LAMPIRAN

1. Listing Program

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

#define s1 PINC.5
#define s2 PINC.4
#define s3 PINC.3
#define out PORTB.0
#define buzzer PORTB.1
#define sensortegangan 2
#define sensorkompres    1
#define sensorbadan      0

// resistansi pada 25 derajat C
#define THERMISTORNOMINAL 10000
// temperatur untuk nominal resistansi ( hampir selalu
25 C )
#define TEMPERATURENOMINAL 25
// banyaknya pengambilan sample
// untuk hasil yang halus
#define NUMSAMPLES 5
```

Lanjut

```
// The beta coefficient of the thermistor (usually
3000-4000)
#define BCOEFFICIENT 3950
// the value of the 'other' resistor
#define SERIESRESISTOR 10000


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


// Declare your global variables here


char buffer[33];
int time[]={10,15,20};
int detik=0,menit=0,start=0;
eeprom int timer=0;
eeprom int suhu=35;


unsigned int samples[NUMSAMPLES];
/*
rumus timer:
(16bit+1)-(1detik*(xtal/prescaller)
TCNT: (65535+1)+(1*(8mhz/1024))
TCNT: 57723
jadikan HEXADESIMAL
TCNT: E17B
```

Lanjut

```
*/


// 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(start==2){


detik--;

if(detik<0){

if(menit>0){menit--;detik=59;}


}


}


}

#define ADC_VREF_TYPE 0x40

// Read the AD conversion result

unsigned int read_adc(unsigned char adc_input)

{
```

Lanjut

```
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;
}

// sumber referensi
https://learn.adafruit.com/thermistor/using-a-
thermistor
float bacasuhu(int ch){
  unsigned int i;
  float average;
  float steinhart;

  for (i=0; i< NUMSAMPLES; i++) {
   samples[i] = read_adc(ch);
   delay_ms(10);

 }
```

Lanjut

```
   average = 0;
  for (i=0; i< NUMSAMPLES; i++) {
     average += samples[i];
  }
  average /= NUMSAMPLES;

  average = 1023 / average - 1;    //.....
  average = SERIESRESISTOR / average;

  steinhart = average / THERMISTORNOMINAL; // (R/Ro)
  steinhart = log(steinhart); // ln(R/Ro)
  steinhart /= BCOEFFICIENT; // 1/B * ln(R/Ro)
  steinhart += 1.0 / (TEMPERATURENOMINAL + 273.15); //
+ (1/To) .........
  steinhart = 1.0 / steinhart;  // Invert
  steinhart -= 273.15;          // convert to C
  return steinhart;
}
// fungsi baca port
float read_volt(){
 float volt,voltout;
 float voltmax=3.65,involtmax=20.0;
 unsigned int data=0;
 int x;
 // rumus sensor tegangan
```

Lanjut

```
// r1= 4,7k r2= 1k
 // involt max= 20.0 volt (sesuai kebutuhan yang
peting hasil perhitungan voltmax tidak lebih dari 5v)
 // mencari voltmax ( max 5V),
voltmax=r2/(r1+r2)*involtmax
 // voltmax=1k/(4,7+1)*20.0
 // voltmax= 3.50 volt
 for(x=0;x<20;x++){
 data=data+read_adc(sensortegangan);
 delay_ms(5);
 }

 data=data/20;

 volt=data*((float)5/1023);  // mengubah nilai adc ke
tegangan

voltout=volt*((float)involtmax/voltmax); // mengubah
nilai tegangan kecil ke tegangan sensor
return voltout;
}
void uji_sensor(){
 float s_badan=bacasuhu(0);
 float s_kompres=bacasuhu(1);
 float s_volt=read_volt();
```

Lanjut

```
 lcd_clear();
 lcd_gotoxy(0,0);
 sprintf(buffer,"K:%.1f",s_kompres);
 lcd_puts(buffer);
 lcd_putchar(0xdf);
 lcd_putchar('C');

 lcd_gotoxy(0,1);
 sprintf(buffer,"B:%.1f",s_badan);
 lcd_puts(buffer);
 lcd_putchar(0xdf);
 lcd_putchar('C');
 lcd_gotoxy(10,1);
 sprintf(buffer,"%.1fV",s_volt);
 lcd_puts(buffer);
 delay_ms(100);
}
void setting(){
 int mode=0;
 lcd_clear();
 out=0;
 buzzer=1;
 delay_ms(200);
 buzzer=0;
 while(1){
```

Lanjut

```
if(s1==0||s2==0||s3==0){
 buzzer=1;
 }
 if(s1==0)mode++;
 if(mode>1)break;

 if(mode==0){
 lcd_clear();
 lcd_gotoxy(0,0);
 sprintf(buffer,"Waktu:%d",time[timer]);
 lcd_puts(buffer);
 if(s2==0)timer++;
 if(s3==0)timer--;
 if(timer>2)timer=0;
 if(timer<0)timer=2;
}
 if(mode==1){
 lcd_clear();
 lcd_gotoxy(0,0);
 sprintf(buffer,"Suhu:%d",suhu);
 lcd_puts(buffer);
 lcd_putchar(0xdf);
 lcd_putchar('C');
 if(s2==0)suhu++;
 if(s3==0)suhu--;
```

Lanjut

```
 if(suhu>45)suhu=41;
 if(suhu<41)suhu=45;
 }
 delay_ms(150);
 buzzer=0;
 }
 buzzer=1;
 lcd_clear();
 delay_ms(200);
 buzzer=0;
}
void programutama(){
 float s_badan=bacasuhu(0);
float s_kompres=bacasuhu(1);
 float s_volt=read_volt();
 if(s1==0) setting();
 if(s1==0||s2==0||s3==0){
 buzzer=1;
 }
 if(s3==0&&start==0){
 menit=time[timer];
 detik=0;
 start=1;
 out=1;
 }
```

Lanjut

```
 if(start==1){
 if(s_kompres>=(suhu-2))start=2;
 }
 if(start==2){

 if(s_kompres>=(suhu-2))out=0;
 if(s_kompres<(suhu-0))out=1;

 if(menit==0&&detik==0){
out=0;
 start=0;
 buzzer=1;
 delay_ms(2000);
 buzzer=0;
 }
 }
 lcd_clear();
 lcd_gotoxy(0,0);
 sprintf(buffer,"K:%.1f",s_kompres);
 lcd_puts(buffer);
 lcd_putchar(0xdf);
 lcd_putchar('C');
 lcd_gotoxy(10,0);
 if(start==0)lcd_putsf("STOP");
 if(start==1)lcd_putsf("HEATER");
```

Lanjut

```
 if(start==2){
 sprintf(buffer,"%02d:%02d",menit,detik);
 lcd_puts(buffer);
 }
 lcd_gotoxy(0,1);
 sprintf(buffer,"B:%.1f",s_badan);
lcd_puts(buffer);
 lcd_putchar(0xdf);
 lcd_putchar('C');
 lcd_gotoxy(10,1);
 sprintf(buffer,"%.1fV",s_volt);
 lcd_puts(buffer);

 delay_ms(100);
 buzzer=0;
}
void main(void)
{
// Declare your local variables here
// Input/Output Ports initialization
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In
Func2=In Func1=OUT Func0=Out
// State7=T State6=T State5=T State4=T State3=T
State2=T State1=0 State0=0
```

Lanjut

```
PORTB=0x00;
DDRB=0x03;

// Port C initialization
// Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State6=T State5=P State4=P State3=P State2=T
State1=T State0=T
PORTC=0x38;
DDRC=0x00;

// Port D 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 State1=T State0=T
PORTD=0x00;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
TCCR0=0x00;
TCNT0=0x00;
// Timer/Counter 1 initialization
```

Lanjut

```
// 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
```

Lanjut

```
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
MCUCR=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
```

Lanjut

```
// 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 - PORTD Bit 0
// RD - PORTD Bit 7
// 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);
```

Lanjut

```
// Global enable interrupts

#asm("sei")

 lcd_clear();

 lcd_gotoxy(0,0);

lcd_putsf("Sandra Monika");

 lcd_gotoxy(0,1);

 lcd_putsf("20153010031");

 delay_ms(1000);

while (1)

      {

      // Place your code here

      //uji_sensor();

      programutama();

      }

}
```

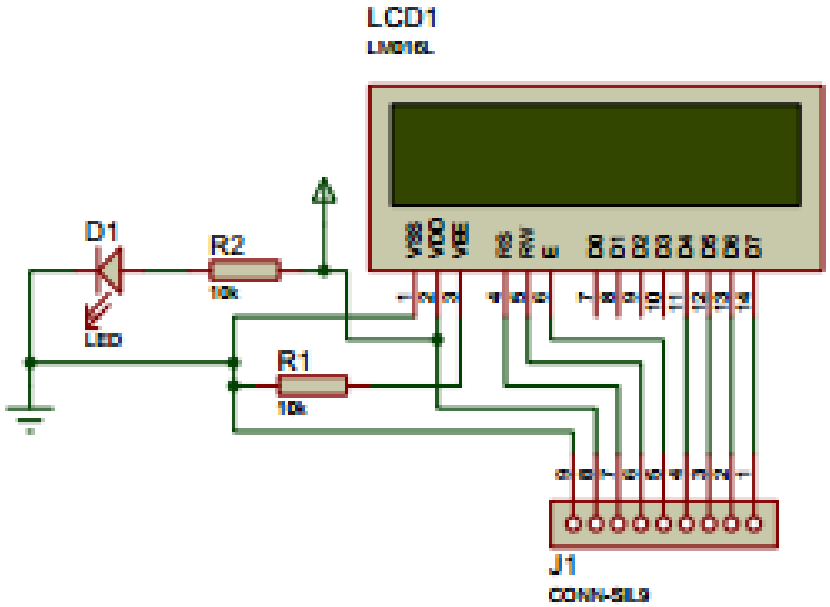
2. Gambar Alat

3. Standar Operasional Prosedure Alat

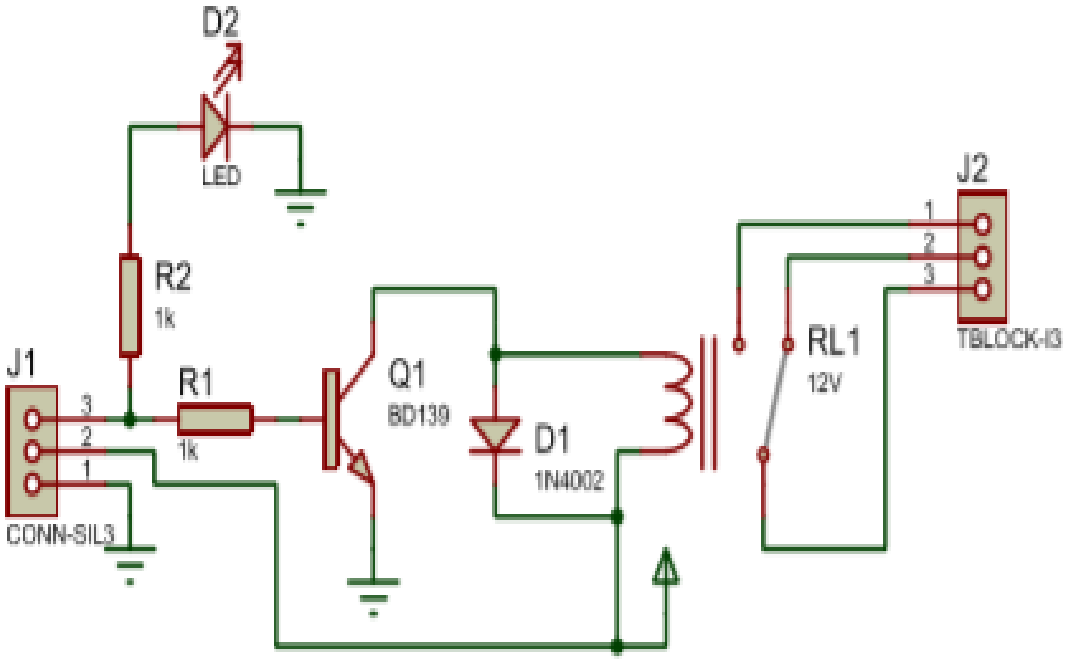
    1. Memasang Kompres di bagian kepala/leher
    2. Menekan tombol ON/OFF pada Alat
    3. Menekan tombol Menu Untuk memilih Settingan kemudian menekan tombol Up/Down untuk mengsetting Suhu dan Timer yang dibutuhkan
    4. Menekan tombol Menu untuk kembali ke tampilan menu, lalu menekan tombol start untuk menyalakan *Heater/* Kompres
    5. Ketika Timer sudah tercapai maka menekan tombol reset
    6. Menekan Tombol ON/OFF untuk mematikan alat
    7. Bersihkan dan rapikan kembali alat

4. Rangkaian Pada Alat

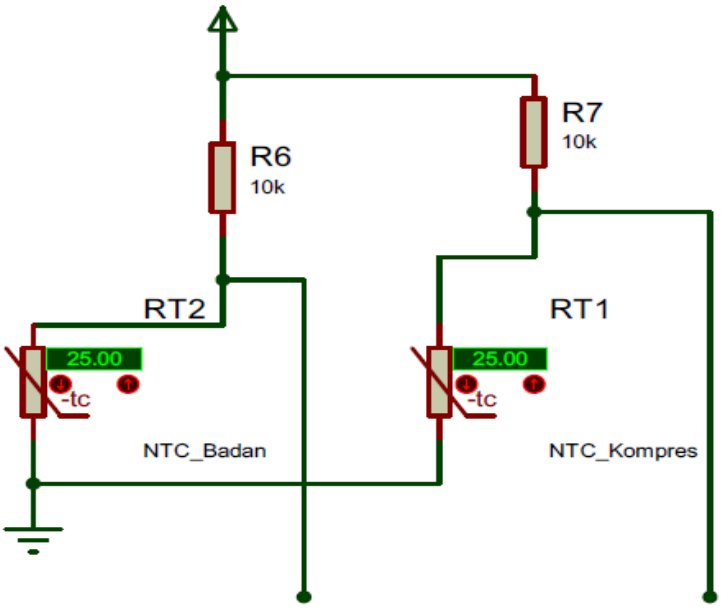
    1. Rngkaian LCD



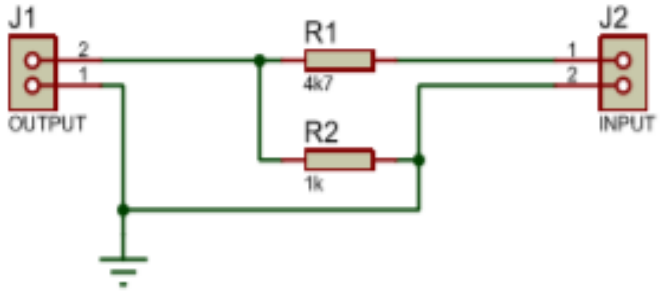
2. Rangkaian Driver Heater



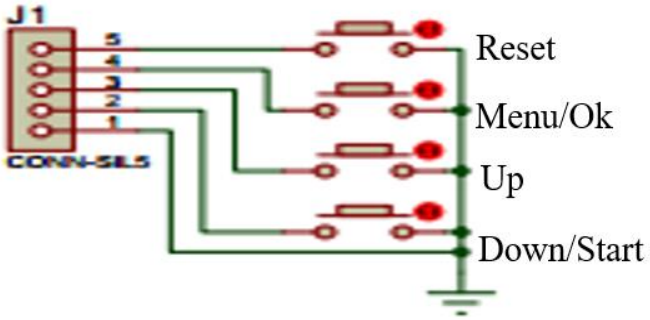
3. Rangkaian Sensor NTC



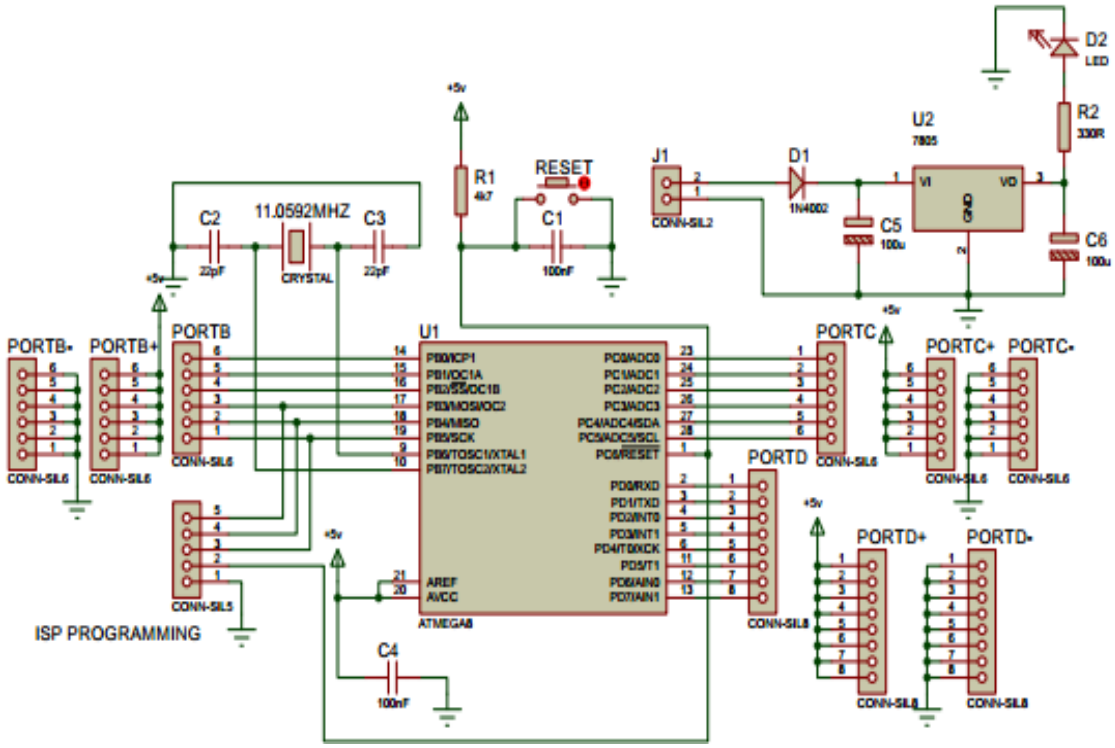
4. Rangkaian Sensor Tegangan



5. Rangkaian PusH Button



6. Rangkaian Minsis



7. Rangkaian Keseluruhan