

## BAB III

### METODOLOGI PENELITIAN

#### 3.1. Alat dan Bahan

Adapun alat dan bahan yang digunakan oleh penulis dalam merancang alat ini adalah sebagai berikut :

##### 3.1.1 Alat

Dalam melakukan penelitian ini penulis menggunakan beberapa peralatan. Pada tabel 3.1 merupakan beberapa peralatan yang digunakan pada modul TA.

Tabel 3. 1 Alat

No	Keterangan	Jumlah
1	Setrika	1
2	Sablon PCB	1
3	Larutan FeCl <sub>3</sub>	1
4	Spidol Permanent	1
5	Mesin Bor duduk	1
6	Mata Bor	1
7	Solder Listrik	1
8	Obeng - +	1
9	Cutter	1
10	Multimeter	1
11	Atraktor	1
12	Tang Potong	1
13	Tang Cucut	1
14	Tang Kombinasi	1
15	Laptop	1

### 3.1.2 Bahan

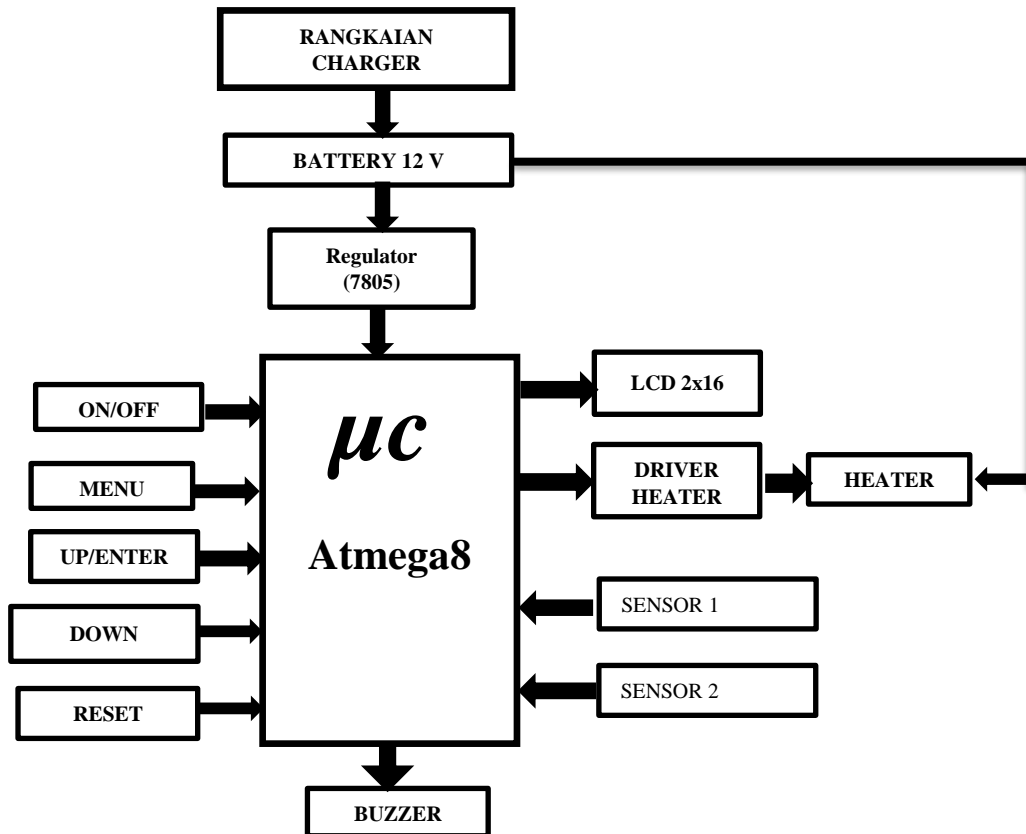
Dalam melakukan penelitian ini penulis menggunakan beberapa bahan elektronika. Pada tabel 3.2 merupakan beberapa bahan elektronika yang digunakan dalam pembuatan modul TA.

Tabel 3. 2 Bahan

No	Nama Komponen	Type/Nilai	Jumlah
1	IC ATmega	8	1
2	<i>Crystal</i>	16 mHz	1
3	Kapasitor	22pF	2
4	Kapasitor	100nF	2
5	Resistor	4k7 $\Omega$	2
6	IC Regulator	7805	1
7	Dioda	IN4002	2
8	Kapasitor	100u	2
9	Resistor	330 $\Omega$	1
10	LED ( <i>Light Emitting Diode</i> )	-	2
11	<i>Relay</i>	5 V/10 A	1
12	T-Block	-	1
13	<i>Transistor</i>	C945 NPN	1
14	<i>Resistor</i>	1k $\Omega$	3
15	<i>Push Button</i>	-	4
16	LCD Character	2x16	1
17	<i>Battery</i>	Lithium Lion 3,7 V	3
18	<i>Buzzer</i>	5 V	1
19	Sensor suhu	NTC	2
20	<i>Peltier</i>	TEC1-12706	1
21	Box Akrilik	-	1

### 3.2 Blok Diagram Alat

Pada Gambar 3.1 merupakan Blok Diagram dari modul TA.



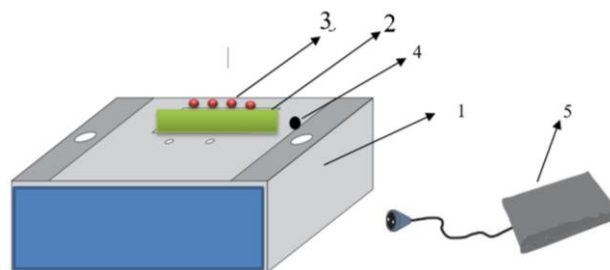
Gambar 3. 1 Diagram Blok Sistem

Mengacu pada Gambar 3.1 dapat dijelaskan sistem kerja dari blok diagram alat kompres elektrik adalah pada saat alat dihidupkan maka tegangan dari baterai akan masuk pada semua blok, dimana baterai akan mendapat daya dari rangkaian *charger*. Begitu alat dihidupkan maka alat langsung mendapat *input* data dari *sensor* suhu dan *sensor* tegangan berupa suhu kompres, suhu badan dan *indicator* baterai.yang akan tertampil di LCD 2 X 16. kemudian *Timer* dan suhu ditentukan dengan menggunakan tombol menu untuk pemilihan *setting*, dan tombol *up/down* sebagai pengatur suhu dan *timer* yang dibutuhkan selama terapi.dengan pilihan *timer* selama 10, 15,20 menit dan 41, 42, 43, 44, 45 derajat

*celsius* untuk suhu *heater* kemudian tekan tombol *enter* untuk memulai terapi. Waktu *count down*, suhu *heater*, dan suhu pasien akan muncul pada *display* LCD. Mikrokontroller akan mengirimkan data yang sudah di atur sebelumnya untuk menyalakan *heater*, *sensor heater* dan *sensor pasien*. *Driver heater* akan menyalakan dan mengontrol suhu *heater* sesuai dengan *setting* agar suhu dapat terjaga stabil nilainya, sedangkan suhu pasien digunakan untuk memonitoring suhu pasien ketika proses terapi berlangsung. Apabila waktu terapi sudah tercapai *buzzer* akan berbunyi dan kerja alat berhenti. dan jika ingin digunakan lagi maka tekan tombol *reset* dan *setting* ulang.

### 3.3 Diagram Mekanis Sistem

Pada gambar 3.2 ini merupakan rancangan alat kompres elektrik atau diagram mekanis sistem dari alat kompres elektrik.

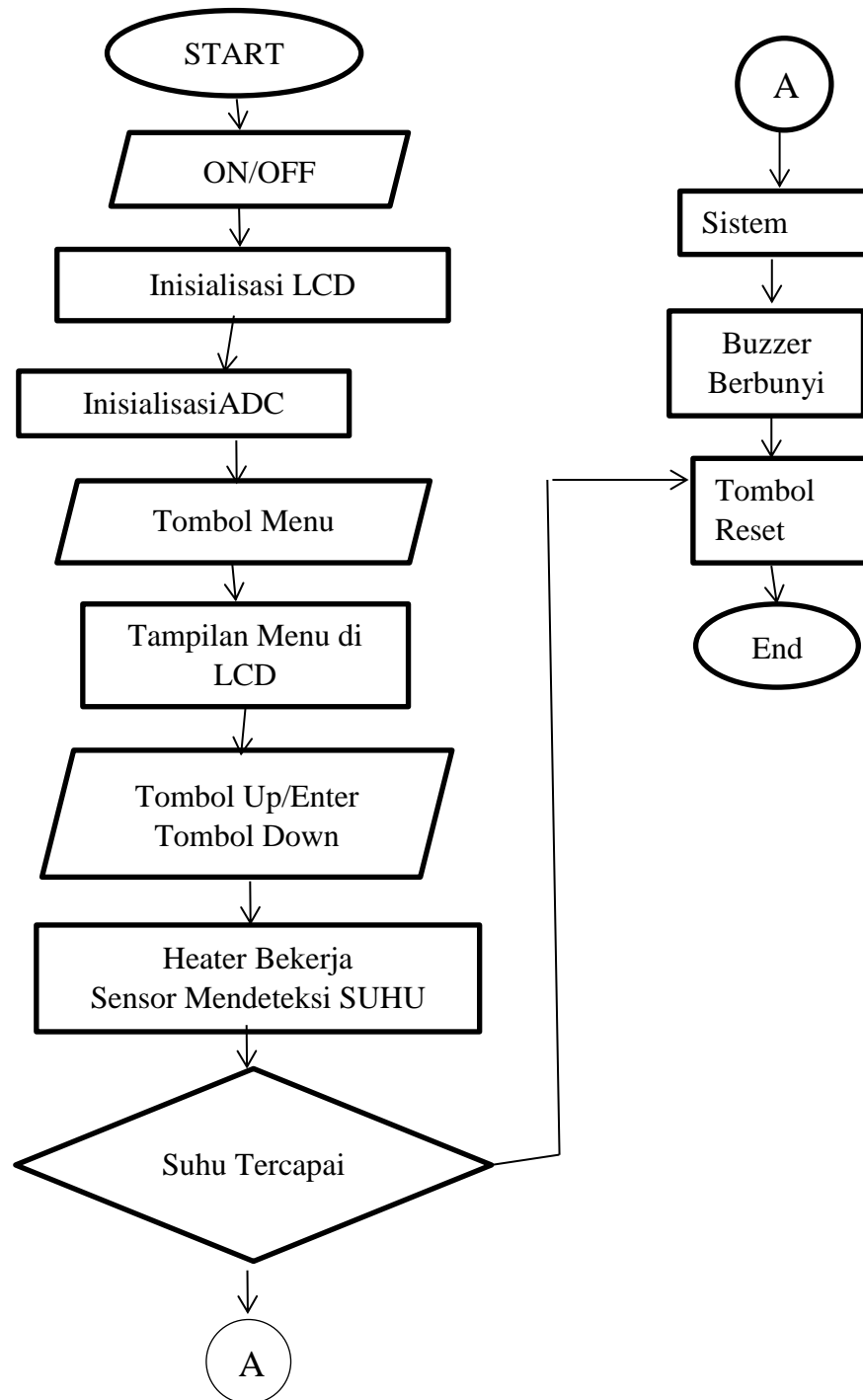


Gambar 3. 2 Kompre Elektrik

Dari Gambar 3.2 dapat dijelaskan, 1 merupakan *body* dari alat untuk pelindung rangkaian untuk membuat alat lebih bernilai ekonomi, 2 merupakan *display* yang berfungsi untuk menampilkan nilai suhu, timer dan tegangan baterai. 3 merupakan tombol *setting* pilihan suhu dan timer yang dibutuhkan. 4 merupakan tombol *ON/OFF* alat. 5 merupakan *peltier* yang berfungsi sebagai kompresnya yang dlapisi oleh kain dan busa sebagai pelindung saat digunakan oleh pasien.

### 3.4 Diagram Alir Alat

Pada gambar 3.3 merupakan diagram alir dari modul TA.



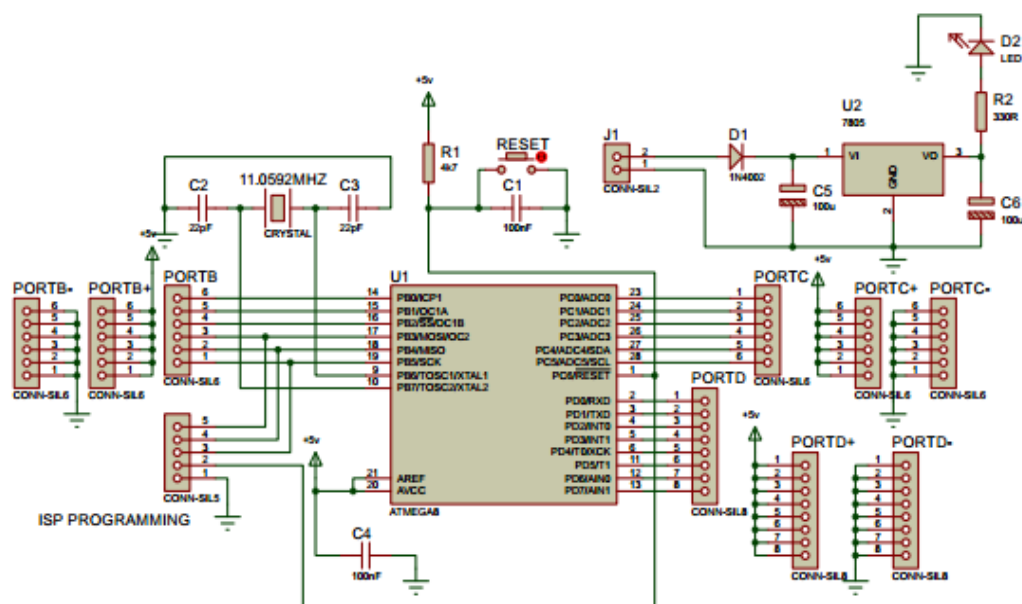
Gambar 3. 3 Diagram Alir Alat

Mengacu pada Gambar 3.3 dapat dijelaskan *Start* untuk memulai program, sebelum mengerjakan program, mikrokontroler melakukan inisialisasi pada LCD kemudian muncul tampilan demo nama dan judul alat. Mikrokontroler akan menginisialisasi ADC yang berfungsi sebagai inputan yang akan ditampilkan pada LCD. Tombol menu untuk pemilihan *setting*, dan tombol *up/down* sebagai pengatur suhu dan *timer* yang dibutuhkan selama terapi dengan pilihan timer selam10, 15,20 menit dan 41, 42, 43, 44, 45 *derajat Celsius*, kemudian tekan tombol *Enter* untuk memulai kerja sistem. *Driver heater* akan menyalakan dan mengontrol suhu *heater* sesuai dengan *setting* agar suhu dapat terjaga stabil nilainya. Apabila waktu terapi sudah tercapai *buzzer* akan berbunyi dan kerja alat berhenti. dan jika ingin digunakan lagi maka tekan tombol *reset* dan *setting* ulang.

### 3.5 Rancangan Perangkat Keras

#### 3.5.1 Rangkaian Minimum Sistem

Pada gambar 3.4 merupakan rangkaian dari Minimum Sistem.

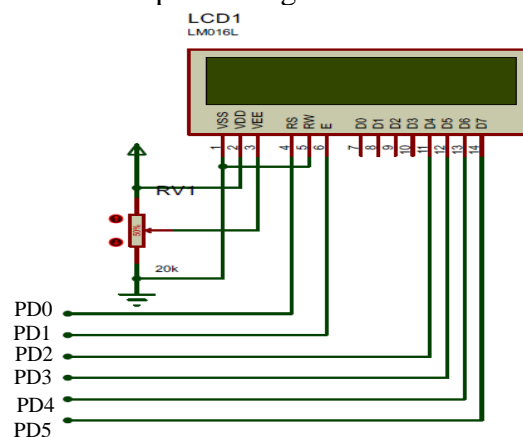


Gambar 3. 4 Rangkaian Minimum Sistem

Mengacu pada Gambar 3.4 dapat dijelaskan rangkaian minimum sistem disini berfungsi sebagai otak dan pengendali segala aktifitas dari alat. Minimum sistem diatas menggunakan ATmega8 yang telah dilengkapi dengan ADC internal sehingga memudahkan sistem dalam *converter analog* menjadi digital. Dan dilengkapi dengan PORT OC/OCR yang digunakan sebagai PWM (*pulse with modulation*). Pada rangkaian minimum sistem juga terdapat port yang disambung ke *downloader* yang berfungsi untuk memasukkan program yang dibutuhkan modul.

### 3.6.2 Rangkaian LCD 2 X 16

Pada gambar 3.5 merupakan rangkaian dari LCD.

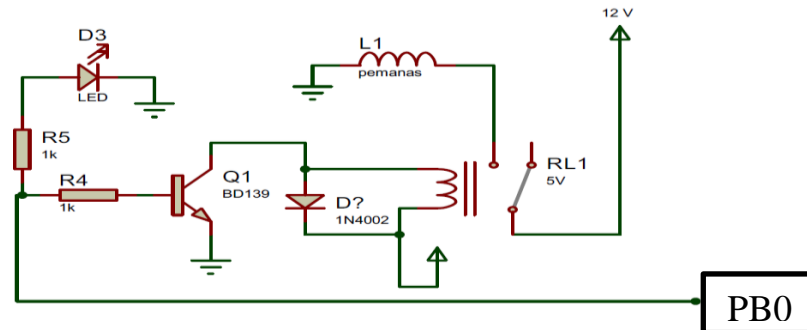


Gambar 3. 5 Rangkaian LCD 2x16

Mengacu pada Gambar 3.5 dapat dijelaskan rangkaian LCD menggunakan tampilan *output* berupa LCD 2x16 , dimana nantinya nilai ADC yang terbaca dalam bentuk nilai *derajat* dan menit akan tertampil pada layar LCD, untuk dapat mnghidupkan LCD diperlukan tegangan *supply* +5V pada pin VDD dan *ground* pada pin VSS dan untuk pengaturan kontras kecerahan LCD dipasang resistor tahanan yang diseri dengan tegangan input +5V, untuk nilai resistornya menggunakan nilai nilai 2.2 kilo ohm karena dengan nilai tersebut kontrasnya dapat pas dan tidak terlalu cerah.

### 3.6.3 Rangkaian Driver Heater

Pada gambar 3.6 merupakan Rangkaian dari *Driver Heater*.



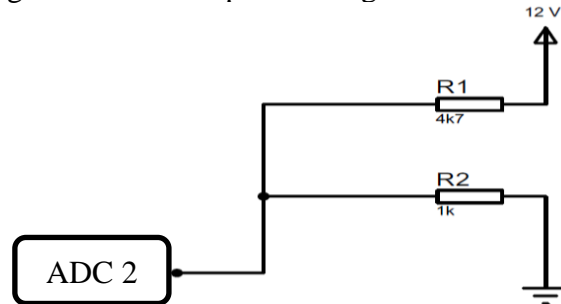
Gambar 3. 6 Rangkaian Driver Heater

Mengacu pada gambar 3.6 dapat dijelaskan Rangkaian *Driver Heater* disini berfungsi sebagai saklar untuk menghidupkan / mematikan peltier menggunakan *relay*. Jika sudah selesai melakukan pengsettingan, tombol *start* ditekan maka adc akan berlogika 1 dan PBO akan memberikan *output* tegangan 5V ke rangkaian *driver*. *Transistor* berfungsi sebagai *switching* untuk mengaktifkan *relay*. *transistor* bisa bekerja apabila mendapat tegangan diatas 0,7 V dan arus tidak boleh melebihi 0,5 A. R1 1k $\Omega$  berfungsi sebagai pengaman agar arus yang masuk di basis *transistor* tidak melebihi  $I_{max}$  *transistor* yaitu 0,5 A. ketika *transistor* mendapat tegangan di atas 0,7 V maka *transistor* akan saturasi atau *collector* & *emitor* akan terhubung. Setelah *transistor* dalam keadaan saturasi, tegangan 5v akan masuk ke *coilnya relay* dan *coil* akan masuk ke *groundnya kolektor* karena ada beda potensial dari + ke - . Kaki COM mendapat tegangan 12 V langsung dari baterai untuk memenuhi tegangan yang dibutuhkan *peltier*. sehingga ketika kaki COM mendapat logika 1 maka kaki NC (*Normaly Close*) akan berpindah ke kaki NO (*Normaly open*) dan akan memberikan tegangan pada *peltier* sehingga menyebabkan *peltier* bekerja. Kaki NO akan berpindah ke NC apabila mendapat logika 0 pada basis *transistor*.



### 3.6.4 Rangkaian Sensor Tegangan

Pada gambar 3.7 merupakan Rangkaian dari *Sensor Tegangan*.



Gambar 3. 7 Rangkaian *Sensor Tegangan*

Mengacu pada Gambar 3.7 dapat dijelaskan rangkaian *sensor tegangan* disini berfungsi untuk mengukur tegangan baterai atau membaca *level* tegangan baterai. Resistor 4k7 dan 1k berfungsi untuk merubah  $V_{in}$  12v Menjadi 2,1v

dengan cara menggunakan rumus pembagi tegangan yaitu :  $\frac{R_{Ground}}{R_{total}} \times V_{in}$  agar

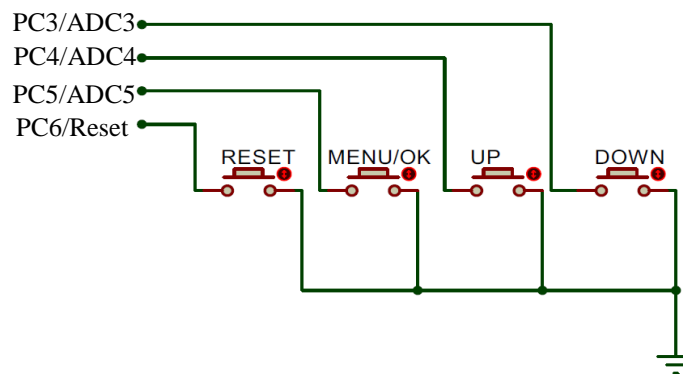
tegangan *input* yang masuk ke ADC tidak melebihi tegangan referensi yaitu 5v.

kemudian untuk mengetahui tegangan *output* ADC yaitu menggunakan rumus :

$$ADC = \frac{V_{input}}{V_{referensi}} \times \text{Lebar data} .$$

### 3.6.5 Rangkaian Push Button

Pada gambar 3.8 ini merupakan Rangkaian dari *Push Button*.

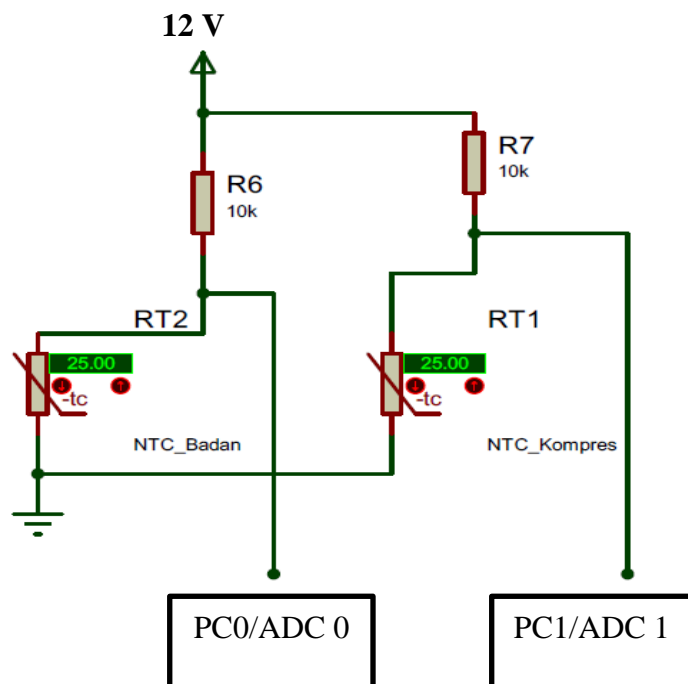


Gambar 3. 8 Rangkaian *Push Button*

Mengacu pada gambar 3.8 dapat dijelaskan rangkaian *push button* berfungsi sebagai tombol untuk menjalankan alat. setiap salah satu kaki *push button* mendapatkan *input* dari rangkaian minimum sistem, dan dimasing-masing kaki yang lain mendapatkan ground.

### 3.6.6 Rangkaian Sensor suhu NTC (*Negative Temperature Coefisien*)

Pada Gambar 3.9 ini merupakan Rangkaian dari *Sensor Suhu NTC*

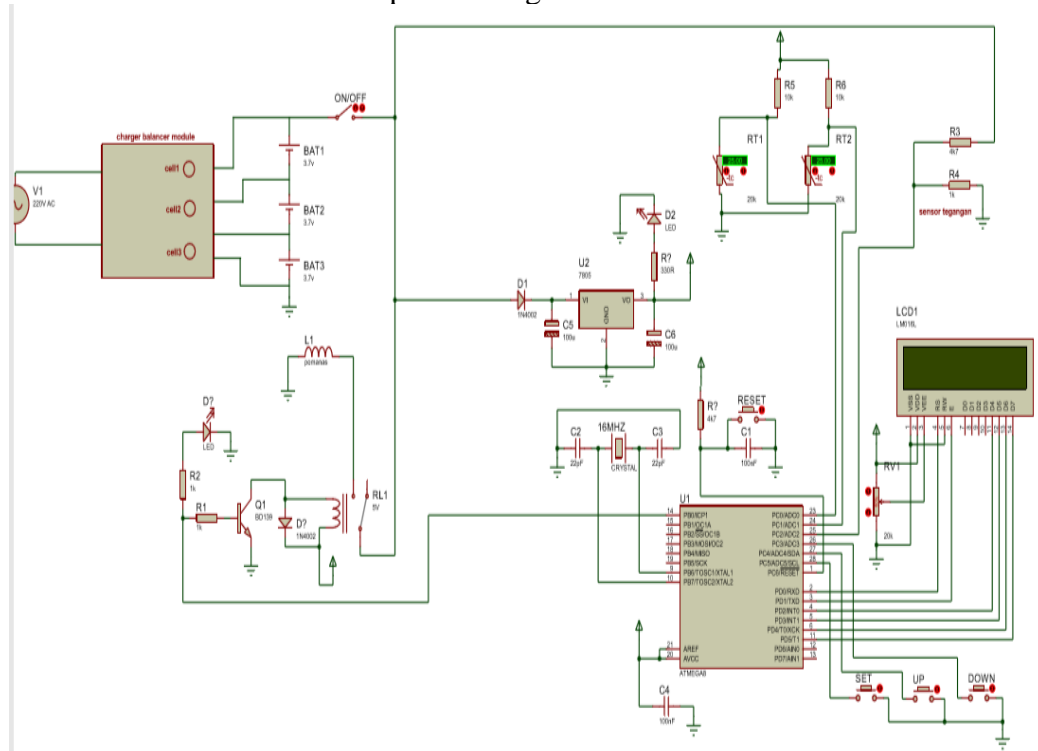


Gambar 3. 9 Rangkaian *Sensor NTC*

Mengacu pada Gambar 3.9 terdapat 2 *sensor NTC* yang berfungsi untuk merubah suhu menjadi resistansi, kemudian *sensor* tersebut dipasang secara seri dengan 2 resistor 10k yang berfungsi untuk mengubah resistansi menjadi tegangan. *output* dari ADC 0 merupakan *sensor* suhu badan yang berfungsi untuk monitoring suhu tubuh. *output* dari ADC1 merupakan *sensor* suhu kompres yang berfungsi untuk memonitoring suhu yang ada di *peltier*.

### 3.6.7 Rangkaian Keseluruhan

Pada Gambar 3.9 merupakan Rangkaian Keseluruhan dari modul TA.



Gambar 3. 10 Rangkaian Keseluruhan

Mengacu pada Gambar 3.9 dapat dijelaskan input PLN 220VAC menjadi *supply* tegangan utama untuk mengaktifkan rangkaian *charger 3 cell battery (By Pass Supply)*, jumlah tegangan yang dibutuhkan adalah sebesar 11,1 VDC maka dibutuhkan 3 baterai dengan masing-masing baterai *output* tegangannya 3,7 VDC/1200 mAh untuk memenuhi kebutuhan arus yang besar sebagai *supply peltier* ketika *aktif*. Saklar *ON/OFF* ditekan ke posisi ON maka *output* rangkaian baterai akan *mensupply* rangkaian *regulator* sebagai *supply* utama beberapa rangkaian diantaranya adalah rangkaian *system minimum*, *sensor* suhu, *sensor* tegangan, *relay* otomatis, dan LCD. Pada rangkaian *regulator* ini dibutuhkan tegangan keluaran/*output* 5VDC oleh karena itu pada rangkaian ini menggunakan IC *Regulator* 7805 dan beberapa kapasitor elektrolit untuk menghasilkan tegangan

DC dengan *noise* yang rendah. *Sensor* tegangan DC digunakan untuk memonitoring *output* baterai dan sebagai indikator *level* tegangan baterai, dimana tegangan baterai ini didapat dari hasil perhitungan 2 resistor (R.4K7 dan R.1K) menggunakan rumus pembacaan sensor tegangan yang sudah ditentukan nilai tegangan maksimal sesuai dengan nilai settingnya, *output sensor* tegangan ini masih *analog* dan perlu dirubah menjadi data *digital* agar dapat ditampilkan di LCD, maka *output* rangkaian sensor tegangan ini masuk ke *input* ADC IC Mikrokontroler menggunakan resolusi pembacaan ADC 10 *Byte*. *Push Button* pada alat terdapat 3 *push button* yang masing-masing mempunyai fungsi sendiri, ketiga *push button* *disetting up/ down*, dimana *push button* ini digunakan untuk memasukkan nilai setting alat (suhu dan *timer*) dan untuk memulai *running* alat. Ketika alat sudah *disetting* suhu dan *timernya*, lalu tombol *start* di tekan maka alat akan bekerja, *output* PB.0 memberikan sinyal keluaran 1 untuk *mentrigger common basis* pada *transistor* BD139 sehingga *common collector transistor* BD139 akan menyaklar jalur *ground* pada *common emittor transistor* BD139, sehingga dioda pada *relay* akan *aktif* karena *ditrigger negatif* pada kaki anodanya sedangkan kaki katodanya sudah terhubung ke *positif* +5VDC, maka *relay* akan aktif dan katup COM akan berpindah jalur ke katup NO sehingga akan menyalakan *peltier*, dimana *supply* tegangan *peltier* adalah langsung dari *output* rangkaian baterai 11,1 VDC. Selama proses awal berjalan adalah pencarian nilai suhu yang ditentukan sesuai dengan nilai *setting* di alat, suhu pada *peltier* akan dipantau menggunakan *sensor* NTC, data *output sensor* NTC menjadi *input* data ADC sensor suhu *peltier*, ketika nilai suhu tercapai maka *output* PB.0 menjadi

sinyal keluaran 0 sehingga akan memutuskan kaki COM *relay* ke kaki NC, *peltier* akan mati lalu *timer count down* akan bekerja sesuai dengan waktu yang *disetting*, pada proses ini *sensor* suhu badan *pasien* akan memantau hasil pembacaan ketika suhu *peltier* sudah sesuai dengan *setting* suhu yang ditentukan, pembacaan ini dilakukan sampai *timer* habis. Ketika *timer* selesai maka *buzzer* akan berbunyi sebagai tanda waktu kompres telah selesai.

### 3.7 Pembuatan Program

Program yang digunakan dalam pembuatan kompres elektrik ini adalah program pada aplikasi AVR. Pada listing 3.1 merupakan program Inisialisasi Awal Alat.

```
// Global enable interrupts
#asm("sei")

1.lcd_clear(); //Refresh Layar LCD
2.lcd_gotoxy(0,0); //Karakter mulai dari baris 0 kolom 0//
3.lcd_putsf("Kompres Elektrik"); //Karakter yang muncul
4.lcd_gotoxy(0,1); //Karakter mulai dari baris 1 kolom 0
5.lcd_putsf("By: Sandra"); //Karakter yang muncul
6.delay_ms(1000); //tunda waktu selama 1000 ms
```

Listing 3. 1 Listing Program Inisialisasi Awal Alat

#### Penjelasan Listing 3.1

Untuk *merefresh* layar LCD terdapat pada line 1. untuk memulai karakter dari baris 0 kolom 0 terdapat pada line 2. untk mengganti karakter yang muncul di awal terdapat pada line ke 3. Untuk menambahkan karakter mulai dari baris 1

kolom 0 terdapat pada line ke 4. Untuk menambahkan karakter yang muncul terdapat pada line ke 5. Untuk melakukan delay 1000 ms terdapat pada line ke 6.

Pada listing 3.2 merupakan Program Deklarasi Variable, Fungsi, dan Type Data.

```

1. #include <mega8.h> // bagian dr header//library// sebuah
perintah yang digunakan untuk memasukkan file *.h yg berisi
perintah" atau konstanta yg digunakan dalam membuat program.
4. #include <stdio.h> // bagian dr header//standart i/o library.
maka semua perintah dan konstanta yg ada dalam file stdio.h dapat
diakses dlm project yg dibuat.
7. #include <delay.h> // bagian dari header// delay.h library//
maka semua perintah dan konstanta yg ada dalam file delay.h dapat
diakses dlm project yg dibuat.
10.#include <math.h> // bagian dr header/ math.h library/ maka
semua perintah dan konstanta yg ada dalam file math.h dapat diakses
dlm project yg dibuat.
13. #define s1 PINC.5 // define merupakan perintah untuk
mendefinisikan nilai,alamat/ dll. s1 PINC.5 artinya s1 merupakan
tombol MENU yang terdapat di PC5 (portc.5)
16. #define s2 PINC.4 // define merupakan perintah untuk
mendefinisikan nilai,alamat/ dll. s2 PINC.4 artinya s2 merupakan
tombol UP yang terdapat di PC4 (portc.4)
19. #define s3 PINC.3 // define merupakan perintah untuk
mendefinisikan nilai,alamat/ dll. s3 PINC.3 artinya s3 merupakan
tombol DOWN yang terdapat di PC3 (portc.3)
22. #define out PORTB.0 // define merupakan perintah untuk
mendefinisikan nilai,alamat/ dll. out PORTB.0 artinya out
merupakan output relay yang terdapat di PORTB.0
25. #define buzzer PORTB.1 // define merupakan perintah untuk
mendefinisikan nilai,alamat/ dll. buzzer PORTB.1 artinya buzzer
terdapat di PORTB.1
28. #define sensortegangan 2 // sensortegangan 2 artinya
sensortegangan berada di 2/PC2
30. #define sensorkompres 1 // sensorkompres 1 artinya
sensorkompres berada di 1/PC1
32. #define sensorbadan 0 // sensorbadan 0 artinya
sensorbadan berada di 0/PC0

// resistansi pada 25 derajat C
35. #define THERMISTORNOMINAL 10000 // THERMISTORNOMINAL 10000
artinya THERMISTORNOMINAL adalah 10k
// temperatur untuk nominal resistansi ( hampir selalu 25 C )
38. #define TEMPERATURENOMINAL 25 // TEMPERATURENOMINAL 25
artinya TEMPERATURENOMINAL adalah 25// tampilan suhu awal di alat
mulai dari 25
41. #define NUMSAMPLES 5 // NUMSAMPLES 5 artinya NUMSAMPLES adalah
5
// The beta coefficient of the thermistor (usually 3000-4000)

```

Lanjut

Lanjut

```

44. #define BCOEFFICIENT 3950 / BCOEFFICIENT 3950 artinya
BCOEFFICIENT adalah 3950
46. #define SERIESRESISTOR 10000 // SERIESRESISTOR 10000 artinya
SERIESRESISTOR adalah 10k

// Alphanumeric LCD functions
49. #include <alcd.h> // bagian dari header// alcd.h library//
maka semua perintah dan konstanta yg ada dalam file alcd.h dapat
diakses dlm project yg dibuat.

// Declare your global variables here

53. char buffer[33]; // char = tipe data / digunakan untuk
menentukan jenis nilai dalam variabel. karakter/ lcd 2x16=32 +1(di
program) =33
56. int time[]={5,10,15,20}; // int = tipe data. tipe data time =
5,10,15,20
58. int detik=0,menit=0,start=0; // int = tipe data . tipe data
detik=0 menit=0 start=0
60. eeprom int timer=0; // variabel= memory/tempat untuk menyimpan
data. eeprom= jenis memory yg digunakan untuk menyimpan data yg
bertujuan untuk diubah-ubah.dan nilainya akan tersimpan meskipun
catu daya dimatikan.
64. eeprom int suhu=35; // memory tipe data suhu = 35

65. unsigned int samples[NUMSAMPLES]; // unsigned int=tipe data.
tipe data samples = [NUMSAMPLES]

```

Listing 3. 2 Program Deklarasi Variable, Fungsi, dan Type Data

### Penjelasan Listing 3.2

Untuk membuat sebuah perintah atau konstanta yang ada dalam file mega8.h agar dapat diakses maka terdapat pada line ke 1. Untuk membuat sebuah perintah atau konstanta yang ada dalam file stdio.h agar dapat diakses maka terdapat pada line ke 4. Untuk membuat sebuah perintah atau konstanta yang ada dalam file delay.h agar dapat diakses maka terdapat pada line ke 7. Untuk membuat sebuah perintah atau konstanta yang ada dalam file math.h agar dapat diakses maka terdapat pada line ke 10. Untuk membuat perintah untuk mendefinisikan tombol MENU yang terdapat pada PORTC.5 maka terdapat pada

line 16. Untuk membuat perintah untuk mendefinisikan tombol UP yang terdapat pada PORTC.4 maka terdapat pada line 19. Untuk membuat perintah untuk mendefinisikan tombol DOWN yang terdapat pada PORTC.3 maka terdapat pada line 22. Untuk membuat perintah untuk mendefinisikan output relay yang terdapat pada PORTB.0 maka terdapat pada line 25. Untuk membuat perintah untuk mendefinisikan *output buzzer* yang berada pada PORTB.1 maka terdapat pada line 28. Untuk membuat perintah untuk mendefinisikan Sensor Tegangan berada di PORTC.2 maka terdapat pada line 30. Untuk membuat perintah untuk mendefinisikan Sensor Kompres berada di PORTC.1 maka terdapat pada line 32. Untuk membuat perintah untuk mendefinisikan Sensor Badan berada di PORTC.0 maka terdapat pada line 35. Untuk membuat perintah untuk mendefinisikan THERMISTORNOMINAL adalah 1000 maka terdapat pada line 38. Untuk membuat perintah untuk mendefinisikan TEMPERATURENOMINAL adalah 25 maka terdapat pada line 41. Untuk membuat perintah untuk mendefinisikan NUMSAMPLES adalah 5 maka terdapat pada line 44. Untuk membuat perintah untuk mendefinisikan BCOEFFICIENT adalah 3950 maka terdapat pada line 46. Untuk membuat perintah untuk mendefinisikan SERIESRESISTOR adalah 10000 maka terdapat pada line 53. Untuk membuat sebuah perintah atau konstanta yang ada dalam file alcd.h agar dapat diakses maka terdapat pada line ke 56. Untuk menentukan jenis nilai dalam variable / karakter lcd 2x16 maka terdapat pada line 58. Untuk menentukan isi tipe data time terdapat pada line 39. untuk menentukan isi tipe data detik terdapat pada line 60. Untuk membuat variable atau memory yang



digunakan untuk menyimpan data terdapat pada line 64. Untuk memanggil variabel dari tipe data unsigned samples maka terdapat pada line ke 65.

Pada listing 3.3 Merupakan Listing Program Utama Keseluruhan Alat

```

1. void programutama(){ // void merupakan bagian dari fungsi yang
akan menjalankan perintah dari inialisasi programutama.
programutama merupakan nama fungsi.
4. float s_badan=bacasuhu(0); // artinya memanggil fungsi dari
s_badan // float merupakan tipe data. s_badan=bacasuhu(0)
merupakan bagian dari fungsi.
7. float s_kompres=bacasuhu(1); // artinya memanggil fungsi dari
s_kompres // float merupakan tipe data. s_kompres=bacasuhu(1)
merupakan bagian dari fungsi.
10. float s_volt=read_volt(); // artinya memanggil fungsi dari
s_volt // float merupakan tipe data. s_volt=read_volt () merupakan
bagian dari fungsi.

13. if(s1==0) setting(); // if merupakan bagian dari kondisi //
memanggil fungsi// jika s1 adalah 0 maka setting
15. if(s1==0||s2==0||s3==0){ // memanggil fungsi// jika s1 adalah
0 s2 adalah 0 s3 adalah 0 makabuzzer adalah 1
17. buzzer=1; } // buzzer adalah 1
18. if(s3==0&&start==0){ // memanggil fungsi// jika s3 adalah 0
dan(logika) start adalah 0
20. menit=time[timer]; //menit adalah waktu (timer)
21. detik=0; // detik sama 0
22. start=1; //start adalah 1
23. out=1;} // out adalah 1
24. if(start==1){ //memanggil fungsi// jika start sama dengan 1
25. if(s_kompres>suhu)start=2;} //memanggil fungsi
(s_kompres>suhu) // jika (s_kompres lebih besar dari suhu maka
start sama dengan 2
28. if(start==2){ //memanggil fungsi dari start // jika
start maka sam dengan 2
30. if(s_kompres>suhu)out=0; //memanggil fungsi dari_kompres //
jika s_kompres lebih besar dari suhu out bernilai 0
32. if(s_kompres<(suhu-1))out=1; //memanggil fungsi dari_kompres
//jika s_kompres lebih kecil maka suhu dikurang satu out bernilai1
34. if(menit==0&&detik==0){ //memanggil fungsi// jika menit sama
dengan 0 dan(logika) detik sama dengan 0
36. out=0; //out adalah 0
37. start=0; // start adalah 0
38. buzzer=1; // buzzer adalah 1
39. delay_ms(2000); // jeda waktu selama 2000ms
40. buzzer=0;} } // buzzer adalah 0

41. lcd_clear(); //Refresh LCD
42. lcd_gotoxy(0,0); //Karakter mulai dari baris 0 kolom 0
43. sprintf(buffer,"K:%.1f",s_kompres);
44. lcd_puts(buffer); // menampilkan STRING yang disimpan pada
SRAM pada LCD.

```

Lanjut

Lanjut

```

46. lcd_putchar(0xdf); // menampilkan karakter c pada LCD (char c)
47. lcd_putchar('C'); // menampilkan karakter c pada LCD (char c)
48. lcd_gotoxy(10,0); //Karakter mulai dari baris 0 kolom 0
49. if(start==0)lcd_putsf("STOP"); // Memanggil fungsi // jika
start sama dengan 0 maka karakter yang muncul adalah STOP
51. if(start==1)lcd_putsf("HEATER"); // memanggil fungsi // jika
start sama dengan 1 maka karakter yang muncul adalah HEATER
53. if(start==2){ //Memanggil fungsi // jika start sama
dengan 2
55. sprintf(buffer,"%02d:%02d",menit,detik);
56. lcd_puts(buffer);} // menampilkan STRING yang disimpan pada
SRAM pada LCD
58. lcd_gotoxy(0,1); //Karakter mulai dari baris 0 kolom 1
59. sprintf(buffer,"B:%.1f",s_badan);
60. lcd_puts(buffer); // menampilkan STRING yang disimpan pada
SRAM pada LCD.
62. lcd_putchar(0xdf); // menampilkan karakter c pada LCD (char c)
63. lcd_putchar('C'); // menampilkan karakter c pada LCD (char c)
64. lcd_gotoxy(10,1); //Karakter mulai dari baris 10 kolom 1
65. sprintf(buffer,"%1fv",s_volt);
66. lcd_puts(buffer); // menampilkan STRING yang disimpan pada
SRAM pada LCD.
68. delay_ms(100); //merupakan jeda waktu selama 100ms
69. buzzer=0;} // buzzer sama dengan 0

```

Listing 3. 3 Program Utama Keseluruhan Alat

## Penjelasan Listing 3.3

Pada line ke 1 *Void* merupakan bagian dari fungsi yang akan menjalankan perintah dari inialisasi program utama. program utama merupakan nama fungsi. Untuk memanggil fungsi dari *s\_badan* terdapat pada line ke 4. Untuk memanggil fungsi dari *s\_kompres* terdapat pada line ke 7. Untuk memanggil fungsi dari *s\_volt* terdapat pada line ke 10. Pada line ke 13 terdapat sebuah kondisi jika *s1* sama dengan 0 maka setting. Pada line ke 15 terdapat sebuah kondisi jika *s1* sama dengan 0 *s2* sama dengan 0 *s3* sama dengan 0. Maka output buzzer adalah 1 terdapat pada line ke 17. Pada line ke 18 terdapat sebuah kondisi jika *s3* sama dengan 0 dan *start* sama dengan 0. Pada line ke 20 variabel timer akan menyimpan *menit* sama dengan *time*, *detik* sama dengan 0 terdapat pada line ke 21, *Start* sama dengan 1 terdapat pada line ke 22. Outputan timer sama dengan 1 terdapat pada

line ke 23. Pada line ke 24 terdapat kondisi jika start sama dengan 1. Pada line ke 25 terdapat kondisi jika s\_kompres lebih besar dari suhu start sama dengan 2. Pada line ke 28 terdapat sebuah kondisi jika start sama dengan 2. Pada line ke 30 terdapat kondisi jika s\_kompres sama lebih besar dari suhu maka out sama dengan 0. Pada line ke 32 terdapat sebuah kondisi jika s\_kompres lebih kecil suhu dikurang 1 output sama dengan 1. Pada line ke 34 terdapat sebuah kondisi jika menit sama dengan 0 dan detik sama dengan 0. pada line ke 36 out sama dengan 0. Pada line ke 37 start sama dengan 0. Pada line ke 38 buzzer sama dengan 1. Pada line 39 delay\_ms (2000) merupakan jeda waktu selama 2000 ms. Buzzer sama dengan 0 terdapat pada line ke 40. Pada line ke 41 berfungsi untuk merefresh LCD. Untuk memulai dari baris 0 kolom 0 terdapat pada line ke 42. Untuk menampilkan STRING yang disimpan pada SRAM pada LCD terdapat pada line ke 43. Untuk menampilkan karakter c pada LCD (char c) terdapat pada line 44. Untuk memulai karakter mulai dari baris 10 kolom 0 terdapat pada line ke 46. Untuk menampilkan karakter c pada LCD (char c) terdapat pada line 47 & 48. Pada line ke 49 merupakan kondisi jika start sama dengan 0 maka karakter yang muncul adalah STOP . Pada line ke 51 merupakan kondisi jika start sama dengan 1 maka karakter yang muncul adalah HEATER. Pada line ke 53 merupakan kondisi jika start sama dengan dua. Pada line ke 58 merupakan perintah untuk menampilkan karakter baris 0 kolom 1. untuk menampilkan STRING yang disimpan pada SRAM pada LCD terdapat pada line ke 60. Pada line ke 62 dan 63 merupakan perintah untuk menampilkan karakter c pada LCD (char c). Pada line ke 64 terdapat perintah karakter mulai dai baris 10 kolom 1. Pada line ke 66

merupakan perintah untuk menampilkan STRING yang disimpan pada SRAM pada LCD. pada line ke 68 merupakan perintah delay 100ms. Pada line ke 69 merupakan perintah buzzer sama dengan 0.

### 3.9 Alat Pemanding (DPM4 PARAMETER TESTER)

Uji fungsi yang penulis lakukan disini adalah membandingkan tampilan nilai suhu yang ada di Prototyp dengan nilai suhu yang ditampilkan oleh alat pemanding (DPM4 Parameter Tester). Spesifikasi alat yang digunakan sebagai pemanding adalah sebagai berikut :

Nama Alat : DPM4 PARAMETER TESTER

Merk : FLUKE Biomedical

Range Temperature : -40° C sampai 200° C

Ukuran : 12,5 cm x 9,3 cm

Tampilan : LCD

Identitas Alat : Milik Rumah Sakit Gadjah Mada Yogyakarta

Pada gambar 3.10 merupakan Alat Pemanding DPM4 PARAMETER TESTER



Gambar 3. 11 Alat Pemanding

### 3.10 Spesifikasi Prototype

Nama	: Alat Kompres Elektrik
Tegangan	: 12,6 VDC
Daya	: 45,36 Watt\
<i>Range Temperature</i>	: 41 °C - 45 °C
<i>Timer</i>	: 10,15,20 menit
<i>Display</i>	: LCD karakter 16x2
<i>Heater</i>	: <i>Peltier</i>
Dimensi	: Panjang =15,5 Cm Lebar =15,5 Cm

Pada gambar 3.11 merupakan Prototype Kompres Elektrik



Gambar 3. 12 Prototype Kompres Elektrik