

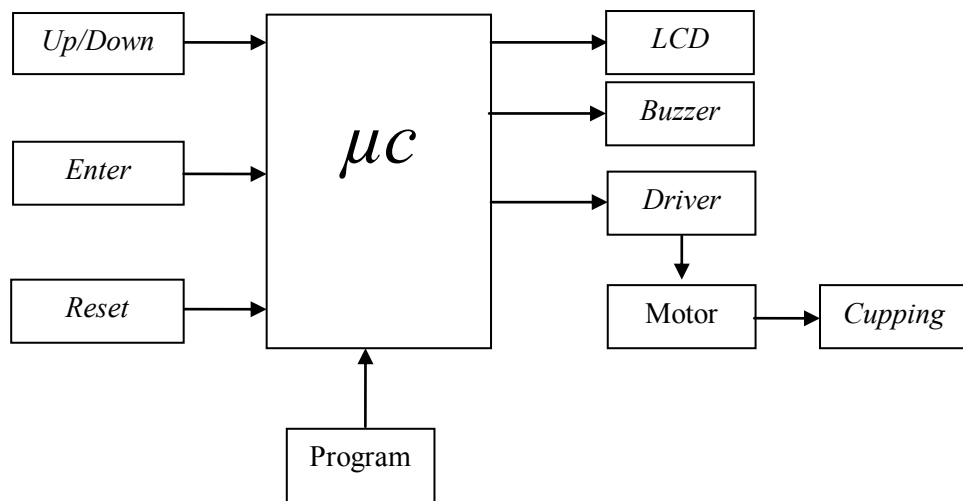
BAB III

METODE PENELITIAN

3.1 Perancangan Perangkat Keras

3.1.1 Blok Diagram

Diagram blok pengembangan *breastpump* elektrik berbasis mikrokontroler *ATMega8535* dilengkapi dengan pengatur waktu dan tekanan dapat dilihat pada Gambar 3.1.



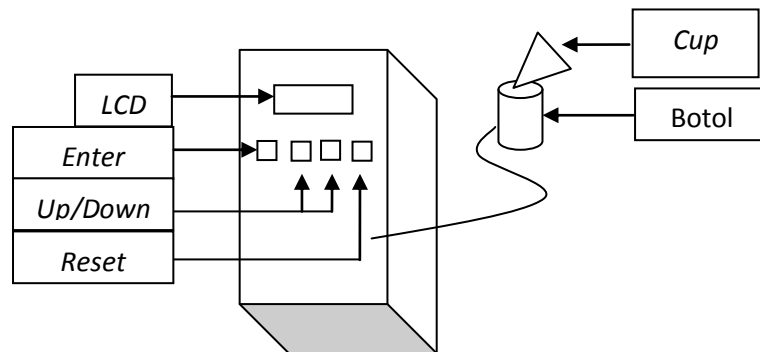
Gambar 3.1 Blok Diagram.

Tombol *Up/Down* digunakan untuk pemilihan *timer*. Tombol *Enter* digunakan untuk mengeksekusi program yang telah dipilih melalui tombol *up/down*. *Reset* digunakan untuk membalikan program ke awal, mulai dari inisialisasi *LCD*. *LCD* sebagai tampilan dan *buzzer* sebagai indikator. Mikrokontroler akan mengeluarkan logika 0 atau 1 dan mengumpulkannya pada *driver* dan *buzzer*. Logika 0 dan 1 dari

mikrokontroler mengatur aktif dan tidaknya *driver*. Ketika *Driver* aktif maka motor bekerja. Ketika motor bekerja, proses *cupping*/penghisapan dimulai.

3.1.2 Diagram Mekanis Sistem

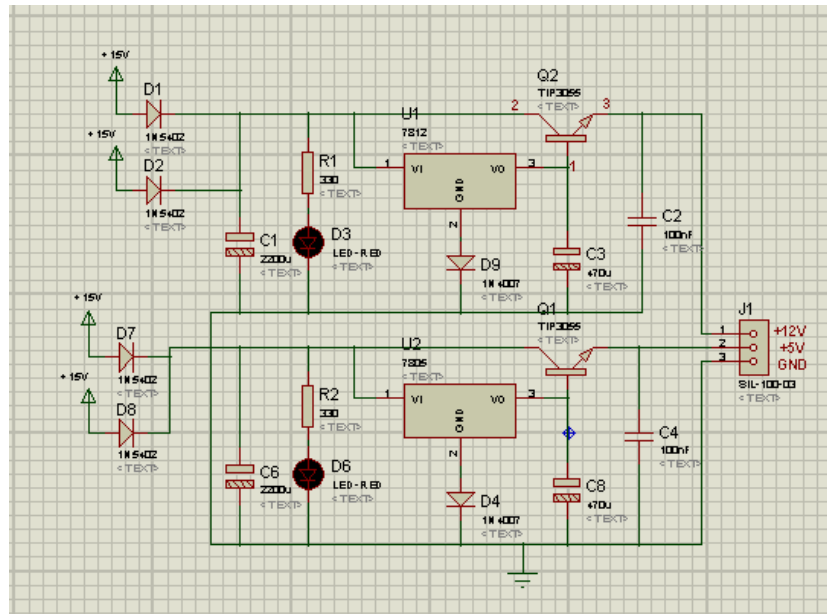
Diagram mekanis sistem dari pengembangan *breastpump* elektrik berbasis mikrokontroler *ATMega8535* dilengkapi dengan pengatur waktu dan tekanan dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram Mekanis Sistem.

3.1.3 Rangkaian *Power Supply*

Rangkaian *power supply* ialah sumber tegangan AC yang akan diubah menjadi tegangan DC kemudian turun menjadi keluaran 9 Volt dan 5 Volt. Skematik rangkaian *power supply* dapat dilihat di Gambar 3.3 dan *layout Power Supply* dapat dilihat di Gambar 3.4.



Gambar 3.3 Skematik Rangkaian *Power Supply*.

Perhitungan untuk pemilihan nilai *resistor* :

$$V_{in} = V_R + V_D$$

$$15 \text{ V} = I_x R + 1,5$$

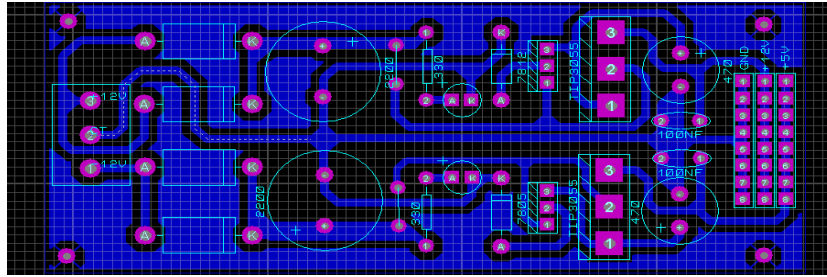
$$15 \text{ V} = 20 \text{ mA} \times R$$

$$15 \text{ V} - 1,5 = 20 \text{ mA} \times R$$

$$13,5 \text{ V} = 200 \text{ mA} \times R$$

$$R = \frac{13,5 \text{ V}}{200 \text{ mA}}$$

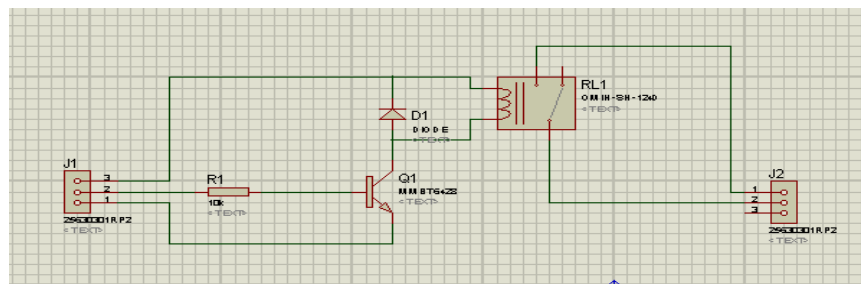
$$R = 675 \text{ Ohm}$$



Gambar 3.4 Layout *Power Supply*.

3.1.4 Rangkaian *Driver*

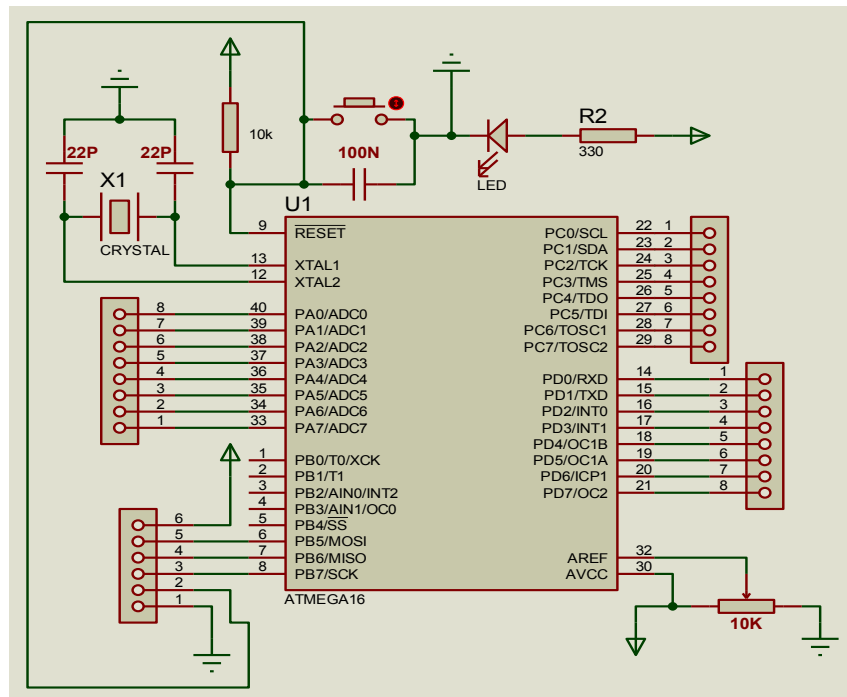
Rangkaian *driver* sebagai pengontak dari perintah sistem ke motor apabila diberi logika 1 maka *driver* akan bekerja dan *relay* mengontak motor untuk bekerja. Skematik dari rangkaian *driver* dapat dilihat di Gambar 3.5.



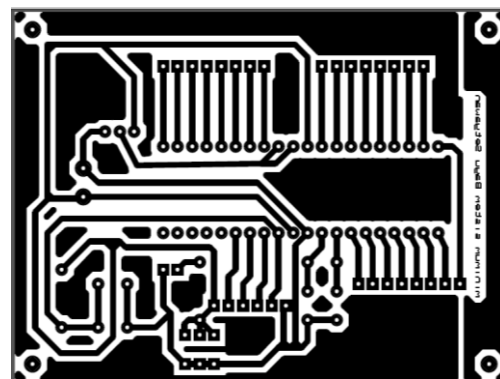
Gambar 3.5 Skematik Rangkaian *Driver*

3.1.5 Rangkaian Minimum Sistem

Minimum sistem digunakan sebagai pengatur atau otak dari alat dan pengatur *driver* motor. Skematik dari rangkaian minimum sistem dapat dilihat di Gambar 3.6 dan *layout* dari minimum sistem dapat dilihat di Gambar 3.7.



Gambar 3.6 Skematik Rangkaian Minimum Sistem.

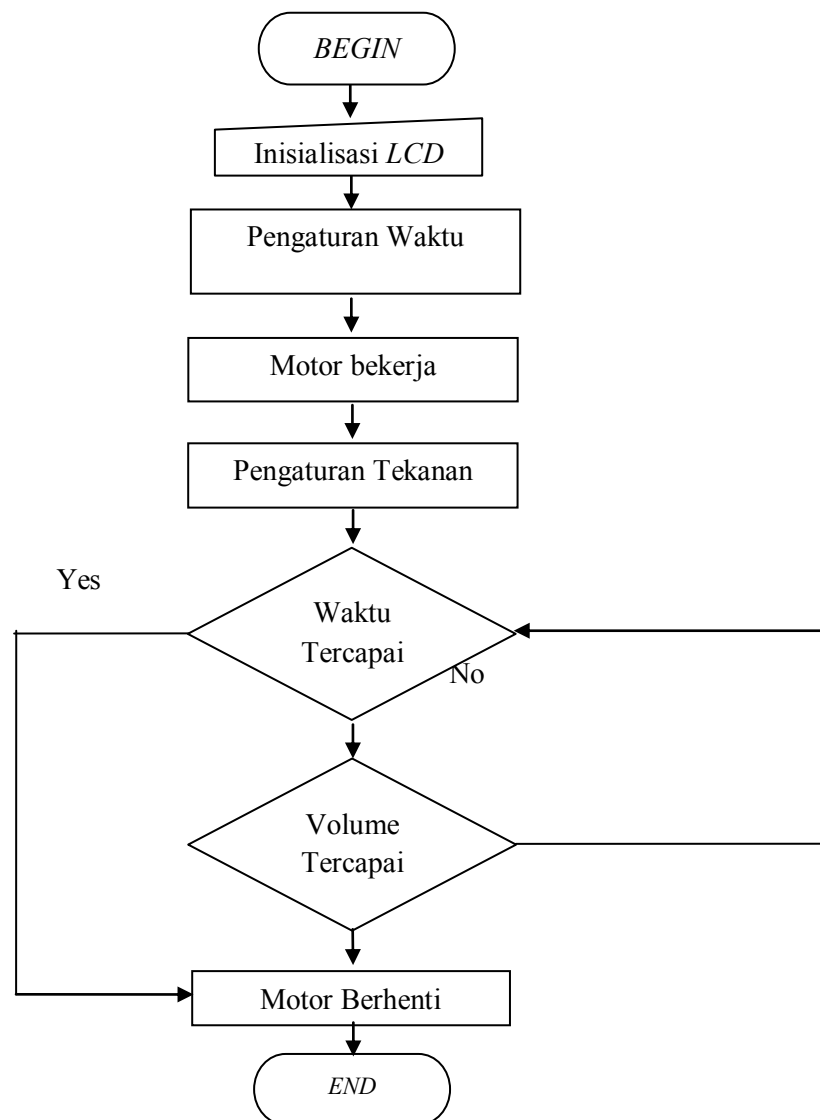


Gambar 3.7 Layout Minimum Sistem

3.2 Perancangan Perangkat Lunak

3.2.1 Diagram Alir

Diagram alir dari *breastpump* elektrik dapat dilihat di Gambar 3.8.



Gambar 3.8 Diagram Alir.

Pertama kita memilih waktu yang akan digunakan (1-15 menit), selanjutnya tekan *enter*, kemudian motor akan bekerja/memulai *pemvakuman*, tekanan mulai bekerja. Setelah itu kita dapat secara manual mengatur tekanan yang diinginkan dengan cara memutar knop, jika waktu

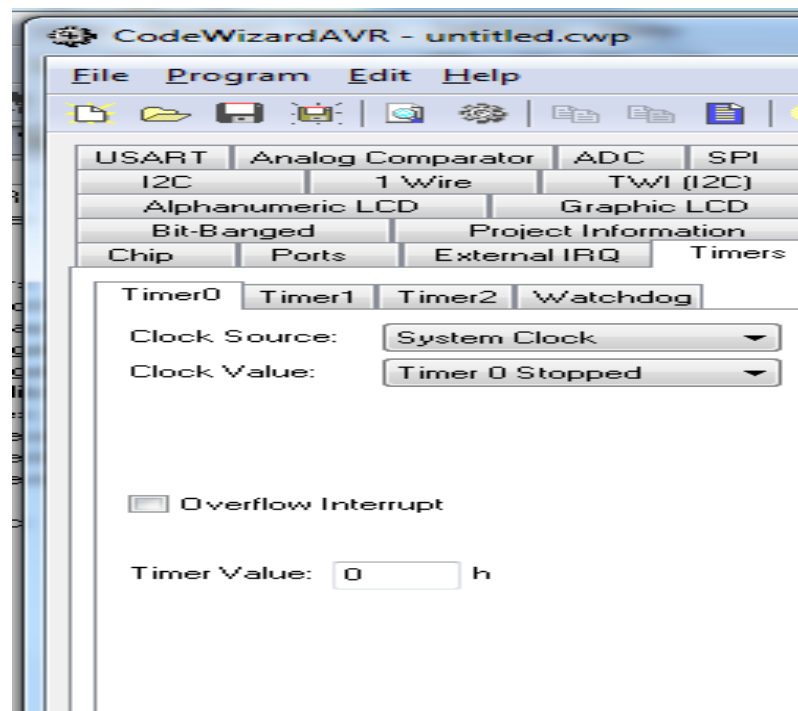
sudah tercapai tetapi volume belum tercapai maka motor akan berhenti, jika waktu belum tercapai dan volume sudah tercapai motor akan berhenti, jika waktu dan volume sudah tercapai, motor akan berhenti.

3.2.2 Program

Untuk pembuatan program pada modul ini menggunakan aplikasi AVR dengan bahasa C. Program yang digunakan ialah program ADC sebagai pengendali *driver* dan *timer* sebagai pengontrol waktunya.

Berikut langkah-langkah *setting timer* :

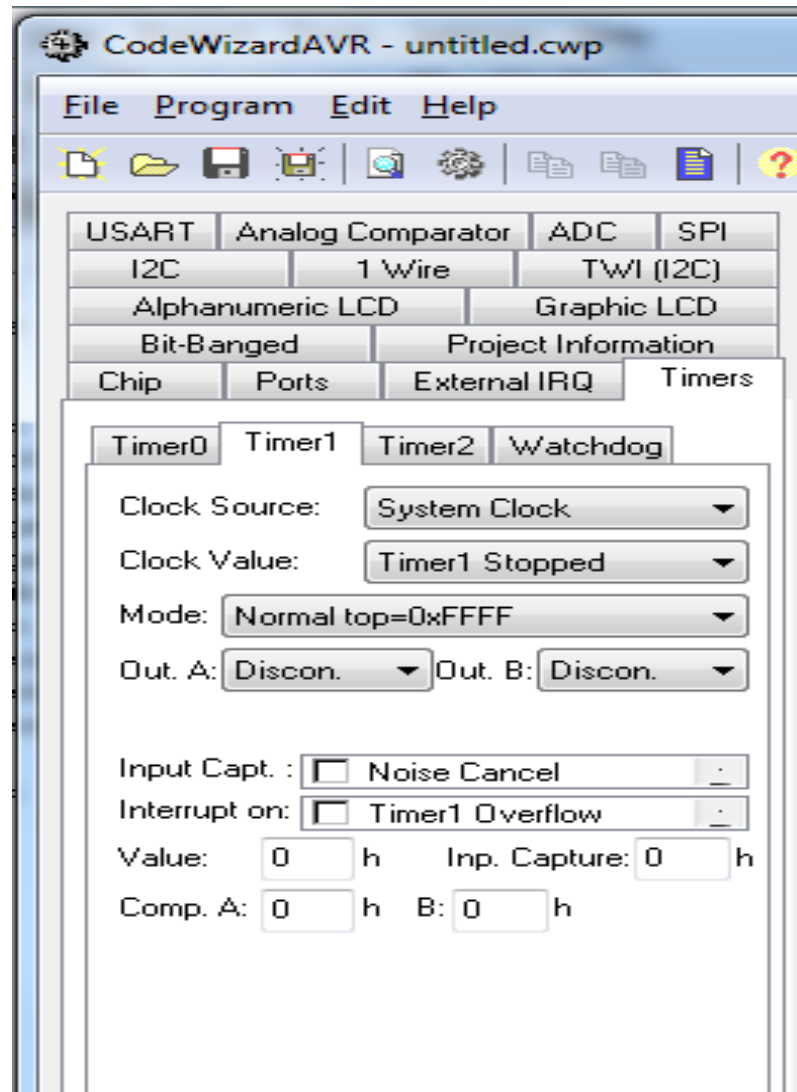
1. Memilih *timer* yang digunakan dalam pengaturan *timer* dapat dilihat pada Gambar 3.9.



Gambar 3.9 Pengaturan *Timer*.

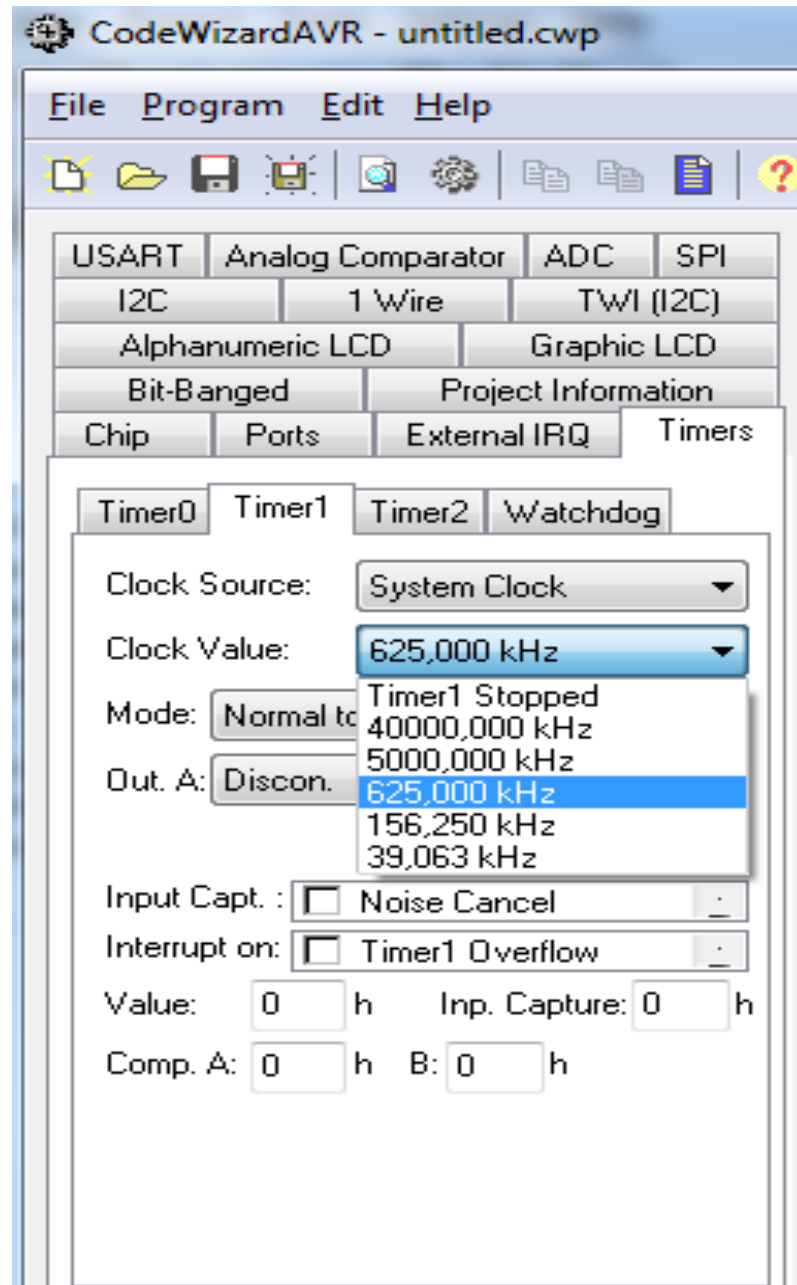
2. Setelah memilih *timer*, kemudian atur *timer* sesuai yang dipilih.

Pemilihan *timer* dapat dilihat pada Gambar 3.10.



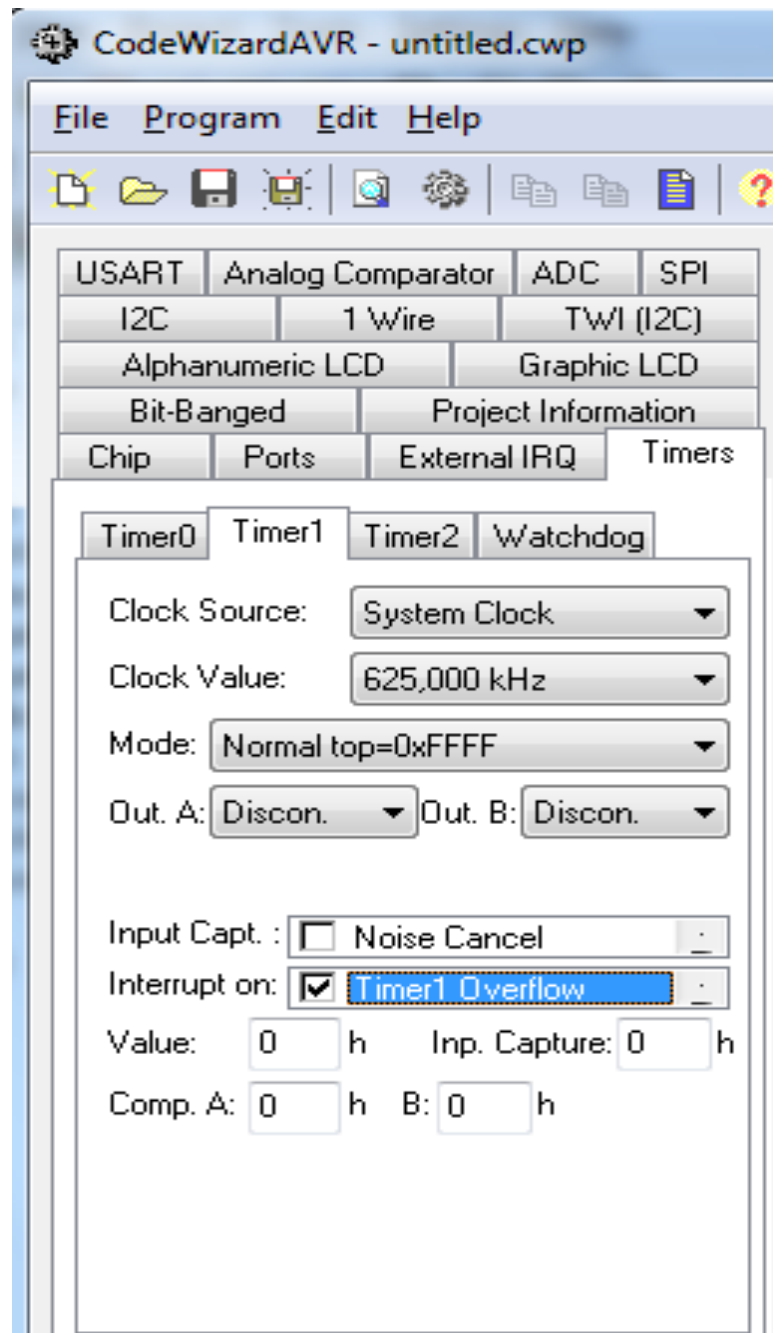
Gambar 3.10 Pemilihan *Timer*.

3. Kemudian atur *clock value* untuk mengaktifkan *interrupt overflow timer1*. *Setting clock value* dapat dilihat pada Gambar 3.11.



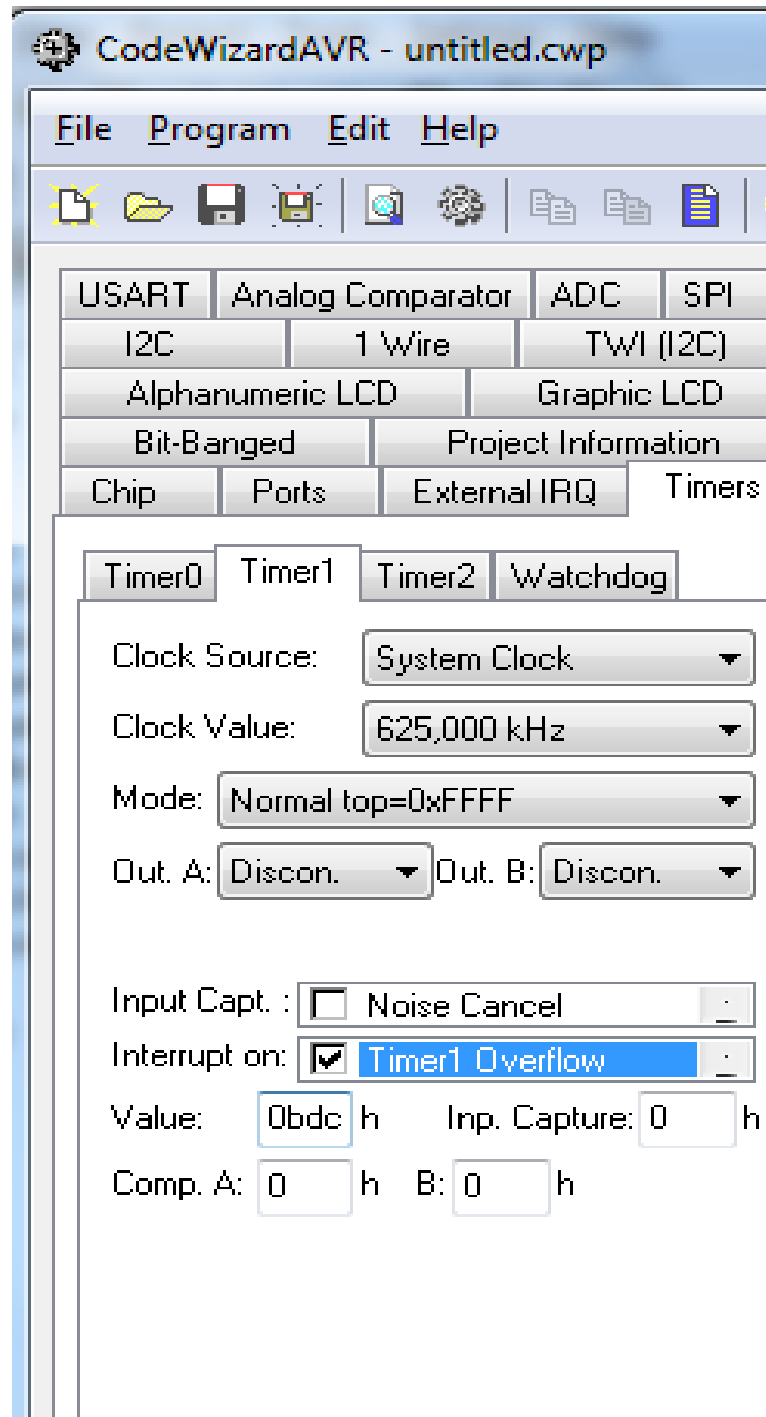
Gambar 3.11 *Setting Clock Value.*

4. Kemudian beri tanda centang pada *interrupt on*, hal ini berfungsi agar ada dua program yang berjalan. *Setting interrupt on* bisa dilihat pada Gambar 3.12.



Gambar 3.12 *Setting Interrupt On.*

5. Kemudian ubah value dengan *setting* 0bdc. *Setting value* dapat dilihat di Gambar 3.13.



Gambar 3.13 *Setting Value.*

Listing program timer diperlihatkan pada *Listing 3.1*.

```

void setting_timer()
{
    if(b==0)
    {
        if(PINB.2==0)
        {
            menit++;delay_ms(500);lcd_clear();
        }
        else if(menit>15)
        {
            menit=0;
        }
        if(PINB.3==0)
        {
            menit--;delay_ms(500);lcd_clear();
        }
        else if(menit<0)
        {
            menit=0;
        }
    }
}

void mulai_timer()
{
    if(timer_aktif==1)
    {
        TCCR1B=0x04;PORTD.2=0;
    }else
    if(timer_aktif==0){TCCR1B=0x00;PORTD.2=1;}}

void stop_timer()
{
    if(menit==0&&detik==0&&c==1)
    {
        timer_aktif=0;
        PORTD.1=1;
    }
}

// Declare your global variables here

void main(void)

```

Listing 3.1 Program *Timer*.

Listing program timer ini digunakan sebagai pengatuan *timer* waktu saat sistem bekerja, waktu yang diatur dalam *listing* program ini yaitu 1 sampai 15 menit dengan metode *counter down*.

Listing Program ADC diperlihatkan pada Listing 3.2.

```

#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 3.2 Program ADC.

Listing ADC (*Analog To Digital Converter*) adalah perangkat elektronika yang berfungsi untuk mengubah sinyal analog (sinyal kontinu) menjadi sinyal digital. Perangkat ADC (*Analog To Digital Conversion*) dapat berbentuk suatu modul atau rangkaian elektronika maupun suatu chip IC. ADC (*Analog To Digital Converter*) berfungsi untuk menjembatani pemrosesan sinyal analog oleh sistem digital (Iswanto, 2008).

3.3 Perancangan Pengujian

3.3.1 Jenis Pengujian

1. Mengukur tegangan menggunakan alat ukur multimeter pada tekanan *breastpump* bertujuan untuk perbandingan.
2. Uji tekanan dengan praktek langsung pada ibu menyusui.
3. Kalibrasi tekanan *breastpump* dengan Dpm.
4. Mengukur waktu dengan menggunakan *stopwatch*.

3.3.2 Pengolahan Data

Jenis penelitian ini menggunakan metode *Pre Eksperimental* dengan jenis “*One group Post Test Design*” yaitu alat *breastpump* ini bekerja dengan tekanan dan *timer* yang di atur kemudian motor akan berhenti apabila tekanan tercapai/waktu telah tercapai kemudian proses selesai. Sehingga penulis hanya melihat hasil tanpa mengukur keadaan sebelumnya.

Desain dapat digambarkan sbb :

X \longrightarrow O

X = *treatment*/perlakuan yang diberikan (variabel independen)

O = Observasi (variabel dependen)

Variabel Penelitian

1. Variabel Bebas

Sebagai variabel bebas yaitu daya *vakum*/tekanan.

2. Variabel Tergantung

Sebagai variabel tergantung yaitu pengontrol untuk tekanan (knop).

3. Variabel Terkendali

Variabel terkendali terdiri dari tampilan tekanan dan waktu yang dikendalikan oleh Mikrokontroler *ATMega8535*.

3.3.3 Sistematika Pengukuran

1. Rata-Rata Pengukuran

Adalah nilai atau hasil pembagian dari jumlah data yang diambil atau diukur dengan banyaknya pengambilan data atau banyaknya pengukuran. Rata-rata pengukuran dirumuskan sebagai berikut :

$$\bar{x} = \frac{\sum x_n}{n} \quad (3-1)$$

dengan :

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

$$\sum x_n = \text{Jumlah } x \text{ sebanyak } n$$

$$n = \text{Banyak data}$$

2. Simpangan (*Error*)

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

$$\text{simpangan} = x_n - \bar{x} \quad (3-2)$$

dengan :

simpangan = Nilai error yang dihasilkan

x_n = Rata – rata data DPM

\bar{x} = Rata – rata data modul

3. *Persentase Error*

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

$$\text{Persentase Error} = \frac{\text{simpangan}}{x_n} \times 100\% \quad (3-3)$$

dengan :

Persentase Error = Besarnya simpangan/nilai error dalam %

x_n = Rata-rata data kalibrator

4. *Standard Deviasi (SD)*

Adalah suatu nilai yang menunjukkan tingkat v(derajat) variasi kelompok data atau ukuran *standard* penyimpanan dari rata-ratanya. Jika *standard* deviasi semakin kecil maka data

tersebut semakin presisi. *Standard deviasi* dirumuskan sebagai berikut :

$$SD = \frac{\sqrt{\sum(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \dots + (x_n - \bar{x})^2}}{n - 1} \quad (3-4)$$

dengan :

SD = Standar deviasi

x = Data x

\bar{x} = Rata-rata

n =Banyak data