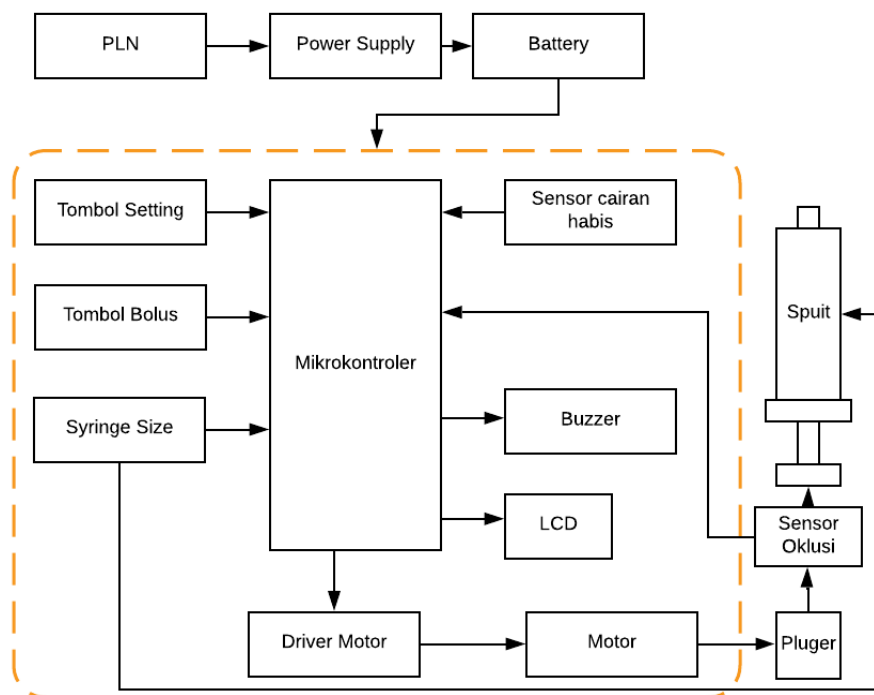


BAB III METODE PENELITIAN

3.1 Diagram Blok Sistem

Syringe pump tersusun dari beberapa rangkaian elektronika yang saling terintegrasi sehingga menciptakan suatu alat yang kompleks dan dapat berfungsi sebagaimana mestinya. Adapun blok diagram yang digunakan untuk menciptakan alat *syringe pump* ditunjukkan pada Gambar 3.2 dibawah ini.



Gambar 3.1 Blok Diagram Alat

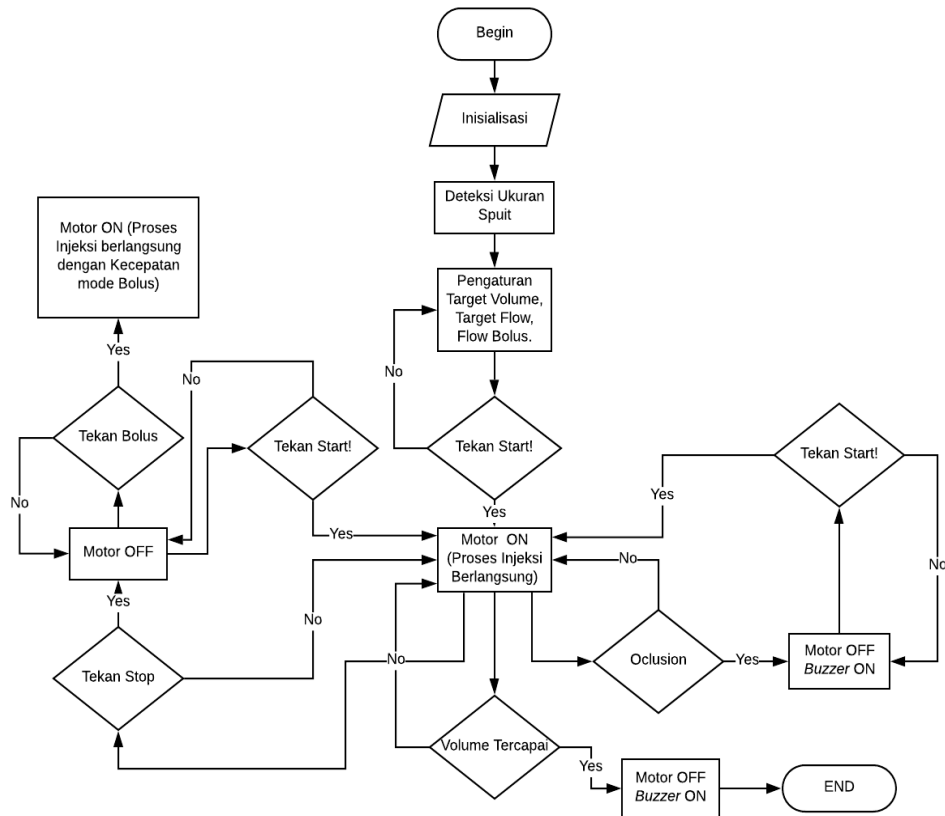
Penjelasan Blok Diagram:

Pertama dari tegangan PLN 220V akan diturunkan dan disearahkan menjadi tegangan DC untuk mensupply semua rangkaian. Tombol *setting*

digunakan untuk mengatur kecepatan aliran dan volume yang diinstruksikan kepada mikrokontroler, dari *setting* kecepatan aliran akan menentukan kecepatan motor *stepper*. sensor *syringe size* juga digunakan untuk menentukan seberapa cepat putaran motor, sebagai contoh *sprit* ukuran 50ml dan 30ml akan memiliki kecepatan motor berbeda meskipun ditentukan kecepatan alirannya sama, hal ini dipengaruhi oleh luas penampang *sprit* yang berbeda. Sensor oklusi sebagai sistem *safety* menggunakan sensor FSR402, apabila terjadi sumbatan maka *sprit* akan semakin berat untuk didorong, dari sinilah diambil nilai tekanannya, sehingga ketika sensor mendeteksi adanya tekanan berlebih maka akan mengintruksikan kepada mikrokontroler untuk menghentikan proses. Setting kecepatan aliran dan volume serta tampilan volume yang telah masuk pada pasien ditampilkan secara terus-menerus selama proses masih berjalan melalui LCD 2 x 16 yang mendapatkan instruksi dari mikrokontroler. Saat tombol bolus ditekan maka motor akan bekerja lebih cepat sesuai dengan *setting* mode *bolus*, mode *bolus* dilakukan dengan cara menekan terus-menerus, ketika tombol tidak ditekan maka proses injeksi akan berhenti, pada mode bolus ini jumlah cairan yang masuk akan tetap terpantau dari LCD. Saat cairan hampir habis maka sensor akan memberikan instruksi kepada mikrokontroler sehingga *buzzer* akan berbunyi beberapa saat sebagai penanda cairan hampir habis. Setelah proses selesai maka *buzzer* akan berbunyi kembali dan LCD 2 x 16 akan menampilkan jumlah akhir volume cairan yang telah masuk pada cairan.

3.2 Diagram Alir proses/program

Sistem yang dirancang oleh penulis menghasilkan sebuah diagram alir proses alat bekerja, yang ditunjukkan pada Gambar 3.2 *Flowchart Program*.



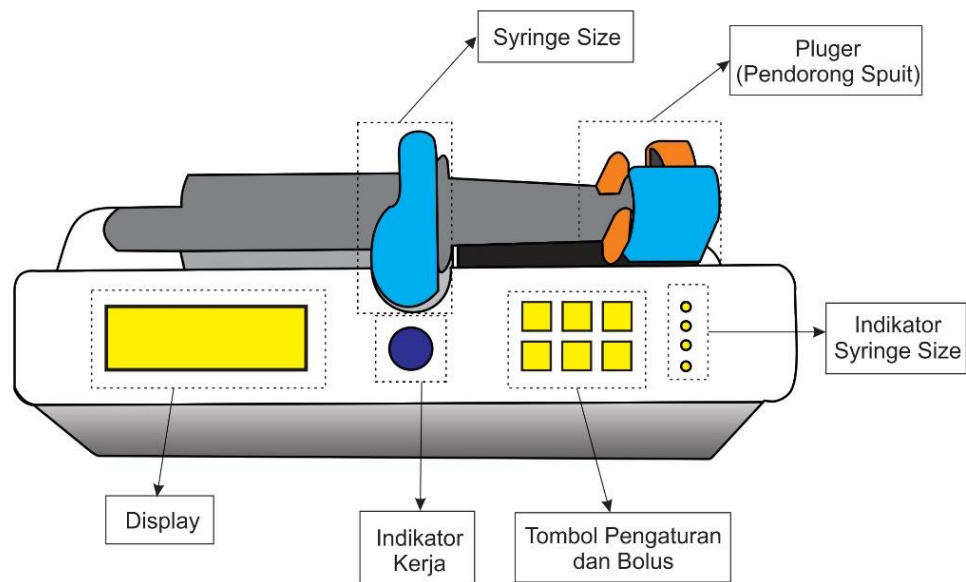
Gambar 3.2 *Flowchart Program*

Saat alat dihidupkan maka alat akan melakukan inisialisasi, setelah proses inisialisasi selesai maka alat akan menampilkan pilihan pengaturan, alat akan mendeteksi keberadaan *s spuit* dan ukuran *s spuit*. setelah dilakukan pengaturan dan ditekan START maka alat akan bekerja motor ON, jika tidak terdeteksi keberadaan *s spuit* atau *syringe size* belum tertutup maka alat tidak akan bisa dijalankan. Program *counter* akan terus mencacah selama proses injeksi berlangsung, nilai dari hasil pencacahan per satu detik akan dikonversi kedalam bentuk nilai jumlah cairan keluar per detik. Ketika terjadi oklusi

maka *buzzer* ON dan motor OFF, jika tidak terjadi sumbatan maka akan melanjutkan proses. Jika alat ditekan tombol *stop* maka motor OFF, setelah motor OFF jika ditekan tombol bolus maka motor akan berputar sesuai kecepatan mode bolus. Setelah volume tercapai maka *buzzer* ON dan motor OFF menandakan proses selesai.

3.3 Diagram mekanik Sistem

Desain bentuk alat yang dirancang oleh penulis ditunjukkan pada Gambar 3.3 Diagram Mekanik dibawah ini.



Gambar 3.3 Diagram Mekanik

3.4 Alat dan Bahan

3.4.1 Alat

1. Multimeter
2. Solder
3. *Project Board*
4. Bor
5. Toolset

6. Sduit
7. Gelas ukur
8. IDA *Analyzer*

3.4.2 Bahan

Bahan-bahan yang dibutuhkan pada penelitian ini ditunjukkan pada Tabel 3.1 dibawah ini.

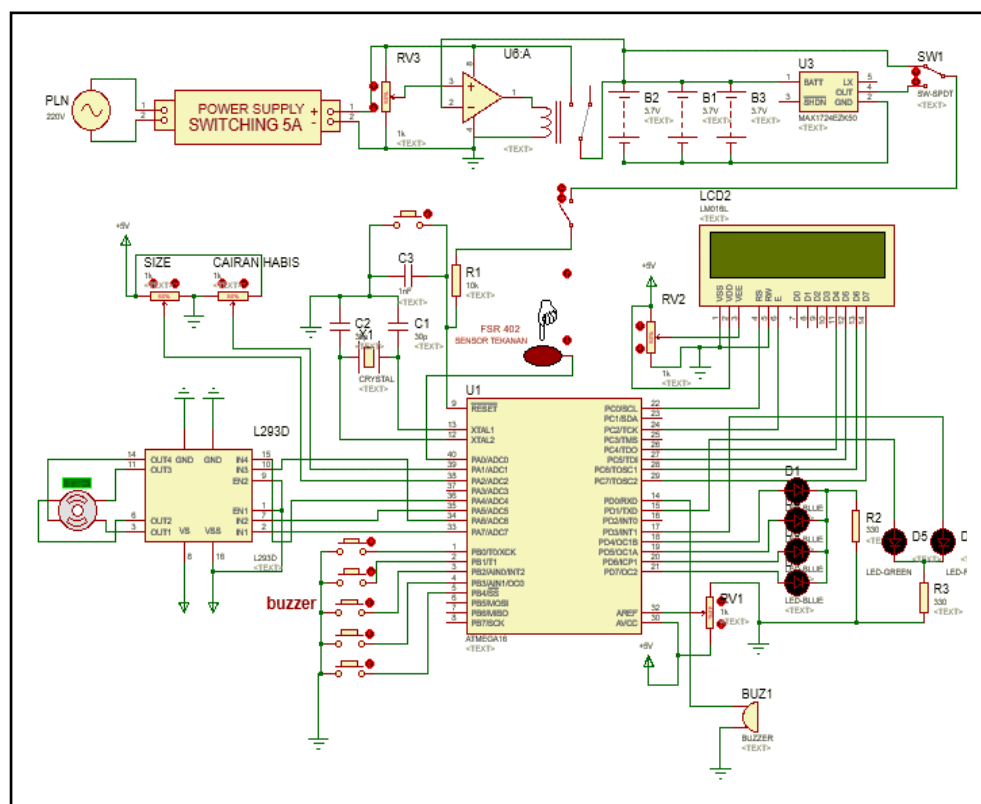
Tabel 3.1 Bahan

No	Nama Komponen	Jumlah
1	IC ATmega 16	1
2	Kristal 12MHz	1
3	Push Button	6
4	Motor Stepper	1
5	LED	12
6	Power ON/OFF	1
7	Fuse	1
8	IC L298N	1
9	Trimpot	2
10	Resistor	Secukupnya
11	Trafo 5A	1
12	Kabel Jumper	Secukupnya
13	Sensor Potensio Geser	2
14	FSR 402	1
15	Lem, isolasi bakar	Secukupnya
16	Modul Charging + Step Up	1

17	PCB	Secukupnya
18	LCD 2x16	1

3.5 Implementasi Perangkat Keras

Pada perancangan alat, dibuatlah sebuah skematik rangkaian elektronika secara keseluruhan yang ditunjukkan pada Gambar 3.4 dibawah ini.



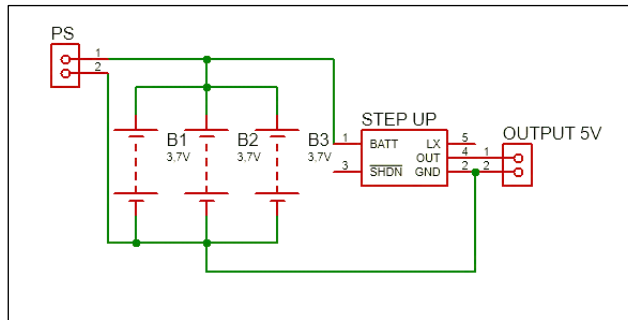
Gambar 3.4 Rangkaian Keseluruhan

Rangkaian keseluruhan diatas memiliki prinsip kerja yaitu setelah alat ON alat akan menjalankan inialisasi program dan memerintahkan untuk melakukan *setting* pada alat. Setelah alat telah melakukan inialisasi dan *setting* maka mikrokontroler akan memberikan sinyal PWM terhadap *driver* motor dan sistem nilai *counter* akan mulai berjalan menunjukkan jumlah

cairan keluar. Kecepatan motor pada masing-masing jenis spuit berbeda meskipun dengan pengaturan kecepatan yang sama, hal ini diatur oleh *syringe size* yang berfungsi mengidentifikasi ukuran spuit dan akan memberikan konstanta yang berbeda setiap ukurannya. Konstanta inilah yang menyebabkan kecepatan motor berbeda. Ketika cairan terdeteksi hampir habis maka mikrokontroler akan memberikan alarm peringatan. Begitupun juga dengan saat kondisi terjadi sumbatan pada saluran injeksi maka alat juga akan memberikan alarm namun pada kondisi ini alat akan otomatis berhenti bekerja.

1. Pembahasan Rangkaian Catu Daya

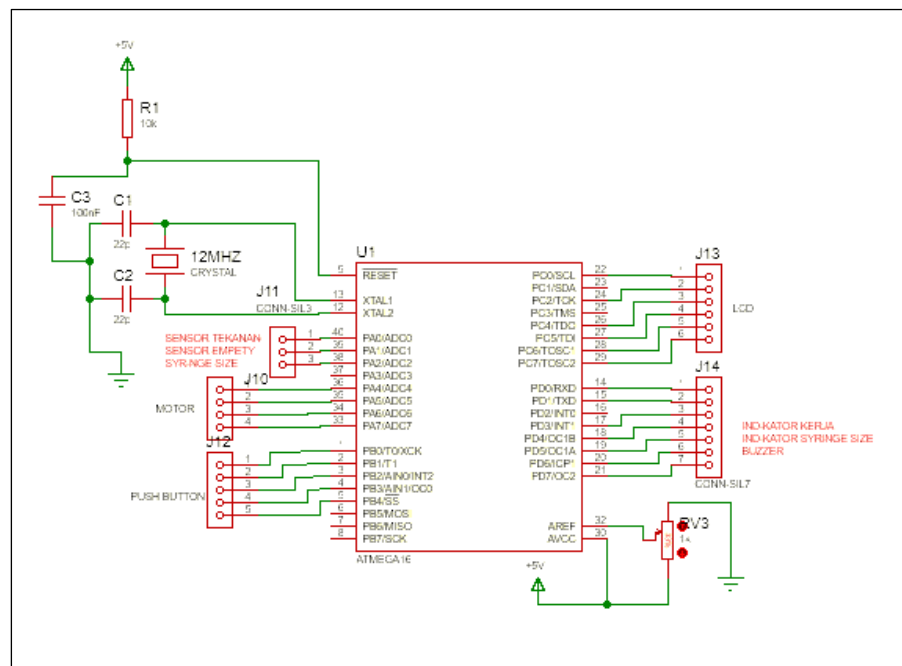
Catu daya merupakan sebuah *supply* listrik yang digunakan untuk menyalakan sebuah peralatan elektronik. Tanpa catu daya peralatan elektronik tidak akan bisa digunakan. Pada perancangan alat ini digunakan 2 mode catu daya yaitu mode PLN dan baterai. Mode PLN merupakan mode dimana alat akan langsung mendapatkan tegangan PLN 220V AC yang nantinya akan diturunkan dan disearahkan pada rangkaian *power supply switching* sehingga menghasilkan keluaran tegangan 5V DC 5A. Mode baterai merupakan mode dimana alat akan mendapatkan *supply* tegangan dari baterai *rechargeable* 3,7V. Baterai 3,7V akan dinaikkan tegangannya hingga keluarannya sebesar 5V. Skematik baterai yang digunakan pada alat ini tercantum pada Gambar 3.5 dibawah ini.



Gambar 3.5 Rangkain Skematik *Battery*

2. Penjelasan Minimum Sistem ATMEGA 16

Pada perancangan alat ini menggunakan basis mikrokontroler Atmega 16, adapun rangkaian skematik minimum sistem Atmega 16 ditunjukkan pada Gambar 3.6 dibawah ini.



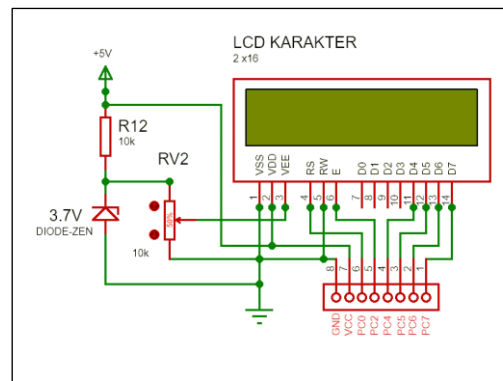
Gambar 3.6 Skematik Minimum Sistem ATMEGA 16

Minimum sistem yang dibuat menggunakan IC mikrokontroler Atmega 16 dengan *Cristal* 12MHz. IC ini dipilih karena memiliki 4 PORT. Minimum sistem merupakan suatu pusat pengendali yang akan mengontrol kerja setiap blok *hardware* yang terintegrasi pada minimum

sistem tersebut. Minimum sistem bisa dikatakan sebuah otak dalam suatu alat, dimana tanpa otak alat tidak akan dapat bekerja. Dengan adanya minimum sistem ini suatu sistem kendali dapat tercipta dengan simple dan efesian karena sistem kendali dirancang dengan bahasa pemrograman yang diinject kedalam IC.

3. Penjelasan Rangkaian LCD

Pada perancangan alat ini menggunakan display LCD karakter 2 x 16, adapun rangkaian skematik LCD *display* ditunjukkan pada Gambar 3.7 dibawah ini.

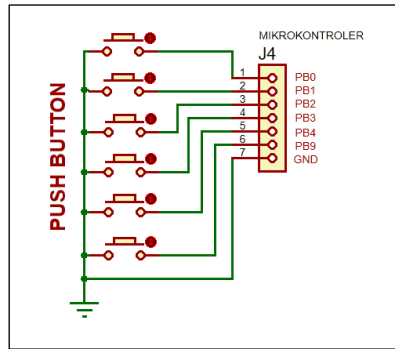


Gambar 3.7 Rangkain Skematik LCD Karakter 2x16

LCD pada pembuatan alat ini digunakan untuk menampilkan beberapa mode setting yang harus diatur sebelum melakukan proses injeksi obat dan digunakan untuk menampilkan cairan yang masuk pada pasien secara *realtime*. LCD nantinya akan dapat menampilkan *display* karena mendapatkan perintah dari mikrokontroler. Dengan memandang kebutuhan yang ada pada penelitian, maka penulis menggunakan LCD karakter yang berukuran 2 x 16. LCD karakter 2x16 terhubung ke seluruh PORTC kecuali PORTC.1 dan PORTC.3.

4. Penjelasan Rangkaian *Pushbutton*

Pada perancangan alat ini menggunakan button pengontrol alat, adapun rangkaian skematik *Pushbutton* ditunjukkan pada Gambar 3.8 dibawah ini.

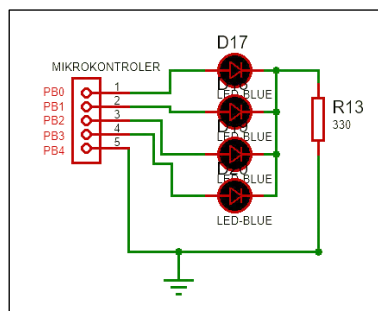


Gambar 3.8 Rangkaian Skematik *Pushbutton*

Gambar 3.8 merupakan rangkaian *pushbutton* yang digunakan untuk memberikan interupsi pada mikrokontroler sehingga PORT yang digunakan untuk *pushbutton* harus disetting sebagai *input* dengan mode *pul up*. Pada penelitian ini push button digunakan untuk melakukan setting sebelum proses dijalankan. *Pushbutton* terhubung dengan PORTB.0 – PORTB.5.

5. Penjelasan Rangkaian Indikator *Syringe Size*

Sebagai sebuah indikator spuit, dibuatlah sebuah rangkaian indikator yang ditunjukkan pada Gambar 3.9 dibawah ini.

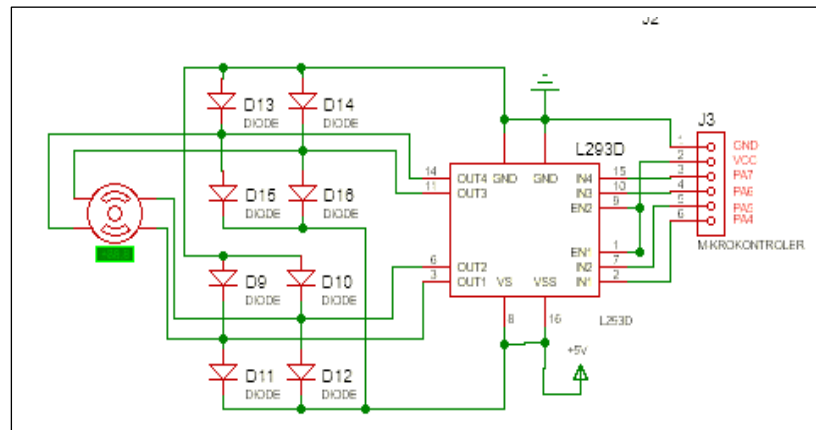


Gambar 3.9 Rangkaian Skematik Indikator *Syringe Size*

Pada rangkaian ini terdiri dari komponen utama yaitu LED, terdapat 4 buah LED dimana masing-masing LED menjadi indikator bagi ukuran spuit, yaitu indikator tidak ada spuit, spuit 10 ml, 20 ml, dan 50 ml. Pada skematik yang dibuat oleh penulis, LED dipasang secara *common cathode* artinya dalam kondisi awal kaki *cathode* LED sudah mendapatkan logika *low* (0) sehingga membutuhkan logika *high* (1) untuk menyalakan masing-masing LED ini. Nyala masing-masing LED diatur oleh mikrokontroler sesuai *spuit* yang terpasang. Indikator *syringe size* terhubung pada PORTD.4 – PORTD.7.

6. Penjelasan Rangkaian *Driver Motor*

Sebagai penggerak utama alat, motor stepper memerlukan sebuah driver untuk menjalankan motor stepper. Adapun rangkaian skematik driver motor ditunjukkan pada Gambar 3.10 dibawah ini.



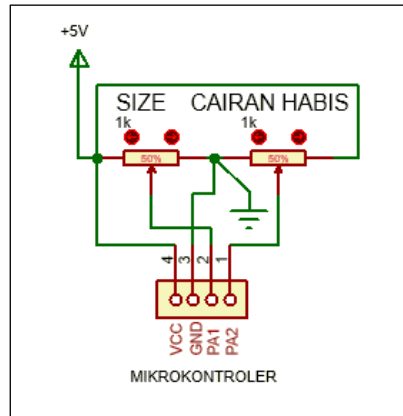
Gambar 3.10 Rangkaian Skematik *Driver Motor*

Motor *stepper* ini memiliki 4 logika yang berbeda dan dijalankan secara berurutan dengan nilai *delay* yang sama pada setiap masukan yang diberikan. Logika yang diberikan pada IC L298 berupa data biner yang

dimasukan melalui mikrokontroler, sehingga motor stepper tersebut dapat bergerak. Keluaran IC L298 diberikan dioda agar tidak ada *feedback* dari motor *stepper*. *Driver* motor terhubung pada PORTA.4 – PORTA.7.

7. Penjelasan Rangkaian Sensor *Syringe Size*

Alat yang dirancang ditambahkan sensor *syringe size* dan sensor cairan hampir habis. Adapun rangkaian skematik sensor *syringe size* dan cairan hampir habis ditunjukkan pada Gambar 3.11 dibawah ini.



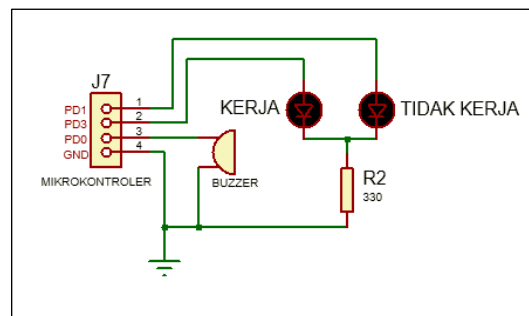
Gambar 3.11 Rangkaian Skematik *Sensor Empty* dan *Syringe Size*

Rangkaian ini menggunakan sensor yang sama yaitu menggunakan potensiometer. dimana potensiometer akan berubah-ubah nilai resistansinya sesuai besar spuit yang digunakan. begitupun pemanfaatan potensiometer sebagai sensor cairan habis. pada nilai resistansi tertentu maka akan menandakan cairan pada spuit hampir habis. Dari nilai resistansi tersebut dapat dirubah menjadi nilai tegangan dengan memanfaatkan sistem pembagi tegangan (tegangan referensi) sehingga akan dapat diolah dalam sistem ADC mikrokontroler. *Sensor empty*

terhubung pada PORTA.1 sedangkan sensor *syringe size* terhubung pada PORTA.2.

8. Penjelasan Rangkaian Indikator dan *buzzer*

Alat yang dirancang ditambahkan *buzzer* dan indikator kerja. Adapun rangkaian skematik indikator kerja dan *buzzer* ditunjukkan pada Gambar 3.12 dibawah ini.

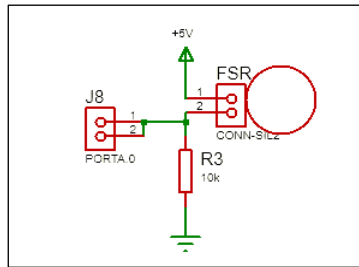


Gambar 3.12 Rangkaian Skematik Indikator dan *buzzer*

Skematik ini merupakan sebuah gambaran rangkaian yang dibuat oleh penulis untuk menunjukkan sebuah indikator alat bekerja/tidak serta *buzzer* sebagai penanda cairan hampir habis/cairan habis. Pada rangkaian ini indikator dan *buzzer* terdiri dari komponen utama berupa LED dan *buzzer*, dimana LED dan *buzzer* dipasang secara *common cathode* artinya indikator atau *buzzer* akan aktif ketika mendapatkan logika *high* dari mikrokontroler. Indikator kerja terhubung pada PORTD.3 sedangkan indikator tidak kerja terhubung pada PORTD.1 dan *buzzer* terhubung pada PORTD.0.

9. Penjelasan Rangkaian Sensor Oklusi

Pada perancangan alat penulis menambahkan sensor oklusi. Adapun rangkaian skematik sensor oklusi ditunjukkan pada Gambar 3.13 dibawah ini.



Gambar 3.13 Rangkain Skematik Sensor Oklusi

Skematik ini merupakan sebuah gambaran rangkaian yang dibuat oleh penulis agar sensor oklusi agar dapat bekerja. Sensor oklusi pada dasarnya merupakan sensor tekanan. Pada penelitian ini sensor tekanan yang digunakan adalah sensor FSR 402, sensor ini merupakan sensor tekanan yang berbasis resistansi. Setiap terjadi sumbatan pada selang atau jalur aliran obat. maka tekanan akan semakin berat, dorongan spuit akan semakin berat. Dengan memandang kondisi tersebut maka peneliti akan mengambil nilai perubahan tekanan tersebut. FSR yang bernilai resistansi maka harus dirubah menjadi output tegangan dengan menggunakan sistem pembagi tegangan. Dari output inilah nilai sensor dapat diolah oleh mikrokontroler. Sensor oklusi terhubung pada PORTA.0.

3.6 Implementasi Perangkat Lunak

1. Code Program Untuk Tampilan

```
void tampil_lcd()
{
    nilai=(float) satuan* (data3/3600.0);
    ftoa (nilai, 2, temp);
    lcd_gotoxy (4, 1);
    lcd_puts (temp);
    lcd_gotoxy (10, 1);
    lcd_putsf ("ml");
    lcd_gotoxy (0, 0);
    lcd_putsf ("-CAIRAN KELUAR-");
}
```

Hasil counter yang nilainya terdapat pada variabel “satuan” dimasukkan sebuah rumus untuk menentukan berapa ml/s. Kemudian variabel “nilai” yang semula bertipe data float dikonversi menjadi data aschi (ftoa) agar nilainya dapat ditampilkan pada LCD dalam bentuk desimal kemudian disimpan dalam tipe data char temp (karakter) dan ditampilkan pada LCD dengan koordinat (4,1) berada di baris ke-2 dan kolom ke-5 serta diikuti tulisan satuan ml pada koordinat (10,1). Terdapat tulisan karakter “-CAIRAN KELUAR-” pada koordinat (0,0) .

2. Code Program Untuk Pengaturan

```

void menu1()
{
  if(!up) {data3=data3+1; delay_ms(200);}
  if(!down){data3=data3-1; delay_ms(200);}
  //tombol setting menu1
  lcd_gotoxy(2,0);
  lcd_putsf("-FLOW RATE-");
  lcd_gotoxy(3,1);
  sprintf(uy,"%d mL/h   ", data3);
  lcd_puts(uy);
  if(data3==d){data3=1;}
  if(data3==0){data3=d-1;}
  //mengatur volume maksimal yaitu data1
}
void menu2()
{
  if(!up) {data6=data6+1; delay_ms(200);}
  if(!down){data6=data6-1; delay_ms(200);}
  //tombol setting menu2
  lcd_gotoxy(2,0);
  lcd_putsf("-BOLUS MODE-");
  lcd_gotoxy(3,1);
  sprintf(huy,"%d mL/h   ", data6);
  lcd_puts(huy);
  if(data6==0){data6=data3+1;}
  if(data6==101){data6=data3+1;}
  if(data6==data3){data6=101-1;}
}
void menu3()
{
  if(!up) {data4=data4+1; delay_ms(200);}
  if(!down){data4=data4-1; delay_ms(200);}
  //tombol setting menu2
  lcd_gotoxy(2,0);
  lcd_putsf("-MAX VOLUME-");
}

```

```

sprintf(buf," %d ml  ", data4);
lcd_puts(buf);
if(data4==0){data4=e;}
if(data4==e+1){ data4=1;}
//mengatur volume maksimal yaitu data1
}

void oklusi()
{
if(!up) {data5=data5+1; delay_ms(500); lcd_clear();}
if(!down){data5=data5-1; delay_ms(500); lcd_clear();}
if(data5==3){data5=0;}
if(data5==-1){data5=2;}
switch(data5)
{
case 0 :low();break;
case 1 :medium();break;
case 2 :high();break;
} //setting menu tampilan awal
}
while (1)
{
adc();
lcd_gotoxy(1,0);
lcd_putsf("__OCCLUSION__");
oklusi();
}
}

```

Program diatas merupakan program untuk mode pengaturan, mula-mula sistem akan menjalankan program pada fungsi “oklusi” untuk melakukan pemilihan level oklusi, dimana ketika tombol up ditekan maka PIN UP (!up) atau artinya tidak sama dengan. Dengan ini data5++ (bertambah satu setiap kondisi tombol UP ditekan) dan ketika tombol down ditekan maka PIN DOWN (!down) atau artinya tidak sama dengan. Dengan ini data5-- (berkurang satu setiap kondisi !down). setiap nilai dari data5 mempunyai isi, yaitu ketika data5=0 maka akan menjalankan program di fungsi “low”, ketika data=1 maka akan menjalankan program difungsi “medium” dan ketika data5=2 maka akan menjalankan program difungsi “high”. Setelah ditekan tombol “START” maka akan menjalankan program sesuai dengan nilai data5.

Setelah menjalankan fungsi *low*, *medium*, atau *high* akan masuk ke fungsi pemilihan *flow rate* dimana pada fungsi ini ketika tombol *up* ditekan (!up) maka nilai akan bertambah 1, sedangkan ketika tombol *down* ditekan (!down). maka nilai data3 akan berkurang 1. Nilai data3 akan ditampilkan pada LCD dengan koordinat (3,1). Pada proses ini ketika ditekan “START” maka akan menyimpan nilai terakhir dari data3 dan akan melanjutkan pengaturan selanjutnya yaitu, *flow rate bolus* dan volume. Proses pengaturan *flow rate bolus* dan volume menggunakan kode program yang sejenis namun berbeda pada penyimpanan nilainya, *flow rate bolus* disimpan kedalam variabel data6 dan volume disimpan pada variabel data4. Semua data akan disimpan dan akan menentukan pergerakan motor, tampilan LCD, dsb.

3. Kode Program Sensor *Nearly Empty*, Oklusi, dan *Syringe Size*

```

void adc()
{
  data1=read_adc(1);
  data2=read_adc(2);
  if (data1>=200 && data1<=300)
  {PORTD.6=1; a=100.0; c=537; d=301; e=50; f=700;
  g=680; h=2; i=1.5;}
  else {PORTD.6=0;}
  if (data1>=450 && data1<=600) {PORTD.5=1; a=214.0;
  c=404; d=101; e=20; f=515; g=422; h=1.5; i=1;}
  else {PORTD.5=0;}
  if (data1>=620 && data1<=700) {PORTD.4=1; a=333.6;
  c=379; d=101; e=10; f=480; g=396; h=1; i=0.5;}
  else {PORTD.4=0;}
  if (data1>700) {PORTD.7=1;}
  else {PORTD.7=0;}
}
void soklusi()
{
  data=read_adc(3);
}

```

Program diatas merupakan kode program pembacaan sensor *nearly empty*, oklusi, dan *syringe size*. *read_adc* merupakan kode program

untuk membaca suatu nilai ADC. Angka dalam kurung dibelakang `read_adc` menunjukkan tempat pembacaan nilai ADC, contohnya `read_adc(1)` berarti melakukan pembacaan nilai ADC pada pin 1 ADC (PORTA.1). Hasil pembacaan sensor akan disimpan pada variabel data `data1` untuk ADC1 (*syringe size*), variabel `data2` untuk ADC2 (*nearly empty*), variabel `data3` untuk ADC3 (oklusi).

4. Kode Program Putaran Motor dan Kecepatan Motor

```
void kecepatan()
{
    hasil_volume=data4*a;
    t=(3600000*data4/data3);
    ts=t/hasil_volume;
    t_tunda=ts/12;
}
void putar()
{
    PORTA=0b00010000;
    delay_ms(t_tunda);
    PORTA=0b00100000;
    delay_ms(t_tunda);
    PORTA=0b01000000;
    delay_ms(t_tunda);
    PORTA=0b10000000;
    delay_ms(t_tunda);
}
```

Kode program diatas merupakan kode program untuk menjalankan motor stepper dan mengatur kecepatan putarannya. motor *stepper* memiliki 4 pin *input*, maka untuk menggerakkan motor *stepper* perlu membuat pulsa pada 4 *input* tersebut, terlihat pada program diatas PORTA.4-PORTA.7 bergantian menghasilkan *output* sesuai *delay* yang diberikan. Dengan *output* dari PORTA.4-PORTA.7 ini membuat alat motor *stepper* berputar. Kecepatan motor *stepper* tergantung dari nilai dari variable “`t_tunda`”. Nilai “`t_tunda`” diperoleh dari rumus yang ada pada fungsi kecepatan, pada rumus terdapat `data3`, dimana `data3`

merupakan nilai *flow rate setting*, maka dari itu hasil *setting* akan terakumulasi pada rumus kecepatan.

5. Kode Program Mode *Bolus*

```
void boluss()
{
    hasil_volume=data4*a;
    tb=(3600000*data4/data6);
    tsb=tb/hasil_volume;
    t_bolus=tsb/12;
}

if(!bolus)
{
    TIMSK=0x01;
    istop=0;
    irun=1;
    tampil_lcdbolus();
    boluss();
    adc();
    soklusi();
    PORTA=0b00010000;
    delay_ms(t_bolus);
    PORTA=0b00100000;
    delay_ms(t_bolus);
    PORTA=0b01000000;
    delay_ms(t_bolus);
    PORTA=0b10000000;
    delay_ms(t_bolus);
}
```

Program diatas merupakan kode program untuk mode *bolus*, dimana isi programnya hampir sama dengan program menggerakkan motor *stepper* sebelumnya hanya saja memiliki perbedaan pada kecepatannya, pada program pengatur kecepatan mode *bolus* tertulis menggunakan *data6* pada rumusnya. Variabel *data6* merupakan nilai *setting* dari mode *bolus* (berupa nilai *flow rate bolus*). Sehingga ketika proses injeksi mode *infusion* saat ditekan tombol STOP dan dilanjutkan ditekan tombol *bolus* maka motor akan berputar dengan kecepatan berbeda dengan kecepatan mode *infusion* karena kecepatan mode *bolus* menggunakan nilai *delay* dari variable *t_bolus*.

3.7 Standar Prosedur Operasional Penggunaan Alat

1. Melakukan pemasangan alat pada tiang infus dan bawa kedekat pasien
2. Menyiapkan spuit dan medikasi pasien
3. Memasang spuit ke *syringe pump* dan menutup kembali *syringe size*
4. Mengaktifkan alat dengan menekan saklar power ke posisi ON
5. Melakukan penyambungan selang penghubung dari spuit ke akses *intravena* pasien
6. Melakukan pengaturan pada alat (level oklusi, *flow rate*, *flow rate bolus* dan volume)
7. Untuk memulai proses dapat menekan tombol “START” apabila sudah selesai melakukan pengaturan (Indikator warna hijau akan menyala tanda alat bekerja)
8. Cara menggunakan mode *bolus* dengan cara menghentikan proses sebelumnya dengan menekan tombol STOP, lalu tombol BOLUS digunakan untuk memulai menginjeksi secara terus menerus. Selama tombol ditekan maka akan terus melakukan injeksi dengan kecepatan mode bolus, ketika tombol BOLUS tidak ditekan maka proses injeksi berhenti. Setelah injeksi manual sudah cukup dapat menekan tombol START untuk kembali ke mode awal
9. Jika terjadi alarm maka hal yang harus dilakukan adalah mencari sumber sumbatan, setelah sumber error ditemukan proses akan dapat dijalankan kembali dengan menekan tombol START.

10. Jika alat telah selesai digunakan maka cara mematikan adalah dengan menekan saklar power ke posisi OFF dan komponen yang digunakan untuk injeksi obat terhadap pasien dapat dilepas.

3.8 Metode Pengukuran *Flow Rate*

Metode pengukuran yang digunakan adalah menggunakan pembandingan berupa alat ukur *Infusion Device Analyzer* yang merupakan alat kalibrasi *infusion pump* dan *syringe pump*. Menggunakan pembandingan ini menjadikan hasil pengukuran menjadi lebih akurat. Adapun langkah-langkah pengukuran sebagai berikut:

1. Menyiapkan *syringe pump* dan alat ukurnya *Infusion Device Analyzer*
2. Menyiapkan lembar pengukuran yang standar kalibrasi
3. Melakukan *setting flow rate* pada nilai 100 ml/jam, 75 ml/jam, 50 ml/jam, 25 ml/jam, dan 10 ml/jam (pada masing-masing ukuran spuit)
4. Melakukan pengambilan data pada masing-masing kecepatan sebanyak 5x dalam jangka waktu 1 menit. Pengambilan data dimulai menit ke-4.
5. Mencatat nilai hasil pengukuran pada lembar pengukuran
6. Menganalisa hasil pengukuran dan membuat kesimpulan
7. Untuk pengujian *flow rate bolus* dengan cara melakukan langkah 1-6, dengan mengatur *flow rate bolus* sebesar 50 ml/jam dan 100ml/jam
8. Pengambilan data sebanyak 3x dengan jangka waktu pengambilan per 1 menit dimulai menit ke-4.