

## LAMPIRAN

```
#include <mega16.h> //menambahkan library atmega16
#include <delay.h> //menambahkan library delay
#define ADC_VREF_TYPE 0x40

// Fungsi untuk mengaktifkan dan membaca nilai adc
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;
}

//Deklarasi variable global
float SUHU2,SUHU,SUHU1, hum, sum;
float data, data2, dataf, data1, datahum;
int pul,sat,koma,pulsk,satsk,komask,pulhum,sathum,ubah;

void ruang(); void skin(); void heater(); void humidity();
void motor(); void motor_mati();

//fungsi untuk mengubah kedalam format 7segment
void ubah_ke_format7segment()
{
    if (ubah==0){ubah=0xc0;}
    if (ubah==1){ubah=0xf9;}
    if (ubah==2){ubah=0xa4;}
    if (ubah==3){ubah=0xb0;}
    if (ubah==4){ubah=0x99;}
    if (ubah==5){ubah=0x92;}
    if (ubah==6){ubah=0x82;}
    if (ubah==7){ubah=0xf8;}
    if (ubah==8){ubah=0x80;}
    if (ubah==9){ubah=0x90;}
}
```

```

//fungsi untuk mengatur tampilan seven segment
void tampil_7segment()
{
    PORTC=pul; //menampilkan nilai puluhan pada PORTC
    PORTD.5=0;PORTD.6=0;PORTD.7=0; //posisi seven segment y0
    delay_us(300); //waktu tunda 300 mikro detik

    PORTC=sat; //menampilkan nilai satuan pada PORTC
    PORTD.5=1;PORTD.6=0;PORTD.7=0; //posisi seven segment y1
    delay_us(300); //waktu tunda 300 mikro detik

    PORTC=koma; //menampilkan nilai koma pada PORTC
    PORTD.5=0;PORTD.6=1;PORTD.7=0; //posisi seven segment y2
    delay_us(300); //waktu tunda 300 mikro detik

    PORTC=pulsk; //menampilkan nilai puluhan pada PORTC
    PORTD.5=1;PORTD.6=1;PORTD.7=0; //posisi seven segment y3
    delay_us(300); //waktu tunda 300 mikro detik

    PORTC=satsk; //menampilkan nilai satuan pada PORTC
    PORTD.5=0;PORTD.6=0;PORTD.7=1; //posisi seven segment y4
    delay_us(300); //waktu tunda 300 mikro detik

    PORTC=komask; //menampilkan nilai koma pada PORTC
    PORTD.5=1;PORTD.6=0;PORTD.7=1; //posisi seven segment y5
    delay_us(300); //waktu tunda 300 mikro detik

    PORTC=pulhum; //menampilkan nilai puluhan pada PORTC
    PORTD.5=0;PORTD.6=1;PORTD.7=1; //posisi seven segment y6
    delay_us(300); //waktu tunda 300 mikro detik

    PORTC=sathum; //menampilkan nilai satuan pada PORTC
    PORTD.5=1;PORTD.6=1;PORTD.7=1; //posisi seven segment y7
    delay_us(300); //waktu tunda 300 mikro detik
}

void motor_mati() //Funfsi untuk mematikan motor
{
    PORTD.1=0; //Inisiasi PORTD.1 dengan nilai 0
    PORTD.2=0; //Inisiasi PORTD.2 dengan nilai 0
    OCR1B=0;
}

void motor() //Fungsi untuk menggerakkan motor
{
    humidity(); //Memanggil fungsi humidity()
    if(sum>70) //Jika nilai humidity lebih dari 70
    {
        PORTD.1=1; //Inisiasi PORTD.1 dengan nilai 1
        PORTD.2=0; //Inisiasi PORTD.2 dengan nilai 0
        OCR1B=250;
    }
}

```

```

    if(PINB.6==0) //Jika PINB.6 mendapatkan logika 0
    {
        motor_mati(); //Menjalankan fungsi motor_mati()
    }
    else if (sum<=60) //Jika nilai humidity kurang dari 60
    {
        PORTD.1=0; //Inisiasi PORTD.1 dengan nilai 0
        PORTD.2=1; //Inisiasi PORTD.2 dengan nilai 1
        OCR1B=250;
    }
    if(PINB.7==0) Jika PINB.7 mendapatkan logika 0
    {
        motor_mati(); //menjalankan fungsi motor_mati
    }
}

void humidity() //fungsi pembacaan sensor RH
{
    datahum=read_adc(3); //membaca nilai dari adc 3
    hum=(float)datahum*5/1023; //merubah nilai adc ke RH
    sum=(hum*33)-15;

    pulhum=(unsigned int) sum/10; //mengambil nilai puluhan
    sathum=(unsigned int) sum%10; //mengambil nilai satuan

    ubah=pulhum;ubah_ke_format7segment();pulhum=ubah;
    ubah=sathum;ubah_ke_format7segment();sathum=ubah;
    tampil_7segment();
}

void ruang() //fungsi pembacaan sensor suhu
{
    SUHU = read_adc(0); //membaca nilai adc 0
    SUHU1 = read_adc(1); //membaca nilai adc 1
    SUHUR = (SUHU+SUHU1)/2; //mengambil rata-rata nilai adc
    suhu_celcius=(float)SUHUR*500/1023; //rumus menghitung suhu
    dataf=suhu_celcius; //inisiasi nilai suhu

    pul=(unsigned int) dataf/10; //mengambil nilai puluhan
    sat=(unsigned int) dataf%10; mengambil nilai satuan
    koma=(unsigned int) (dataf*10)%10; //mengambil nilai koma

    //merubah ke format seven segment
    ubah=koma;ubah_ke_format7segment();koma=ubah;
    ubah=sat;ubah_ke_format7segment();sat=ubah;
    ubah=pul;ubah_ke_format7segment();pul=ubah;
    tampil_7segment();
}

```

```

        {
            PORTD.0=0;
        }
        else
        {
            PORTD.0=1;
        }
    }
    //setting batas suhu pada 35 derajat
    else if (PINB.3==0)
    {
        if (dataf<35)
        {
            PORTD.0=0;
        }
        else
        {
            PORTD.0=1;
        }
    }
    //setting batas suhu pada 36 derajat
    else if (PINB.4==0)
    {
        if (dataf<36)
        {
            PORTD.0=0;
        }
        else
        {
            PORTD.0=1;
        }
    }
    //setting batas suhu pada 37 derajat
    else if (PINB.5==0)
    {
        if (dataf<37)
        {
            PORTD.0=0;
        }
        else
        {
            PORTD.0=1;
        }
    }
}

void main(void)
{

```

```

State1=T State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=P State6=P State5=P State4=P State3=P State2=P
State1=P State0=P
PORTB=0xFF;
DDRB=0x00;

// Port C initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out
Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0
State1=0 State0=0
PORTC=0x00;
DDRC=0xFF;

// Port D initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=In Func2=Out
Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=T State2=0
State1=0 State0=0
PORTD=0x00;
DDRD=0xF7;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 1000,000 kHz
// Mode: Fast PWM top=0x00FF
// OC1A output: Discon.
// OC1B output: Non-Inv.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off

```

```
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=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: 500,000 kHz
// ADC Voltage Reference: AVCC pin
// ADC Auto Trigger Source: Free Running
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0xA1;
SFIOR&=0x1F;
```

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

while(1) //looping
{
    humidity();
    motor();
    ruang();
    heater();
    skin();
}
}
```

