

LAMPIRAN

Listing Program

```
#include <mega16.h>
#include <delay.h>
#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;
}

float      SUHU2, SUHU, SUHU1,      hum,      sum, suhu_celcius,
suhu_celcius1, SUHUR;
float data, data2, dataf, data1=0, datahum;
int pul, sat, koma, pulsk, satsk, komask, pulhum, sathum, ubah;

void ruang(); void skin(); void heater(); void humidity();
void motor(); void motor_mati();
void ubah_ke_format7segment()//fungsi untuk mengubah kedalam
format 7segment
{
    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;}
}
void tampil_7segment()
{
    PORTC=pul;
    PORTD.5=0;PORTD.6=0;PORTD.7=0;
    delay us(300);
}
```

```
    PORTC=sat;
    PORTD.5=1;PORTD.6=0;PORTD.7=0;
    delay_us(300);

    PORTC=koma;
    PORTD.5=0;PORTD.6=1;PORTD.7=0;
    delay_us(300);

    PORTC=pulsk;
    PORTD.5=1;PORTD.6=1;PORTD.7=0;
    delay_us(300);

    PORTC=satsk;
    PORTD.5=0;PORTD.6=0;PORTD.7=1;
    delay_us(300);

    PORTC=komask;
    PORTD.5=1;PORTD.6=0;PORTD.7=1;
    delay_us(300);

    PORTC=pulhum;
    PORTD.5=0;PORTD.6=1;PORTD.7=1;
    delay_us(300);

    PORTC=sathum;
    PORTD.5=1;PORTD.6=1;PORTD.7=1;
    delay_us(300);
}

void motor_mati()
{
    PORTD.1=0;
    PORTD.2=0;
    OCR1B=0;
}

void motor()
{
    humidity();
    if(sum>60)
    {
        PORTD.1=1;
        PORTD.2=0;
        OCR1B=250;
    }
}
```

```

if(PINB.6==0)
{
    motor_mati();
}

else if (sum<=50)
{
    PORTD.1=0;
    PORTD.2=1;
    OCR1B=250;
}
if(PINB.7==0)
{
    motor_mati();
}
}

void humidity()
{

    datahum=read_adc(3);
    hum=(float) datahum*5/1023;
    sum=(hum*33)-15;

    pulhum=(unsigned int) sum/10;
    sathum=(unsigned int) sum%10;

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

void ruang()
{
    SUHU = read_adc(0);
    SUHU1 = read_adc(1);
    SUHUR=(SUHU+SUHU1)/2;
    suhu_celcius = (float)SUHUR*500/1023;
    dataf=suhu_celcius;
    pul=(unsigned int) dataf/10;
    sat=(unsigned int) dataf%10;
    koma=(unsigned int) (dataf*10)%10;
    ubah=koma;ubah_ke_format7segment();koma=ubah;
    ubah=sat;ubah_ke_format7segment();sat=ubah;
    ubah=pul;ubah_ke_format7segment();pul=ubah;
    tampil_7segment();
}

```

```
void skin()
{
    SUHU2 = read_adc(2);
    suhu_celcius1 = (float)SUHU2*500/1023;
    data1=suhu_celcius1;
    pulsk=(unsigned int) data1/10;
    satsk=(unsigned int) data1%10;
    komask=(unsigned int) (data1*10)%10;

    ubah=komask;ubah_ke_format7segment();komask=ubah;
    ubah=satsk;ubah_ke_format7segment();satsk=ubah;
    ubah=pulsk;ubah_ke_format7segment();pulsk=ubah;
    tampil_7segment();
}

void heater()
{
    ruang();
    DDRB=0x00; PORTB=0xff;

    if(PINB.0==0)
    {
        if(dataf<32)
        {
            PORTD.0=0;
        }
        else
        {
            PORTD.0=1;
        }
    }
    else if(PINB.1==0)
    {
        if(dataf<33)
        {
            PORTD.0=0;
        }
        else
        {
            PORTD.0=1;
        }
    }
}
```

```
else if(PINB.2==0)
{
    if(dataf<34)
    {
        PORTD.0=0;
    }
    else
    {
        PORTD.0=1;
    }
}
else if(PINB.3==0)
{
    if(dataf<35)
    {
        PORTD.0=0;
    }
    else
    {
        PORTD.0=1;
    }
}
else if(PINB.4==0)
{
    if(dataf<36)
    {
        PORTD.0=0;
    }
    else
    {
        PORTD.0=1;
    }
}
else if(PINB.5==0)
{
    if(dataf<37)
    {
        PORTD.0=0;
    }
    else
    {
        PORTD.0=1;
    }
}
}
```

```

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

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T
Statel=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
Statel=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
Statel=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
Statel=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
// Compare B Match Interrupt: Off
TCCR1A=0x21;
TCCR1B=0x09;
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: 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;

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

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

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