

b. Sistematika Pengukuran

Setelah melakukan pengukuran modul stetoskop elektronik dilengkapi perhitungan denyut jantung permenit 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 :

$$\bar{X} = \frac{\sum X_n}{n} \quad (3.2)$$

dengan :

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

$$x_1 \cdots x_n = \text{nilai data}$$

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

2. Simpangan (*Error*)

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

$$\text{simpangan} = x_n - \bar{x} \quad (3.3)$$

dengan :

$$\text{simpangan} = \text{Nilai error yang dihasilkan}$$

$$x_n = \text{Rata – rata data } SpO_2$$

$$\bar{x} = \text{Rata – rata data modu}$$

1

3. *Persentase Error*

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

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

dengan :

\bar{Y} = rata-rata data kalibrator

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

4. *Standart Deviasi*

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

Rumus *Standart Deviasi* adalah:

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

dengan:

SD = standart deviasi

\bar{X} = rata-rata

$x_1 \dots x_n$ = nilai data

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

3.1 Perancangan Pengujian

3.1.1 Listing Program

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

```
#include <mega8535.h>

#include <stdlib.h>

unsigned char detak=0,temp[3],detik=0;

bit a,b;

// Declare your global variables here

// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Reinitialize Timer1 value
TCNT1H=0xBDC >> 8;
TCNT1L=0xBDC & 0xff;
// Place your code here
detik++;
}

void main(void)
{
```

```
lcd_putsf("Timer:");  
    lcd_gotoxy(5,0);  
    itoa(detik,temp);  
    lcd_gotoxy(5,0);  
    lcd_puts(temp);  
    lcd_gotoxy(0,1);  
    lcd_putsf("Detak:");  
    lcd_gotoxy(5,1);  
    itoa(detak,temp);  
    lcd_gotoxy(5,1);  
    lcd_puts(temp);  
    }  
} itoa(detik,temp);  
    lcd_gotoxy(5,0);  
    lcd_puts(temp);  
    lcd_gotoxy(0,1);  
    lcd_putsf("Detak:");  
    lcd_gotoxy(5,1);  
    itoa(detak,temp);  
    lcd_gotoxy(5,1);  
    lcd_puts(temp);  
  
    }  
}
```