LAMPIRAN A *Listing Program*

```
/*****************************************************

Chip type               : ATmega16
Program type            : Application
AVR Core Clock frequency: 8,000000 MHz
*****************************************************/
#include <mega16.h>
#include <stdio.h>
#include <delay.h>
#include <stdlib.h>

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

#define sound PORTD
#define sw    PINB.1
#define play  PORTB.0

unsigned char angka[10]={
0b00000000, //0
0b00110111, //1
0b01000010, //2
0b01100011, //3
0b01111010, //4
0b10010000, //5
0b10110001, //6
0b00010110, //7
0b00000001, //8
0b00100000,  //9
};
unsigned char tigapuluh=0b01011000;
unsigned char koma=0b11101011;
unsigned char derajatcelcius=0b11000111;
unsigned char suhuanda=0b00001011;

char buff[33];
unsigned int vin;
float suhu;
int timer=300;  // atur waktu detik
int detik=0;
int flag=0;

int depan_koma,depan_koma_satuan,belakang_koma;

int delay1=3000,delay2=3000,delay3=3000;
int delay4=3000,delay5=3000,delay6=3000;

// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Reinitialize Timer1 value
TCNT1H=0xE17B >> 8;
TCNT1L=0xE17B & 0xff;
// Place your code here
if(flag==1)detik++;
else detik=0;
}
```

```
#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;
}

// Declare your global variables here

void read_lm35dz(){
vin=read_adc(0);      // read analog
suhu=(float)vin*5/1023;     // convert analog to celcius
suhu=suhu*100;
}

void pemisah_angka(){
if(suhu>=30.0&&suhu<40){
depan_koma=(int)suhu/10;
depan_koma_satuan=(int)suhu%10;
belakang_koma=(int)(suhu*10)%10;
}}
```

```
void ngomong(){
sound=suhuanda;
delay_ms(100);
play=0;
delay_ms(500);
play=1;
delay_ms(delay1);
sound=0x00;

sound=tigapuluh;
delay_ms(100);
play=0;
delay_ms(500);
play=1;
delay_ms(delay2);
sound=0x00;
if(depan_koma_satuan>0){
sound=angka[depan_koma_satuan];
delay_ms(100);
play=0;
delay_ms(500);
play=1;
delay_ms(delay3);
sound=0x00;
}
if(belakang_koma>0){
sound=koma;
delay_ms(100);
play=0;
delay_ms(500);
play=1;
delay_ms(delay4);
sound=0x00;

sound=angka[belakang_koma];
delay_ms(100);
play=0;
delay_ms(500);
play=1;
delay_ms(delay5);
sound=0x00;
}
sound=derajatcelcius;
delay_ms(100);
play=0;
delay_ms(500);
play=1;
delay_ms(delay6);
sound=0x00;

}
void main(void)
{
// 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: 7,813 kHz
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: On
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x05;
TCNT1H=0xE1;
TCNT1L=0x7B;
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=0x04;


// 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=0x83;
// SPI initialization
// SPI disabled
SPCR=0x00;

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

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 7
// EN - PORTC Bit 1
// D4 - PORTC Bit 2
// D5 - PORTC Bit 3
// D6 - PORTC Bit 4
// D7 - PORTC Bit 5
// Characters/line: 16
lcd_init(16);
lcd_gotoxy(0,0);
lcd_putsf("TERMOMETER DIGITAL");
delay_ms(2000);
lcd_gotoxy(0,1);
lcd_putsf("Ready...");
delay_ms(2000);
lcd_clear();

// Global enable interrupts
```

```
#asm("sei")
while (1)
      // Place your code here
      while (1)
      {
      if(sw==0)flag=1;
      if(detik>=timer){
      flag=0;
      ngomong();
      }
      read_lm35dz();
      pemisah_angka();
      //lcd_clear();
      lcd_gotoxy(0,0);
      sprintf(buff,"SUHU AND: %.1f C",suhu);
      lcd_puts(buff);
      lcd_gotoxy(0,1);
      sprintf(buff,"Timer: %i ",detik);
      lcd_puts(buff);
      delay_ms(100);
      }
}
```

LAMPIRAN FOTO PENELITIAN TERMOMETER DIGITAL DENGAN OUTPUT SUARA BERBASIS ATMEGA 16

(Gambar Lampiran Saat Pembuatan dan Penelitian Alat)