

BAB III

METODOLOGI PENELITIAN

Pada bab ini dijelaskan langkah-langkah yang digunakan dalam menyelesaikan perangkat keras (*hardware*) yang berupa komponen fisik sebagai penunjang seperti IC ATmega 16, selain menjelaskan perangkat keras pada bab ini juga menjelaskan perangkat lunak (*software*) yang berisikan program yang digunakan untuk alat *water bath*. Adapun pelaksanaannya dilakukan dengan cara menentukan spesifikasi secara umum, melakukan perancangan dan realisasi perangkat keras (*hardware*) dan perangkat lunak (*software*).

3.1. Perancangan Rangkaian Alat *Water Bath*

3.1.1. Spesifikasi Fungsi

Sebelum merealisasikan perencanaan-perencanaan yang dijabarkan nanti, maka terlebih dahulu dijabarkan tentang spesifikasi dari alat ini. Rangkaian *Water Bath* ini dirancang untuk dapat menghasilkan suhu yang konstan sesuai dengan yang dibutuhkan oleh pengguna alat.

3.1.2. Spesifikasi Alat

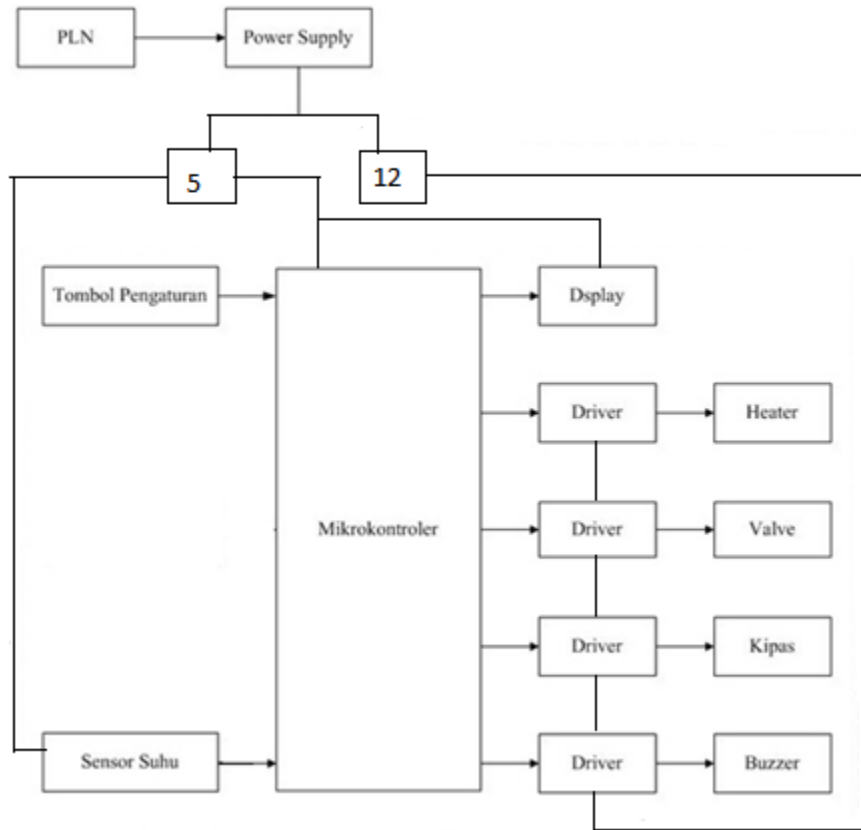
Untuk merealisasikan proses pelaksanaannya, maka perlu dirancang rangkaian-rangkaian yang sesuai dengan spesifikasi yang diperlukan, seperti:

1. Tegangan jala-jala yang dibutuhkan 220 V AC.
2. Catu daya yang dibutuhkan sebesar 12 V dan 5 V DC.
3. *Liquid Cristal Display* (LCD) 2 x 16 sebagai *display*.
4. ATmega 16 sebagai kontrol/pengendali sistem.
5. *Solenoid Valve* sebagai kran pembuangan air secara otomatis.
6. Sensor LM35 sebagai pendeteksi suhu.

3.2. Perancangan Rangkaian Secara Diagram Blok

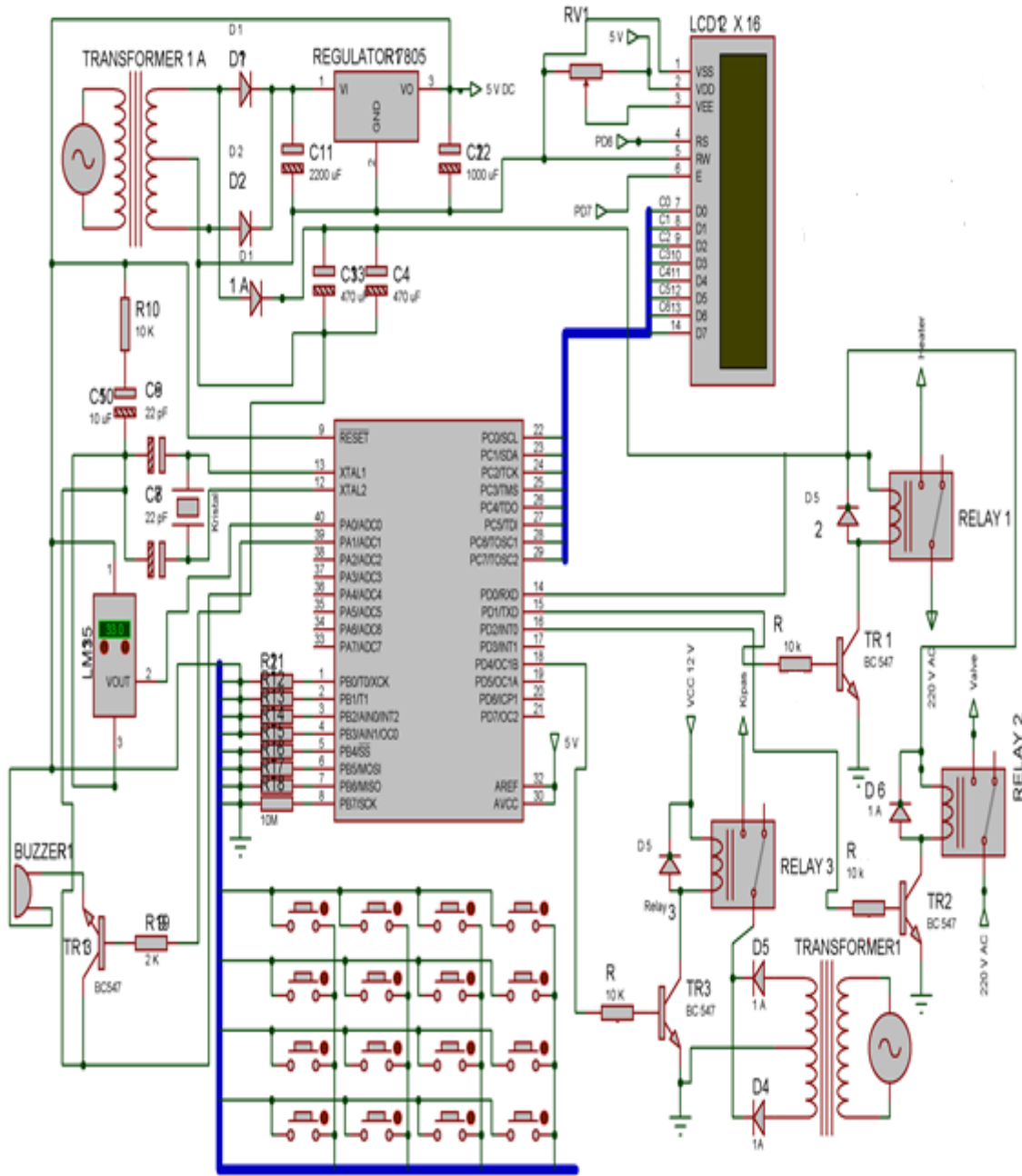
Pertama-tama tegangan dari jala-jala PLN menjadi masukan pada *power supply*, *power supply* akan menghasilkan dua keluaran yaitu 12 V DC dan 5VDC. Keluaran 5VDC menyuplai tegangan untuk rangkaian mikrokonroller, sensor suhu, LCD dan sedangkan

keluaran 12VDC menyuplai tegangan untuk relay dimana relay ini berfungsi untuk menghidupkan heater dan valve. Diagram blok perancangan alat *water bath* dapat dilihat pada Gambar 3.1.



Gambar 3.1. Diagram Blok Water Bath

3.3. Rangkaian Keseluruhan

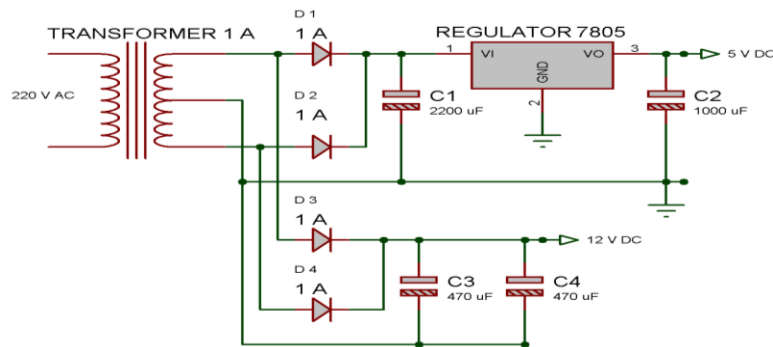


Gambar 3.2 rangkaian

3.4 Perancangan Perangkat Keras dan Program yang Digunakan

3.5.1. Perancangan Rangkaian *Power Supply*

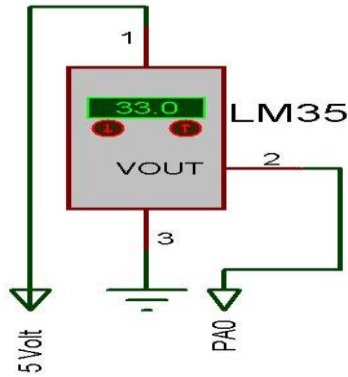
Rangkaian *power supply* ini berfungsi untuk memberikan tegangan ke seluruh rangkaian pada alat *water bath* sesuai dengan yang dibutuhkan oleh masing-masing rangkaian. Pada rangkaian *power supply* ini menggunakan beberapa komponen antara lain trafo CT, 4 buah dioda 1 A, 4 buah kapaitor 470 μF , dan 1 buah regulator 7805 untuk menghasilkan keluaran 5 V DC dan 12 V DC. Perancangan rangkaian *power supply* dapat dilihat pada Gambar 3.2.



Gambar 3.3. Rangkaian *Power Supply*

3.5.2. Perencanaan Rangkaian Kontrol Suhu dan Program Pengolahan Suhu

Dalam perencanaan rangkaian kontrol suhu pada alat *water bath* merupakan rangkaian yang berfungsi untuk mengubah besaran suhu menjadi besaran listrik dalam bentuk tegangan. Sensor LM35 mempunyai tegangan keluaran yang linear terhadap suhu dalam derajat celcius. Artinya dalam setiap kenaikan suhu 1 °Celcius maka akan terjadi perubahan tegangan pada *output* sensor sebesar 10 mV. Pada kaki 2 sensor LM35 akan terhubung dengan PORT0 pada mikrokontroler. Besar tegangan input pada PORT0 tergantung dari suhu yang dideteksi oleh sensor LM35. Bentuk rangkaian dari sensor LM35 dapat dilihat pada Gambar 3.3.



Gambar 3.4. Bentuk Rangkaian Sensor LM35

Proses pengolahan suhu dengan sensor LM35 menghasilkan tegangan sebagai keluaran dari LM35. Sensor LM35 akan mendeteksi suhu $-55\text{ }^{\circ}\text{C}$ sampai $150\text{ }^{\circ}\text{C}$ dengan skala keluaran sebesar $10\text{ mV}/^{\circ}\text{C}$. Pada saat sensor LM35 pada suhu $+150\text{ }^{\circ}\text{C}$ akan menghasilkan keluaran tegangan $+1500\text{ mV}$, pada suhu $+25\text{ }^{\circ}\text{C}$ akan menghasilkan keluaran tegangan sebesar $+250\text{ mV}$, pada suhu $-55\text{ }^{\circ}\text{C}$ akan menghasilkan keluaran tegangan sebesar -550 mV .

Untuk mendapatkan tampilan suhu pada alat dilakukan penghitungan keluaran dari LM35 dengan pin ADC, dimana dalam program ditulis dengan sintak sebagai berikut:

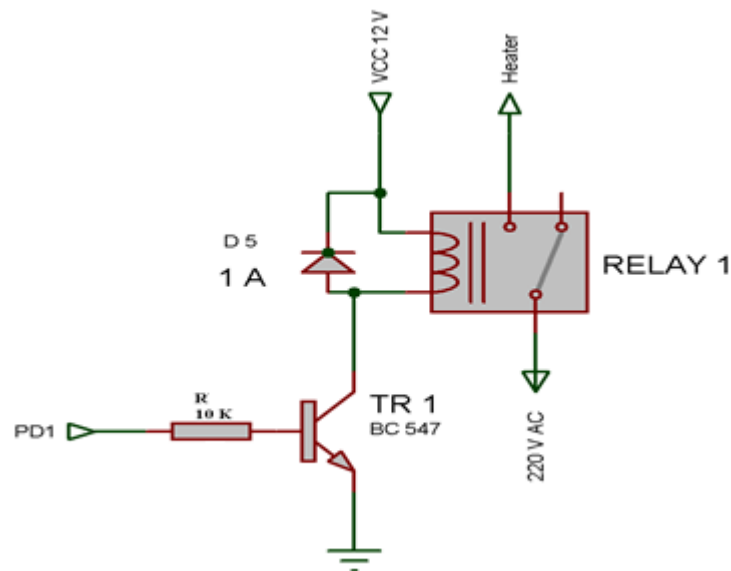
```
ADMUX=0;
r=0;
for (c=0;c<250;c++)
    {ADCSRA=(1<<ADEN)|(1<<ADSC)|(5<<ADPS0);
    while ((ADCSRA & 0x10) == 0);
    {ADCSRA=(1<<ADIF);
    r+=ADC;}
    }
c=r/25;
```

Program sintak adalah program untuk pembuatan rumus untuk mendapatkan tegangan keluaran dari sensor LM35 berdasarkan suhu di sekitarnya. Tegangan keluaran dari hasil perhitungan tersebut selanjutnya diubah menjadi tegangan, dimana skala keluaran dari sensor LM35 adalah $10\text{ mV}/^{\circ}\text{C}$ untuk melakukan perubahan tegangan menjadi suhu.

PORTA0 pada mikrokontroler selain menerima inputan dari output sensor LM35, juga sebagai jalur untuk pengambilan data ADC, untuk pemilihan jalur dilakukan dengan memberikan nilai pada register ADMUX. ADC dikerjakan pada mode *single conversion*, sehingga setiap akan memulai proses untuk konversi maka register control ADC (ADCSRA) harus diberi nilai untuk memulai proses konversi. Program akan menunggu Bit ADIF yang bernilai 1 yaitu proses konversi selesai dan data telah di-*update*.

3.5.3. Perencanaan Rangkaian *Driver Heater* dan Program Penggerak *Heater*

Perancangan rangkaian *driver heater* atau pengendali *heater* merupakan rangkaian yang berfungsi sebagai pengendali *heater* yang bekerja apabila kaki PD1 pada mikrokontroler memberikan *input* pada *driver heater* seperti yang ditunjukkan pada Gambar 3.4.



Gambar 3.5. Rangkaian *Driver Heater*

Rangkaian *driver heater* menggunakan *relay* sebagai pensaklar untuk memutus dan menyambung tegangan 220 AC yang akan disuplai ke *heater*. Selain menggunakan *relay*, pada rangkaian *driver* ini juga menggunakan transistor jenis NPN yang berfungsi untuk untuk memutus dan menyambung *ground* yang akan diloloskan ke *relay*, sehingga bekerjanya *relay* tergantung pada transistor. Jika basis pada transistor mendapatkan tegangan 5 V DC dari kaki PD1 mikrokontroler (dengan hambatan arus 2 K Ω) maka *ground* akan diloloskan dari kaki kolektor ke emitor

yang kemudian diteruskan ke *relay* sehingga *relay* bekerja. Dengan bekerjanya *relay*, tegangan 220 V AC akan diloloskan ke *heater* sehingga *heater* bekerja. Kode program yang digunakan untuk penggerak *heater* sebagai berikut:

```
c=(r*500)/1023;

if ((menu==2)&(c>set))

{PORTD|=0b00001000; PORTD&=0b11111101;

if (got==0)

    {got=1;

    TCCR1B=(1<<WGM12); TCCR1B|=(1<<CS12);} }

else if ((menu==2)&(c>(set-5))&(got==1))

{PORTD&=0b11111101;}

else if ((menu==2)&(c<(set-5)))

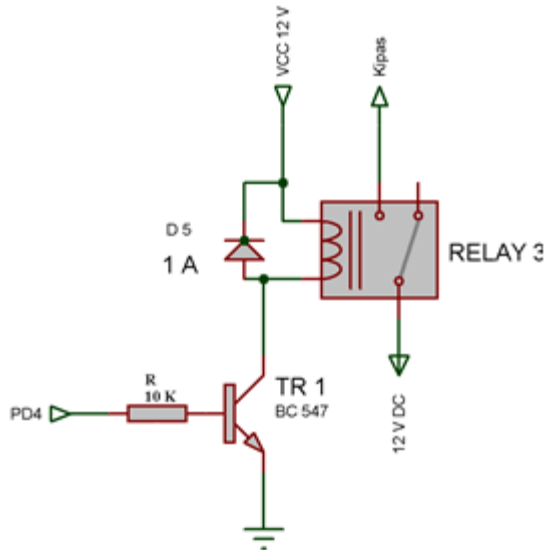
{PORTD|=0b00000010;    PORTD&=0b11110111;}

}
```

Bekerjanya rangkaian *driver heater* tergantung dari pemberian logika pada masing-masing keluaran yang terhubung dengan penggerak pada *heater*. Jika dilakukan pemberian logika 1 pada pin keluaran yang terhubung dengan penggerak maka *driver heater* yang mendapatkan keluaran logika 1 akan aktif atau bekerja. Jika pemberian logika 0 pada keluaran yang terhubung dengan penggerak *diver heater* maka *heater* tidak aktif atau tidak bekerja. Keluaran pada mikrokontroler PD1 terhubung dengan penggerak *heater*.

3.5.4. Perencanaan Rangkaian *Driver Kipas*

Rangkaian kipas pada alat *water bath* merupakan rangkaian yang berfungsi untuk mendinginkan *heater* pada kondisi *heater* tidak bekerja ketika suhu sudah tercapai. Kipas akan bekerja berdasarkan suhu yang terdeteksi oleh sensor LM35. Pada saat suhu kurang dari suhu *setting* maka kipas akan berhenti bekerja dan akan bekerja kembali ketika suhu lebih dari suhu *setting*. Bentuk dari rangkaian *driver* kipas dapat dilihat pada Gambar 3.5.



Gambar 3.6 Rangkaian *Driver* Kipas

Pada rangkaian *driver* kipas menggunakan beberapa komponen seperti resistor 2 K Ω , transistor BC547, dioda 1 A, relay 12 V, dan kipas 12 V DC. Rangkaian *driver* kipas pada mikrokontroler terhubung dengan PORT D4. Bekerjanya rangkaian kipas tergantung dari sinyal yang dikeluarkan oleh mikrokontroler yang terhubung dengan rangkaian *driver* kipas. Jika mikrokontroler memberikan sinyal keluaran dengan logika 1 atau dengan tegangan 5 V DC. Maka basis pada transistor akan mendapatkan tegangan (dengan hambatan 2 K Ω), maka *groud* yang *standbay* pada transistor akan diteruskan ke *relay* sehingga *relay* bekerja. Untuk mengatasi terjadinya arus bolak balik pada *relay* maka digunakan dioda 1 A sebagai pengamanan pada *relay*.

```

if ((menu==2)&(c>set))
    {PORTD|=0b00001000; _delay_ms(100);
    PORTD&=0b11111101;
    if (got==0)
        {got=1;
        TCCR1B=(1<<WGM12); TCCR1B|=(1<<CS12);}
    }
else if ((menu==2)&(c>(set-5))&(got==1))
    {PORTD&=0b11111101;
    }
else if ((menu==2)&(c<(set-5)))

```

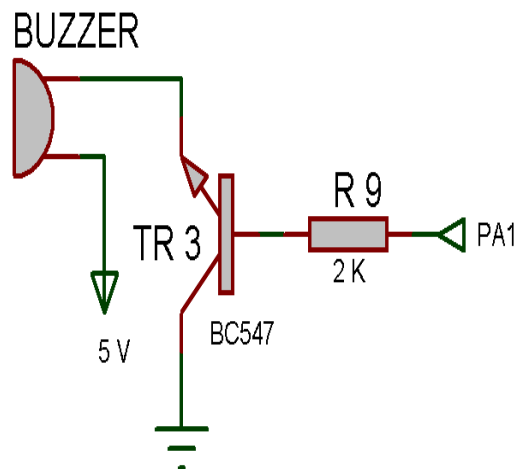


```
{PORTD|=0b00000010; _delay_ms(100);  
PORTD&=0b11110111;}
```

Untuk dapat menjalankan rangkaian *driver* kipas maka dibutuhkan sebuah kode program untuk mengatur kapan rangkaian *driver* kipas bekerja atau tidak.

3.5.5. Perancangan Rangkaian *Buzzer* dan Program.

Penggerak Pada perancangan ini *buzzer* berfungsi untuk indikator bunyi atau penanda apabila waktu pengaturan lamanya alat bekerja sudah selesai. *Buzzer* terhubung pada Port PA1 mikrokontroler, rangkaian *buzzer* menggunakan transistor NPN BC547. Pada dasarnya *buzzer* dihubungkan ke tegangan Vcc 5 Volt (dengan batasan arus oleh resistor 2 k Ω) karena adanya transistor, maka *buzzer* mendapatkan arus atau tidaknya tergantung dari kondisi transistor saat itu. Jika transistor ON (karena adanya arus rendah pada basis, dengan pemberian logika '0'), maka *buzzer* mendapatkan tegangan Vcc, namun sebaliknya jika transistor OFF (karena adanya arus tinggi pada basis, dengan pemberian logika '1'), maka *buzzer* juga OFF. Gambar rangkaian *buzzer* dapat dilihat pada Gambar 3.6.



Gambar 3.7. Rangkaian *Buzzer*

Bekerjanya *driver buzzer* tergantung dari pemberian logika pada mikrokontroler yang terhubung dengan penggerak pada *buzzer*. Jika dilakukan pemberian logika 1 pada pin keluaran yang terhubung dengan penggerak maka *driver buzzer* yang mendapatkan keluaran logika 1 akan aktif atau berbunyi. Jika pemberian logika 0 pada keluaran yang terhubung dengan penggerak maka penggerak yang

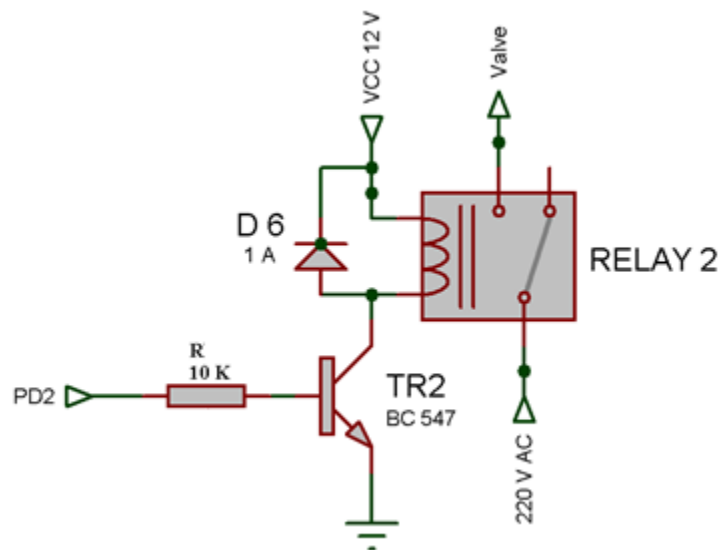
mendapatkan logika 0 tidak aktif atau tidak berbunyi. Keluaran pada mikrokontroler yang terhubung dengan penggerak *buzzer* PA1. Kode program yang digunakan untuk penggerak *buzzer* adalah sebagai berikut:

```
DDRA|=0b00000010;
```

```
DDRA&=0b11111101;
```

3.5.6. Perancangan Rangkaian *Driver Valve* dan Program Penggerak *Valve*

Perancangan rangkaian *driver valve* atau pengendali *valve* merupakan rangkaian yang berfungsi sebagai pengendali *valve* yang bekerja apabila kaki PD2 pada mikrokontroler memberikan *input* pada *driver valve*. Rangkaian *driver valve* dapat dilihat pada Gambar 3.7.



Gambar 3.8. Rangkaian *Driver Valve*

Prinsip kerja dari *driver valve* sama dengan prinsip kerja dari *driver heater* hanya memiliki perbedaan di inputan basis dan tegangan yang diloloskan pada kontaktor *relay*, jika pada *driver heater* keluaran dari kontaktor *relay* untuk mengaktifkan *heater* dan pada *driver valve* keluaran dari kontaktor *relay* untuk mengaktifkan *valve*. Dengan bekerjanya *valve* maka proses pembuangan air pada *chamber* alat berlangsung.

Bekerjanya *driver valve* tergantung dari pemberian logika pada masing-masing keluaran yang terhubung dengan penggerak pada *relay*. Jika dilakukan pemberian logika 1 pada pin keluaran yang terhubung dengan penggerak maka driver yang

mendapatkan keluaran logika 1 akan aktif atau bekerja. Jika pemberian logika 0 pada keluaran yang terhubung dengan penggerak maka penggerak yang mendapatkan logika 0 tidak aktif atau tidak bekerja.

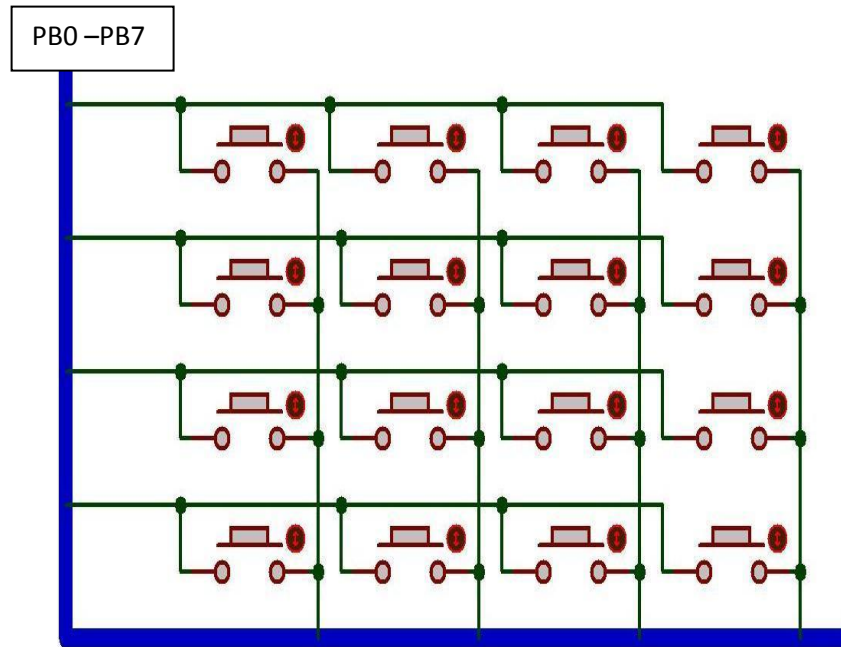
Keluaran pada mikrokontroler yang terhubung dengan penggerak *valve* PD2. Kode program yang digunakan untuk penggerak *valve* pada perancangan alat *water bath* adalah sebagai berikut:

```
DDRD|=0b00000100;  
DDRD&=0b11111011;
```

3.5.7. Perencanaan Rangkaian *Keypad* dan Program Pendeteksi *Keypad*.

Perencanaan rangkaian *keypad* pada alat *water bath* menggunakan jenis *keypad* matriks. *Keypad* matriks adalah tombol-tombol yang disusun secara maktriks (baris x kolom) sehingga dapat mengurangi penggunaan pin *input*. Dalam penggunaan *Keypad* Matriks 4×4 cukup menggunakan 8 pin untuk 16 tombol. Hal tersebut dimungkinkan karena rangkaian tombol disusun secara horizontal membentuk baris dan secara vertikal membentuk kolom. Prinsip kerja dari *keypad* matriks sebenarnya adalah menggunakan proses scanning aris dan Kolom. Pada *keypad* matriks 4×4 memiliki 4 buah baris dan 4 buah kolom.

Jika digunakan program scanning manual, maka salah satu dari baris atau kolom tersebut harus diatur sebagai *output* dan yang lainnya sebagai *input* (pada pin mikrokontroler yang terhubung dengan *keypad*). Bentuk dari rangkaian *keypad* dapat dilihat pada Gambar 3.8.



Gambar 3.9. Bentuk Rangkaian Keypad Matriks 4 x 4

Pada perancangan alat *water bath* ini pin pada keypad matriks akan terhubung dengan pin pada mikrokontroler pada PORTB0 sampai PORTB7. Agar tombol dapat terdeteksi oleh mikrokontroler maka pendeteksian tombol dilakukan dengan pertanyaan pada pin-pin PINA, pin-pin akan berlogika 0 jika pin-pin tersebut diberi tegangan yang mendekati 0 V dan berlogika 1 jika pin-pin tersebut diberi tegangan yang mendekati 5 V DC. Jika pin berlogika 1 dari sebelumnya yang berlogika 0, maka dinyatakan terjadi penekanan pada tombol. Penekanan diikuti dengan proses-proses yang mengikutinya berdasarkan keadaan pada saat penekanan.

Kode program yang digunakan untuk mendeteksi tombol-tombol adalah sebagai berikut:

```
unsigned char _key[]={8,4,2,1};
```

```
for (i=0;i<4;i++)
```

```
{k=_key[i];
```

```
key=0;
```

```
again:
```

```
PORTB=k;
```

```
asm("nop");
```

```

k=PINB&0b11110000;

j=k/16;

if (j!=0)
    {key=j;
    goto again;}
}

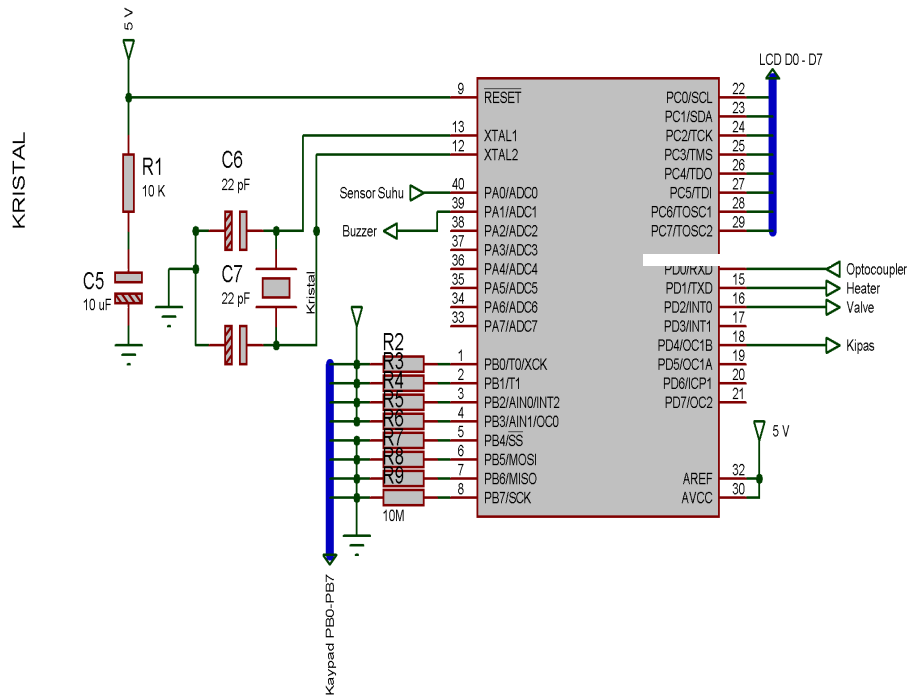
```

Proses *scanning* diulang sebanyak empat kali dengan variabel terdapat pada variabel *i*. Setiap kali *scanning* program mengambil satu data tabel *_key* untuk diumpungkan ke *PORTB*. Data tersebut merupakan data 8-bit dengan salah satu bit bernilai 1 dan terletak pada nibble bawah. *Scanning* selanjutnya mendeteksi data pada nibble atas, data salah satu bit pada nibble atas ikut bernilai 1 ketika salah satu tombol tertekan.

Pemasukan data dari tombol dilakukan melalui *PINB*, data dibagi dengan 16 untuk menggeser nibble atas agar berada pada nibble bawah. Jika didapatkan hasil *scanning* tidak bernilai 0 atau terdapat tombol yang tertekan maka program menyimpan data penekanan pada variabel *k*. Selanjutnya program mengulang *scanning* sampai dideteksi tombol dilepas.

3.5.8. Perancangan Sistem Mikrokontroler ATmega 16

Mikrokontroler pada perancangan alat *water bath* ini mengatur jalannya proses pemanasan air sehingga mendapatkan suhu air yang stabil serta sebagai penampil informasi selama bekerjanya alat. Mikrokontroler ATmega 16 akan memperoleh masukan dan keluaran yang ada pada peralatan ini, pengontrolan tersebut dilakukan melalui pengaktifan masing-masing pin pada kaki mikrokontroler tersebut, baik pengaktifan secara paralel ataupun secara per-bit mikrokontroler dalam setiap portnya. Untuk melakukan proses pengaktifan pada pin-pin atau port yang terdapat di dalam mikrokontroler tersebut dilakukan dengan menggunakan perangkat lunak (*software*) yang ditempatkan pada *flash* program memori internal tanpa menggunakan program memori eksternal. Proses perancangan mikrokontroler untuk alat ini dapat dilihat pada Gambar 3.10.



Gambar 3.10. Perancangan Mikrokontroler ATmega 16.

Untuk mengaktifkan Mikrokontroler ATmega 16 maka perlu diberikan tegangan catu daya + 5 V DC pada pin 10 dan pemberian tegangan nol (*ground*) pada pin 11. Disamping itu diperlukan juga pengaktifan osilator internal yang terdapat pada mikrokontroler. Untuk mengaktifkan osilator internal tersebut dalam perancangan ini digunakan kristal 12 MHz untuk memperoleh kecepatan pelaksanaan instruksi per-siklus sebesar 12 MHz.

Port PA0 digunakan untuk pembacaan interupsi *eksternal* dari rangkaian sensor suhu. Perancangan *water bath* ini menampilkan suhu yang diatur, waktu serta indikator-indikator, hasil pembacaan yang menggunakan LCD 2 x 16 sebagai *display*. Port PA1 digunakan untuk *control data buzzer*. Port PB0 sampai PB7 digunakan sebagai port data masukan *kaypad*. Port PC0 sampai PC7 digunakan sebagai port data untuk masukan LCD. Port PD1 digunakan untuk *control data heater/driver heater*. Port PD2 digunakan untuk *control data valve/driver valve* dan Port PD4 digunakan untuk kontrol kipas angin.

Untuk dapat melakukan pemakaian fasilitas yang ada pada mikrokontroler diperlukan program inisialisasi yang merupakan suatu program untuk mendefinisikan dan memulai pemakaian fasilitas pada mikrokontroler. Fasilitas-fasilitas yang

digunakan pada mikrokontroler berupa I/O, ADC, *Timer*, dan interupsi. Port I/O merupakan Port yang digunakan sebagai jalur antar muka untuk mikrokontroler dengan perangkat penampil formasi proses dan waktu. *Timer* digunakan untuk memberikan waktu lamanya proses pemanasan air dengan suhu yang konstan. Interupsi diperlukan sebagai sarana penyalaan dari suatu kondisi ke program yang sedang berjalan. Pemakaian LCD diinisialisasi melalui pemanggilan rutin inisialisasi LCD. Pemanggilan rutin tersebut akan memberikan data ke LCD untuk pemakaian LCD pada kedua barisnya. Untuk kode program yang digunakan dalam pemrograman mikrokontroler sebagai berikut:

```
DDRA=0b00000010; PORTA=0x00;
```

```
DDRB=0b00001111; PORTB=0x00;
```

```
DDRC=0b11111111; PORTC=0x00;
```

```
DDRD=0b11111110; PORTD=0x00;
```

```
init_LCD();
```

```
TIMSK|=(1<<OCIE1A);
```

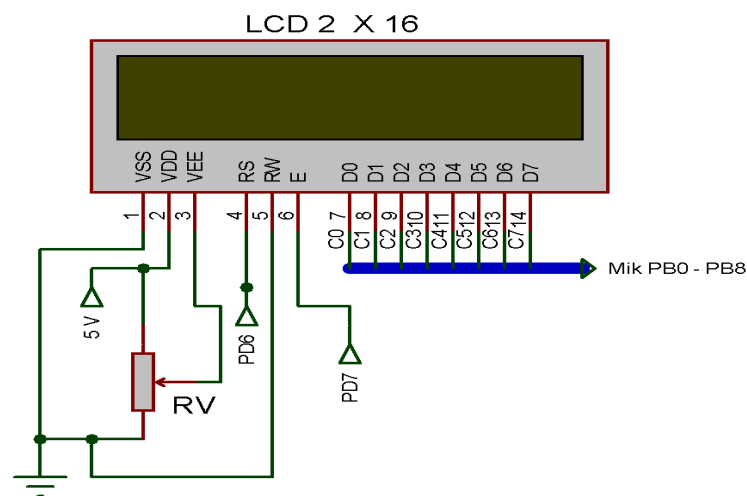
```
OCR1A=0xB71B;
```

```
TCCR1B=0;
```

Pin pada DDRA0 diberi nilai logika 0 maka *port* A0 berfungsi sebagai *inputan* dan DDRA1 diberi nilai logika 1 maka *port* A1 berfungsi sebagai *output*. Semua pin DDRB diberi nilai logika 0 sehingga semua pin pada *port* B berfungsi sebagai *input*. Semua pin DDRC diberi nilai logika 1 sehingga pin pada *port* C berfungsi sebagai *output*. DDRD0 diberi nilai logika 0 maka *port* D0 berfungsi sebagai *inputan* dan DDRD0 diberi nilai logika 1 maka *port* D0 berfungsi sebagai *output*. DDRD1 diberi nilai logika 0 maka *port* D1 berfungsi sebagai *inputan* dan DDRD1 diberi nilai logika 1 maka *port* D1 berfungsi sebagai *output*. DDRD2 diberi nilai logika 0 maka *port* D2 berfungsi sebagai *input* dan DDRD2 diberi nilai logika 1 maka *port* D2 berfungsi sebagai *output*.

1.4.9 Perancangan Rangkaian *Liquid Cristal Display (LCD)* dan Program Tampilan.

Rangkaian *Liquid Cristal Display (LCD)* merupakan sebuah rangkaian yang digunakan dalam perancangan alat *water bath*. Tampilan sistem yang bekerja pada aplikasi mikrokontroler sebagai alat penampil informasi suhu, waktu, waktu pembuangan air, dan indikator *valve* dalam perancangan ini digunakan suatu tampilan berupa LCD. Bentuk dari rangkaian LCD dapat dilihat pada Gambar 3.10. Pada rangkaian penampil *water bath* ini menggunakan komponen LCD 2 x 16, dan resistor variabel 50 K Ω .. Resistor variable pada rangkian LCD ini berfungsi sebagai pengatur kecerahan pada LCD.



Gambar 3.11. Rangkaian LCD 2 x 16

Dalam pemrograman tampilan dilakukan dengan cara memberikan data ke LCD dalam bentuk dua buah format. Format yang pertama adalah untuk mengirimkan kolom lokasi tempat untuk penulisan karakter, sedangkan untuk format yang kedua adalah untuk mengirimkan karakter yang akan dituliskan. Dalam melakukan pengiriman data karakter menggunakan Rutin `wr_inst` sedangkan untuk melakukan pengiriman data karakter menggunakan rutin `wr_data`.

Kode program yang digunakan dalam perancangan LCD pada alat adalah sebagai berikut:

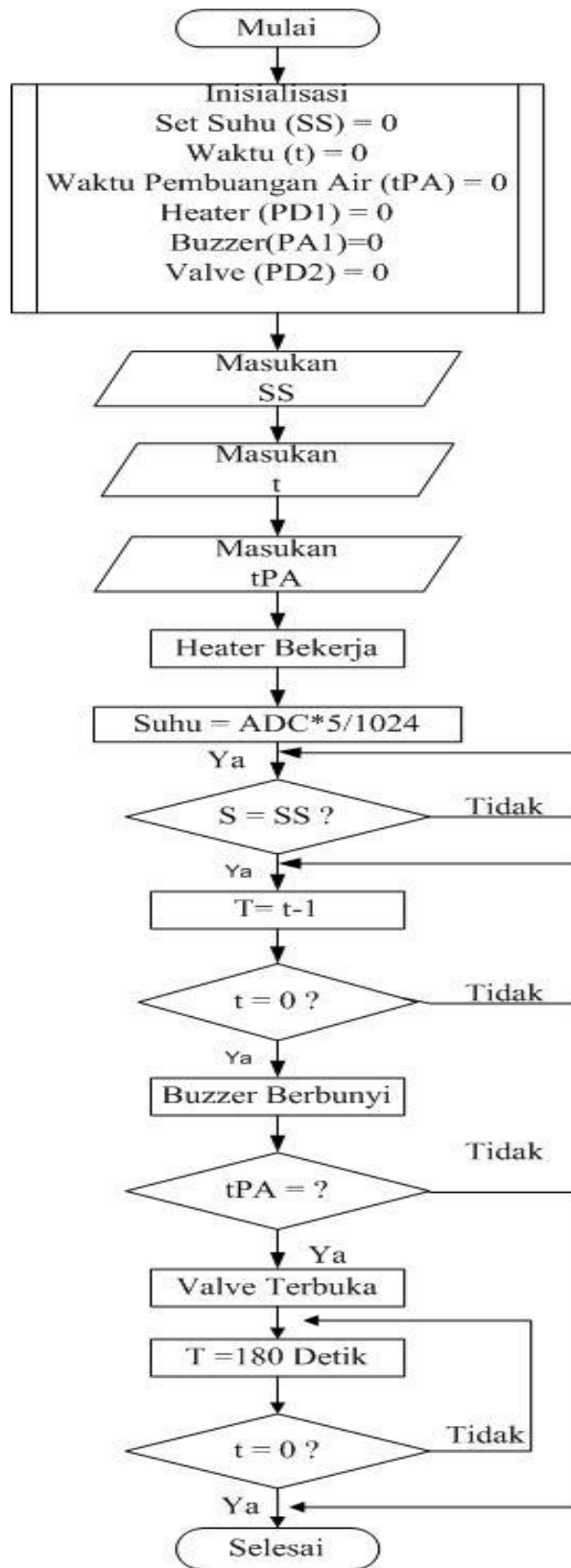
```
void LCD(unsigned char dat, char RS)
{
    unsigned char i;
    PORTC=dat;
```



```
if (RS==0) PORTD|=0b10000000;
else    PORTD|=0b11000000;
for (i=0;i<70;i++)  {asm("nop");}
PORTD&=0b00111111;
for (i=0;i<70;i++)  {asm("nop");}}
```

3.4.10 Flow Chart Perancangan Perangkat Lunak

Dalam perancangan perangkat lunak, dibuat dengan menggunakan Bahasa C yang ditulis pada editor *Advance Versatile RIS (AVR) Studio 4*. Algoritma dari perancangan perangkat lunak ditunjukkan dalam diagram alur (*flowchart*) sebagaimana yang ditunjukkan pada Gambar 3.11.



Gambar 3.12. Diagram Alir Perancangan Perangkat Lunak

Algoritma proses dijabarkan sebagai berikut:

- a. Proses inisialisasi dilakukan untuk menetapkan penggunaan fasilitas mikrokontroler I/O, Seting Suhu, Timer, Pembuangan Air, dan Interupsi.
- b. Untuk proses inputan diberi tampilan awal berupa inputan SS, t, dan tPA untuk memberikan nilai seting masing-masing berupa suhu, waktu, dan waktu pembuangan air.
- c. Nilai suhu yang terdeteksi merupakan perkalian ADC dengan konstanta 5 kemudian dibagi dengan 1024.
- d. Apakah suhu sama dengan suhu *setting*?
- e. Jika ya maka *heater* akan bekerja untuk melakukan proses pemanasan air kemudian akan melakukan perhitungan waktu.
- f. Jika waktu perhitungan sudah selesai dan batas waktu pembuangan air belum tercapai maka *buzzer* berbunyi menandakan waktu seting telah habis dan proses pemanasan sudah selesai.
- g. Jika batas waktu pembuangan air tercapai, *valve* akan terbuka dan proses pembuangan air berlangsung selama 180 detik.
- h. Jika waktu pembuangan sama dengan nol, maka proses pemanasan telah selesai.

3.6. Jenis Penelitian

Jenis penelitian yang penulis gunakan adalah jenis penelitian eksperimental, artinya meneliti, mencari, menjelaskan, dan membuat suatu instrument dimana instrument ini dapat langsung dipergunakan oleh pengguna.

3.7. Sistematika Pengukuran

Pengukuran tegangan pada beberapa titik *test point* dilakukan beberapa kali dalam percobaan. Kemudian hasil pengukuran tersebut dibandingkan dengan angka *standart* dan berapa nilai rata-rata, *standart deviasi (SD)*, ketidakpastian dan *error* dengan rumus seperti dibawah ini:

3.7.1. Rata-rata

Rata-rata dalam perkataan sehari-hari, orang sudah menafsirkan dengan rata-rata hitung. Dan arti sebenarnya adalah bilangan yang di dapat dari hasil pembagian jumlah nilai data oleh banyaknya data dalam kumpulan pengukuran tersebut.

Dinyatakan dengan rumus :

$$\boxed{= \frac{\sum x_i}{n}} \dots\dots\dots (4.1)$$

Keterangan :

- x_i : Jumlah X sebanyak i
- n : Banyak data
- \bar{x} : Rata-rata

3.7.2. Simpangan (*error*)

Merupakan selisih dari rata-rata nilai terhadap masing-masing nilai yang di ukur. Dinyatakan dengan rumus :

$$\boxed{\text{Simpangan} = x - \bar{x}} \dots\dots\dots (4.2)$$

Keterangan :

- X : Data x
- \bar{x} : Rata-rata

3.7.3. Error (%)

Merupakan nilai persen dari simpangan (*error*) terhadap nilai yang di kehendaki. Dinyatakan dengan rumus :

$$\boxed{\text{Error} (\%) = \frac{\text{Simpangan}}{\bar{x}} \times 100} \dots\dots\dots (4.3)$$

Keterangan :

- Error* : Besaran simpangan/nilai error dalam%
- X : Data x
- \bar{x} : Rata-rata