

Lampiran

1. Listing Program

```
Chip type           : ATmega8
Program type        : Application
AVR Core Clock frequency: 16.000000 MHz
Memory model        : Small
External RAM size   : 0
Data Stack size     : 256
*****/

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

// Alphanumeric LCD Module functions
#include <alcd.h>

#define ADC_VREF_TYPE 0x40

#define sensor1 0
#define sensor2 1
#define sensor3 2

#define tombola PINB.0
#define tombolb PINB.1
#define tombolc PINB.2

#define motor OCR2

#define buzzer PORTD.7
#define relay PORTD.6

char buff[33];
int mapvolume[]={0,80,150,240};
int vol=1,pwm=0,interval=1000;
int level=0,str;

// 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;
}

void nyeting()
{
    int index=0;
    int x=0;
    lcd_clear();
    delay_ms(200);
```

```

while(1)
{
if(tombolc==0) index++;
if(index==0)
{
lcd_clear();
lcd_gotoxy(0,0);
sprintf(buff,"Volume:%dml ",mapvolume[vol]);
lcd_puts(buff);

if(tombola==0) vol++; // naik turunkan angka
if(tombolb==0) vol--;
if(vol>3)vol=1; // batasi angka
if(vol<1)vol=3;

}
if(index==1)
{
lcd_clear();
lcd_gotoxy(0,0);
sprintf(buff,"Daya Hisap ",pwm);
lcd_puts(buff);

if(tombola==0) x++;
if(tombolb==0) x--;

if(x>2)x=0;
if(x<0)x=2;

if(x==0)pwm=100;
if(x==1)pwm=180;
if(x==2)pwm=255;

if (x==0)
{
lcd_gotoxy(0,1);
lcd_putsf("Low");
}
if (x==1)
{
lcd_gotoxy(0,1);
lcd_putsf("Medium") ;
}
if (x==2)
{
lcd_gotoxy(0,1);
lcd_putsf("High") ;
}
}

if(index==2)
{
lcd_clear();
lcd_gotoxy(0,0);
sprintf(buff,"Interval:%dms ",interval);

```

```

    lcd_puts(buff);

    if(tombola==0) interval=interval+100;
    if(tombolb==0) interval=interval-100;
    if(interval>2000)interval=500;
    if(interval<500)interval=2000;

}

if(index>2) break; // keluar dari menu

delay_ms(150); // interval siklus program

}
lcd_clear();
delay_ms(200);

}

// program yang paling utama
void go_run()
{
    int level1=0,level2=0,level3=0; // dari nilai adc akan menjadi
logika 1 / 0 dengan memberi nilai parameter sesuai kepekaan
sensor level volume

    if( read_adc(sensor1) < 500 ) level1=1;
    else level1=0;

    if( read_adc(sensor2) < 500 ) level2=1;
    else level2=0;

    if( read_adc(sensor3) < 500 ) level3=1;
    else level3=0;

// menjumlahkan pembacaan sensor
level= level1 + level2 + level3;

    if(tombolc==0) nyeting(); // masuk setingan
    if(tombola==0) str=1; // start algoritma

// kondisi saat start
    if(str==1)
    {
// jika mode vol = 0 ( 80ml )
        if(vol==1)
        {
// sensor level kena susu
            if(level1==1)
            {
                buzzer=1; // buzer on
                relay=0; // selenoid off
                pwm=0; // motor off
                lcd_clear();
                lcd_gotoxy(0,0);
            }
        }
    }
}

```

```

        lcd_puts("Asi Penuh !");
        delay_ms(3000); // jeda 3 detik
        buzzer=0; // buzzer off
        while(1);
    }
}

if(vol==2)
{
    if(level1==1&&level2==1)
    {
        buzzer=1;
        relay=0;
        pwm=0;
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_puts("Asi Penuh !");
        delay_ms(3000);
        buzzer=0;
        while(1);
    }
}

if(vol==3)
{
    if(level1==1&&level2==1&&level3==1)
    {
        buzzer=1;
        relay=0;
        pwm=0;
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_puts("Asi Penuh !");
        delay_ms(3000);
        buzzer=0;
        while(1);
    }
}

}

lcd_clear();
lcd_gotoxy(0,0);
sprintf(buff,"Volume:%dml",mapvolume[vol]);
lcd_puts(buff);
lcd_gotoxy(0,1);
sprintf(buff,"Vol:%dml",mapvolume[level]);
lcd_puts(buff);
lcd_gotoxy(11,1);
if(str==1)lcd_putsf("Start"); // display saat start
else lcd_putsf("Stop");
delay_ms(100); // interval siklus program

if(str==1)
{ // saat start ( motor dc dan selenoid aktif )
    relay=1;
}

```

```

    motor=pwm;
    delay_ms(interval); // jeda
    relay=0;
    motor=0;
    delay_ms(interval/2); // ambil nili interval setengah saja
}

}

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=Out Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=0 State2=P
State1=P State0=P
PORTB=0x07;
DDRB=0x08;

// Port C initialization
// Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=Out Func6=Out Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=0 State6=0 State5=T State4=T State3=T State2=T
State1=T State0=T
PORTD=0x00;
DDRD=0xC0;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
TCCR0=0x00;
TCNT0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off

```

```

TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: 250.000 kHz
// Mode: Phase correct PWM top=0xFF
// OC2 output: Non-Inverted PWM
ASSR=0x00;
TCCR2=0x64;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
MCUCR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// USART disabled
UCSRB=0x00;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC Clock frequency: 1000.000 kHz
// ADC Voltage Reference: AVCC pin
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

// Alphanumeric LCD initialization
// Connections specified in the

```

```

// Project|Configure|C Compiler|Libraries|Alphanumeric LCD
menu:
// RS - PORTD Bit 0
// RD - PORTB Bit 6
// EN - PORTD Bit 1
// D4 - PORTD Bit 2
// D5 - PORTD Bit 3
// D6 - PORTD Bit 4
// D7 - PORTD Bit 5
// Characters/line: 16
lcd_init(16);

lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("POMPA ASI");
lcd_gotoxy(0,1);
lcd_putsf("ELEKTRIK");
delay_ms(1000);
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("Novia wandasari");
lcd_gotoxy(0,1);
lcd_putsf("20153010002");
delay_ms(1000);

while (1)
{
// Place your code here
go_run();
}
}

```

2. Gambar Alat

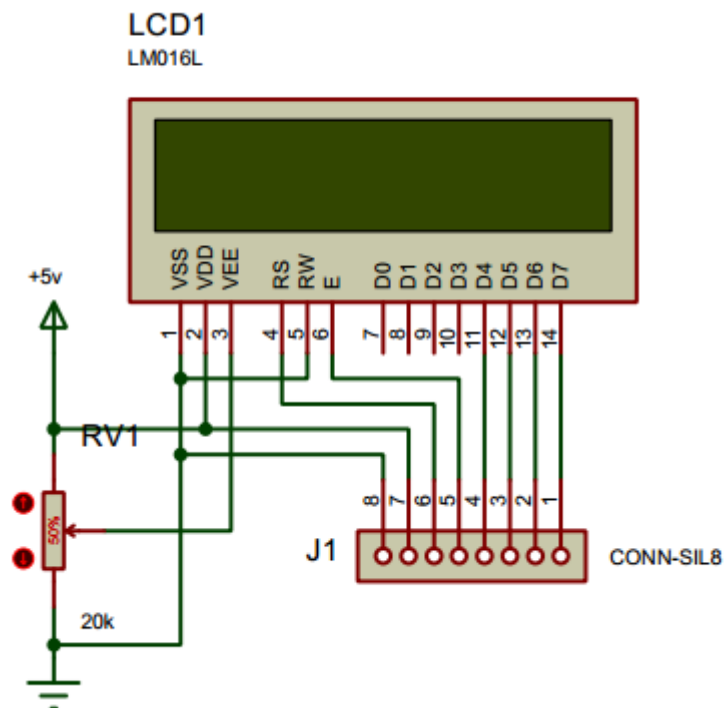


3. Standar Operasional Prosedure Alat

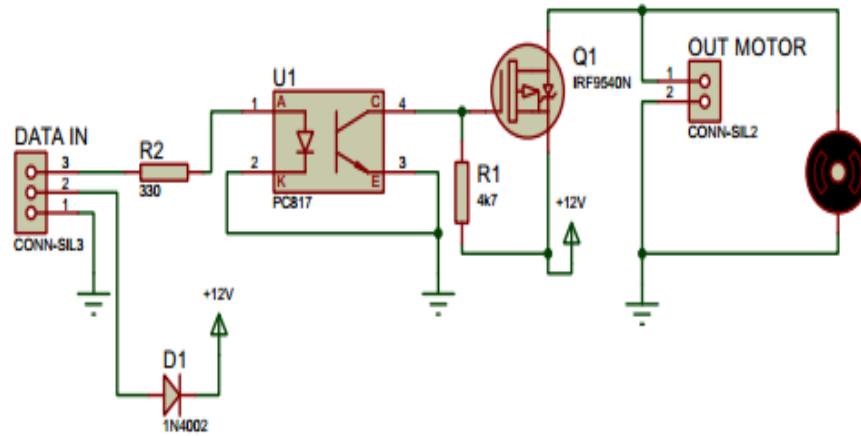
- a. Menekan tombol *on/off* pada alat
- b. Memasukan botol susu pada wadah sensor volume
- c. Memasang cup terhadap payudara
- d. Menekan tombol menu atur nilai volume dengan tombol *up/down*
- e. Menekan lagi tombol menu atur nilai daya hisap dengan tombol *up/down*
- f. Menekan lagi tombol menu atur nilai interval dengan tombol *up/down*
- g. Menekan tombol menu untuk kembali ke tampilan menu, tekan *start*
- h. Ketika ASI sudah terdeteksi dan alat mati menekan tombol *reset*
- i. Menekan tombol *on/off* untuk mematikan alat
- j. Membersihkan dan rapikan kembali alat

4. Rangkaian Pada Alat

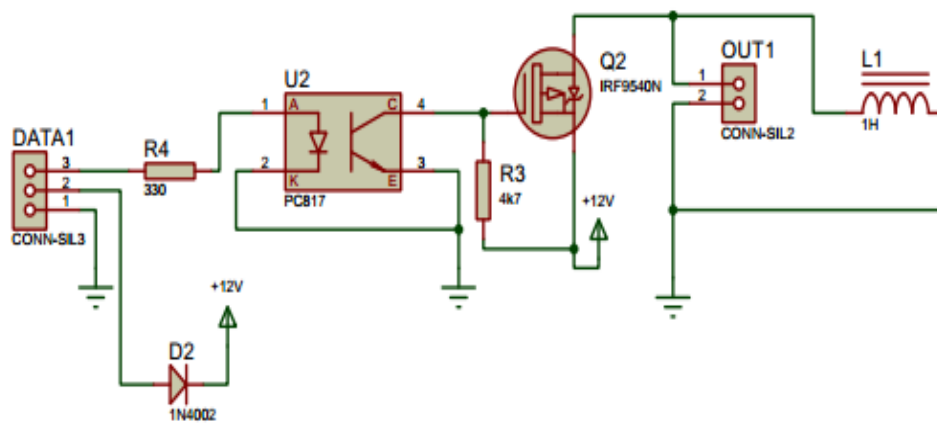
➤ Rangkaian LCD



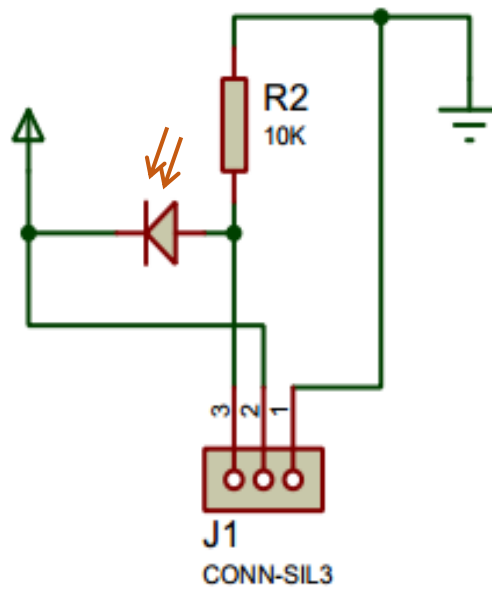
➤ **Rangkaian Driver Motor**



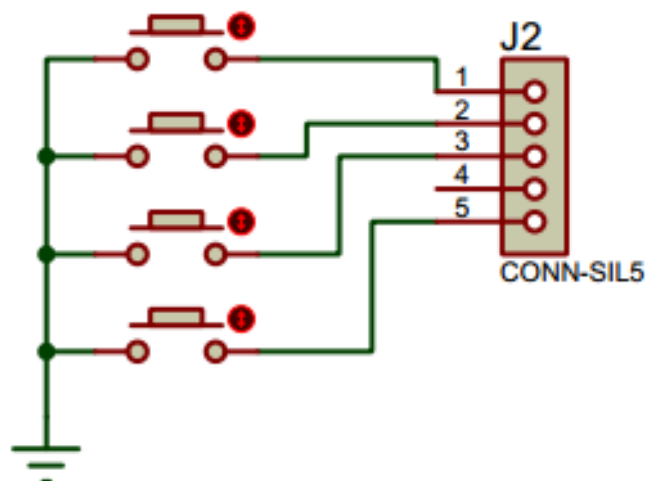
➤ **Rangkaian Driver Selenoid**



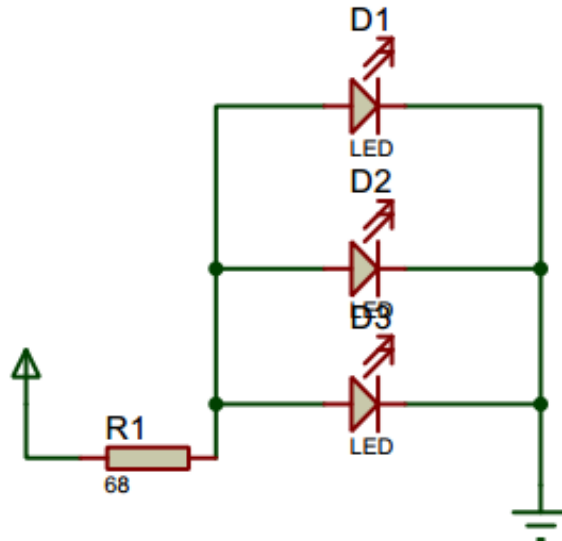
➤ **Rangkaian Photodioda**



➤ **Rangkaian Push Button**



➤ **Rangkaian LED**



➤ **Rangkaian Minsis**

