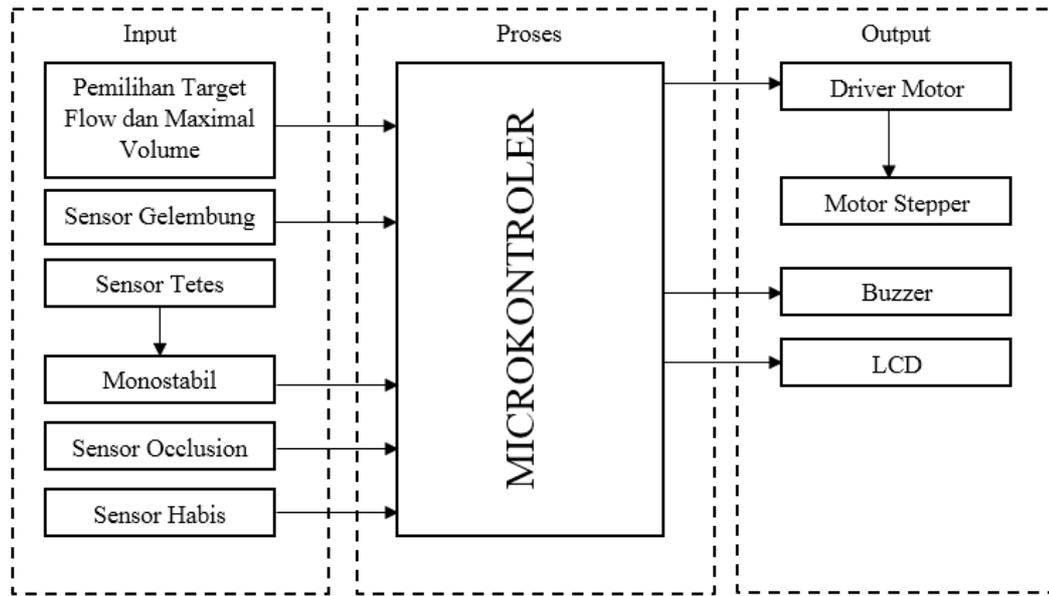


BAB III METODOLOGI PENELITIAN

3.1 Diagram Block Sistem



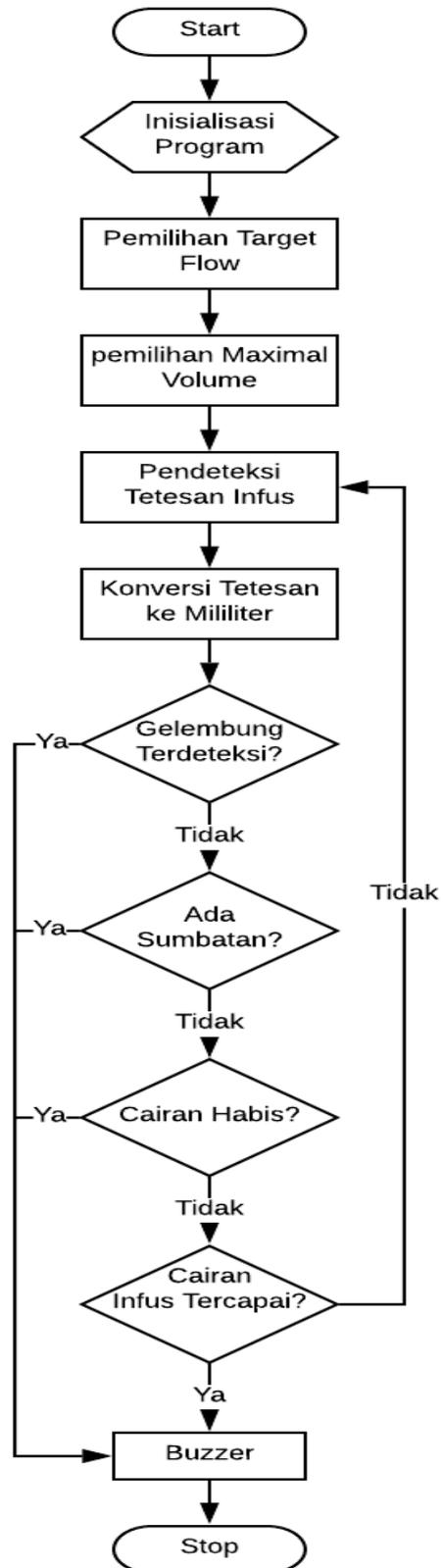
Gambar 3.1 Diagram Block Alat

Setelah alat dihidupkan maka akan terjadi proses pemilihan target aliran dan maksimal volume. Setelah melakukan pemilihan maka mikrokontroler akan memberikan sinyal logika pada rangkaian driver *motor stepper* agar memerintahkan *motor stepper* untuk bekerja. Saat *motor stepper* bekerja maka sensor tetes akan mendeteksi tetesan cairan infus. Sinyal listrik yang dihasilkan sensor tetes akan masuk pada rangkaian monostabil. Setelah masuk pada rangkaian monostabil maka sinyal listrik akan menuju rangkaian mikrokontroler, sinyal listrik akan diubah menjadi karakter berupa angka yang ditampilkan ke LCD. Ketika sensor gelembung mendeteksi adanya gelembung pada selang infus maka akan

mengirimkan sinyal listrik pada mikrokontroler sehingga mikrokontroler akan memerintahkan *motor stepper* untuk berhenti dan *buzzer* akan berbunyi. Ketika sensor *occlusion* mendeteksi adanya sumbatan pada selang infus maka akan mengirimkan sinyal listrik pada mikrokontroler sehingga mikrokontroler akan memerintahkan *motor stepper* untuk berhenti dan *buzzer* akan berbunyi. Dan Ketika sensor *Empty* mendeteksi bahwa cairan infus telah habis maka akan mengirimkan sinyal listrik pada mikrokontroler sehingga mikrokontroler akan memerintahkan *motor stepper* untuk berhenti dan *buzzer* akan berbunyi.

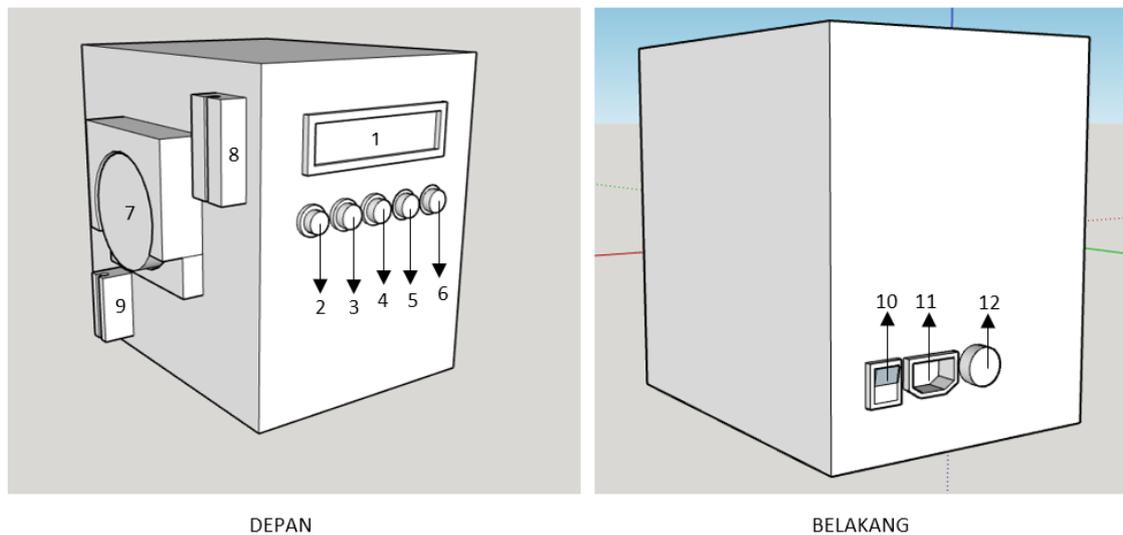
3.2 Diagram Alir Program

Pada saat alat dihidupkan maka akan terjadi inisialisasi program, setelah inisialisasi selesai maka akan terjadi proses pemilihan Kecepatan *Flow* dan melanjutkan pemilihan maksimal volume. Setelah melakukan pemilihan, pendeteksi cairan infus akan bekerja. Hasil dari pendeteksian tetesan infus selanjutnya akan dikonversikan ke dalam mililiter. Ketika terdapat gelembung pada selang infus maka *buzzer* akan berbunyi dan menghentikan alat, pada saat tidak ada gelembung maka akan melanjutkan proses. Ketika terdapat sumbatan pada selang infus maka *buzzer* akan berbunyi dan menghentikan alat, pada saat tidak ada sumbatan maka akan melanjutkan proses. Ketika cairan infus telah habis maka *buzzer* akan berbunyi dan menghentikan alat, pada saat cairan infus masih ada maka akan melanjutkan proses. Ketika cairan infus sudah tercapai maka *buzzer* akan berbunyi dan akan mematikan *motor stepper*, ketika cairan belum tercapai maka alat akan kembali pada proses pendeteksi cairan infus.



Gambar 3.2 Diagram Alir Program

3.3 Diagram Mekanis Sistem



Gambar 3.3 Diagram Mekanis Sistem

Keterangan :

1. LCD
2. Tombol *Start/Mode*
3. Tombol *Stop*
4. Tombol *Up*
5. Tombol *Down*
6. Tombol Reset
7. Motor
8. Sensor Gelembung
9. Sensor *Occlusion*
10. Saklar ON/OFF
11. Socket *Power*
12. *Fuse*

3.4 Alat dan Bahan

3.4.1 Alat

Berikut ini adalah daftar peralatan yang digunakan dalam pembuatan modul alat, yakni :

Tabel 3.1 Alat

No	Nama alat	Jumlah
1	Bor PCB	1
2	Obeng	1
3	Solder	1
4	Gergaji besi	1
5	Tang potong	1
6	Tang jepit	1
7	Atraktor	1

3.4.2 Bahan

Berikut ini adalah daftar bahan yang digunakan dalam pembuatan modul alat, yakni :

Tabel 3.2 Bahan

No	Nama Bahan	Jumlah
1	IC ATmega328	1
2	Kristal 16 Mhz	1
3	Kapasitor 22 pf	2
4	Kapasitor 10 nf	3
5	Saklar <i>Push button</i>	4
6	LED	2
7	Pin baris	Secukupnya
8	LCD 16x2	1

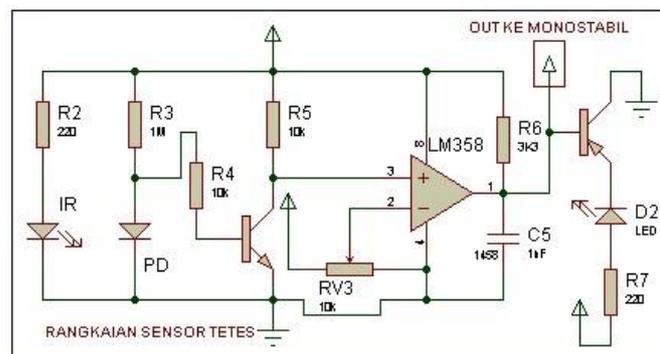
9	<i>Motor stepper</i> Nema 17hs401	1
10	Driver motor stepper A4988	1
11	Loadcell	1
12	Transistor 2N3904	2
13	Transistor 2N3906	2
14	LM 358	3
15	NE 555	1
16	Infrared	2
17	<i>Photodiode</i>	2
18	HX711	1

3.5 Proses Pembuatan Perangkat Keras

3.5.1 Rangkaian Counter Tetes

Spesifikasi rangkaian *counter* tetes yakni sebagai berikut :

1. Tegangan kerja rangkaian 5 Volt DC.
2. Menggunakan transistor NPN 2N3904
3. Menggunakan transistor PNP 2N3906
4. Menggunakan IC LM358
5. Menggunakan sensor *photodiode*
6. Menggunakan kapasitor MKM 10 nf



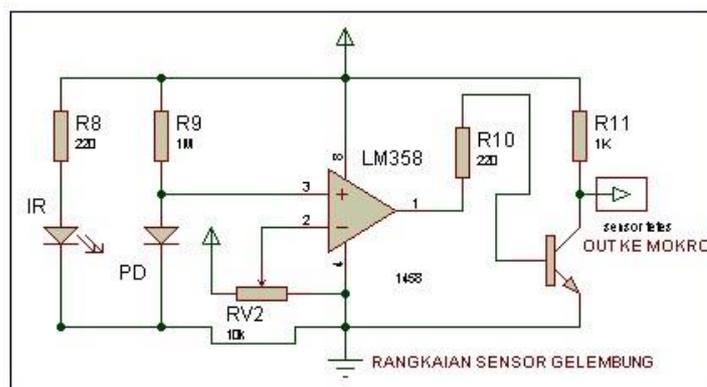
Gambar 3. 4 Rangkaian Sensor Tetes

Infra red diatas sebagai pemancar sinar yang nantinya sinar tersebut akan diterima oleh sensor *photodiode*, ketika komparator pada pin 3 mendapat tegangan lebih kecil dari tegangan pada pin 2 maka keluaran dari komparator pada pin 1 akan menjadi *low*. Keluaran dari komparator diberikan rangkaian *pull up* agar keluaran memiliki polatitas positif selama tidak ada *trigger* dari *input* pada pin 3. Pada keluaran komparator juga diberikan kapasitor yang berfungsi untuk menyaring *noise* sehingga keluaran menjadi lebih baik.

3.5.2 Rangkaian Sensor Gelembung

Spesifikasi rangkaian sensor gelembung antara lain, yakni :

1. Menggunakan IC LM358
2. Menggunakan sensor *photodiode*
3. Menggunakan transistor NPN 2N3904



Gambar 3. 5 Rangkaian Sensor Gelembung

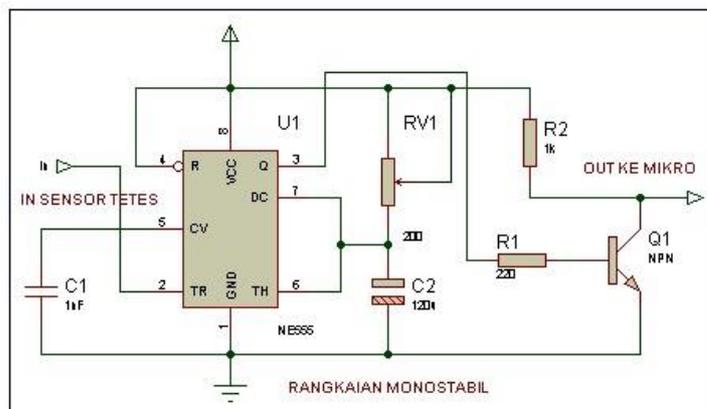
Infra red berfungsi sebagai pemancar sinar yang akan diterima oleh sensor *photodiode*, ketika sensor *photodiode* mendapat perbedaan pancaran sinar maka sensor akan memberikan *trigger* kepada IC

komparator, pada saat keluaran dari komparator mengeluarkan sinyal *high* maka transistor pada kaki *colector* akan mengeluarkan sinyal *low*. Sinyal *low* tersebut akan masuk ke dalam mikrokontroler pada PIN 6

3.5.3 Rangkaian Monostabil

Spesifikasi rangkaian monostabil yakni sebagai berikut :

1. Tegangan kerja rangkaian 5 Volt DC.
2. Menggunakan IC NE555.
3. Menggunakan kapasitor elco 100 uf
4. Menggunakan kapasitor MKM 10 nf
5. Menggunakan transistor 2N3904



Gambar 3. 6 Rangkaian Monostabil

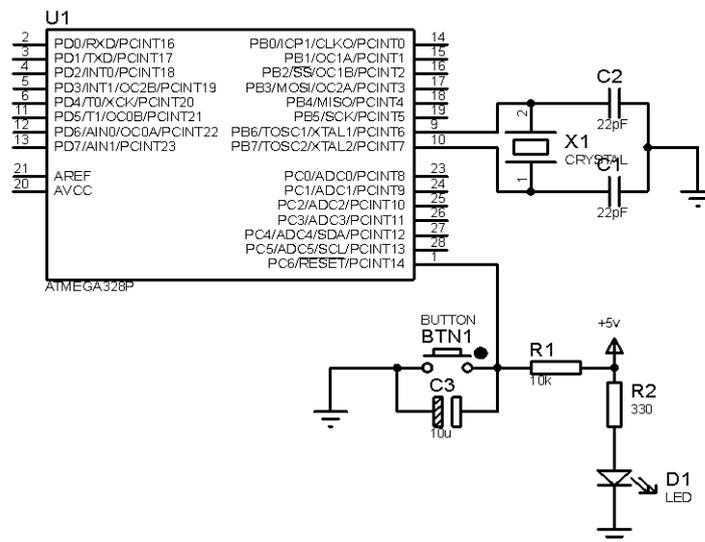
Ketika *trigger* yang dihasilkan sensor *counter* masuk ke pin 2 maka keluaran IC pada pin 3 akan mengeluarkan sinyal *high*, keluaran sinyal tersebut akan diubah polaritasnya didalam transistor, sehingga menjadi sinyal *low*. Lamanya waktu sinyal *high* yang keluar dari monostabil ditentukan oleh besarnya nilai RV1 dan C2, semakin besar RV1 dan C2

maka akan semakin lama sinyal *high* yang keluar. Sinyal yang keluaran dari kaki *colectro* pada transistor akan masuk ke dalam IC mikrokontroler pada kaki PIN 5

3.5.4 Rangkaian Minimum Sistem Arduino

Spesifikasi rangkaian mikrokontroler yakni :

1. Menggunakan IC mikrokontroler ATmega 328p.
2. Menggunakan kristal 16 Mhz.
3. Mengnakan 2 buah kapasitor 22pF.



Gambar 3. 7 Rangkaian Sistem Minimum Arduino

Mikrokontroler ATmega 328p adalah otak dari rangkaian sistem minimum ini. Dimana didalamnya sudah dilengkapi CPU, ROM, RAM, I/O, clock dan peralatan internal lainnya yang sudah saling terhubung dan terorganisasi dengan baik. Pada ATmega 328p memiliki 28 kaki, dan memiliki 3 PORT. Untuk membuat

sebuah minimum sistem Arduino dibutuhkan juga kristal atau yang sering disebut osilator (pembangkit frekuensi).

3.6 Pembuatan Program

Pada alat ini menggunakan bahasa pemrograman Arduino dengan menggunakan ATmega328P sebagai minimum sistem, Program ini digunakan untuk mengendalikan semua sistem yang ada di alat.

3.6.1 Listing program pembacaan tetesan

```
void tetesan()
{
  habis();
  TCCR1B=0x06;
  tetes=TCNT1;
  lcd.setCursor(1,0);
  lcd.print("-Cairan Keluar-");
  lcd.setCursor(0,1);
  lcd.print("Volume :");
  tetes=tetes/19.3;
  lcd.print(tetes);
  lcd.print("ml");
}
```

Gambar 3. 8 Listing program pembacaan sensor tetesan

Pada program void tetesan() terdiri dari program untuk memanggil listing program sensor *empty*. Pada “TCCR1B=0x06” merupakan program untuk mengaktifkan mode pembacaan tetesan pada alat. Pada “tetes=TCNT1;” merupakan program untuk menguah variable dari data yang tersimpan di dalam TCNT1 menjadi variable “tetes”. Lalu dibawahnya ada listing program untuk

menampilkan dan menempatkan tulisan “cairan keluar” dan “volume :” pada lcd. Pada “tetes=tetes/19.3;” merupakan program untuk konversi dari tetesan menjadi milliliter. Lalu pada program dibawahnya ada program untuk menampilkan hasil konversi dan satuan yang digunakan pada layar LCD.

3.6.2 Program sensor *occlusion*

```
void occl()
{
  scale.set_scale(calibration_factor);
  float occ = scale.get_units();
  if (occ>2.5)
  {
    mod=3;error=3;delay(100);lcd.clear();
  }
}
```

Gambar 3. 9 Listing program *Occlusion*

Litsing program diatas merupakan program untuk sensor *occlusion*.

Pada “scale.set_scale(calibration_factor);” merupakan program untuk memasukan factor kalibrasi yang sudah ditentukan. Pada “float occ = scale.get_units();” merupakan program untuk membaca data dari load cell lalu disimpan pada variable ber tipe float dengan nama “occ”. Perintah if dibawahnya digunakan untuk membatasi jika data yang dibaca oleh sensor melebihi data yang ditentukan maka program akan masuk ke mode = 3 dan error = 3 yang merupakan program untuk menampilkan telah terjadi error pada sensor oklusi.

3.6.3 Litsing Program *motor stepper*

Litsing program dibawah merupakan program untuk mengontrol *motor stepper*. Pada “digitalWrite(stepPin,HIGH);” merupakan perintah pada pin untuk menghasilkan sinyal HIGH atau tegangan 5V. Pada “digitalWrite(stepPin,LOW);” merupakan perintah pada pin untuk menghasilkan sinyal LOW atau tegangan 0V. Pada “delay(kecmoto);” merupakan pengaturan *delay* untuk setingan kecepatan motor, kecepatan motor diseting dalam litsing program pemilihan *flow*. Pada 3 program if dibawahnya merupakan program untuk memanggil litsing program error.

```
void moto()
{
  digitalWrite(stepPin,HIGH);
  delay(kecmoto);
  digitalWrite(stepPin,LOW);
  delay(kecmoto);
  if(buttonstop == LOW)
  {mod=3;error=1;delay(100);lcd.clear();}
  if(bubble == LOW)
  {mod=3;error=2;delay(100);lcd.clear();}
  if(occ == LOW)
  {mod=3;error=3;delay(100);lcd.clear();}
}
```

Gambar 3. 10 Litsing program pengendalian motor

3.6.4 Listing Program mode

```

if(buttonstart == LOW)
{
    mod=mod+1;
    delay(200);
    data=0;
    lcd.clear();
    if (mod>2){mod=2;}
}

```

Gambar 3. 11 Listing Program Mode

Listing program diatas adalah program yang akan di kerjakan ketika tombol start ditekan. Pada “mod=mod+1;” meruakan perintah ketika tombol start ditekan maka mode akan bertambah 1, mulai dari mode 0 yang merupakan listing program pemilihan kecepatan *flow*, mode 1 merupakan perintah pemilihan volume target, mode 2 merupakan perintah start, dan mode 3 merupakan listing program untuk perintah berhenti. Pada “delay(200);” merupakan perintah *delay* agar ketika tombol start ditekan maka perubahan pada mode tidak terlalu cepat. Pada “data=0;” merupakan perintah agar variable data bernilai 0. Pada “lcd.clear();” merupakan perintah untuk LCD agar menghilangkan tulisan pada setiap perpindahan mode. Pada penggunaan program if dibawahnya digunakan untuk membatasi mode agar tidak melebihi mode yang sudah ditentukan.

3.6.5 Listing program pemilihan *flow*

Listing program dibawah ini merupakan program yang berjalan ketika mode=0. Pada mode=0 akan dilakukan perintah pemilihan *flow* speed. Pada perintah “if(buttonup == LOW)” ketika tombol pemilihan naik ditekan maka

smode *flow* akan bertambah. Pada perintah “`if(buttondown == LOW)`” ketika tombol pemilihan turun ditekan maka mode *flow* akan berkurang. Program LCD dibawahnya merupakan program untuk menampilkan pemilihan pada LCD. Pada perintah “`if (flow<=25)`” merupakan perintah ketika pemilihan kecepatan *flow* dibawah 25 maka kecepatan motor diatur dengan perhitungan dibawahnya. Pada perintah “`if (flow>25)`” merupakan perintah ketika pemilihan kecepatan *flow* diatas 25 maka kecepatan motor diatur dengan perhitungan di bawahnya.

```

if (mod == 0)
{
  if(buttonup == LOW)
  {lcd.clear(); flow=flow+1; delay(200);
   if (flow>60){flow=20;}}
  if(buttondown == LOW)
  {lcd.clear(); flow=flow-1; delay(200);
   if (flow<20){flow=60;}}
  lcd.setCursor(0,0);
  lcd.print("Kecepatan Flow");
  lcd.setCursor(0,1);
  lcd.print(flow);
  lcd.print("ml/jam");
  if (flow>25)
  {kecmoto=(-1.0667*flow)+77;}
  if (flow<=25)
  {kecmoto=(-1.0667*flow)+105; }
}

```

Gambar 3. 12 Listing Program pemilihan *flow*

3.6.6 Listing program pemilihan volume

```
if (mod == 1)
{
  if(buttonup == LOW)
  {
    lcd.clear();
    volume=volume+1;
    delay(200);
    if (volume>500){volume=1;}
  }
  if(buttondown == LOW)
  {
    lcd.clear();
    volume=volume-1;
    delay(200);
    if (volume<1){volume=500;}
  }
  lcd.setCursor(0,0);
  lcd.print("Set Volume");
  lcd.setCursor(0,1);
  lcd.print(volume);
  lcd.print("ml");
}
```

Gambar 3. 13 Listing program pemilihan volume

Listing program diatas merupakan program yang berjalan ketika mode=1. Pada mode=1 akan dilakukan perintah pemilihan volume target. Pada perintah “if(buttonup == LOW)” ketika tombol pemilihan naik ditekan maka volume akan bertambah. Pada perintah “if(buttondown == LOW)” ketika tombol pemilihan turun ditekan maka volume akan berkurang. Pada perintah “if

`(volume>500){volume=1;}`” dan `“if (volume<1){volume=500;}`” merupakan perintah untuk membatasi jumlah volume, ketika volume lebih dari 500 maka volume akan kembali lagi ke 1, dan jika volume kurang dari 1 maka volume akan menjadi 500. Program LCD digunakan untuk menampilkan settingan yang akan di pilih di LCD.

3.6.7 Listing program sensor *empty*

Pada listing program `“void habis()”` merupakan program untuk mendeteksi ketika cairan infus sudah habis. Pada `“int a=digitalRead(PIN);”` untuk mejadikan variable `“a”` sebagai pembacaan dari `“PIN”`. Pada perintah `“if (a==HIGH)”` ketika `“a”` sedang dalam keadaan `“HIGH”` atau dalam keadaan 5V maka data akan terus bertambah, dan pada perintah `“if (data==12)”` digunakan untuk membatasi data sampai pada angka 12, jika data sudah sampai angka 12 maka mode akan berubah menjadi mode 3, dan menmpilkan error 5. Jika `“a”` tidak dalam keadaan `“HIGH”` maka data akan menjadi 0.

```
void habis()
{
  int a=digitalRead(PIN);
  if (a==HIGH)
  {
    data++;
    if (data==150){mod=3;error=5;}
  }else{data=0;}
}
```

Gambar 3. 14 Listing program sensor *empty*

3.6.8 Listing program perintah mulai

```

if (mod==2)
{
    error=0;
    tetesan();
    ml=TCNT1/19.3;
    if (ml>=volume) {mod=3;error=4;lcd.clear();}
    moto();
    noTone(buzzer);
}

```

Gambar 3. 15 Listing program perintah mulai

Listing program diatas merupakan program yang berjalan ketika mode=2. Pada mode=2 akan dilakukan perintah untuk memulai menjalankan *motor stepper* dan menghitung tetesan. Pada “error=0;” maka isi dari variable error adalah 0. Pada baris bertuliskan “tetesan();” digunakan untuk memanggil “Void tetesan()” agar dapat menghitung nilai dari tetesan. Pada “ml=TCNT1/19.3;” merupakan konversi data yang tetesan pada TCNT1 agar dapat menjadi milliliter dan hasil dari konversi disimpan pada variable “ml”. Pada “if (ml>=volume)” merupakan program ketika jumlah yang berada di variable “ml” lebih tinggi dari volume maka mode akan menjadi 3 dan akan menampilkan error 4. Pada “moto();” merupakan perintah untuk memanggil “void moto()” agar dapat menjalankan *motor stepper*. Pada “noTone(buzzer);” merupakan program agar *buzzer* tidak berbunyi.

3.6.9 Program Berhenti

```

if (mod==3)
{
  TCCR1B=0x00;
  tetes=TCNT1;

  if (error==1){lcd.setCursor(0,0); lcd.print("    -STOP-");
tone(buzzer,1000);}

  if (error==2){lcd.setCursor(0,0); lcd.print("    ERROR");
lcd.setCursor(0,1); lcd.print("    BUBBLE"); tone(buzzer,1000);}

  if (error==3){lcd.setCursor(0,0); lcd.print("    ERROR");
lcd.setCursor(0,1); lcd.print("    OCCLUSION");
tone(buzzer,1000);}

  if (error==4){lcd.setCursor(0,0); lcd.print(" PROSES
SELESAI");tone(buzzer,1000);}

  if (error==5){lcd.print("
"); lcd.setCursor(0,0); lcd.print("    ERROR");
lcd.setCursor(0,1); lcd.print("    CAIRAN HABIS");
tone(buzzer,1000);}
}

```

Gambar 3. 16 Listing Program Berhenti

Listing program diatas merupakan program yang akan dilakukan pada saat mode=3. Pada saat mode=3 maka akan ada perintah untuk menghentikan semua proses baik itu karena cairan sudah tercapai atau terjadi error. Pada “TCCR1B=0x00;” merupakan perintah untuk menghentikan perhitungan pada pembacaan sensor tetes. Pada “tetes=TCNT1;” merupakan perintah agar data dalam variable “tetes” sama dengan data pembacaan sensor tetes pada TCNT1. Pada perintah “if(error==1)” merupakan perintah berhenti ketika tombol stop ditekan, dilayar LCD akan ditampilkan tulisan “STOP” dan *buzzer* akan berbunyi. Pada perintah “if(error==2)” merupakan perintah berhenti ketika sensor gelembung mendeteksi adanya gelembung udara didalam selang, dilayar LCD akan ditampilkan tulisan “ERROR BUBBLE” dan *buzzer* akan berbunyi. Pada perintah

“`if(error==3)`” merupakan perintah berhenti ketika sensor oklusi mendeteksi adanya sumbatan pada selang, dilayar LCD akan ditampilkan tulisan “*ERROR OCCLUSION*” dan *buzzer* akan berbunyi. Pada perintah “`if(error==4)`” merupakan perintah berhenti ketika cairan yang dideteksi oleh sensor tetes sama dengan atau lebih dari setingan volume yang ditentukan, dilayar LCD akan ditampilkan tulisan “*PROSES SELESAI*” dan *buzzer* akan berbunyi. Pada perintah “`if(error==5)`” merupakan perintah berhenti ketika sensor *empty* mendeteksi bahwa cairan infus sudah habis, dilayar LCD akan ditampilkan tulisan “*CAIRAN HABIS*” dan *buzzer* akan berbunyi.

3.7 Teknik Analisis Data

3.7.1 Rata-rata

Rata-rata merupakan nilai atau hasil pembagian dari jumlah data yang diambil atau diukur dengan banyaknya pengambilan data atau banyaknya pengukuran. Dirumuskan sebagai berikut:

$$\bar{x} = \frac{\sum x_n}{n} \quad (3-1)$$

Keterangan:

\bar{x} = Rata-rata

x_n = Jumlah nilai data

n = banyaknya data (1,2,3,4,...n)

3.7.2 Simpangan

Simpangan (*deviation*) adalah perbedaan antara nilai yang sebenarnya (x_s) dengan nilai rata-rata (\bar{x}).

$$D = x_s - \bar{x} \quad (3-2)$$

Keterangan:

D = deviasi (+ atau -)

x_s = nilai sebenarnya

\bar{x} = Rata-rata

3.7.3 Presentase *error*

Presentase simpangan adalah nilai persen dari simpangan (*error*) terhadap nilai yang dikehendaki. Dirumuskan sebagai berikut:

$$\% \text{ simpangan} = \frac{x_s - \bar{x}}{x_s} \times 100\% \quad (3-3)$$

Keterangan:

x_s = nilai sebenarnya

\bar{x} = Rata-rata

3.8 Teknik Pengujian

3.8.1 Pengujian Kecepatan *Flow*

Kecepatan *Flow* diuji dengan menggunakan gelas ukur dan stopwatch. Gelas ukur digunakan untuk mengukur cairan yang keluar dalam 1 menit. Pengujian ini berfungsi untuk mengetahui apakah Kecepatan *Flow* sudah sesuai pada

pengaturan atau belum. Sebelumnya dilakukan konversi satuan dari ml/jam menjadi ml/menit. Pengujian ini dilakukan sebanyak 10 kali.

3.8.2 Pengujian Volume Maximal

Pengujian ini dilakukan dengan cara mengukur *volume* cairan yang keluar dari alat dengan menggunakan gelas ukur. Pengujian ini berfungsi untuk mengetahui apakah *volume* yang keluar dari alat sudah terpenuhi. Pengujian ini dilakukan sebanyak 10 kali.

3.8.3 Sensor Gelembung

Pengujian ini dilakukan dengan cara membuat gelembung pada selang infus yang ukurannya ditentukan dan diukur dengan jangka sorong lalu melewatkannya pada sensor gelembung. Pengujian ini dilakukan sebanyak 10 kali.

3.8.4 Sensor Occlusion

Pengujian ini dilakukan dengan cara mengukur tekanan yang ditimbulkan dari alat dengan menggunakan alat *pressure gauge*. Pengujian ini berfungsi untuk mengetahui apakah sensor *occlusion* sudah sesuai pengaturan atau belum. Pengujian ini dilakukan sebanyak 10 kali.

3.8.5 Sensor Empty

Pengujian ini dilakukan untuk mengetahui waktu yang diperlukan antara infus yang sudah habis dengan alarm yang berbunyi pada alat. Pengujian ini dilakukan dengan menggunakan stopwatch. Pengujian ini dilakukan sebanyak 10 kali.