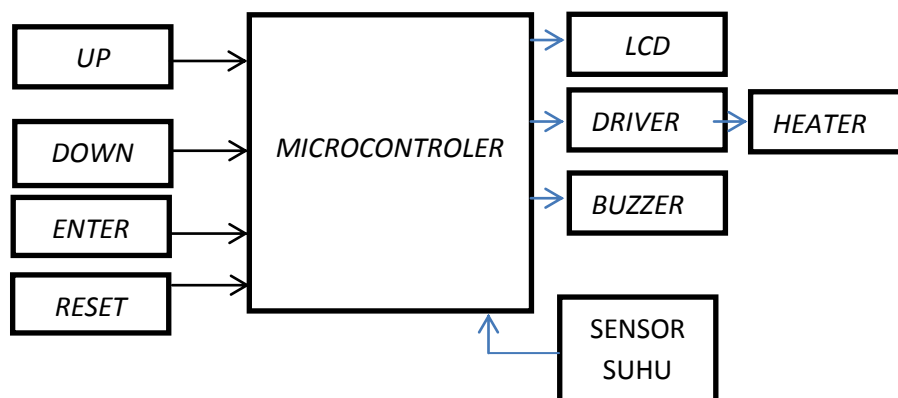


BAB III

METODE PENELITIAN

3.1 Diagram Blok Sistem

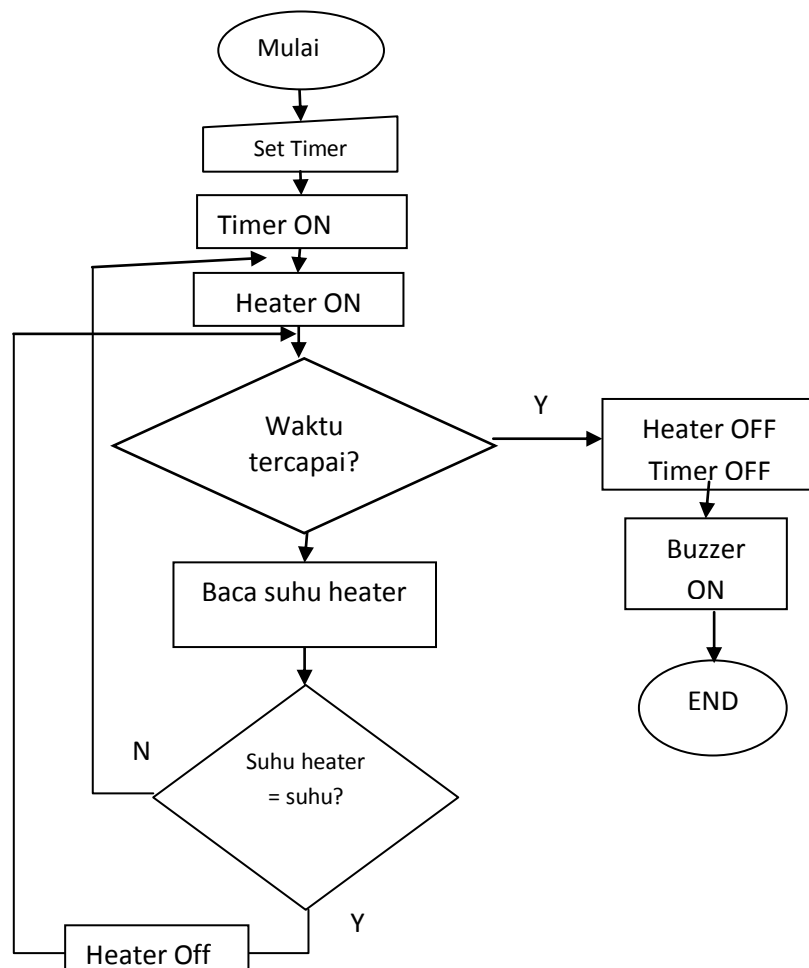
Tegangan *PLN AC 220* akan di turunkan dengan menggunakan *trafo step down* untuk menyuplay rangkaian. *Timer* dan suhu ditentukan dengan menggunakan tombol *UP* dan *DOWN* dengan pilihan selama 5, 10, dan 15 menit untuk waktu dan untuk suhu 40 derajat *celsius*. Waktu dan suhu terapi akan muncul pada *display LCD*. Kemudian tekan tombol *ENTER* untuk memulai terapi. *Microcontroller* akan mengirimkan data yang sudah di atur sebelumnya untuk menyalakan *HEATER* dan *SENSOR SUHU*. Apabila waktu terapi sudah tercapai *BUZZER* akan bunyi dan kerja alat berhenti.



Gambar 3.1. Diagram Blok

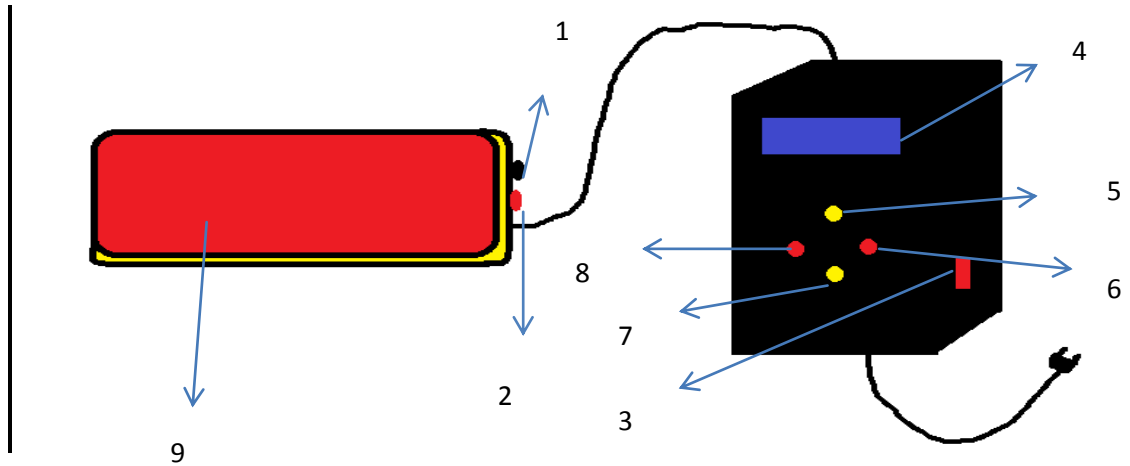
3.2 Diagram Alir

Ketika saat *on* akan ada pilihan untuk *setting timer*, karena suhu sudah konstan di 40 derajat *celcius*. Setelah *setting timer* lalu tekan *enter*. *Heater* dan *timer* mulai *on* lalu panas terdeteksi oleh *sensor* suhu, panas akan di atur dengan suhu yang sudah di sesuaikan. Setelah *timer* sudah tercapai *heater* akan *off* dan *buzzer* akan menyala untuk memberitaukan kalau terapi yang sedang berlangsung sudah selesai.



Gambar 3.2. Diagram Alir

3.3. Diagram Mekanis Alat



Gambar 3.3. Diagram Mekanis

Keterangan:

1. Pengaman (*Fuse*)
2. Indikator pemanas
3. Tombol *power*
4. *LCD display*
5. Tombol *UP*
6. Tombol *ENTER*
7. Tombol *DOWN*
8. Tombol *RESET*
9. Pemanas *heater*

3.4. Perakitan Rangkaian *Driver*

3.4.1. Alat

1. Papan *pcb*
2. Solder
3. Timah
4. Penyedot timah

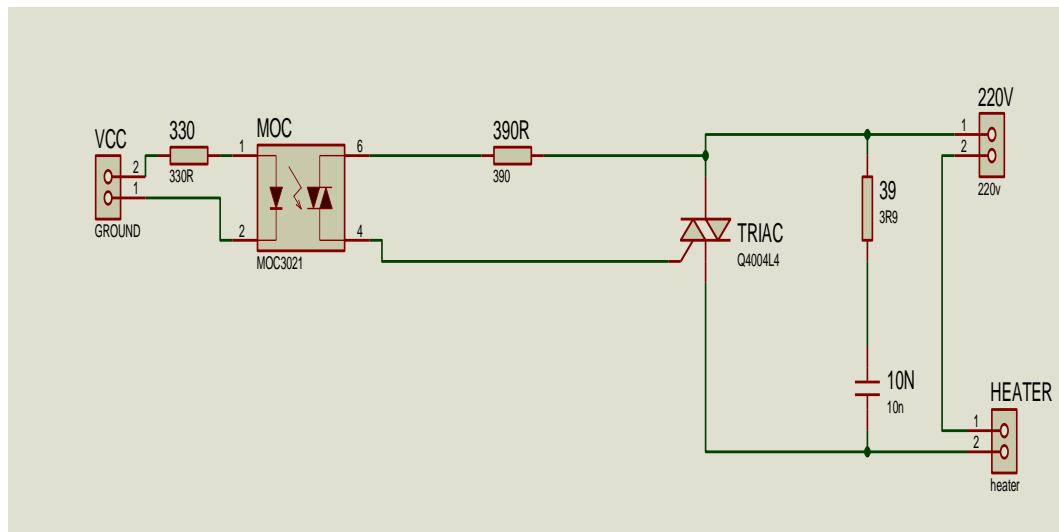
3.4.2. Komponen

1. MOC 3041
2. *Triac* 4004 LT
3. Kapasitor 10 nano 400 V
4. R 330 *ohm*
5. R 390 *ohm* 2 watt
6. R 39 5 watt

3.4.3. Langkah Perakitan

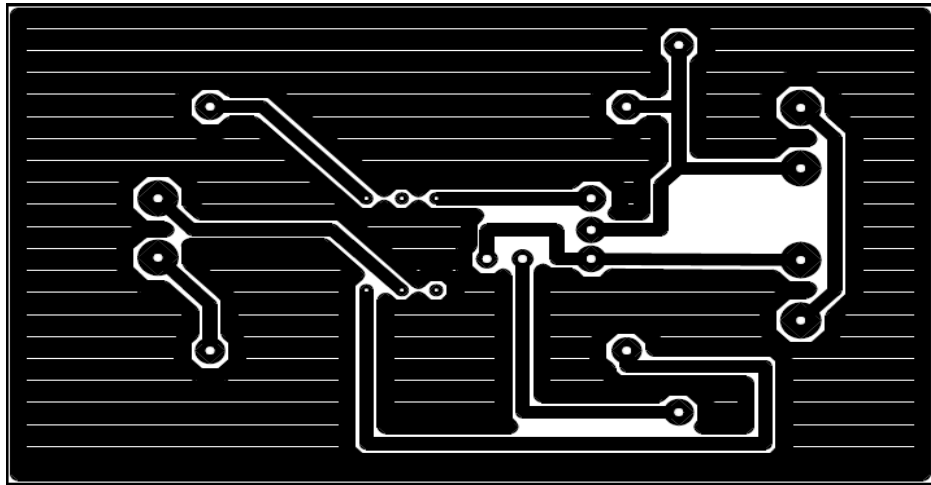
1. Rangkai sistematis rangkaian *driver* dengan menggunakan aplikasi pada laptop, aplikasi yang digunakan pada pembuatan modul ini adalah *proteus*.

Untuk gambar sistematis rangkaian *driver* pada aplikasi dapat dilihat pada gambar 3.4. di bawah ini:



Gambar 3.4. Sistematis Rangkaian Driver

Rangkaian *driver* ini dalam menentukan R1 dengan cara $R = (V_s - V_d) / I$ dan menentukan R2 dengan cara $R_i = V_{cc} - V_d / I_d$. Tegangan led di dalam MOC 1,6V untuk tegangan Vcc 5 V , dan arus yang di butuhkan $10\text{mA} = 0,010\text{A}$. Pengambilan R1 dan R2 itu yang mendekati perhitungan tersebut. Setelah sistematis rangkaian jadi, tahap selanjutnya membuat *lay out* nya dan disablon ke papan *pcb*. Untuk gambar *lay out driver* pada papan *pcb* dapat dilihat pada gambar 3.5. di bawah ini:

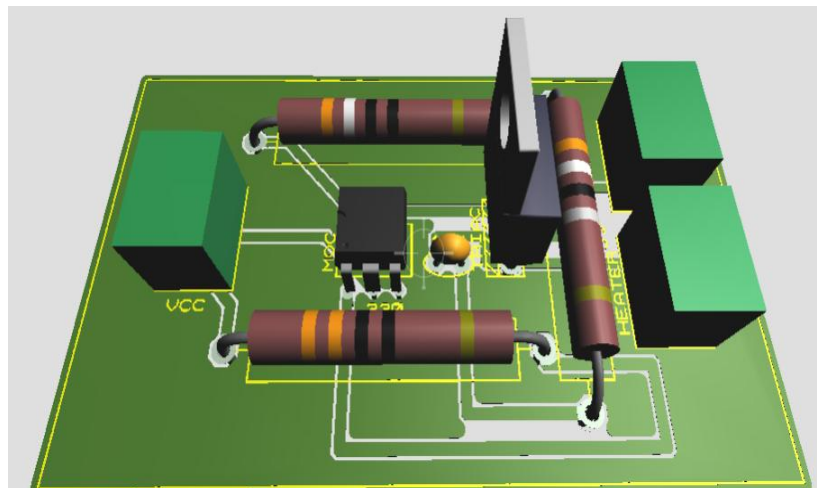


Gambar 3.5.Lay OutDriver

2. Rakit komponen yang dibutuhkan dengan menggunakan timah dan solder.

3.4.4. Gambar RangkainDriver

Gambar rangkaian *driver* dapat dilihat pada gambar 3.6. di bawah ini:



Gambar 3.6. Rangkaian Driver

Rangkaian *driver* pada modul ini berfungsi sebagai kontak dari tegangan DC ke tegangan AC. Prinsip kerjanya dengan memanfaatkan fungsi MOC 3041 yaitu, ketika MOC 3041 mendapat tegangan dari *microcontroller* maka akan saturasi sehingga dapat menghidupkan *triac Q4004LT* dengan kontak AC. Dan ketika MOC tidak mendapatkan tegangan sehingga *triac* akan mati karena *Gate* tidak mendapatkan tegangan dari MOC (Barry Wollard, 2006).

3.5. Perakitan Rangkaian Minimum Sistem

3.5.1. Alat

1. Papan *pcb*
2. Solder
3. Timah
4. Penyedot timah

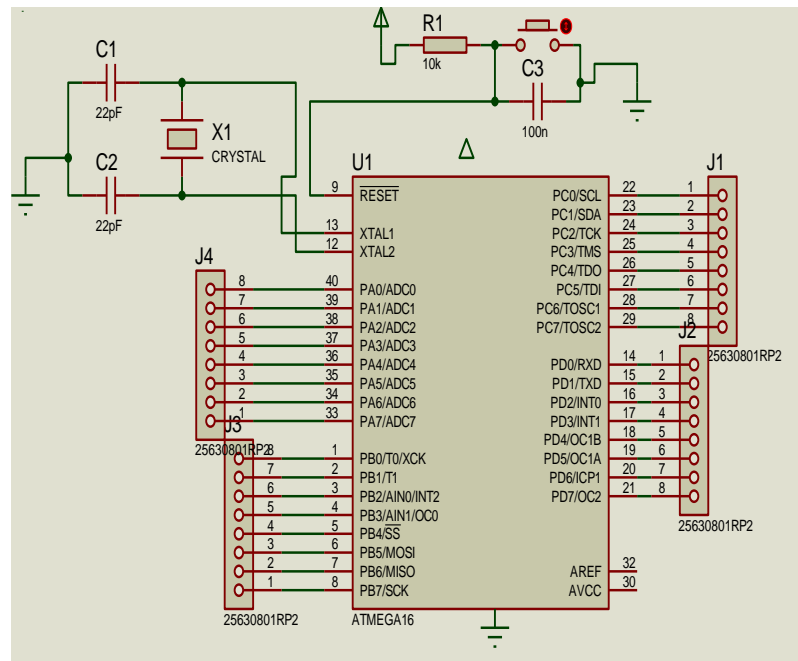
3.5.2. Komponen

1. Atmega 16
2. Kapasitor *nonpolar* 22Pf
3. *Crystal 16000*

3.5.3. Langkah perakitan

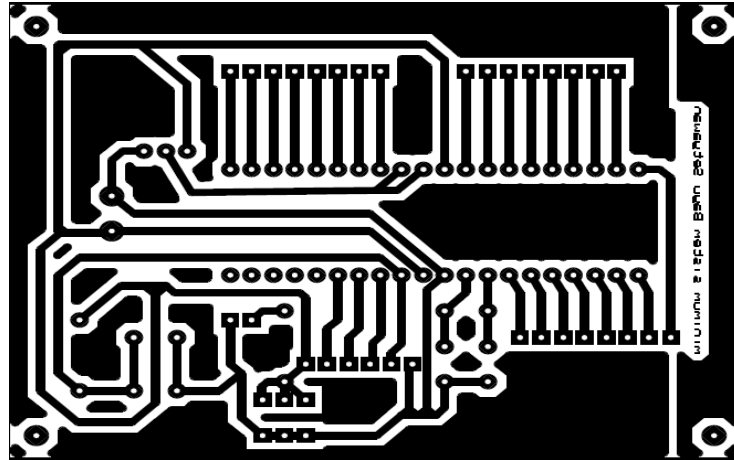
1. Rangkai sistematis rangkaian minimum sistem dengan menggunakan aplikasi pada laptop, aplikasi yang digunakan pada pembuatan modul ini adalah *proteus*.

Untuk gambar sistematis rangkaian minimum sistem pada aplikasi dapat dilihat pada gambar 3.7. di bawah ini:



Gambar 3.7. Sistematis Minimum Sistem

2. Setelah sistematis rangkaian jadi, tahap selanjutnya membuat *lay out* dan disablon ke papan *pcb*. Untuk gambar *lay out* minimum sistem pada papan *pcb* dapat dilihat pada gambar 3.8. di bawah ini:



Gambar 3.8. Lay Out Rangkaian Minimum Sistem

3. Rakit komponen yang dibutuhkan dengan menggunakan solder.

3.5.4. Gambar Minimum Sistem

Untuk gambar minimum sistem dapat dilihat pada gambar 3.9. di bawah ini:



Gambar 3.9. Minimum Sistem

Rangkaian minimum sistem pada modul ini berfungsi sebagai kontrol kerja modul secara keseluruhan. Cara kerja rangkaian minimum sistem ini dengan memanfaatkan kapasitas penyimpanan yang dimiliki oleh *IC ATmega 16*. Pada *IC ATmega 16* ini diberi program yang akan mengontrol sistem kerja modul secara keseluruhan. Adapun program yang digunakan pada modul ini adalah *ADC* sebagai pembaca tegangan dari *sensor* suhu *LM35* dan program *timer* sebagai pengendali waktu pada modul.

3.6. Perakitan Rangkaian Power supply

3.6.1. Alat

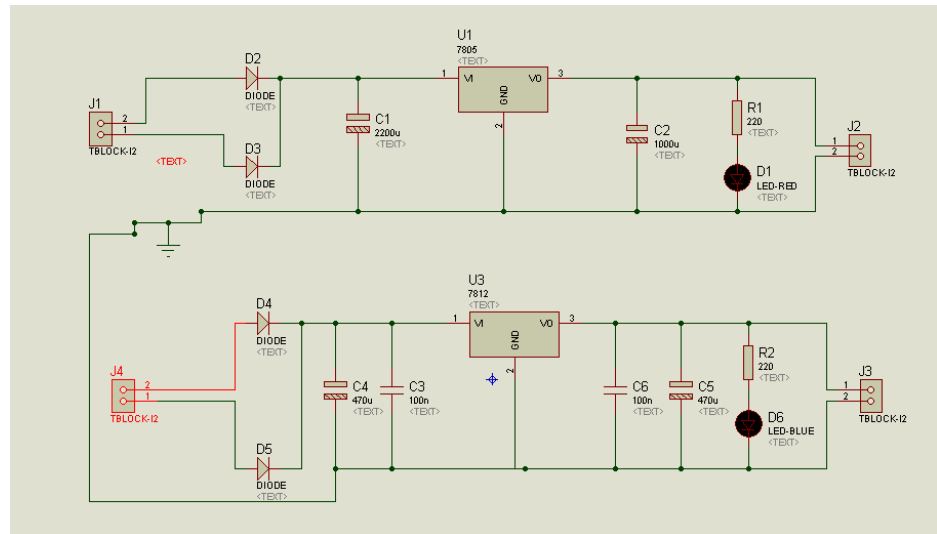
1. Papan *pcb*
2. Solder
3. Timah
4. Penyedot timah

3.6.2. Bahan

1. Dioda 4A
2. Travo 1 A
3. Kapasitor 3300 μf (2)
4. Kapasitor 470 μf (2)
5. Kapasitor *non polar* 104 (4)
6. *IC* regulator 7805 dan 7812
7. *LED* (2)
8. *T-blok*

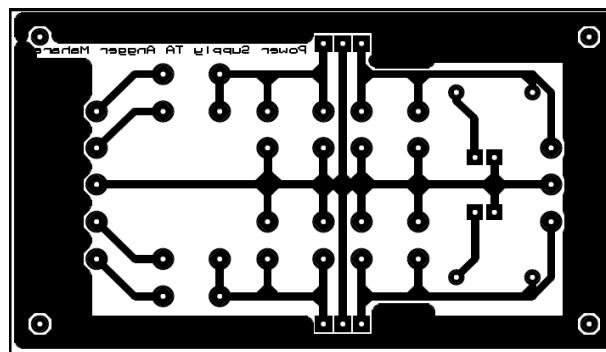
3.6.3. Langkah Perakitan

1. Rangkai sistematis rangkaian *power supply* dengan menggunakan aplikasi pada laptop, aplikasi yang digunakan pada pembuatan modul ini adalah *proteus*.



Gambar 3.10. Sistematis *power supply*

2. Setelah sistematis rangkaian jadi, tahap selanjutnya membuat *lay out* nya dan disablon ke papan *pcb*. Untuk gambar *lay out power supply* pada papan *pcb* dapat dilihat pada gambar 3.11. di bawah ini:



Gambar 3.11. *Lay Out Power supply*

3. Rakit komponen yang dibutuhkan dengan menggunakan solder.



Gambar 3.12. *Power Supply*

Rangkaian *power supply* pada modul ini berfungsi sebagai *supply* tegangan ke semua rangkaian yang menggunakan tegangan DC. Prinsip kerja *power supply* adalah mengubah tegangan AC menjadi tegangan DC dengan menggunakan *transformator* sebagai penurun tegangan dan dioda sebagai komponen yang berfungsi sebagai penyearah tegangan. Pada modul ini *power supply* akan mengubah tegangan AC menjadi DC sebesar 5 VDC dan 12 VDC dengan menggunakan *IC regulator* 7805 dan 7812. Adapun tegangan 5 VDC digunakan untuk rangkaian *minimum sistem* sedangkan tegangan 12 VDC digunakan untuk *relay* 12 VDC.

3.7. Perakitan Rangkaian *Sensor suhu*

3.7.1. Alat

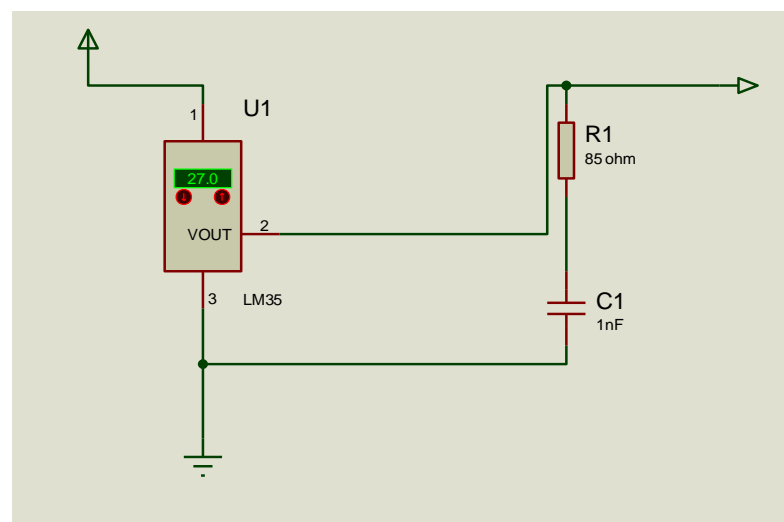
1. Papan *pcb*
2. Solder
3. Timah
4. Penyedot timah

3.7.2. Komponen

1. IC LM 35
2. Kapasitor 1 nf
3. Resistor 85 ohm

3.7.3. Langkah perakitan

1. Rangkai sistematis rangkaian *sensor suhu* dengan menggunakan aplikasi pada laptop, aplikasi yang digunakan pada pembuatan modul ini adalah *proteus*.



Gambar 3.13. sistematis *sensor suhu*

Rangkaian *sensor* suhu pada modul ini berfungsi sebagai pengatur suhu dalam modul ini. Prinsip kerja *sensor* suhu adalah mengubah besaran suhu ke besaran listrik dalam bentuk tegangan dan juga sebagai pengendali suhu yang sudah ditentukan. Pada modul ini *sensor* suhu akan memberikan perintah ke *driver heater* untuk menyala dan mati apabila suhu yang ditentukan sesuai atau belum. Fungsi dari rangkaian RC pada rangkaian *sensor* suhu adalah sebagai penyaringan sinyal dengan memberikan tahanan atau blok. Tahanan tersebut dihasilkan oleh resistor melalui kemampuan resistansi. Selanjutnya sinyal juga akan disimpan dalam kapasitor melalui efek kapasitansi. Dalam proses penyaringan sinyal (*filtering*), resistor berfungsi sebagai hambatan untuk menahan arus listrik, sedangkan kapasitor berfungsi untuk menyimpan arus listrik secara sementara. Rangkaian filter (rangkaiannya penyaring) merupakan rangkaian yang didesain hanya untuk memperbolehkan suatu frekuensi pada rentang tertentu memiliki nilai redaman (*atenuasi*) yang kecil (disebut sebagai '*Pass Band*'), sedangkan pada rentang frekuensi lainnya memiliki nilai redaman yang sangat besar (disebut sebagai '*Attenuation Band*' atau '*Stop Band*').

Sebuah rangkaian filter bisa terdiri hanya dari komponen-komponen pasif dan biasa disebut sebagai rangkaian filter pasif (*Passive Filter Network*). Ada juga rangkaian filter yang

Pushbutton ini digunakan untuk pemelihan *timer*, *enter*, dan *reset*. *Pushbutton* disambungkan pada *Port B pin 0.1.2.9* dan yang satu di *ground microcontroller* sebagai inputan pada saat di tekan ada beberapa perintah yang sudah di program di dalam *microcontroller* agar sistem bekerja dengan sempurna.

3.9. Perakitan Rangkaian LCD

3.9.1. Alat

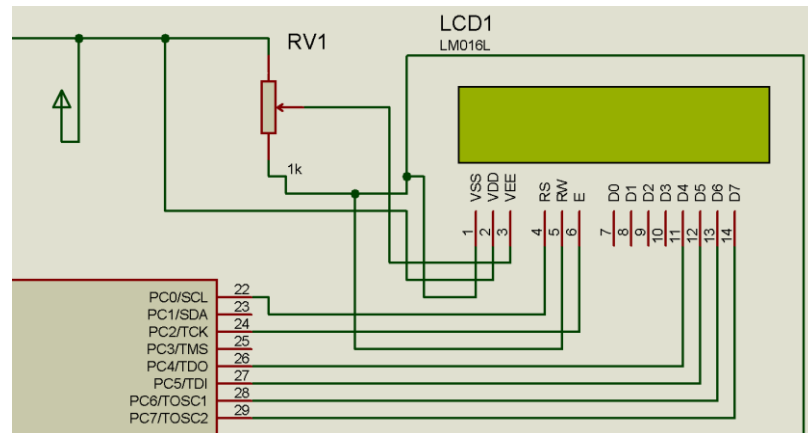
1. Solder
2. Timah
3. Penyedot timah

3.9.2. Komponen

1. *LCD 16x2*
2. Kabel pelangi *female* dan *male*

3.9.3. Langkah perakitan

1. Pasang kabel pelangi pada pin di *LCD*.
2. Pasang kabel yang sudah terhubung dengan *LCD* ke *Port c minimum sistem* sesuai konfigurasi yang sudah di tentukan.
3. Sesuai sistematik di bawah ini :



Gambar 3.17. Rangkaian *LCD*

Rangkaian *LCD* 16x2 ini digunakan untuk menampilkan beberapa karakter penulisan yang sudah di program agar muncul dan terlihat, di dalam *LCD* ini akan di tampilkan 2 buah tampilan antara lain suhu yang terbaca dan *timer* perhitungan 5 menit, 10 menit, 15 menit perhitungan secara *counter down*.

3.10. Perakitan Rangkaian *Buzzer*

3.10.1. Alat

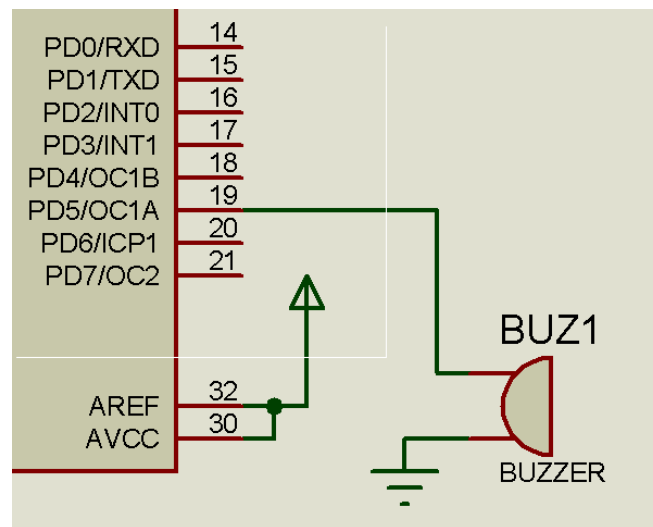
1. Solder
2. Timah
3. Penyedot timah

3.10.2. Komponen

1. *Buzzer*
2. Kabel pelangi *female* dan *male*

3.10.3. Langkah perakitan

1. Pasang kabel pelangi ke kaki *buzzer*.
2. Hubungkan kabel *positif* ke *Port D pin 5* dan kabel *negatif* pasang ke *ground*.
3. Sesuai sistematik di bawah ini :



Gambar 3.18. Rangkaian *Buzzer*

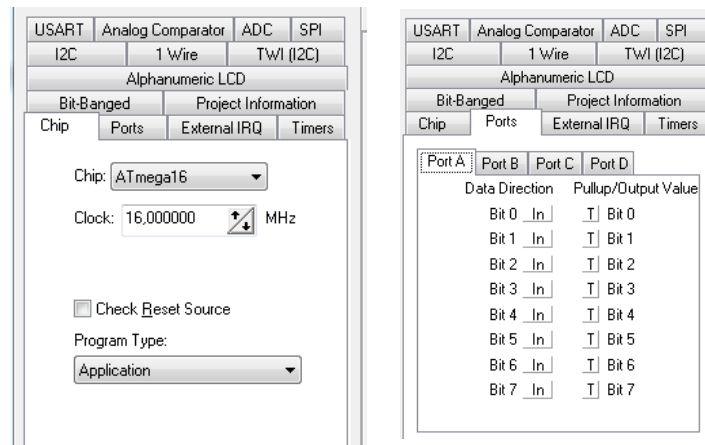
Rangkaian *Buzzer* ini di gunakan untuk memberikan peringatan apabila semua sistem yang sudah berjalan selesai sesuai *timer* yang di tentukan, *buzzer* ini di sambungkan langsung dengan *mikrokontroller* dengan kaki positif ke *Port D pin 5* dan kaki negatif di sambungkan ke *ground* sudah dapat bekerja dengan baik.

3.11. Rangkaian keseluruhan modul

Rangkaian ini tersusun dari beberapa blok-blok PCB yang sudah terpasang komponen-komponen sesuai fungsi dari blok tersebut dan di

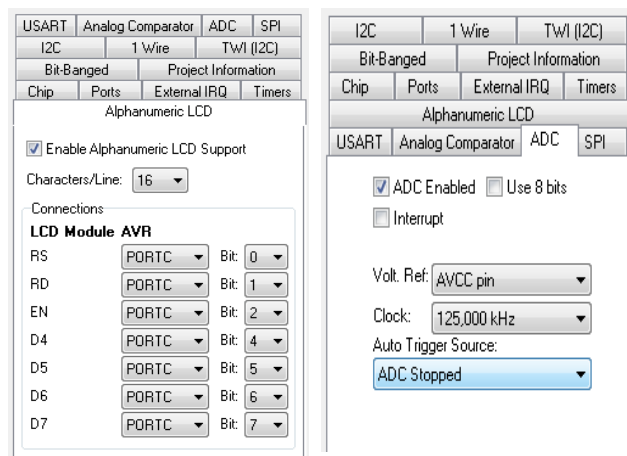
3.12. Pembuatan Program sistem.

3.12.1. Untuk pembuatan program pada modul ini menggunakan aplikasi AVR dengan bahasa C dan atur dahulu pembuatan dengan cara:



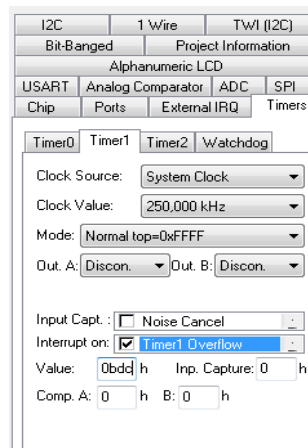
Gambar 3.20. Pengaturan program *Chip* dan *Port*.

Sebelum pembuatan program terlebih dahulu mengatur tata sistem yang akan di gunakan, antara lain adalah mengatur *chip/ ATmega* yang di gunakan, penulis menggunakan *ATmega16* dan *clock/ kristal* yang di gunakan 16000 MHz. Setelah itu mengatur *Port-Port* yang akan di gunakan sebagai *input/output*, agar sesuai dengan sistem yang di butuhkan penulis. *Port* yang di gunakan sebagai *input* antara lain *Port A* di gunakan sebagai penempatan *ADC sensor* suhu *LM35* dan *Port B* di gunakan sebagai penempatan *Pushbutton*.



Gambar 3.21. Pengaturan program *LCD* dan *ADC*

Setelah mengatur *Chip* dan *Port* dilakukan pengaturan *LCD* di *Port C* di karakter 16 dikarenakan *LCD* ini hanya menampilkan ruang penampikan 16x2 saja. Setelah pengaturan *LCD* diikuti dengan pengaturan *ADC*, penulis menggunakan *ADC* dari *VCC* karena tidak ada pengaturan *AREF*.



Gambar 3.22. Pengaturan program *timer*

Pengaturan *timer* ada 2 macam *timer* yang sering di gunakan antara lain *Timer1* (16 bit) dan *Timer2* (8 bit). Untuk

perbedaan dan cara kerja masing-masing *timer*, dapat dilihat pada *datasheet*. Pengaturan *timer* yang digunakan adalah *timer1* (16 bit) didapatkan persamaan antara lain :

$$TCNT = (2^n) - \frac{T_{timer} \times f_{clk}}{N} \dots\dots\dots(3-1)$$

Keterangan:

TCNT : Nilai yang diberikan pada *registertimer* (TCNT)

Fclk : *frekuensi clock (crystal)* yang digunakan

Ttimer : waktu *timer* yang diinginkan untuk *overflow*

n : Jumlah *bit* pada *TIMER/COUNTER*

N : *Prescaler* (1, 8, 64, 256, 1024)

Perhitungan yang harus di hitung dalam *timer1* 16 bit yang dapat bekerja selama 1 detik, dengan *frekuensi clock* sebesar 16 MHz dan *prescaler* 64 adalah :

$$TCNT = (65.536) - \frac{1-16000000}{64} . .$$

$$TCNT = 184,464$$

Konfigurasi pada *codevision*, *clock value* didapatkan dengan pembagian, *f clock* 16MHz dibagi dengan *prescaler* 64 sehingga didapatkan *frekuensi clock* ke *TIMER/COUNTER* 250,000 kHz. Setelah pengaturan *Timer, ADC, Chip, Port, dan LCD* sudah diatur simpan pengaturan program pada *Generate program, save and exit*.

3.12.2. Pembuatan *list* program dengan cara manual untuk menambahkan perintah dan logika pada program ini agar berjalan sesuai yang diinginkan penulis, antara lain:

```

//seting timer
void setting_timer()
{
    if(b==0)
    {
        if(PINB.0==0)
        {
            a++;delay_ms(500);lcd_clear();
        }
        if(a<1)
        {
            a=1;
        }
        if(a>3)
        {
            a=1;
        }
        else if(PINB.1==0)
        {
            a--;delay_ms(500);lcd_clear();
        }
        if(a==1)
        {
            menit=5;
        }
        else if(a==2)
        {
            menit=10;
        }
        else if(a==3)
        {
            menit=15;
        }
    }
}

void mulai_timer()

```

Listing 3.1 Listing program Timer.

Listing program timer ini digunakan sebagai pengatutan *timer* waktu saat sistem bekeja ada 3 waktu yang diatur dalam *listing* program ini antara lain 5,10,15 menit dengan metode *counter down*.


```
// Pembacaan sensor suhu
void baca_suhu()
{
  lcd_gotoxy(0,0);
  lcd_putsf("SUHU:");
  data=read_adc(0);
  suhu=(data*5)/1024;
  suhu=suhu*100;
  itoa(suhu,temp3);
  lcd_gotoxy(0,1);
  lcd_puts(temp3);
  delay_ms(100);
}
// Pengaktifan driver
void driver_set()
{
  if(timer_aktif==1)
  {
    if(suhu>=40)
    {
      PORTD.5=0;
    }else{PORTD.5=1;}}
}
```

Listing 3.2 Listing program sensor suhu.

Listing program *sensor* suhu ini digunakan sebagai pengatuan pembacaan *sensor* suhu LM35 saat sistem bekerja ada beberapa yang di atur dalam *listing* program ini antara lain rumus pembacaan *sensor* dan batas *sensor* suhu pembacaan untuk mengontrol *driver* hetaer.