV4115VC6U*

```

void Vmmhg2() {           //Vacuum
detiks=0;
timers=0;
memMM2=0;
memV=0;
while(1){

    lcd_clear();
    adc1=0;
    for(i=0;i<100;i++) //sampling 100x
    {
        data_adc1 = read_adc(3);
        Vpress=(float)data_adc1*4.9/1023;
        adc1=adc1+Vpress;
    }
    adc1=adc1/100;
    //KPA=(float)((adc1/4.75)-0.92)/0.007652;
    //mmHg1=(float)(KPA*7.5);

    if(adc1>4.536){ KPA=0;}

    lcd_gotoxy(0,0);
    lcd_putsf("Tekanan :");
    ftoa(KPA,1,buf);
    lcd_puts(buf);

    lcd_gotoxy(0,1);
    sprintf(buf,"0:%d ",detiks);
    lcd_puts(buf);
}

```

```

lcd_gotoxy(10,1);
ftoa(adc1,3,buf);
lcd_puts(buf);
delay_ms(250);

if(memMM2==1){saveMM2=KPA;saveV2=adc1;}
if(memMM2==1){delay_ms(500);break;}
if(!ok){delay_ms(200);break;}
lcd_clear();
} }

```

### 3.9 Langkah Pengujian Alat

### 3.10 Rumus Statistik

#### 3.10.1 Rata-rata

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

$$\text{Rata - Rata } (\bar{X}) = \frac{\sum X_i}{n}$$

Dengan keterangan sebagai berikut:

$$\bar{X} = \text{rata - rata}$$

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

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

#### 3.10.2 Simpangan

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

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

Dengan keterangan sebagai berikut:

Simpanan = nilai *error* yang dihasilkan

$X_n$  = rata-rata data alat pembanding

$\bar{X}$  = rata-rata data alat penulis

### 3.10.3 Error (%)

Error (kesalahan) adalah selisih antara mean terhadap masing-masing data. Error dapat dirumuskan sebagai berikut:

$$\% \text{Error} = \frac{X_n - \bar{X}}{X_n} \times 100\%$$

### 3.10.4 Standar Deviasi

Standar deviasi adalah suatu nilai yang menunjukkan tingkat (derajat) variasi kelompok data atau ukuran standar penyimpangan dari *mean*nya.

Rumus standar deviasi (SD) adalah:

$$SD = \sqrt{\frac{(X_1 - \bar{X})^2 + (X_2 - \bar{X})^2 + \dots + (X_n - \bar{X})^2}{n - 1}}$$

Dimana = Standar Deviasi

n = Banyaknya data

x = rata-rata

$X_1, \dots, X_n$  = nilai data

### 3.10.5 Ketidakpastian ( $U_a$ )

Ketidakpastian adalah kesangsian yang muncul pada tiap hasil.

Atau pengukuran biasa disebut, sebagai kepresisian data satu dengan data yang lain. Rumus ketidakpastian adalah sebagai berikut.

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

Dimana:

STDV = standar deviasi

n = banyaknya data