

**RANCANG BANGUN *SYRINGE PUMP* BERBASIS MIKROKONTROLER
ATMEGA8535 DILENGKAPI DETEKTOR OKLUSI**

SKRIPSI



NADA FITRIEYATUL HIKMAH

**PROGRAM STUDI S1 TEKNOBIOMEDIK
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS AIRLANGGA
2012**

**RANCANG BANGUN SYRINGE PUMP BERBASIS MIKROKONTROLER
ATMEGA8535 DILENGKAPI DETEKTOR OKLUSI**

SKRIPSI

Sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Teknik Bidang Teknobiomedik
Pada Fakultas Sains dan Teknologi
Universitas Airlangga

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Imam Sapuan, S.Si, M.Si
NIP. 19720118 199903 1 001

Triwiyanto, S.Si, M.T
NIP. 19730502 200312 1 002

LEMBAR PENGESAHAN NASKAH SKRIPSI

Judul : Rancang Bangun *Syringe Pump* Berbasis
Mikrokontroler ATmega8535 Dilengkapi Detektor
Oklusi

Penyusun : Nada Fitrieyatul Hikmah

NIM : 080810293

Tanggal Ujian : 11 Juli 2012

Disetujui oleh :

Pembimbing I,

Pembimbing II,

Imam Sapuan, S.Si, M.Si
NIP. 19720118 199903 1 001

Triwiyanto, S.Si, M.T
NIP. 19730502 200312 1 002

Mengetahui :

Ketua Departemen Fisika
Fakultas Sains dan Teknologi
Universitas Airlangga

Ketua Prodi S-1 Teknobiomedik
Fakultas Sains dan Teknologi
Universitas Airlangga

Drs. Siswanto, M.Si
NIP . 19640305 198903 1 003

Dr. Retna Apsari, M.Si
NIP . 19680626 199303 2 003

PEDOMAN PENGGUNAAN SKRIPSI

Skripsi ini tidak dipublikasikan, namun tersedia di perpustakaan dalam lingkungan Universitas Airlangga, diperkenankan untuk dipakai sebagai referensi kepustakaan, tetapi pengutipan harus seizin penyusun dan harus menyebutkan sumbernya sesuai kebiasaan ilmiah. Dokumen skripsi ini merupakan hak milik Universitas Airlangga.



Kepada Umi, Ayah, dan keluargaku tercinta.....

Kupersembahkan segala dukungan dan do'a kalian berupa tinta dan kertas, serta hasil alat dalam penelitian ini.....

Kuyakin tidak ada yang tidak mungkin dilakukan di dunia ini, kuncinya adalah belajar, berdo'a, dan berusaha.....

- prinsipku-

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT yang senantiasa memberikan hidayah, inayah, dan rahmat-Nya sehingga penyusun dapat menyelesaikan naskah skripsi yang berjudul **“Rancang Bangun *Syringe Pump* Berbasis Mikrokontroler ATmega8535 Dilengkapi Detektor Oklusi”**.

Naskah skripsi ini disusun untuk memenuhi salah satu syarat kelulusan. Pada kesempatan ini, penyusun menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Ketua Departemen Fisika, Bapak Drs. Siswanto, M.Si, yang telah memberikan fasilitas laboratorium untuk pelaksana penelitian skripsi ini.
2. Ketua Program Studi S1 Teknobiomedik, Ibu Dr. Retna Apsari, M.Si, yang telah memberikan informasi tentang penyusunan naskah skripsi ini.
3. Bapak Imam Sapuan, S.Si, M.Si selaku pembimbing I yang telah membimbing penyusun dalam penyusunan laporan tugas akhir ini dengan dukungan dan saran yang diberikan.
4. Bapak Triwiyanto, S.Si, M.T selaku pembimbing II yang telah bersedia membantu dalam hal konsultasi, bimbingan, bantuan dan dukungan serta ide-ide yang diberikan selama penyusunan dalam mengerjakan tugas akhir ini.
5. Bapak H. Rahmatullah, S.T dan Bapak Rofiq yang telah banyak memberikan masukan untuk penyusun.

6. Bapak Heri RS Darmo yang telah membantu penyusun mencari mekanik bekas alat *syringe pump*.
7. Mas Iqbal distributor alat *syringe pump* Terumo yang telah bersedia meluangkan waktu bagi penyusun untuk berkonsultasi mengenai alat *syringe pump* dan meminjamkan alat untuk dipelajari mekanismenya.
8. Rekan kerja setiaku, mas Farid Wahyudi yang selalu menemani penyusun dan memberikan motivasi.
9. Teman-teman lab di Atem, Made, Cahyo, Dita, Wawan, dan teman-teman Atem lainnya, terima kasih atas bantuan dan kerjasamanya selama di lab.
10. Riant Adzandy yang telah membantu penyusun menganalisis mekanik yang digunakan.
11. Sahabat-sahabatku yang senantiasa memberi motivasi, Adista, Bibah, Taufiq, Wida, Reza.
12. Mas Lucky yang membantu memeriksa hasil rangkaian yang telah dibuat.
13. Teman SMP-ku Bajonk, terima kasih sudah memberikan waktu luang untuk berkonsultasi masalah elektronika.
14. Fitrotin Nufus yang telah memberi masukan-masukan kepada penyusun.
15. Aditya Iman Rizky yang telah membantu membenahi penulisan abstrak versi Bahasa Inggris.
16. Semua teman-teman Teknobiomedik angkatan 2008 yang selalu mendukung dan membantu baik dalam proses perkuliahan maupun yang lain.

17. Serta semua pihak yang tidak dapat disebutkan disini atas dukungan dan do'a kepada penyusun selama penyelesaian laporan tugas akhir ini.

Penyusun menyadari bahwa naskah skripsi ini masih banyak kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk perbaikan naskah skripsi ini.

Surabaya, Juni 2012

Penyusun

Nada Fitriyatul Hikmah



Hikmah, Nada Fitriyatul, 2012, Rancang Bangun *Syringe Pump* Berbasis Mikrokontroler ATmega8535 Dilengkapi Detektor Oklusi. Skripsi di bawah bimbingan Imam Sapuan, S.Si, M.Si dan Triwiyanto, S.Si, M.T, Departemen Fisika, Fakultas Sains dan Teknologi, Universitas Airlangga, Surabaya

ABSTRAK

Telah dilakukan penelitian merancang alat *syringe pump* dilengkapi dengan detektor oklusi dengan *load cell* sebagai komponen sensor. Penelitian ini bertujuan untuk menghasilkan *syringe pump* yang dilengkapi dengan menu volume obat dan *flow rate* untuk injeksi sehingga memudahkan perawat dalam mengontrol obat, serta dilengkapi dengan rangkaian untuk mendeteksi terjadinya oklusi. Pada penelitian ini sistem operasi alat sudah diprogram dengan sistem pergerakan motor sehingga perawat hanya menentukan dosis volume obat dengan rentang 1 ml hingga 50 ml dan *flow rate* dengan rentang 1 ml/jam hingga 50 ml/jam yang perlu diberikan kepada pasien untuk selanjutnya diolah oleh mikrokontroler. *Syringe pump* yang telah dibuat dilengkapi dengan buzzer yang digunakan sebagai alarm *nearly empty*, oklusi, dan volume cairan obat telah selesai diinjeksikan. Alat ini mempunyai tingkat keakuratan yang tinggi dalam menginjeksikan volume cairan obat, sedangkan untuk variabel *flow rate* mempunyai tingkat akurasi sebesar 98,92% dan tingkat presisi 99,88%. Pada sistem detektor oklusi menggunakan sensor *load cell* mendeteksi terjadinya oklusi pada tekanan 100 mmHg dengan tingkat akurasi sebesar 91,60%.

Kata kunci : *syringe pump*, oklusi, volume, flow rate, load cell

Hikmah, Nada Fitrieyatul, 2012, Design of Microcontroller ATmega8535-Based Syringe Pump Equipped with Occlusion Detector. Final project was under guidance Imam Sapuan, S.Si, M.Si and Triwiyanto, S.Si, M.T, Department of Physics, Fakultas Sains dan Teknologi, Airlangga University, Surabaya

ABSTRACT

A research has been conducted to design a syringe pump tool equipped with an occlusion detector as well as a load cell as the sensor component. This study aimed to produce a syringe pump equipped with a menu of drug volume and flow rate for injection that makes it easier for nurses to control drug, and is equipped with a circuit for detecting the occurrence of occlusion. In this study, the device operating system is programmed with motor movement system so that nurses only need to determine the drug volume dosage that will be given to the patients in the range of 1 ml to 50 ml and flow rate in the range of 1 ml/hour to 50 ml/hour to further be processed by the microcontroller . Syringe pump that has been created, is equipped with buzzer that is used as an alarm informing nearly empty and occlusion stages, and that the drug fluid volume has been injected. This device has a high degree of accuracy in injecting drug fluid volume, while the flow rate variable have an accuracy rate of 98.92% and precision rate of 99,88%. In the occlusion detector system, a load cell sensor was used, and was able to detect the occurrence of occlusion at a pressure of 100 mmHg with an accuracy rate of 91.60%.

Keywords : syringe pump, occlusion, volume, flow rate, load cell

DAFTAR ISI

LEMBAR JUDUL	i
LEMBAR PERNYATAAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR PEDOMAN PENGGUNAAN SKRIPSI.....	iv
KATA PENGANTAR	v
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xv
DAFTAR GAMBAR	xvi
DAFTAR LAMPIRAN	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang Permasalahan	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
BAB II TINJAUAN PUSTAKA.....	6
2.1 <i>Syringe Pump</i>	6
2.1.1 Tinjauan Umum <i>Syringe Pump</i>	6
2.1.2 Prinsip Dasar Penggunaan <i>Syringe Pump</i>	7
2.1.3 Volume dan <i>Flow Rate</i>	8

2.1.4 Sistem Oklusi Intravena	9
2.2 Mikrokontroler ATmega8535	10
2.3 LCD (<i>Liquid Crystal Display</i>).....	11
2.4 Motor Stepper	14
2.4.1 Tinjauan Umum Motor Stepper	14
2.4.2 Prinsip Kerja Motor Stepper	15
2.4.2.1 Mode <i>Wave</i>	16
2.4.2.2 Mode <i>Full Step</i>	16
2.4.2.3 Mode <i>Half Step</i>	17
2.4.3 <i>Driver Motor</i> L298	17
2.5 Sensor Optocoupler	18
2.6 Transduser <i>Load Cell</i>	20
2.6.1 Tinjauan Umum <i>Load Cell</i>	20
2.6.2 Prinsip Kerja <i>Load Cell</i>	20
2.6.3 Konfigurasi <i>Wheatstone Bridge</i>	23
2.7 Sensor Potensio Geser	25
2.8 Komponen Pendukung	27
2.8.1 IC LM358	27
2.8.2 Transistor PNP BC557	28
2.8.2.1 Prinsip Kerja Transistor PNP	28
2.8.2.2 Daerah Operasi Transistor	29
2.8.3 Buzzer	30
2.8.4 <i>Instrumentation Amplifier</i> AD620	31
2.8.4.1 Tinjauan <i>Instrumentation Amplifier</i>	31
2.8.4.2 <i>Gain</i> AD620	33

2.9 Rumus Dasar Roda Gigi	34
2.10 Pemrograman CodeVision AVR	36
2.10.1 Tinjauan CodeVision AVR	36
2.10.2 Bahasa Pemrograman CodeVision AVR	37
BAB III METODE PENELITIAN.....	40
3.1 Tempat dan Waktu Penelitian	40
3.2 Peralatan dan Bahan Penelitian	40
3.2.1 Peralatan Penelitian	40
3.2.2 Bahan Penelitian	41
3.3 Prosedur Penelitian.....	41
3.3.1 Tahap Persiapan	43
3.3.2 Tahap Perancangan	45
3.3.2.1 Analisis Mekanik	45
3.3.2.2 Perancangan Perangkat Keras (<i>Hardware</i>).....	51
3.3.2.2.1 Rangkaian Catu Daya	51
3.3.2.2.2 Rangkaian Minimum Sistem AVR ATmega8535	51
3.3.2.2.3 Rangkaian <i>Driver Motor</i> L298	53
3.3.2.2.4 Rangkaian LCD	53
3.3.2.2.5 Rangkaian Buzzer.....	54
3.3.2.2.6 Rangkaian Potensio Geser	55
3.3.2.2.7 Rangkaian Optocoupler dan Komparator	56
3.3.2.2.8 Rangkaian <i>Load Cell</i>	57
3.3.2.3 Perancangan Perangkat Lunak (<i>Software</i>).....	58
3.3.3 Tahap Pengujian.....	60
3.3.4 Tahap Pengambilan Data	62

3.3.5 Analisis Data.....	62
3.3.5.1 Presisi (Ketelitian)	62
3.3.5.2 Akurasi (<i>Recovery</i>)	63
BAB IV HASIL DAN PEMBAHASAN	64
4.1 Hasil Perancangan Alat	64
4.1.1 Analisis Mekanik	64
4.1.2 Perangkat Keras (<i>Hardware</i>)	65
4.1.2.1 Rangkaian Catu Daya	66
4.1.2.2 Rangkaian <i>Driver Motor</i> L298	66
4.1.2.3 Rangkaian Optocoupler dan Komparator	67
4.1.2.4 Rangkaian LCD	68
4.1.2.5 Rangkaian Buzzer	69
4.1.2.6 Rangkaian Minimum Sistem AVR ATmega8535	70
4.1.2.7 Rangkaian Potensio Geser	71
4.1.2.8 Rangkaian <i>Load Cell</i>	72
4.1.3 Perangkat Lunak (<i>Software</i>)	75
4.1.3.1 Program Push Button <i>Up-Down</i>	76
4.1.3.2 Program <i>Setting Volume</i>	77
4.1.3.3 Program <i>Setting Flow Rate</i>	77
4.1.3.4 Program Menjalankan Motor Stepper	78
4.1.3.5 Program Deteksi <i>Nearly Empty</i>	79
4.1.3.6 Program Deteksi Oklusi.....	80
4.1.3.7 Program Tampilan LCD	80
4.2 Hasil Pengujian Alat dan Analisis Data	81
4.2.1 Kalibrasi Motor Stepper.....	81

4.2.2 Hasil Pengujian Volume	82
4.2.3 Hasil Pengujian Waktu dan <i>Flow Rate</i>	84
4.2.4 Hasil Pengujian Tekanan	86
BAB V KESIMPULAN DAN SARAN.....	89
5.1 Kesimpulan.....	89
5.2 Saran.....	89
DAFTAR PUSTAKA	91
LAMPIRAN	



DAFTAR TABEL

Nomor	Judul Tabel	Halaman
2.1	Fungsi pin pada LCD	12
2.2	Langkah pada mode <i>wave</i>	16
2.3	Langkah pada mode <i>full step</i>	16
2.4	Langkah pada mode <i>half step</i>	17
4.1	Hasil pengukuran tegangan keluaran rangkaian <i>load cell</i>	73
4.2	Hasil pengujian <i>count</i> untuk volume 1 ml	82
4.3	Hasil pengujian volume.....	83
4.4	Hasil pengujian volume, waktu, dan <i>flow rate</i>	84
4.5	Data persentase kesalahan dari <i>flow rate</i>	85
4.6	Hasil pengujian pada tekanan 100 mmHg	86

DAFTAR GAMBAR

Nomor	Judul Gambar	Halaman
2.1	Bagian-bagian <i>syringe pump</i> J-1047	7
2.2	<i>Syringe pump</i> dengan sistem aliran fluida pada <i>intravenous line</i>	9
2.3	Konfigurasi pin mikrokontroler ATmega8535	10
2.4	Rangkaian LCD.....	13
2.5	Motor stepper bipolar.....	14
2.6	Prinsip kerja motor stepper	15
2.7	Rangkaian dasar <i>driver motor</i> L298	18
2.8	Sensor optocoupler.....	19
2.9	Penempatan optocoupler pada piringan sensor	19
2.10	Konfigurasi <i>load cell</i>	20
2.11	<i>Strain gauge</i>	22
2.12	Rangkaian <i>strain gauge full bridge</i>	23
2.13	Konfigurasi <i>strain gauge</i>	24
2.14	Potensio geser	26
2.15	Rangkaian pembagi tegangan	26
2.16	Konfigurasi pin pada LM358	27
2.17	Rangkaian komparator tegangan sederhana.....	28
2.18	Kurva I_C terhadap V_{CE}	29
2.19	Bentuk fisik buzzer	30

DAFTAR GAMBAR

Nomor	Judul Gambar	Halaman
2.20	Rangkaian <i>instrumentation amplifier</i>	31
2.21	Rangkaian <i>instrumentation amplifier</i> pada IC AD620	33
2.22	Konfigurasi pin pada AD620	33
2.23	Sebuah <i>gear</i> yang berotasi terhadap sumbu tetap melalui pusatnya.....	34
2.24	Dua <i>gear</i> yang saling bersinggungan.....	36
2.25	Kotak dialog 1	38
2.26	Kotak dialog 2	38
2.27	<i>Code Wizard</i> AVR dan <i>Generate Program</i>	39
3.1	Diagram alir prosedur penelitian.....	42
3.2	Diagram blok <i>syringe pump</i>	43
3.3	Mekanik sistem <i>gear</i> tampak depan.....	46
3.4	Mekanik sistem <i>gear</i> tampak belakang.....	46
3.5	Rancangan sistem mekanik alat <i>syringe pump</i>	47
3.6	Rangkaian catu daya	51
3.7	Rangkaian minimum sistem AVR ATmega8535	52
3.8	Rangkaian <i>driver motor</i> L298	53
3.9	Rangkaian LCD.....	54
3.10	Rangkaian buzzer	55
3.11	Rangkaian potensio geser.....	55

DAFTAR GAMBAR

Nomor	Judul Gambar	Halaman
3.12	Rangkaian optocoupler dan komparator	56
3.13	Rangkaian <i>load cell</i>	57
3.14	<i>Flowchart</i> perangkat lunak (<i>software</i>) <i>syringe pump</i>	59
3.15	Pengujian tekanan dengan tabung pemodelan intravena	61
4.1	Perangkat mekanik alat <i>syringe pump</i>	64
4.2	Susunan <i>gear</i> pada perangkat mekanik	65
4.3	Rangkaian catu daya	66
4.4	Pemasangan optocoupler pada piringan sensor	67
4.5	Rangkaian <i>driver</i> motor, optocoupler dan komparator, <i>buzzer</i> , LCD, dan minimum sistem alat <i>syringe pump</i>	70
4.6	Sensor potensio geser	71
4.7	Hubungan linieritas volume dengan resistansi potensio geser	71
4.8	Hubungan linieritas tegangan keluaran <i>load cell</i> dengan tekanan	72
4.9	Bentuk fisik <i>load cell</i> 3 kg	74
4.10	Hasil rangkaian <i>instrumentation amplifier</i> dan <i>low pass filter</i>	74
4.11	Proses <i>compile</i> program pada CodeVisionAVR	75
4.12	Tampilan PROGISP (Ver 1.68) <i>software downloader</i>	76
4.13	Grafik linieritas volume	83
4.14	Grafik linieritas <i>flow rate</i>	85

DAFTAR LAMPIRAN

Nomor	Judul Lampiran
1	Skematik Rangkaian <i>Syringe Pump</i>
2	Hasil Alat <i>Syringe Pump</i>
3	<i>Listing Program</i>
4	<i>Datasheet</i> AD620
5	<i>Datasheet</i> L298
6	<i>Datasheet</i> LM358
7	<i>Datasheet Load Cell</i>
8	Data Hasil Perhitungan Presisi Variabel <i>Flow Rate</i>
9	Data Tegangan Keluaran <i>Load Cell</i> Terhadap Tekanan

BAB I

PENDAHULUAN

1.1 Latar Belakang Permasalahan

Rumah sakit merupakan sistem yang sangat kompleks sehingga sulit untuk mengontrol setiap pasien. Dengan semakin banyaknya kasus malpraktik dan seiring perkembangan teknologi di bidang kedokteran, tuntutan masyarakat akan keselamatan jiwa semakin meningkat yang berkaitan dengan reputasi rumah sakit. Hal tersebut menyebabkan perlunya peningkatan kualitas pada bidang medis. Perkembangan dunia medis dibangun oleh dua faktor penting yaitu faktor manusia dan faktor perlengkapan medis. Faktor perlengkapan medis meliputi obat-obatan dan peralatan medis yang menunjang kebutuhan medis. Salah satu alat medis yang sangat penting dan mendasar adalah jarum suntik. Jarum suntik difungsikan sebagai alat medis untuk memudahkan cairan obat masuk ke dalam tubuh pasien (*MedicineNet Dictionary*, 2011).

Bagi pasien yang membutuhkan pengobatan ekstra dan intensif, maka diperlukan suatu alat yang dapat mengontrol dosis volume penggunaan obat dan *flow rate* obat yang akan diinjeksikan. *Flow rate* adalah banyaknya atau volume fluida yang mengalir diukur per satuan waktu. Alat medis yang dapat melakukan injeksi secara otomatis adalah *syringe pump*. Dalam hal ini, perawat hanya memberi *input* pada alat berupa volume obat yang dibutuhkan serta *flow rate* yang dikondisikan dengan kebutuhan pasien mendapatkan volume obat per jamnya (ml/jam).

Dalam kasus penyakit jantung dan saraf misalnya, pemberian cairan obat harus tepat dan konstan atau dengan kata lain jumlah cairan yang diberikan sesuai dengan yang dibutuhkan oleh pasien, terutama untuk pasien yang dalam keadaan kritis sehingga tidak terjadi ketidaksetimbangan cairan pada tubuh pasien yang dapat membahayakan bagi pasien yang sedang menjalani perawatan intensif atau yang sedang menjalani operasi (Royan, 2007). Cairan obat dimasukkan ke dalam tubuh pasien melalui injeksi *intravenous* untuk durasi waktu yang lama dengan *flow rate* disesuaikan dengan tingkat yang tepat sehingga diperlukan jarum suntik otomatis yang dapat diprogram. *Syringe pump* merupakan alat medis yang difungsikan untuk melakukan injeksi cairan obat secara lambat dan terus-menerus dengan tujuan terapeutik maupun diagnostik (Saidi *et al.*, 2010).

Sistem *syringe pump* dirancang dengan mekanisme pergerakan motor (Kobayashi, 2006). Pergerakan motor akan menyebabkan ulir maju sehingga mendorong *plunger* (pendorong suntikan) dan proses injeksi mulai terjadi. Secara keseluruhan, sistem *syringe pump* terdiri dari *plunger*, sebuah motor, mekanisme pompa, pengontrol mekanisme pompa, dan alarm (Wang, 2010). Mekanisme pompa menggunakan gaya yang mendorong *plunger* sehingga cairan obat pada selang terdorong menuju pembuluh darah pasien.

Masalah yang sering timbul saat penggunaan *syringe pump* adalah oklusi (penyumbatan) selama mekanisme pompa. Penggunaan *syringe pump* yang dipasang secara berkelanjutan dapat menyebabkan terjadinya oklusi yang menyebabkan cairan obat yang masuk ke dalam tubuh tidak mengalir secara konstan dan terbentuk tekanan besar pada *syringe* dan aliran cairan (Wang, 2010)

yang jika dibiarkan akan terjadi pembengkakan. Oklusi dipengaruhi oleh sifat darah pasien yaitu mudahnya terjadi koagulasi (penggumpalan), selang yang terjepit, dan adanya penggumpalan darah di jarum menuju pembuluh darah pasien. Meninjau dari hal tersebut, maka dirancang *syringe pump* dilengkapi dengan mekanisme alarm deteksi oklusi.

Peralatan medis termasuk *syringe pump* yang terdapat di rumah sakit merupakan produk impor. Oleh karena itu, penulis ingin menghasilkan *syringe pump* produk lokal yang harapannya dapat dikembangkan oleh produsen instrumentasi medis di Indonesia yaitu *syringe pump* dilengkapi dengan alarm sebagai indikasi adanya oklusi dan *nearly empty* volume obat pada sistem tersebut. Alarm *nearly empty* merupakan indikasi untuk volume obat yang mendekati habis pada alat suntik yang terpasang di *syringe pump*.

Pada penelitian ini akan dikembangkan *syringe pump* berbasis mikrokontroler ATmega8535 dilengkapi dengan detektor oklusi dan *nearly empty*. Setelah terwujudnya *syringe pump* ini diharapkan rumah sakit dapat memberikan pelayanan optimal kepada pasien dan terlepas dari ketergantungan terhadap produk impor sehingga dapat meningkatkan pasar lokal di Indonesia yang harapannya meningkatkan daya saing sumber daya manusia di Indonesia dengan produk impor.

1.2 Rumusan Masalah

Berdasarkan latar belakang permasalahan, dirumuskan masalah sebagai berikut:

1. Bagaimana hasil membangun *syringe pump* berbasis mikrokontroler ATmega8535 sebagai pengendali volume cairan obat dan *flow rate*?
2. Bagaimana hasil rangkaian detektor tekanan oklusi pada *syringe pump*?

1.3 Batasan Masalah

Untuk membatasi permasalahan sehingga tidak meluas, maka penulis perlu membatasi masalah sebagai berikut :

1. Ukuran *syringe* yang digunakan adalah jenis Terumo 50 ml dengan pemilihan volume injeksi dari 1 ml hingga 50 ml.
2. Jangkauan menu *flow rate* dari 1 ml/jam hingga 50 ml/jam.
3. Level untuk deteksi tekanan oklusi hanya untuk 100 mmHg.
4. Kemampuan untuk mendeteksi oklusi dengan menggunakan sensor *load cell*.
5. Mekanik yang digunakan adalah *syringe pump* merek Terumo TE-331.

1.4 Tujuan Penelitian

Tujuan penelitian ini meliputi tujuan umum dan tujuan khusus. Tujuan umum penelitian ini adalah merancang *syringe pump* berbasis mikrokontroler ATmega8535 yang dilengkapi detektor oklusi dan *nearly empty*. Adapun tujuan khusus adalah sebagai berikut :

1. Merancang *syringe pump* yang dilengkapi dengan menu volume obat dan *flow rate* untuk injeksi sehingga memudahkan perawat dalam mengontrol pasien.
2. Merancang *syringe pump* dengan rangkaian detektor oklusi.

1.5 Manfaat Penelitian

1. Rancang bangun *syringe pump* berbasis mikrokontroler ATmega8535 yang dikembangkan pada penelitian ini memudahkan pengontrolan dalam pemberian cairan obat kepada pasien.
2. Rangkaian detektor oklusi yang dibangun pada *syringe pump* sangat bermanfaat untuk mendeteksi terjadinya penyumbatan sehingga tidak membahayakan pasien.

BAB II

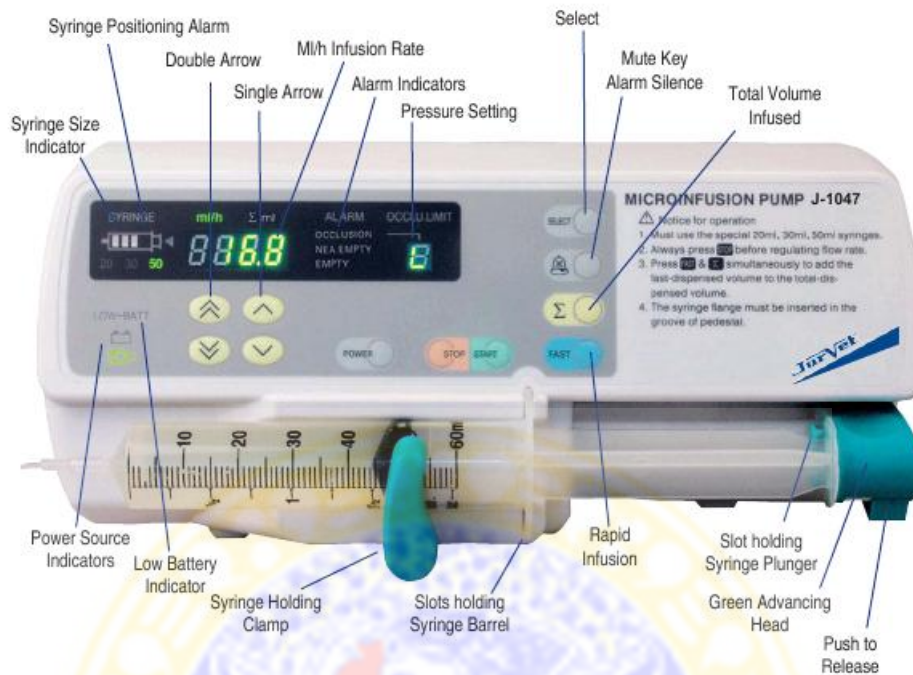
TINJAUAN PUSTAKA

2.1 *Syringe Pump*

2.1.1 Tinjauan Umum *Syringe Pump*

Penggunaan *syringe pump* untuk mengelola obat telah dipraktekkan selama bertahun-tahun. Alat *infusion pump* untuk pengiriman antibiotik secara berkesinambungan dilakukan pada tahun 1945 (Dickenson, 1983). Pada sistem penemuan ini melibatkan mekanisme jam alarm sederhana dimana rotasi kumparan jam memutar sekrup yang secara bertahap mendorong *plunger* suntikan. Desain lainnya diproduksi oleh Ruben dan Mohelsky pada tahun 1954 (Wright, 2003), dengan dipasang dua pegas baja untuk mendorong *plunger* jarum suntik dengan kecepatan diatur oleh jam alarm yang bekerja melalui sistem katrol.

Syringe pump difungsikan untuk memasukkan cairan obat ke dalam tubuh pasien. *Syringe pump* dapat mengawasi dan mengontrol dirinya sendiri yaitu mengontrol pemberian obat ke dalam *intravenous* pasien (Graham, 2005). Perawat hanya memberi *input* pada *syringe pump* berupa dosis volume obat (ml) dan *flow rate* (ml/jam). *Flow rate* merupakan banyaknya atau volume fluida yang mengalir diukur per satuan waktu. *Syringe pump* diaplikasikan bagi pasien yang membutuhkan perawatan khusus yaitu pemberian obat harus dilakukan secara terus-menerus dengan volume yang konstan. Pada Gambar 2.1 dijelaskan bagian-bagian *syringe pump* J-1047.



Gambar 2.1 Bagian-bagian *syringe pump* J-1047 (Jorgensen Laboratories, 2004)

2.1.2 Prinsip Dasar Penggunaan *Syringe Pump*

Pada *syringe pump* sistem operasinya sudah diprogram dengan sistem pergerakan motor, maka perawat hanya menentukan dosis volume obat (ml) dan *flow rate* (ml/jam) yang perlu diberikan kepada pasien dimana penentuan dosis ini sudah mengatur kecepatan motor untuk menekan *plunger syringe* (pendorong suntikan) guna memasukkan obat ke dalam tubuh pasien (Kobayashi, 2006). *Syringe pump* dapat mengontrol pemberian obat melalui dosis volume cairan obat dan *flow rate* penggunaan obat tersebut.

Pada dasarnya, alat-alat medis mengutamakan keselamatan pasien saat alat tersebut kontak dengan tubuh pasien. Oleh karena itu, untuk menunjang hal tersebut maka *syringe pump* dilengkapi dengan alarm sebagai indikasi adanya detektor oklusi dan *nearly empty*. Detektor oklusi diperlukan untuk mencegah

terjadinya pembengkakan pembuluh darah pasien akibat penyumbatan. Oklusi dipengaruhi oleh sifat darah pasien yaitu mudahnya terjadi koagulasi (penggumpalan), selang yang terjepit, dan adanya penggumpalan darah di jarum menuju pembuluh darah pasien. Pada *syringe pump* selain alarm detektor oklusi, juga terdapat alarm *nearly empty* yang merupakan indikasi untuk volume obat yang mendekati habis pada alat suntik yang terpasang di *syringe pump*.

2.1.3 Volume dan Flow Rate

Dosis volume obat dan jenis obat ditentukan oleh dokter berdasarkan penyakit yang dialami pasien. Perhitungan *flow rate* dipengaruhi oleh dosis volume obat dan berat badan pasien. Adapun perhitungan *flow rate* ditentukan oleh Persamaan (2.1).

$$Flow\ rate\ \left(\frac{ml}{jam}\right) = \frac{Dosis\ (\mu g/kgBB/menit) \times kgBB \times Volume\ pelarut\ (ml)}{Sediaan\ obat\ yang\ dipakai\ (mg) \times 1000} \times 60 \quad (2.1)$$

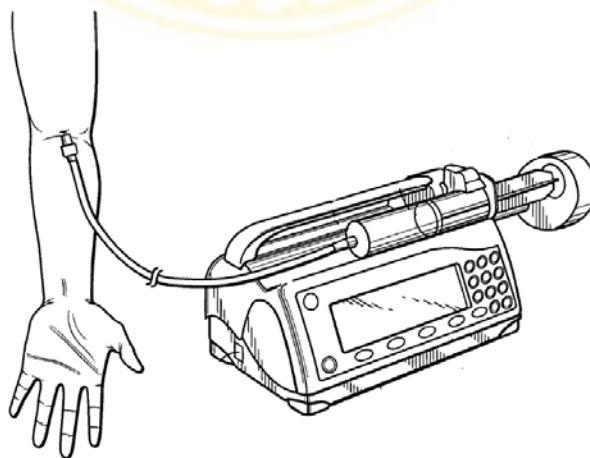
Pada contoh kasus kenaikan tekanan darah (hipertensi) pada pasien menjelang operasi, maka perlu diberikan konsumsi obat Perdipine secara terus-menerus dengan dosis tertentu. Adapun cara pemberian Perdipine adalah :

- Dosis yang dipakai :
 Untuk hipertensi emergensi : 0,5 – 6 μ g/kg BB/menit
 Terapi emergensi : 2- 10 μ g/kg BB/menit
- Volume cairan pelarut : 50 ml untuk *syringe pump*
- Sediaan Perdipine yang dipakai : 2 mg atau 10 mg
- Estimasi lama pemakaian/ampul : $(50 / \text{jumlah tetesan terpakai}) = n \text{ jam}$

Jenis obat yang digunakan akan berbeda pada setiap kasus. Setiap jenis obat berkaitan dengan dosis terpakai, volume cairan pelarut yang digunakan, sediaan massa obat yang terpakai, serta BB pasien. Dalam hal ini, faktor-faktor tersebut akan mempengaruhi perhitungan *flow rate* (ml/jam) pada *syringe pump*.

2.1.4 Sistem Oklusi Intravena

Secara umum, oklusi dapat diartikan sebagai penyumbatan. Aliran fluida antara pompa intravena dan sistem vaskuler pasien dapat mengalami hambatan yang disebabkan oleh beberapa faktor. Hambatan mekanis yang sepenuhnya menghalangi aliran fluida intravena disebut sebagai oklusi menyeluruh atau *hard occlusion* (Butterfield, 2010). Hal ini dapat disebabkan oleh selang katup *three way* yang tertutup, dapat pula disebabkan karena kateter intravena yang tertekuk. *Hard occlusion* biasanya terjadi secara tiba-tiba. Oklusi sebagian atau *soft occlusion* dapat terjadi akibat terbentuknya lapisan endapan (presipitat) atau emboli dalam kateter, tekanan dari kateter, atau penetrasi kateter ke dalam ruang jaringan pembuluh darah (Butterfield, 2010). *Syringe pump* dengan sistem aliran fluida pada *intravenous line* ditunjukkan pada Gambar 2.2.

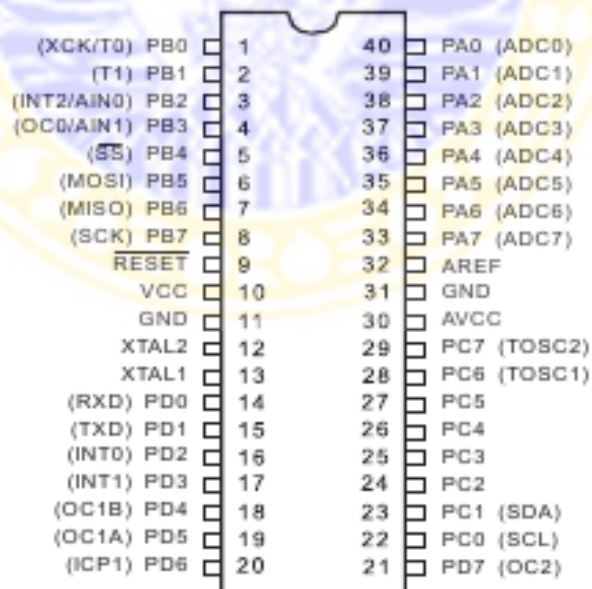


Gambar 2.2 *Syringe pump* dengan sistem aliran fluida pada *intravenous line*

Penentuan nilai tekanan pada saat terjadi oklusi ditentukan berdasarkan nilai hipertensi tekanan darah diastolik, yaitu 100 mmHg (Weir, 2005). Tekanan diastolik adalah tekanan darah pada saat jantung tidak sedang berkontraksi. Jika diukur dengan menggunakan tensimeter, maka tekanan diastolik merupakan bunyi detakan yang terakhir kali terdengar pada saat darah mulai mengalir. Oklusi awalnya ditandai dengan terjadinya sedikit penyumbatan yang prinsipnya sama dengan tekanan diastolik yang diukur jika menggunakan tensimeter.

2.2 Mikrokontroler ATmega8535

(Heryanto, 2008) Mikrokontroler ATmega8535 merupakan salah satu mikrokontroler buatan Atmel Corporation. Berikut adalah konfigurasi pin ATmega8535 yang dikemas dalam *Dual Inline Package* (DIP) 40 pin.



Gambar 2.3 Konfigurasi pin mikrokontroler ATmega8535

(Atmel Corporation Datasheet, 2006)

Adapun penjelasan dari konfigurasi pin mikrokontroler ATmega8535 pada Gambar 2.3 adalah sebagai berikut.

1. *Power*, VCC, dan GND.
2. *Port A* (*Port A*₀₋₇), merupakan pin IO dua arah dan berfungsi khusus sebagai pin masukan ADC.
3. *Port B* (*Port B*₀₋₇), merupakan pin IO dua arah dan berfungsi khusus sebagai pin *Timer/Counter*, komparator analog, dan SPI.
4. *Port C* (*Port C*₀₋₇), merupakan pin IO dua arah dan fungsi khusus.
5. *Port D* (*Port D*₀₋₇), merupakan pin IO dua arah dan fungsi khusus.
6. Reset adalah pin untuk mereset mikrokontroler.
7. XTAL1 dan XTAL2 merupakan pin untuk *external clock*.
8. AVCC adalah pin masukan untuk tegangan ADC.
9. AREF adalah pin masukan untuk tegangan referensi eksternal ADC.

2.3 LCD (*Liquid Crystal Display*)

LCD adalah sebuah *display* dot matrix yang difungsikan untuk menampilkan tulisan berupa angka atau huruf sesuai dengan yang diinginkan (sesuai dengan program yang digunakan untuk mengontrolnya). Pada tugas akhir ini penulis menggunakan LCD dot matrix karakter 2 x 16. LCD ini hanya memerlukan daya yang kecil, tegangan yang dibutuhkan juga rendah yaitu +5VDC. Adapun fungsi masing-masing pin pada LCD karakter 2x16 akan dijelaskan pada Tabel 2.1.

Tabel 2.1 Fungsi pin pada LCD

No.	Simbol	Level	Keterangan
1	Vss	-	Dihubungkan ke 0 V (<i>Ground</i>)
2	Vcc	-	Dihubungkan dengan tegangan <i>supply</i> +5V dengan toleransi $\pm 10\%$.
3	Vee	-	Digunakan untuk mengatur tingkat kontras LCD.
4	RS	H/L	Bernilai logika „0” untuk input instruksi dan bernilai logika „1” untuk input data.
5	R/W	H/L	Bernilai logika „0” untuk proses „write” dan bernilai logika „1” untuk proses „read”.
6	E	H	Merupakan sinyal <i>enable</i> . Sinyal ini akan aktif pada <i>failing edge</i> dari logika „1” ke logika „0”.
7	DB0	H/L	Pin data D0
8	DB1	H/L	Pin data D1
9	DB2	H/L	Pin data D2
10	DB3	H/L	Pin data D3
11	DB4	H/L	Pin data D4
12	DB5	H/L	Pin data D5
13	DB6	H/L	Pin data D6
14	DB7	H/L	Pin data D7
15	V+BL	-	<i>Back Light</i> pada LCD ini dihubungkan dengan tegangan sebesar 4 – 4,2 V dengan arus 50 – 200 mA
16	V-BL	-	<i>Back Light</i> pada LCD ini dihubungkan dengan <i>ground</i>

Cara kerja menjalankan LCD akan dijelaskan sebagai berikut.

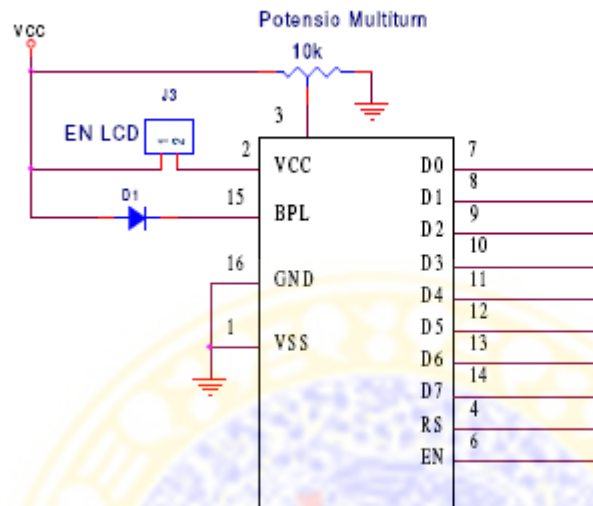
Langkah 1 : Inisialisasi LCD.

Langkah 2 : Arahkan pada alamat yang dikehendaki (lihat tabel alamat).

Langkah 3: Tuliskan data ke LCD, maka karakter akan tampil pada alamat tersebut.

Jalur EN dinamakan *Enable*. Jalur ini digunakan untuk memberitahu LCD sedang mengirimkan sebuah data. Untuk mengirimkan data ke LCD, maka melalui program EN harus dibuat logika *low* (0) dan diatur pada dua jalur control yang lain RS dan RW. Ketika dua jalur yang lain telah siap, mengatur EN dengan

logika (1) dan tunggu untuk sejumlah waktu tertentu (sesuai dengan *datasheet* dari LCD tersebut) dan berikutnya mengatur EN ke logika *low* (0) lagi.



Gambar 2.4 Rangkaian LCD

Jalur RS adalah jalur *Register Select*. Ketika RS berlogika *low* (0), data akan dianggap sebagai sebuah perintah atau instruksi khusus (seperti *clear screen*, posisi kursor, dan lainnya). Ketika RS berlogika *high* (1), data yang dikirim adalah data teks yang akan ditampilkan pada tampilan LCD. Sebagai contoh, untuk menampilkan huruf “T” pada layar LCD maka RS harus diatur pada logika *high* (1).

Jalur RW adalah jalur kontrol *Read/ Write*. Ketika RW berlogika *low* (0), maka informasi pada bus data akan dituliskan pada layar LCD. Ketika RW berlogika *high* (1), maka program akan melakukan pembacaan memori dari LCD. Sedangkan pada aplikasi umum pin RW selalu diberi logika *low* (0).

2.4 Motor Stepper

2.4.1 Tinjauan Umum Motor Stepper

Motor stepper adalah motor DC yang gerakannya bertahap (*step per step*) dan memiliki akurasi yang tinggi yang dikendalikan dengan pulsa-pulsa digital, tidak dengan memberikan tegangan secara terus-menerus. Deretan pulsa diterjemahkan menjadi putaran *shaft*, dimana setiap putaran membutuhkan jumlah pulsa yang ditentukan. Satu pulsa menghasilkan satu kenaikan putaran atau *step*, yang merupakan bagian dari satu putaran penuh. Oleh karena itu, perhitungan jumlah pulsa dapat diterapkan untuk mendapatkan jumlah putaran yang diinginkan. Perhitungan pulsa secara otomatis menunjukkan besarnya putaran yang telah dilakukan, tanpa memerlukan informasi balik (*feedback*).

(Heryanto, 2008) Motor stepper berputar secara bertahap (*step*) yang tetap dari satu posisi ke posisi berikutnya. Besar pergeseran *step* tergantung dari konstruksi motor. Besar *step* ini disebut dengan derajat *step* atau *step angle* (SA). Angka ini berkisar antara 1,8 – 2,5 – 3,75 – 7,5 – 15 dan 30°. Secara umum terdapat dua jenis motor stepper, yaitu bipolar dan unipolar. Pada rancang bangun alat *syringe pump*, penulis menggunakan motor stepper jenis bipolar. Adapun bentuk fisik dari motor stepper bipolar ditunjukkan pada Gambar 2.5.

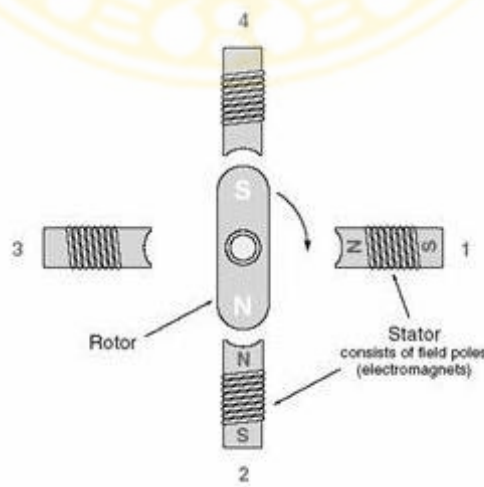


Gambar 2.5 Motor stepper bipolar

2.4.2 Prinsip Kerja Motor Stepper

Pada umumnya motor stepper hanya mempunyai kumparan pada statornya sedangkan pada bagian rotornya merupakan magnet permanen. Gambar 2.6 menunjukkan prinsip kerja dari motor stepper. Jika kumparan mendapat tegangan, dengan analogi mendapat logika '1', maka akan dibangkitkan kutub magnet yang berlawanan dengan kutub magnet tetap pada rotor. Hal ini menyebabkan posisi kutub magnet rotor akan ditarik mendekati lilitan yang menghasilkan kutub magnet berlawanan tersebut.

Jika langkah berikutnya lilitan yang bersebelahan diberi tegangan, sedangkan catu daya pada lilitan sebelumnya dilepas, maka magnet tetap pada rotor akan berpindah posisi menuju kutub magnet lilitan yang dihasilkan. Hal tersebut menyebabkan terjadinya gerakan 1 *step*. Jika langkah ini diulang secara terus-menerus dengan memberikan tegangan secara bergantian ke lilitan-lilitan yang bersebelahan, maka rotor pada motor stepper akan berputar (Pitowarno, 2006).



Gambar 2.6 Prinsip kerja motor stepper (Pitowarno, 2006)

Pada dasarnya motor stepper mempunyai tiga mode atau cara dalam pengoperasiannya, mode langkah tergantung dari perintah yang diberikan, berikut tiga mode langkah yang terdapat pada motor stepper :

1. Mode *Wave*
2. Mode *HalfStep*
3. Mode *Fullstep*

2.4.2.1 Mode *Wave*

Mode *wave* mempunyai torsi yang lebih rendah dibandingkan dengan mode *fullstep*. Mode *wave* lebih stabil pada kecepatan tinggi dan mengkonsumsi daya yang jauh lebih rendah dibandingkan mode *fullstep* dan mode *halfstep*. Langkah pada mode *wave* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Langkah pada mode *wave*

<i>Step</i>	<i>Coil 1</i>	<i>Coil 2</i>	<i>Coil 3</i>	<i>Coil 4</i>
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

2.4.2.2 Mode *Full Step*

Mode *full step* mempunyai resolusi yang sangat lambat akan tetapi mempunyai *self-lock* yang sangat kuat. Untuk berputar searah dan berlawanan arah dengan jarum jam cukup dengan membalikan kombinasi digital yang dimasukkan. Langkah pada mode *full step* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Langkah pada mode *full step*

<i>Step</i>	<i>Coil 1</i>	<i>Coil 2</i>	<i>Coil 3</i>	<i>Coil 4</i>
1	1	0	1	0
2	1	0	0	1
3	0	1	1	0
4	0	1	0	1

2.4.2.3 Mode *Half Step*

Mode *half step* mempunyai torsi yang paling baik dan paling stabil diantara tiga mode yang terdapat pada motor stepper. Mode *half step* merupakan gabungan dari mode *wave* dan *full step*. Langkah pada mode *half step* dapat dilihat pada Tabel 2.4.

Tabel 2.4 Langkah pada mode *half step*

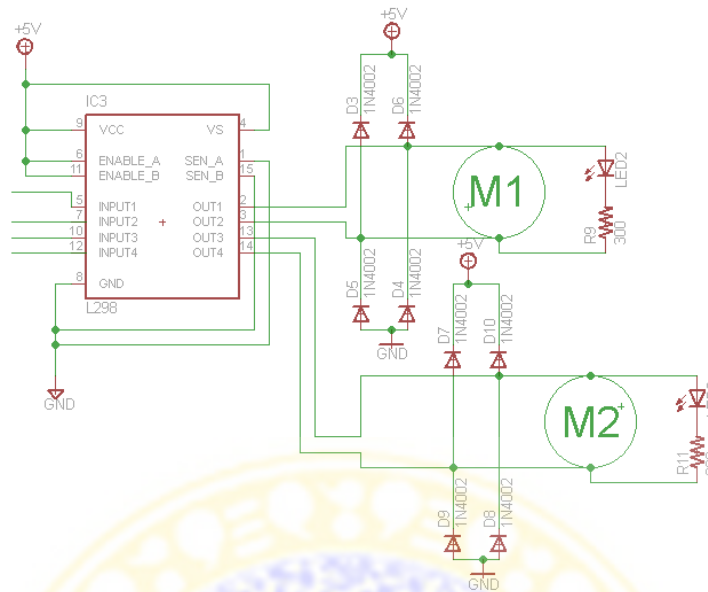
<i>Step</i>	<i>Coil 1</i>	<i>Coil 2</i>	<i>Coil 3</i>	<i>Coil 4</i>
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	1	0	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

2.4.3 Driver Motor L298

(Pitowarno, 2006) Motor stepper memiliki beberapa kebutuhan standar yang harus dipenuhi agar dapat bekerja dengan baik, di antaranya:

1. Tegangan/arus yang memadai untuk setiap lilitan untuk langkah per *step*.
2. Lama tegangan/arus harus diberikan untuk setiap langkah atau *step*. Hal ini diperlukan untuk memberikan waktu yang cukup bagi torsi untuk memindah posisi kutub ke posisi yang paling dekat dengan kutub stator (lilitan).

L298 adalah jenis IC *driver motor* yang dapat mengendalikan arah putaran dan kecepatan motor DC ataupun motor stepper. *Driver motor* L298 mampu mengeluarkan tegangan keluaran untuk motor DC dan motor stepper sebesar 50 V. Masing-masing dapat mengantarkan arus hingga 2 A. Gambar 2.7 adalah rangkaian dasar *driver motor* L298.

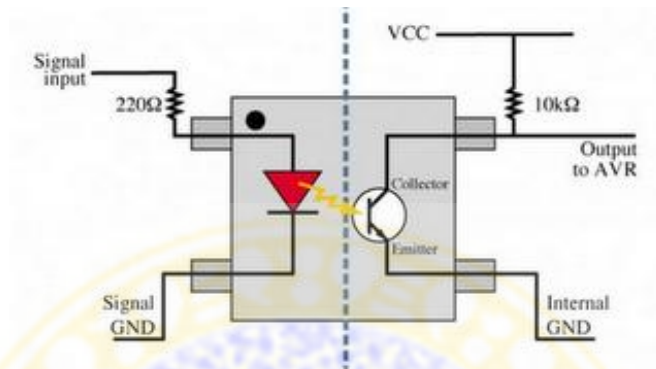
Gambar 2.7 Rangkaian dasar *driver motor* L298

2.5 Sensor Optocoupler

Optocoupler merupakan sebuah sensor yang terdiri dari dua bagian yaitu *transmitter* dan *receiver*. *Transmitter* (pengirim) merupakan bagian yg terhubung dengan rangkaian *input* atau rangkaian kontrol. Pada bagian ini terdapat sebuah LED inframerah (IR LED) yang berfungsi untuk mengirimkan sinyal pada *receiver*. Jika dibandingkan dengan menggunakan LED biasa, LED inframerah memiliki ketahanan yang lebih baik terhadap sinyal tampak.

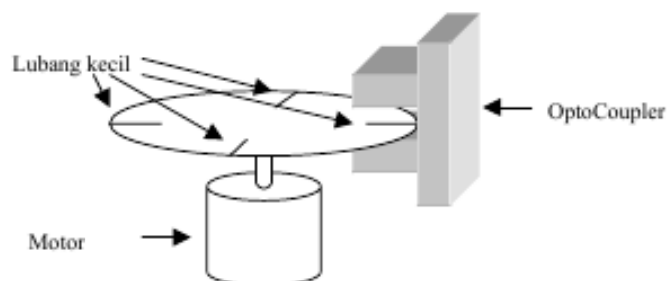
Receiver merupakan bagian yg terhubung dengan rangkaian *output* atau rangkaian beban, dan berisi komponen penerima cahaya yang dipancarkan oleh *transmitter*. Pada bagian *receiver* dibangun dengan dasar komponen fototransistor. Suatu sumber cahaya menghasilkan energi panas, begitu pula dengan spektrum inframerah. Spektrum inframerah mempunyai efek panas yang lebih besar dari cahaya tampak. Oleh karena itu, fototransistor lebih peka untuk menangkap

radiasi dari sinar inframerah. Adapun simbol optocoupler dapat dilihat pada Gambar 2.8.



Gambar 2.8 Sensor optocoupler

Penggunaan optocoupler dapat diaplikasikan untuk mendeteksi putaran motor stepper. Sensor putaran dengan optocoupler ini menghasilkan pulsa dari putaran motor stepper. Pulsa ini yang akan digunakan sebagai masukan pada mikrokontroler ATmega8535 untuk dihitung berapa pulsa yang diterima pada satuan waktu tertentu. Sensor putaran dengan optocoupler ini disusun menggunakan optocoupler jenis U. Gambar 2.9 menunjukkan pemasangan optocoupler untuk mendeteksi putaran pada motor stepper.

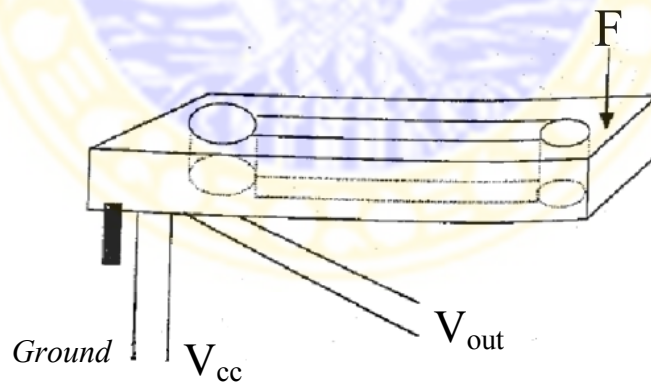


Gambar 2.9 Penempatan optocoupler pada piringan sensor.

2.6 Transduser *Load Cell*

2.6.1 Tinjauan Umum *Load Cell*

(Basuki, 2002) *Load cell* adalah transduser yang digunakan untuk mengkonversi gaya menjadi sinyal listrik. Konversi ini tidak langsung dan terjadi dalam dua tahap. Sebuah *load cell* terdiri dari empat *strain gauge* dalam konfigurasi jembatan Wheatstone. Melalui pengaturan mekanik, gaya menyebabkan deformasi *strain gage*. *Strain gage* mengukur deformasi (*strain*) sebagai sinyal listrik karena *strain* mengubah resistansi listrik efektif kawat. Keluaran tegangan listrik dari *load cell* dalam kisaran beberapa milivolt dan membutuhkan penguatan oleh sebuah penguat instrumentasi sebelum dapat digunakan. Pada *load cell* terdapat 4 kabel yang digunakan, 1 kabel untuk catu daya, 1 kabel untuk ground, dan 2 kabel sebagai keluaran tegangan dari *load cell*. Gambar 2.10 menunjukkan konfigurasi dari *load cell*.



Gambar 2.10 Konfigurasi *load cell* (Basuki, 2002)

2.6.2 Prinsip Kerja *Load Cell*

Berdasarkan penjelasan sebelumnya, *load cell* merupakan sebuah transduser gaya yang bekerja berdasarkan prinsip deformasi sebuah material akibat adanya tegangan mekanis yang bekerja. Untuk menentukan tegangan

mekanis didasarkan pada hasil penemuan Robert Hooke (Purwanto, 2010), bahwa hubungan antara tegangan mekanis dan deformasi yang diakibatkan disebut regangan. Regangan ini terjadi pada lapisan kulit dari material sehingga memungkinkan untuk diukur menggunakan *strain gauge*.

Strain gauge adalah transduser yang mengubah suatu pergeseran mekanis menjadi perubahan tahanan. Perubahan tahanan ini sebanding dengan regangan yang diberikan dan diukur dengan sebuah jembatan *wheatstone* yang dipakai secara khusus. Sensitivitas sebuah *strain gage* dijelaskan dengan suatu karakteristik yang disebut faktor *gage*, yang didefinisikan sebagai perubahan satuan tahanan dibagi perubahan satuan panjang.

$$G = \frac{\Delta R/R}{\Delta L/L} \dots\dots\dots(2.2)$$

Oleh karena $\Delta L/L$ adalah regangan ϵ , sehingga Persamaan (2.2) dapat ditulis menjadi Persamaan (2.3) :

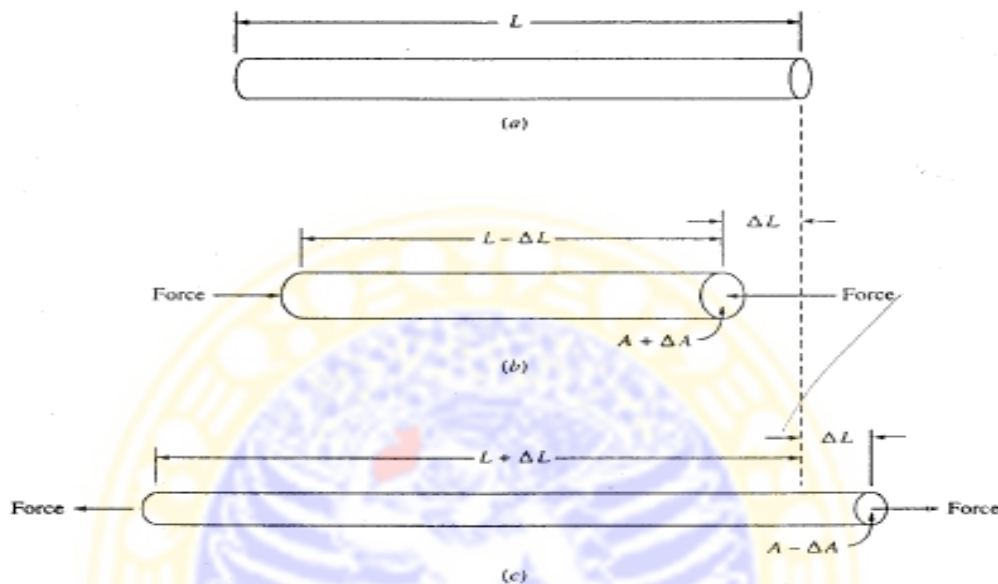
$$G = \frac{\Delta R/R}{\epsilon} \dots\dots\dots(2.3)$$

Perubahan ΔR pada *strain gauge* yang memiliki resistivitas bahan ρ dan panjangnya L dapat dihitung menggunakan persamaan tahanan pada Persamaan (2.4) :

$$R = \rho \frac{L}{A} \dots\dots\dots(2.4)$$

Tarikan (*tension*) terhadap *strain gauge* menyebabkan pertambahan panjang ΔL dan pengurangan luas penampang ΔA secara bersamaan, demikian juga sebaliknya. Penekanan (*compression*) terhadap *strain gauge* menyebabkan

pengurangan panjang ΔL dan pertambahan luas penampang ΔA secara bersamaan. Gaya *tension* dan *compression* yang diberikan pada *strain gauge* dapat dilihat pada Gambar 2.11.



Gambar 2.11 *Strain gauge* : (a) *Strain gauge* tidak diberikan gaya, (b) Gaya *tension* terhadap *strain gauge*, dan (c) Gaya *compression* terhadap *strain gauge*

Jika diberi gaya *compression*, resistansi *strain gauge* (R_s) berubah dan dapat dihitung dengan Persamaan (2.5) :

$$R_s = \rho \frac{(L+\Delta L)}{(A-\Delta A)} \dots\dots\dots(2.5)$$

Resistansi *strain gauge* akan berubah jika terdapat gaya yang bekerja padanya. Jika beban F diberikan maka regangan yang terjadi ditunjukkan pada Persamaan (2.6):

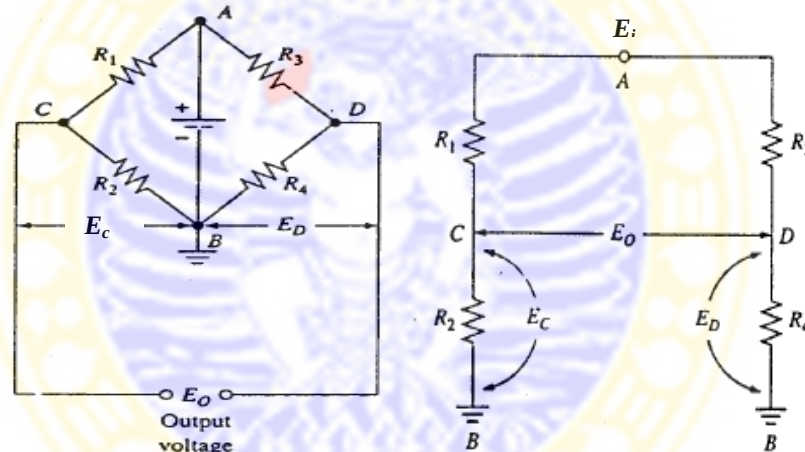
$$\frac{F}{A} = \varepsilon E$$

$$\varepsilon = \frac{F}{AE} \dots\dots\dots(2.6)$$

Pada Persamaan (2.6) variabel ε adalah regangan, F adalah gaya yang diberikan, A adalah luas penampang, dan E adalah modulus Young.

2.6.3 Konfigurasi *Wheatstone Bridge*

Perubahan regangan yang terjadi pada *strain gauge* sebanding dengan perubahan tahanan pada *strain gage*, sehingga dapat diukur dengan jembatan *wheatstone*. Pada Gambar 2.12 menunjukkan konfigurasi rangkaian *strain gauge full bridge*. Jika pada titik A dan B dihubungkan dengan tegangan masukan E_i , maka pada titik C dan D akan menimbulkan tegangan keluaran E_o .



Gambar 2.12 Rangkaian *strain gauge full bridge*

Berdasarkan Gambar 2.12 dapat dihitung masing-masing tegangan pada titik C dan D yang masing-masing diberikan pada Persamaan (2.7) dan (2.8):

$$E_C = \frac{R_2}{R_1 + R_2} \times E_i \dots\dots\dots(2.7)$$

$$E_D = \frac{R_4}{R_3 + R_4} \times E_i \dots\dots\dots(2.8)$$

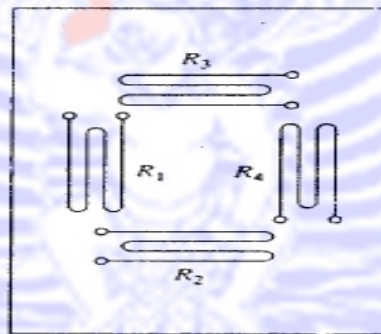
Tegangan keluaran E_o dapat dihitung dengan menggunakan Persamaan (2.7) dan (2.8) sehingga menjadi Persamaan (2.9).

$$E_o = E_C - E_D$$

$$E_o = \frac{E_i R_2}{R_1 + R_2} - \frac{E_i R_4}{R_3 + R_4}$$

$$E_o = E_i \left(\frac{R_2}{R_1 + R_2} - \frac{R_4}{R_3 + R_4} \right) \dots\dots\dots(2.9)$$

Asumsikan semua nilai tahanan pada *strain gauge* adalah sama ketika tidak ada gaya yang dikenakan ($R_1 = R_2 = R_3 = R_4 = R$). Saat gaya diaplikasikan pada transduser maka R_1 dan R_4 mengalami gaya *compression*, sedangkan R_2 dan R_3 mengalami gaya *tension*. Gambar 2.13 menunjukkan konfigurasi *strain gauge*.



Gambar 2.13 Konfigurasi *strain gauge*

Selisih antara nilai resistansi pada saat dikenakan gaya dengan nilai resistansi awal disimbolkan dengan ΔR . Saat R_1 dan R_4 mengalami gaya *compression*, maka nilai resistansinya berubah menjadi $R + \Delta R$. Pada R_2 dan R_3 yang mengalami gaya *tension*, maka nilai resistansinya berubah menjadi $R - \Delta R$. Berdasarkan Persamaan (2.9) dapat diperoleh tegangan keluaran *strain gauge* saat diaplikasikan gaya yang ditunjukkan pada Persamaan (2.10).

$$E_o = E_i \left[\frac{(R - \Delta R)}{(R + \Delta R) + (R - \Delta R)} - \frac{(R + \Delta R)}{(R - \Delta R) + (R + \Delta R)} \right]$$

$$E_o = E_i \left[\frac{(R-\Delta R)}{2R} - \frac{(R+\Delta R)}{2R} \right]$$

$$E_o = -E_i \left(\frac{\Delta R}{R} \right) \dots\dots\dots(2.10)$$

Dengan mensubstitusikan Persamaan (2.3) ke Persamaan (2.10) maka akan diperoleh hubungan antara tegangan keluaran *strain gauge* dengan regangan pada Persamaan (2.11).

$$E_o = -G \varepsilon E_i \dots\dots\dots(2.11)$$

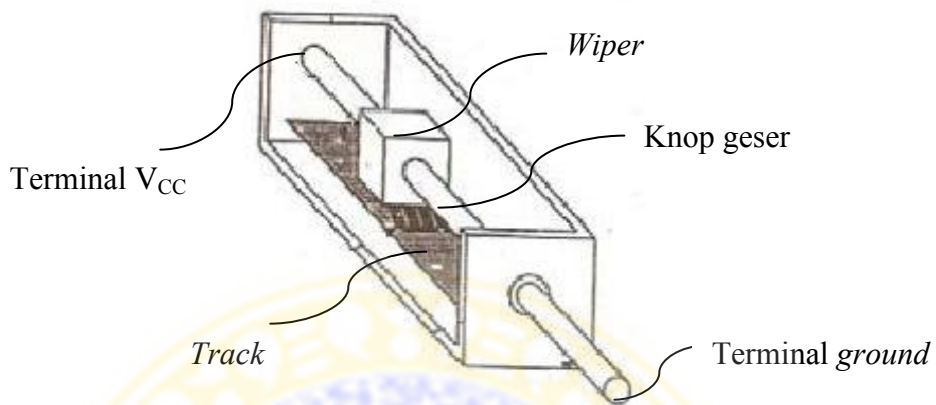
Tanda negatif pada Persamaan (2.11) menunjukkan R₁ dan R₄ mengalami gaya *compression*, sedangkan R₂ dan R₃ mengalami gaya *tension*. Berdasarkan Persamaan (2.11) dan Persamaan (2.6) dapat diperoleh hubungan antara besar gaya yang diberikan dengan tegangan keluaran yang dihasilkan *strain gauge* oleh Persamaan (2.12).

$$E_o = -G \left(\frac{F}{A E} \right) E_i \dots\dots\dots(2.12)$$

2.7 Sensor Potensio Geser

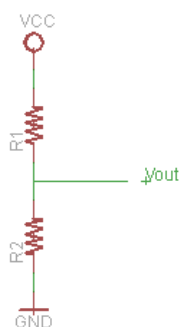
(Suryono, 2008) Potensio geser merupakan salah satu jenis resistor variabel yaitu resistor yang dapat diubah nilai tahanannya. Salah satu jenis potensiometer adalah potensiometer geser, yaitu potensiometer yang nilai tahanannya dapat diubah dengan cara menggeser *wiper* pada knop geser potensiometer tersebut. Potensio geser memiliki sebuah pita film, disebut sebagai jalur (*track*), yang terbuat dari karbon. Sedangkan knop gesernya terbuat dari bahan keramik yang bersifat konduktif. Ujung-ujung jalur terhubung ke dua buah

terminal potensiometer, yaitu V_{CC} dan *ground*. Bentuk fisik potensiometer ditunjukkan pada Gambar 2.14.



Gambar 2.14 Potensio geser (Suryono, 2008)

Terminal ketiga dari potensiometer disebut dengan *wiper*. *Wiper* adalah sebuah *strip* (lempengan kecil dan tipis) logam yang bersifat lentur, yang menempel dan menekan kuat pada jalur karbon untuk membentuk suatu hubungan listrik. *Wiper* dipasang pada sebuah knop geser, yang digunakan untuk memindahkan posisi *wiper* di sepanjang jalur karbon. Jika knop geser diubah secara translasi, maka akan menghasilkan perubahan resistansi. Prinsip rangkaian potensiometer sama dengan rangkaian pembagi tegangan yang ditunjukkan pada Gambar 2.15.



Gambar 2.15 Rangkaian pembagi tegangan

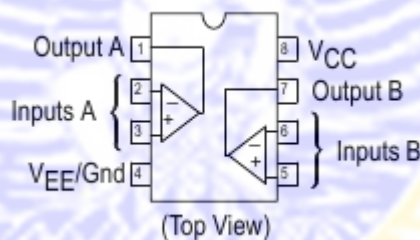
Berdasarkan ilustrasi rangkaian pembagi tegangan sederhana yang ditunjukkan pada Gambar 2.15 maka dapat dihitung tegangan keluaran dengan Persamaan (2.13).

$$V_{out} = \frac{R_2}{R_1 + R_2} \times V_{cc} \dots\dots\dots(2.13)$$

2.8 Komponen Pendukung

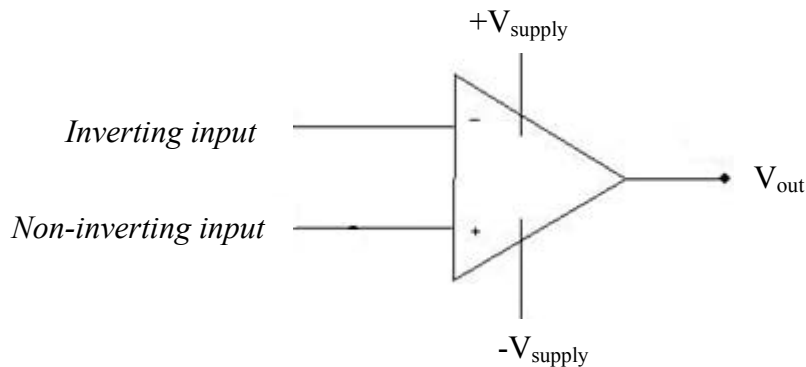
2.8.1 IC LM358

(Motorola Datasheet, 1996) LM358 merupakan *operational amplifier* yang memiliki 2 *output* dengan *input inverting* (-) dan *non-inverting* (+). Konfigurasi pin pada IC LM358 ditunjukkan pada Gambar 2.16.



Gambar 2.16 Konfigurasi pin pada LM358 (Lampiran 6)

Konfigurasi *open-loop* pada *op-amp* dapat difungsikan sebagai komparator. Jika kedua *input* pada *op-amp* pada kondisi *open-loop*, maka *op-amp* akan membandingkan kedua saluran *input* tersebut. Gambar 2.17 menunjukkan rangkaian komparator tegangan sederhana. Sebuah rangkaian komparator akan membandingkan nilai tegangan pada kedua masukannya. Jika masukan *inverting* (-) lebih besar dari masukan *non-inverting* (+), maka keluaran dari *op-amp* akan sama dengan $-V_{supply}$. Jika tegangan masukan *inverting* (-) lebih kecil dari masukan *non-inverting* (+), maka keluaran *op-amp* akan sama dengan $+V_{supply}$.



Gambar 2.17 Rangkaian komparator tegangan sederhana

2.8.2 Transistor PNP BC557

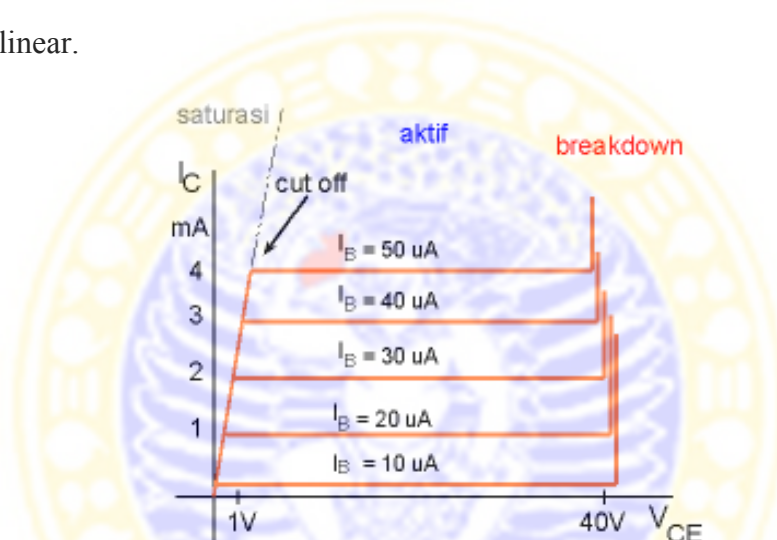
2.8.2.1 Prinsip Kerja Transistor PNP

(Sutrisno, 1986) Pada transistor dwikutub sambungan *p-n* antara emitor dan basis diberi *forward bias* sehingga arus mengalir dari emitor ke basis. *Bias* adalah tegangan dan arus *dc* harus diberikan agar transistor dapat bekerja. Oleh adanya *forward bias* antara emitor dan basis, pembawa muatan dari emitor akan tertarik masuk basis dan terus tersapu ke kolektor. Adanya arus setelah kolektor akan membuat kolektor mempunyai tegangan positif terhadap basis, sehingga sambungan *p-n* antara kolektor dan basis juga akan mendapat *forward bias*.

Hal ini akan menarik arus I_{CB} dari kolektor ke basis, berlawanan dengan arus dari emitor, yaitu arus I_{BC} . Kolektor harus berada pada tegangan jauh di bawah basis untuk menghindari terjadinya arus balik I_{CB} . Oleh karena itu, pada kaki kolektor dipasang tegangan *reverse bias* melalui catu daya $-V_{cc}$. Dengan demikian, fungsi dari terminal transistor antara lain emitor berasal dari kata '*emitter*' yang berarti pengeluar, basis berasal dari kata '*base*' yang berarti landasan, dan kolektor berasal dari kata '*collector*' yang berarti pengumpul.

2.8.2.2 Daerah Operasi Transistor

Sebuah transistor memiliki empat daerah operasi transistor, di antaranya daerah aktif, daerah *cut off*, daerah saturasi, daerah *breakdown*. Daerah kerja transistor yang normal adalah pada daerah aktif, dimana arus I_C konstan terhadap berapapun nilai V_{CE} . Berdasarkan kurva pada Gambar 2.18 diperlihatkan bahwa arus I_C hanya tergantung dari besar arus I_B . Daerah kerja ini disebut juga dengan daerah linear.



Gambar 2.18 Kurva I_C terhadap V_{CE} (Sutrisno, 1986)

Daerah saturasi adalah mulai dari $V_{CE} = 0$ Volt sampai kira-kira 0,7 Volt (transistor silikon), yaitu akibat dari efek dioda kolektor-base dimana tegangan V_{CE} belum mencukupi untuk dapat menyebabkan aliran elektron.

Jika tegangan V_{CC} dinaikkan perlahan-lahan hingga tegangan V_{CE} tertentu, arus I_C mulai konstan. Pada saat perubahan ini, daerah kerja transistor berada pada daerah *cut-off* yaitu dari kondisi saturasi (*off*) lalu menjadi daerah aktif (*on*). Perubahan ini dipakai pada sistem digital yang hanya mengenal angka biner 1 dan 0 yang tidak lain dapat dipresentasikan oleh status transistor *off* dan *on*.

Berdasarkan kurva kolektor, terlihat bahwa jika tegangan V_{CE} lebih dari 40V, arus I_C menanjak naik dengan cepat. Transistor pada daerah ini disebut berada pada daerah *breakdown*. Seharusnya transistor tidak boleh bekerja pada daerah ini, karena akan dapat merusak transistor.

2.8.3 Buzzer

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah energi listrik menjadi energi suara. Pada dasarnya prinsip kerja buzzer hampir sama dengan *loud speaker*. Buzzer juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet. Kumparan tersebut akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya.

Kumparan yang dipasang pada diafragma akan menyebabkan setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. Buzzer biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (alarm). Gambar 2.19 berikut merupakan bentuk fisik dari komponen buzzer.

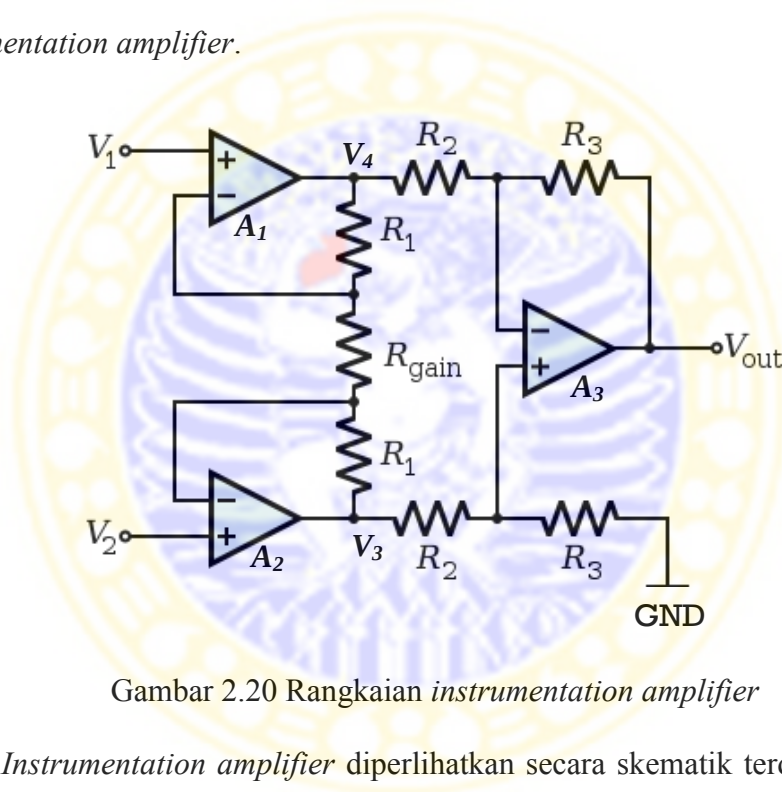


Gambar 2.19 Bentuk fisik buzzer

2.8.4 Instrumentation Amplifier AD620

2.8.4.1 Tinjauan Instrumentation Amplifier

(Carr, 1981) *Instrumentation amplifier* adalah jenis penguat diferensial yang memiliki karakteristik antara lain DC *offset* sangat rendah, *drift* rendah, *noise* rendah, *infinite open-loop voltage gain*, *common-mode rejection ratio* tinggi, dan impedansi *input* yang sangat tinggi. Gambar 2.20 merupakan rangkaian *instrumentation amplifier*.



Gambar 2.20 Rangkaian *instrumentation amplifier*

Instrumentation amplifier diperlihatkan secara skematik terdiri dari 3 *op-amp* yang dirancang sedemikian rupa sehingga terdapat dua *op-amp* (A_1 dan A_2) yang dihubungkan dengan konfigurasi *non-inverting follower* dan satu *op-amp* (A_3) dengan konfigurasi penguat diferensial. Asumsikan bahwa V_1 berdasarkan Gambar 2.20 diaplikasikan pada masukan *non-inverting* A_1 , V_2 diaplikasikan pada masukan *non-inverting* A_2 , V_3 adalah tegangan keluaran A_2 , dan V_4 adalah tegangan keluaran A_1 . Tegangan V_1 dan V_2 juga merupakan masukan *inverting*

dari A_1 dan A_2 . Tegangan keluaran dari V_3 dan V_4 dapat dihitung berdasarkan Persamaan (2.14) dan (2.15).

$$V_3 = V_2 \times \left(\frac{R_1}{R_{gain}} + 1 \right) - \left(V_1 \times \frac{R_1}{R_{gain}} \right) \dots\dots\dots(2.14)$$

$$V_4 = V_1 \times \left(\frac{R_1}{R_{gain}} + 1 \right) - \left(V_2 \times \frac{R_1}{R_{gain}} \right) \dots\dots\dots(2.15)$$

Berdasarkan Persamaan (2.14) dan (2.15) dapat diperoleh tegangan diferensialnya sehingga diperoleh Persamaan (2.16).

$$(V_3 - V_4) = (V_2 - V_1) \left(\frac{R_1}{R_{gain}} + 1 \right) + (V_2 - V_1) \left(\frac{R_1}{R_{gain}} \right)$$

$$(V_3 - V_4) = (V_2 - V_1) \left(\frac{R_1}{R_{gain}} + 1 + \frac{R_1}{R_{gain}} \right)$$

$$(V_3 - V_4) = (V_2 - V_1) \left(\frac{2R_1}{R_{gain}} + 1 \right) \dots\dots\dots(2.16)$$

Sehingga *gain* dari penguatan A_1 dan A_2 dihasilkan pada Persamaan (2.17).

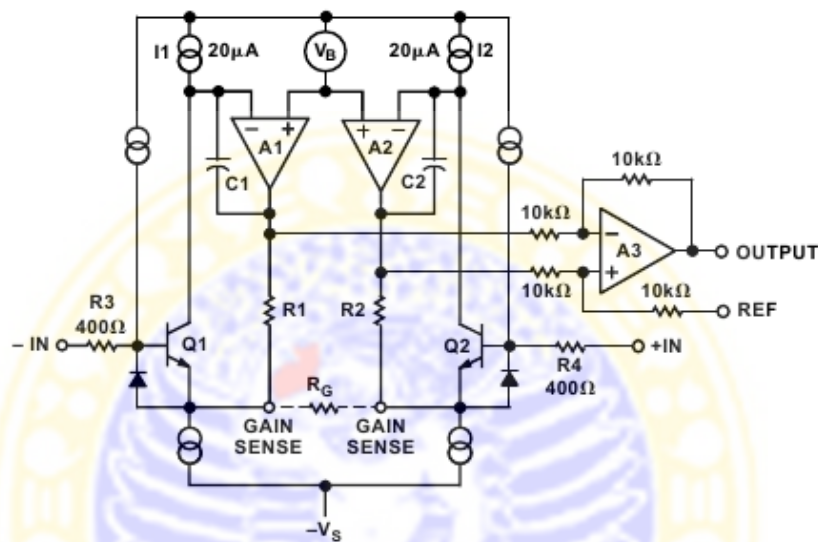
$$A_{12} = \frac{2R_1}{R_{gain}} + 1 \dots\dots\dots(2.17)$$

Gain dari rangkaian *instrumentation amplifier* secara keseluruhan diperoleh dengan menggabungkan *gain* pada A_3 dengan Persamaan (2.17) sehingga akan diperoleh Persamaan (2.18).

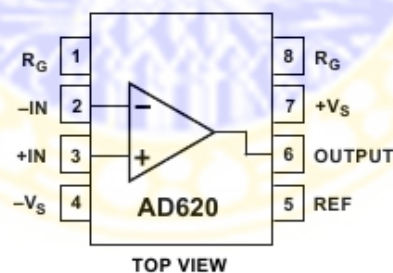
$$A_{out} = \left(\frac{2R_1}{R_{gain}} + 1 \right) \left(\frac{R_3}{R_2} \right) \dots\dots\dots(2.18)$$

2.8.4.2 Gain AD620

(Analog Devices Datasheet, 1999) AD620 merupakan IC yang berfungsi sebagai *instrumentation amplifier*. Gambar 2.21 berikut merupakan rangkaian *instrumentation amplifier* yang terdapat pada IC AD620.



Gambar 2.21 Rangkaian *instrumentation amplifier* pada IC AD620 (Lampiran 4)



Gambar 2.22 Konfigurasi pin pada AD620 (Lampiran 4)

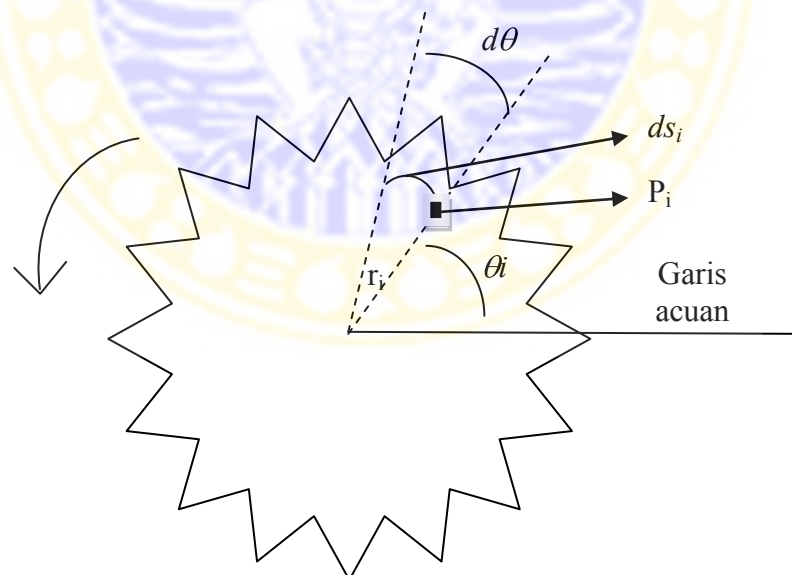
AD620 adalah penguat instrumentasi yang dibentuk tiga *operational amplifier*. Gain pada AD620 diatur dengan menghubungkan resistor eksternal tunggal, R_G :

$$G = 1 + \frac{49,4 \text{ k}\Omega}{R_G} \dots\dots\dots(2.19)$$

Nilai $49,4 \text{ k}\Omega$ dalam Persamaan (2.19) merupakan hasil jumlah dari dua resistor *internal feedback* (R_1 dan R_2 pada Gambar 2.21) dimana masing-masing bernilai $24,7 \text{ k}\Omega$. Kestabilan dan suhu dari resistor eksternal, R_G , juga akan mempengaruhi *gain*. Nilai resistor yang rendah diperlukan untuk memperoleh hasil *gain* yang tinggi.

2.9 Rumus Dasar Roda Gigi

(Tipler, 1991) Gambar 2.23 menunjukkan sebuah roda gigi (*gear*) yang bebas berotasi terhadap sumbunya, yaitu melalui pusatnya. Misalkan terdapat sebuah partikel bermassa m_i pada *gear*, maka dapat dinyatakan posisi partikel P_i dengan jaraknya r_i dari pusat gear dan sudut θ_i antara garis dari pusat ke partikel itu dan sebuah garis acuan yang tetap dalam ruang.



Gambar 2.23 Sebuah *gear* yang berotasi terhadap sumbu tetap melalui pusatnya

Selama selang waktu yang singkat dt , partikel bergerak sepanjang busur sebuah lingkaran sejauh ds_i yang diberikan oleh Persamaan (2.20).

$$ds_i = v_i dt \dots\dots\dots(2.20)$$

Dimana v_i adalah kelajuan partikel.

Selama selang waktu ini, garis radial ke partikel menempuh sudut $d\theta$ relatif terhadap garis tetap dalam ruang. Ukuran sudut $d\theta$ bila dinyatakan dalam radian adalah panjang busur ds_i dibagi dengan r_i dalam Persamaan (2.21).

$$d\theta = \frac{ds_i}{r_i} \dots\dots\dots(2.21)$$

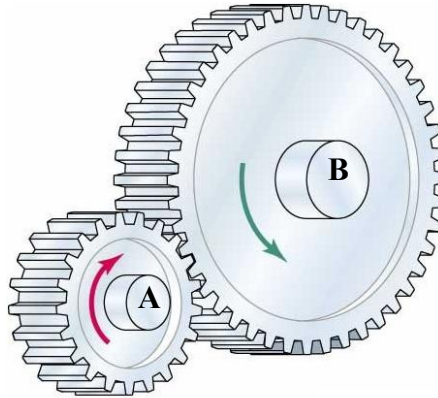
Walaupun jarak ds_i berubah dari partikel ke partikel, sudut $d\theta$ yang ditempuh dalam waktu tertentu adalah sama untuk semua partikel *gear*. Ini dinamakan dengan perpindahan angular *gear*. Laju perubahan sudut terhadap waktu, $d\theta/dt$, adalah sama untuk semua partikel *gear*. Ini dinamakan kecepatan angular ω *gear* dengan Persamaan (2.22).

$$\omega = \frac{d\theta}{dt} \dots\dots\dots(2.22)$$

Satuan kecepatan angular adalah radian per sekon. Besarnya kecepatan angular dinamakan kelajuan angular. Gerakan angular sebuah *gear* dapat juga dinyatakan dengan satuan putaran per menit, dengan 1 putaran = 2π rad. Kecepatan tangensial partikel di *gear* dengan kecepatan angular *gear* dapat dihubungkan dengan menggabungkan Persamaan (2.20) dan (2.21).

$$v_{it} = \frac{ds_i}{dt} = \frac{r_i d\theta}{dt} = r_i \omega \dots\dots\dots(2.23)$$

Pada Gambar 2.24 menunjukkan dua buah *gear* yang duhubungkan serta tidak seporos. Oleh karena dua buah *gear* tersebut tidak seporos, maka *gear* A dan *gear* B memiliki kecepatan angular yang berbeda.

Gambar 2.24 Dua *gear* yang saling bersinggungan

Jika terdapat dua buah *gear* yang bersinggungan, maka kecepatan tangensial partikel, v_{it} , *gear* A dan *gear* B adalah sama, sehingga diperoleh Persamaan (2.24) perbandingan kecepatan *gear* (Deutsman, 1975). Persamaan (2.24) berlaku untuk dua roda gigi jenis *spur gear*, tidak seporos, serta giginya saling bersinggungan.

$$R_v = \frac{\omega_B}{\omega_A} = \frac{Nt_A}{Nt_B} = \frac{d_A}{d_B} \dots\dots\dots(2.24)$$

Dimana:

ω_A, ω_B = putaran roda gigi (rpm)

Nt_A, Nt_B = jumlah gigi

d_A, d_B = diameter roda gigi

2.10 Pemrograman CodeVision AVR

2.10.1 Tinjauan CodeVision AVR

(Heryanto, 2008) CodeVisionAVR menyediakan sebuah editor yang didesain untuk menghasilkan program C secara otomatis untuk mikrokontroler AVR. Program C yang akan diimplementasikan menggunakan standar ANSI C

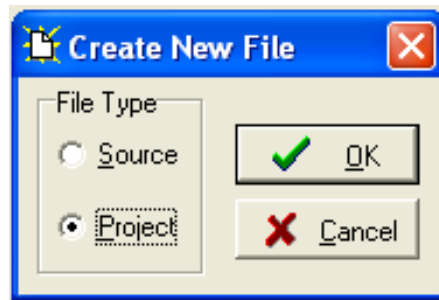
yang sesuai dengan arsitektur AVR. CodeVisionAVR adalah sebuah *compiler* C yang telah dilengkapi dengan fasilitas *Integrated Development Environment* (IDE) dan didesain agar dapat menghasilkan kode program secara otomatis untuk mikrokontroler Atmel AVR.

Integrated Development Environment (IDE) telah dilengkapi dengan fasilitas pemrograman chip melalui metode *In-System Programming* sehingga dapat secara otomatis mentransfer *file* program ke dalam *chip* mikrokontroler AVR setelah sukses dikompilasi. *Software In-System Programmer* didesain untuk bekerja ketika dihubungkan dengan *development board* STK500, STK600, AVRISP, AVRISP mkII, AVR Dragon, AVRProg (AVR910 *application note*), Atmel JTAGICE mkII, Kanda *System* STK200+STK300, Dontronics DT006, Vogel Elektronik VTEC-SIP, Futurlec JRAVR and MicroTronics ATCPU, dan Mega2000.

2.10.2 Bahasa Pemrograman CodeVision AVR

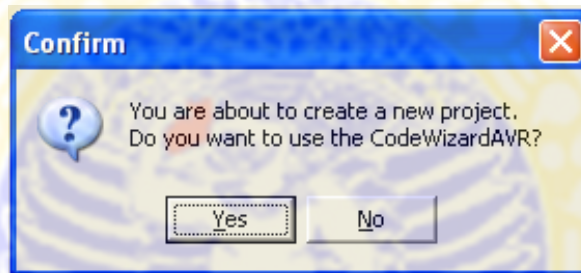
Program harus diterjemahkan dahulu ke dalam kode mesin agar suatu program dapat dimengerti oleh mikrokontroler. Adapun penerjemah yang digunakan bisa berupa kompilasi. Kompilasi adalah suatu jenis penerjemah yang menerjemahkan baris per baris instruksi program dalam bahasa C hingga menjadi program yang *executable* (dapat dieksekusi secara langsung).

Untuk memulai bekerja dengan CodeVision AVR pilih pada menu File -> New. Maka akan muncul kotak dialog pada Gambar 2.25.



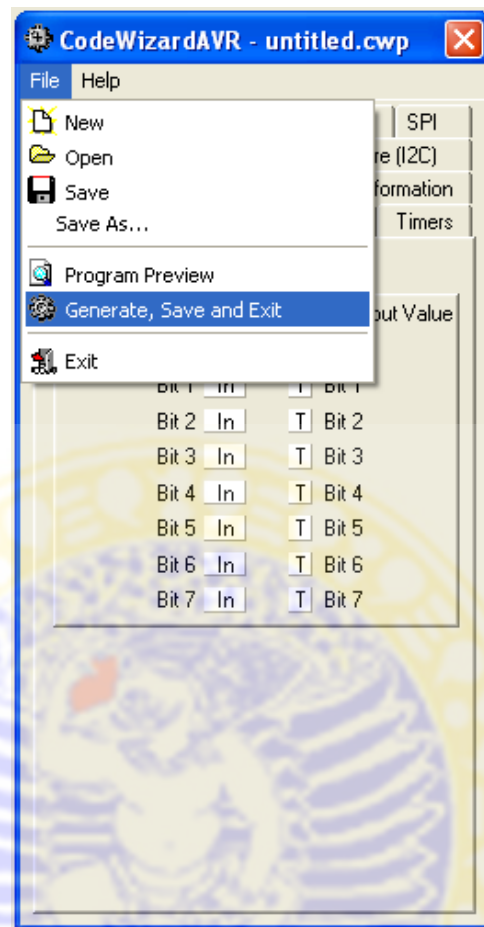
Gambar 2.25 Kotak dialog 1

Pilih Project kemudian tekan OK, maka akan muncul kotak dialog 2 pada Gambar 2.26.



Gambar 2.26 Kotak dialog 2

Pilih Yes untuk menggunakan CodeWizardAVR. CodeWizardAVR digunakan untuk membantu dalam men-generate program, terutama dalam konfigurasi chip mikrokontroler, baik itu konfigurasi Port, Timer, penggunaan fasilitas-fasilitas seperti LCD, interrupt, dan sebagainya. CodeWizardAVR ini sangat membantu programmer untuk setting chip sesuai keinginan. Untuk selanjutnya fasilitas-fasilitas lainnya dapat disetting sesuai kebutuhan dari pemrograman. Setelah selesai dengan CodeWizard AVR, selanjutnya pada menu *File*, pilih *Generate, Save and Exit* dan simpan pada direktori yang diinginkan, dapat dilihat pada Gambar 2.27.



Gambar 2.27 Code Wizard AVR dan Generate Program

BAB III

METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian

Kegiatan penelitian ini dilakukan pada bulan Desember 2011 sampai bulan Juni 2012 yang dilaksanakan di laboratorium Elektronika dan Robotika Departemen Fisika Fakultas Sains dan Teknologi Universitas Airlangga, serta laboratorium Mikrokomputer Teknik Elektromedis Surabaya.

3.2 Peralatan dan Bahan Penelitian

3.2.1 Peralatan Penelitian

Sebagai penunjang dalam melaksanakan pembuatan, pengukuran, pengamatan, maupun pengujian *syringe pump*, akan digunakan beberapa alat sebagai berikut :

1. Multimeter analog
2. Solder
3. Bor
4. Obeng
5. Tang
6. Downloader ISP untuk AVR ATmega8535
7. Penyedot timah
8. Gunting
9. Lem bakar

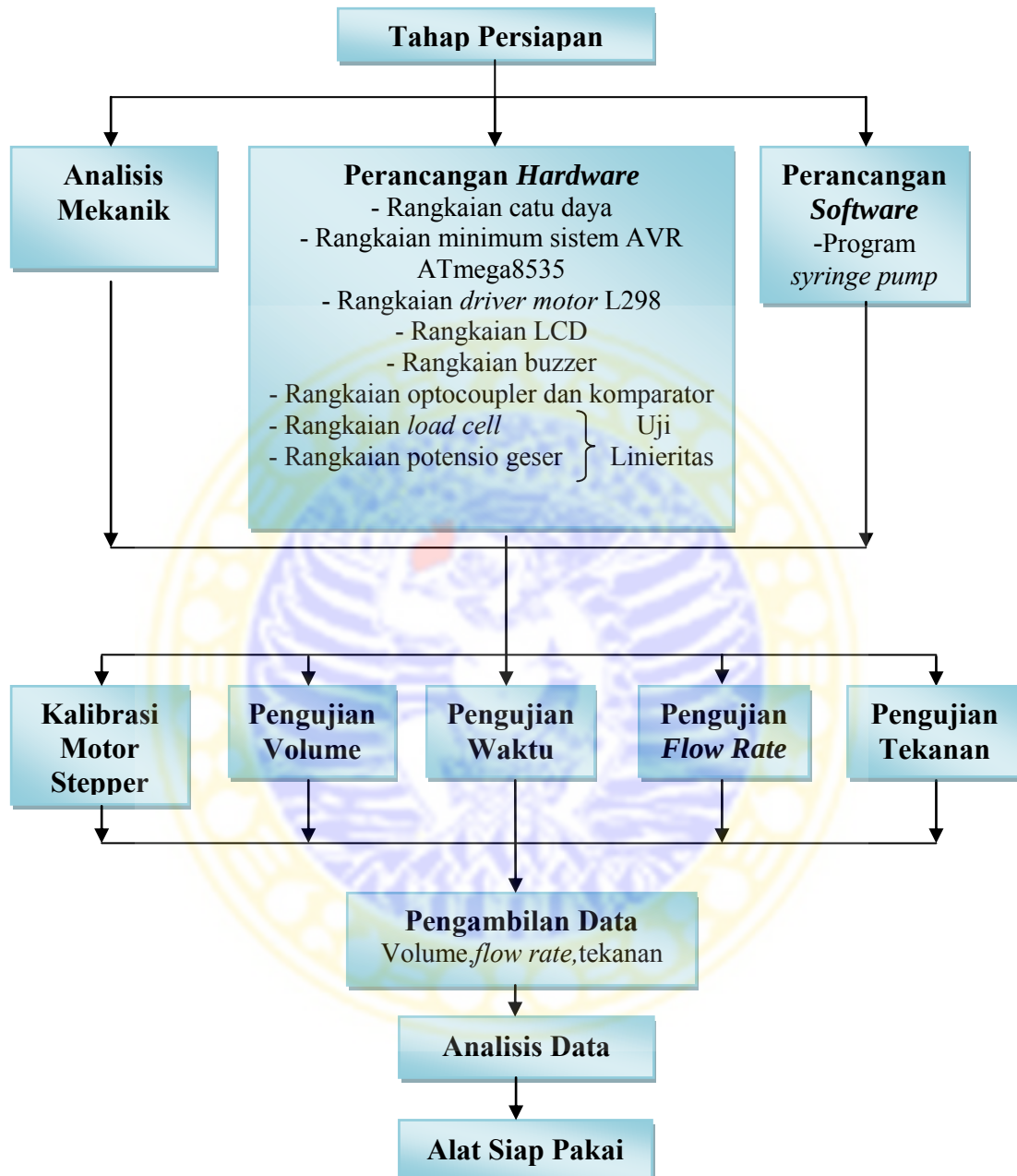
3.2.2 Bahan Penelitian

Adapun yang perlu diperhatikan dalam kegiatan ini di antaranya adalah karakteristik komponen elektronika, harga, dan faktor ada tidaknya komponen tersebut di pasaran. Berikut ini disampaikan data bahan-bahan yang diperlukan dalam pembuatan *syringe pump* :

1. LCD (*Liquid Crystal Display*) character 2x16
2. Motor stepper bipolar
3. Buzzer 5 V
4. Trafo
5. *Push button*
6. Sensor : *Load cell*, potensio geser, optocoupler.
7. Komponen aktif : IC mikrokontroler AVR ATmega8535, IC L298, IC LM358, IC AD620, IC regulator 7805, IC regulator 7905, transistor PNP BC557, dioda.
8. Komponen pasif : Resistor variabel, resistor, kapasitor.
9. Perangkat mekanik *syringe pump* Terumo TE-331

3.3 Prosedur Penelitian

Prosedur penelitian pada “Rancang Bangun *Syringe Pump* Berbasis Mikrokontroler ATmega8535 Dilengkapi Detektor Oklusi” akan dilaksanakan dalam beberapa tahap pelaksanaan. Diagram alir prosedur penelitian secara lengkap dapat dilihat pada Gambar 3.1.

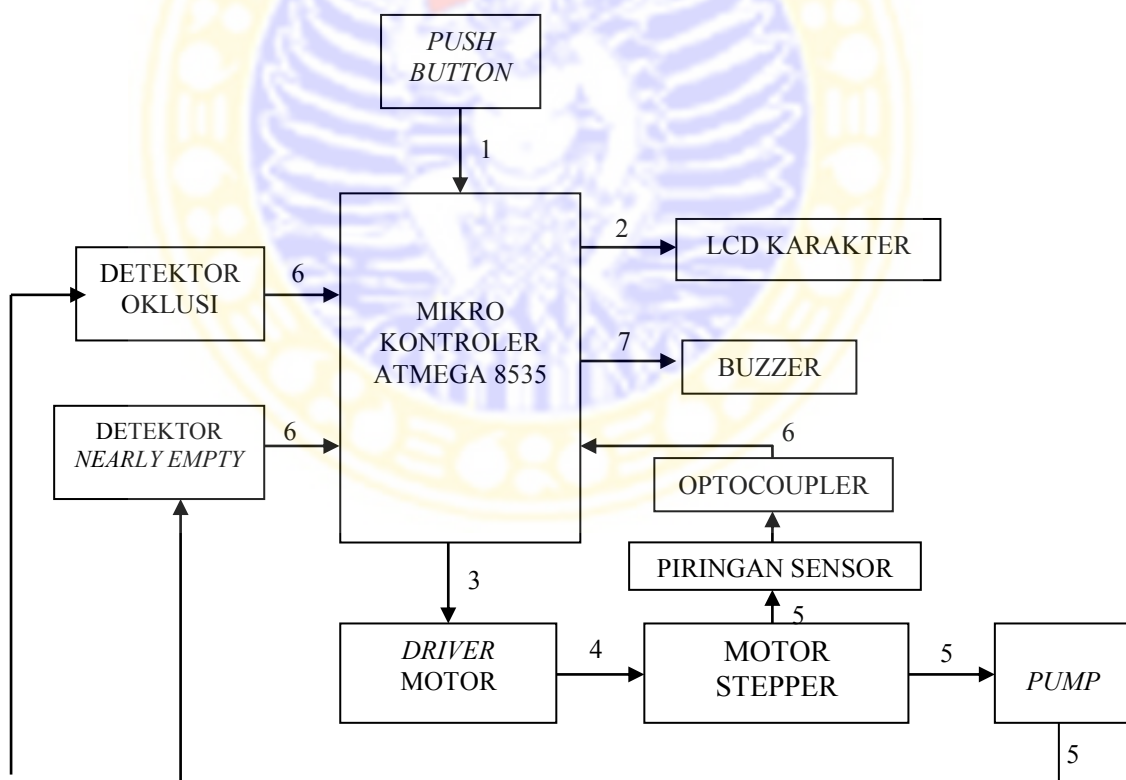


Gambar 3.1 Diagram alir prosedur penelitian

3.3.1 Tahap Persiapan

Pada tahap ini, proses yang dilakukan adalah pencarian informasi dengan studi pustaka pada beberapa jurnal ilmiah dan tugas akhir yang berhubungan dengan *syringe pump* serta melakukan observasi lapangan ke rumah sakit pada ruang ICU (*Intensive Care Unit*).

Pada tahap persiapan, semua rancangan rangkaian (*layout*) yang akan dirancang terlebih dahulu dibuat diagram bloknya sehingga pada tahap perancangan akan disesuaikan dengan diagram blok *syringe pump*. Adapun diagram blok *syringe pump* ditunjukkan pada Gambar 3.2.



Gambar 3.2 Diagram blok *syringe pump*

Pada diagram blok *syringe pump* menggambarkan sistem *syringe pump* secara keseluruhan dan hubungan antara rangkaian pendukung dengan rangkaian

minimum sistem mikrokontroler. Pemberian angka pada diagram blok *syringe pump* tersebut dimaksudkan untuk mengetahui prioritas kerja pada rangkaian penyusun *syringe pump*. Adapun penjelasan mengenai diagram blok *syringe pump* pada Gambar 3.2 yaitu :

1. *Push button* untuk menentukan jumlah obat yang akan diinjeksikan pada pasien dengan rentang volume dari 1 ml hingga 50 ml serta untuk menentukan *flow rate* proses penginjeksian obat dengan laju 1 ml/jam hingga 50 ml/jam.
2. LCD digunakan sebagai tampilan volume (ml) dan *flow rate* (ml/jam) penginjeksian obat.
3. Saat proses injeksi dimulai, *driver* motor berfungsi untuk menguatkan arus kendali dari mikrokontroler ke motor stepper.
4. Mikrokontroler akan menggerakkan motor stepper dengan *driver* motor sesuai dengan masukan volume dan *flow rate*.
5. Optocoupler mendeteksi putaran motor stepper untuk mengetahui jumlah cairan obat yang telah diinjeksi. Selama mekanisme pompa berjalan, sensor potensio geser mendeteksi sistem *nearly empty* dan sensor *strain gauge* mendeteksi adanya oklusi.
6. Jumlah *count* yang telah dideteksi oleh optocoupler dikirim ke mikrokontroler untuk dibandingkan dengan *input* volume. Selama proses ini, sensor potensio geser dan *load cell* mengirimkan data bit hasil ADC ke mikrokontroler untuk diproses adanya indikasi *error*.

7. Mikrokontroler mengolah data-data bit yang diterima dari sensor optocoupler, potensiometer geser, dan *load cell*. Buzzer akan berbunyi jika nilai *counter* yang dideteksi optocoupler sama dengan *input volume*, potensiometer geser mendeteksi *nearly empty*, atau *strain gauge* mendeteksi adanya oklusi yang nilainya telah ditentukan.

3.3.2 Tahap Perancangan

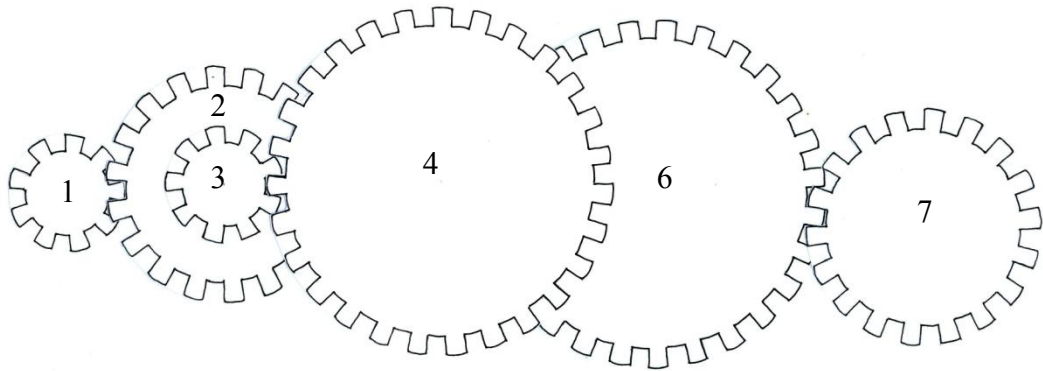
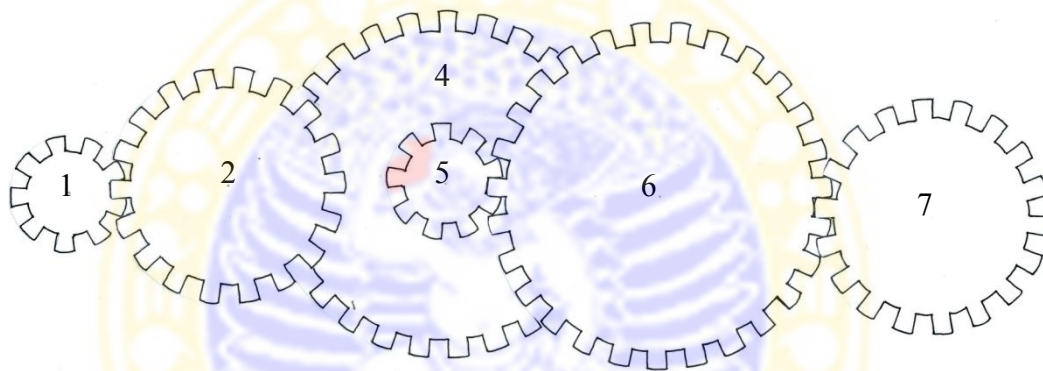
Tahap perancangan alat *syringe pump* terbagi dalam tiga bagian, yaitu analisis mekanik, perancangan perangkat keras (*hardware*), dan perancangan perangkat lunak (*software*).

3.3.2.1 Analisis Mekanik

Mekanik *syringe pump* yang akan digunakan adalah mekanik dari Terumo *syringe pump* tipe TE-331. Pada mekanik alat *syringe pump* digunakan jenis roda gigi lurus (*spur gear*) dengan alasan sebagai berikut :

1. Jenis ulir yang digunakan adalah *feed screw*, maka jenis roda gigi yang sesuai digunakan adalah roda gigi lurus.
2. Putaran yang relatif rendah sesuai dengan karakteristik roda gigi lurus.

Jumlah roda gigi lurus (*spur gear*) yang digunakan pada mekanik sejumlah 7 buah *gear* yang terdiri dari 4 buah *gear* tumpuk yang berporos sama dan 3 buah *gear* tunggal. Masing-masing *gear* yang digunakan memiliki diameter yang berbeda-beda. Adapun desain sistem *gear* pada alat *syringe pump* ditunjukkan pada Gambar 3.3 dan Gambar 3.4.

Gambar 3.3 Mekanik sistem *gear* tampak depanGambar 3.4 Mekanik sistem *gear* tampak belakang

Setiap *gear* saling bersinggungan antara satu dengan yang lain. *Gear* nomor 1 terhubung langsung dengan motor stepper (termasuk *gearbox motor*). *Gear* nomor 7 terhubung dengan *feed screw* yang akan menyebabkan pergeseran ulir sehingga menyebabkan alat suntik terdorong. Besar diameter setiap *gear* adalah sebagai berikut :

$$d_1 = 6,60 \text{ mm}$$

$$d_2 = 15,00 \text{ mm}$$

$$d_3 = 7,00 \text{ mm}$$

} seporos

$$d_4 = 25,50 \text{ mm}$$

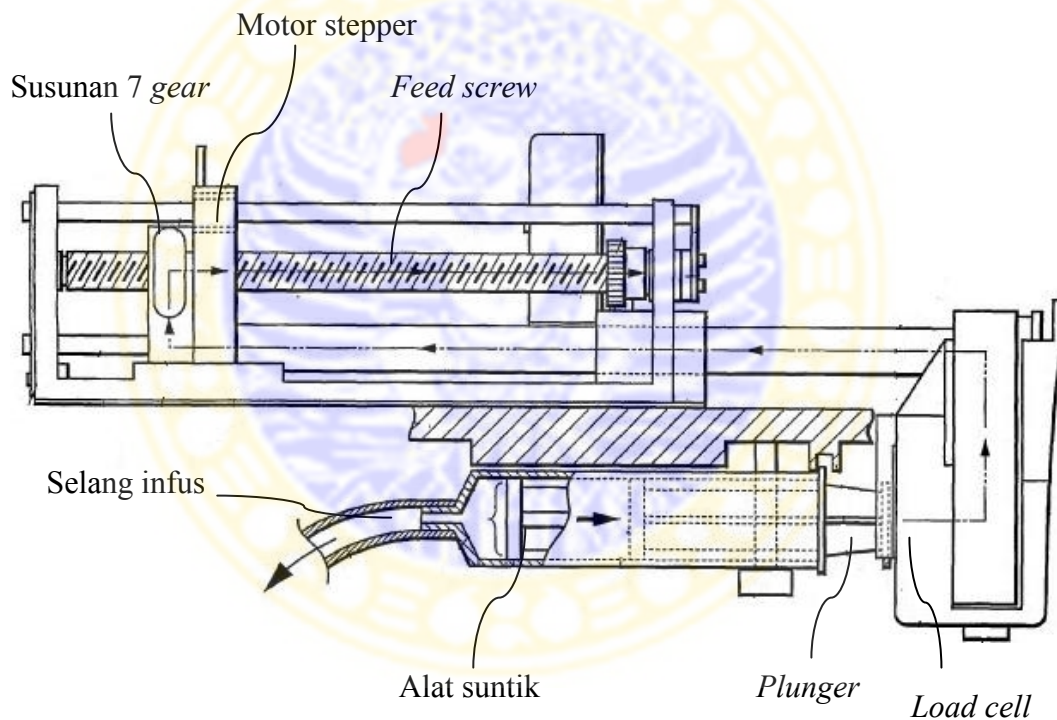
$$d_5 = 8,60 \text{ mm}$$

} seporos

$$d_6 = 25,00 \text{ mm}$$

$$d_7 = 13,00 \text{ mm}$$

Berdasarkan hubungan sistem mekanik *gear* tersebut maka dapat diketahui perbandingan antara kecepatan motor (rpm) dengan kecepatan *gear* nomor 7 yang secara langsung akan menyebabkan pergeseran ulir menggunakan Persamaan (2.24). Secara keseluruhan desain sistem mekanik alat *syringe pump* ditunjukkan pada Gambar 3.5 (sketsa menggunakan program *Corel Draw 12*).



Gambar 3.5 Rancangan sistem mekanik alat *syringe pump*

Berdasarkan hubungan *gear* pada perangkat mekanik alat *syringe pump* maka dapat diperoleh analisis perbandingan kecepatan motor (rpm) dengan kecepatan *gear* yang berhubungan langsung dengan *feed screw*. Persamaan perbandingan kecepatan ini diperlukan untuk kalibrasi motor stepper sehingga diperoleh jumlah *count* untuk menghasilkan nilai volume sebesar 1 ml.

Perhitungan kecepatan *gear* ke-2 :

$$\frac{\omega_2}{\omega_1} = \frac{d_1}{d_2}$$

$$\frac{\omega_2}{\omega_1} = \frac{6,6}{15} \implies \omega_2 = \frac{6,6 \omega_1}{15}$$

Gear ke-2 dan *gear* ke-3 seporos, maka :

$$\omega_3 = \omega_2 = \frac{6,6 \omega_1}{15}$$

Perhitungan kecepatan *gear* ke-4 :

$$\frac{\omega_4}{\omega_3} = \frac{d_3}{d_4}$$

$$\frac{\omega_4}{\frac{6,6 \omega_1}{15}} = \frac{7}{25,5} \implies \omega_4 = \frac{46,2 \omega_1}{382,5}$$

Gear ke-4 dan *gear* ke-5 seporos, maka :

$$\omega_5 = \omega_4 = \frac{46,2 \omega_1}{382,5}$$

Perhitungan kecepatan *gear* ke-6 :

$$\frac{\omega_6}{\omega_5} = \frac{d_5}{d_6}$$

$$\frac{\omega_6}{\frac{46,2 \omega_1}{382,5}} = \frac{8,6}{25} \implies \omega_6 = \frac{397,32 \omega_1}{9562,5}$$

Perhitungan kecepatan *gear* ke-7 :

$$\frac{\omega_7}{\omega_6} = \frac{d_6}{d_7}$$

$$\frac{\omega_7}{\frac{397,32 \omega_1}{9562,5}} = \frac{25}{13} \quad \Longrightarrow \quad \omega_7 = \frac{9933 \omega_1}{124312,5}$$

Perbandingan kecepatan motor stepper(rpm) dengan kecepatan *gear* ke-7 :

$$\frac{\omega_7}{\omega_1} = \frac{9933}{124312,5} = \frac{1}{12,5} \dots\dots\dots(3.1)$$

Volume 1 ml diperoleh dari perhitungan pergeseran *feed screw* yang sama dengan pergeseran alat suntik. Pergeseran *feed screw* untuk volume 1 ml ini diperoleh berdasarkan analisis perbandingan *gear* (roda gigi) yang digunakan dalam sistem mekanik alat *syringe pump*.

Alat suntik yang digunakan berbentuk tabung, maka perhitungan untuk nilai volume :

$$V = \pi r^2 \cdot Y \dots\dots\dots(3.2)$$

Dimana V adalah volume cairan obat dalam tabung, r adalah jari-jari tabung, dan Y adalah pergeseran *feed screw* atau sama dengan pergeseran alat suntik. Berdasarkan Persamaan (3.2) untuk memperoleh nilai pergeseran *feed screw* maka :

$$Y = \frac{V}{\pi r^2}$$

$$Y = \frac{1 \text{ ml}}{22/7 \times (1,501)^2 \text{ cm}}$$

$$Y = 0,1413 \text{ cm} = 1,413 \text{ mm}$$

Setiap penginjeksian sebesar 1 ml cairan obat maka *feed screw* mengalami pergeseran sebesar 1,413 mm. Nilai pergeseran tersebut sebanding dengan gerak *gear* nomor 7 yang berhubungan dengan *feed screw*. *Gear* nomor 7 dengan *feed*

screw memiliki hubungan seporos, sehingga kecepatan *gear* ke-7 sama dengan kecepatan *feed screw* ($\omega_7 = \omega_8$). Jarak satu *pitch* pada *feed screw* adalah sebesar 1mm. Satu *pitch* membutuhkan gerak rotasi satu putaran penuh pada *gear* yang seporos dengan *feed screw* yaitu *gear* ke-7. Oleh karena 1 ml membutuhkan gerak *pitch* sebesar 1,413 mm, maka *gear* ke-7 harus berputar sebanyak 1,413 putaran atau rotasi.

Berdasarkan perbandingan kecepatan motor stepper (rpm) dengan kecepatan *gear* ke-7 pada Persamaan (3.1), maka dapat dihitung banyaknya rotasi yang diperlukan motor stepper agar *gear* ke-7 berotasi 1,413 kali.

$$\frac{\omega_7}{\omega_1} = \frac{1}{12,5}$$

$$\omega_1 \left(\frac{\text{rotasi}}{\text{menit}} \right) = 12,5 \omega_7 \left(\frac{\text{rotasi}}{\text{menit}} \right)$$

$$n_1 \text{ rotasi} = 12,5 (1,413 \text{ rotasi})$$

$$n_1 \text{ rotasi} = 17,6625 \text{ rotasi} \dots\dots\dots(3.3)$$

Rotasi yang diperlukan motor stepper agar *gear* ke-7 berotasi 1,413 kali adalah sebesar 17,6625 rotasi. Adapun spesifikasi besar pergeseran derajat motor stepper yang digunakan adalah $7,5^\circ/\text{step}$, sehingga untuk menghasilkan satu rotasi penuh memerlukan $360^\circ/7,5^\circ = 48 \text{ step/rotasi}$. Berdasarkan Persamaan (3.3) dapat diperoleh banyaknya *step* yang diperlukan untuk memperoleh volume 1 ml.

$$\text{Banyaknya step motor} = 17,6625 \text{ rotasi} \times 48 \text{ step/rotasi}$$

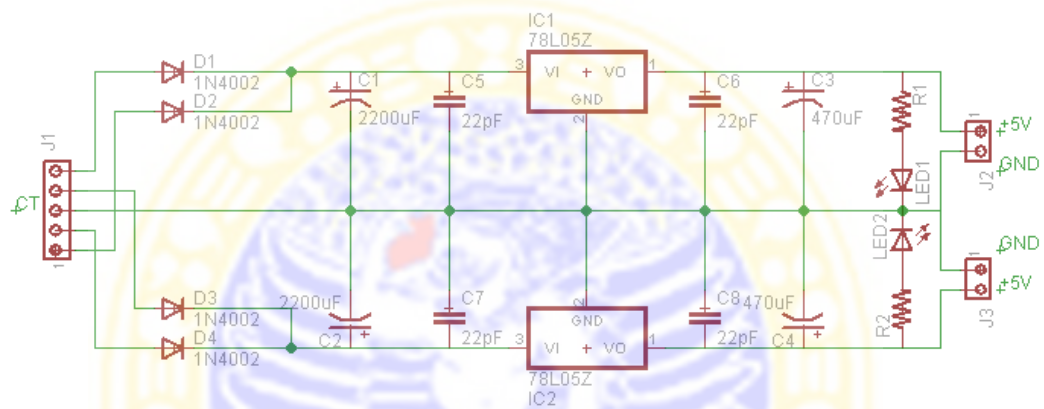
$$\text{Banyaknya step motor} = 848 \text{ step}$$

Dengan demikian diperlukan gerak motor stepper sebanyak 848 *step* untuk menghasilkan volume sebesar 1 ml.

3.3.2.2 Perancangan Perangkat Keras (*Hardware*)

3.3.2.2.1 Rangkaian Catu Daya

Catu daya yang digunakan pada alat *syringe pump* ini adalah 5V. Rangkaian catu daya terdiri dari trafo 12V/2A, regulator 7805 untuk menstabilkan tegangan menjadi 5V, diode 1N4002, kapasitor, resistor, dan LED. Skema rangkaian catu daya dapat dilihat pada Gambar 3.6.



Gambar 3.6 Rangkaian catu daya

3.3.2.2.2 Rangkaian Minimum Sistem AVR ATmega8535

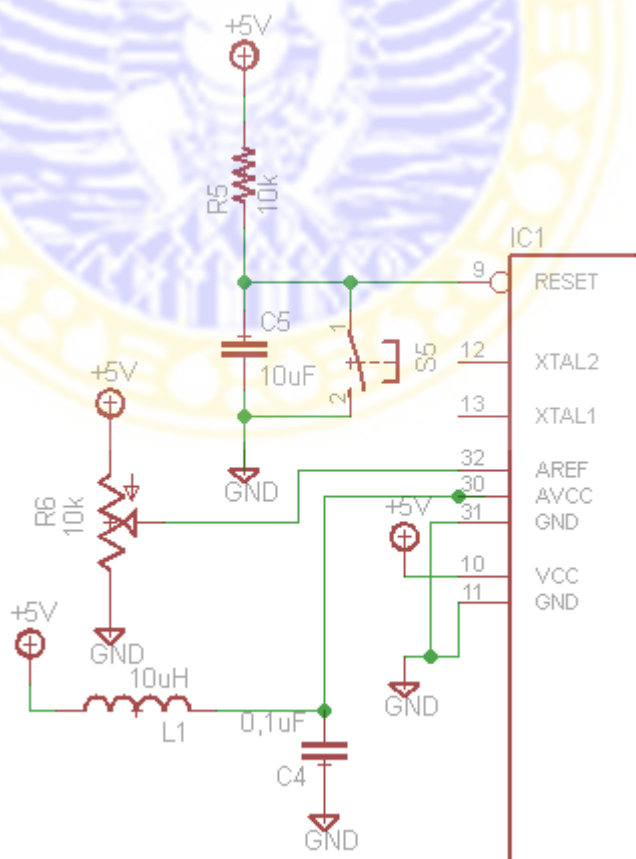
Fungsi mikrokontroler adalah sebagai otak dari suatu alat sehingga mampu menjalankan proses yang telah diprogram. Pada rangkaian mikrokontroler membutuhkan rangkaian RESET yang berfungsi untuk membuat mikrokontroler memulai kembali pembacaan program. Hal tersebut dibutuhkan pada saat mikrokontroler mengalami gangguan dalam mengeksekusi program.

Pada rangkaian minimum sistem ini, semua *port I/O* digunakan untuk mengontrol rangkaian pendukung pada *syringe pump*. Berikut koneksi pin pada mikrokontroler dengan rangkaian lainnya.

1. Pin A.0 difungsikan sebagai ADC untuk rangkaian potensiometer geser

2. Pin A.1 difungsikan sebagai ADC untuk rangkaian *strain gauge*
3. Pin A.2 dihubungkan ke rangkaian buzzer
4. Pin PB.1 dihubungkan ke rangkaian optocoupler
5. Pin PB.2, pin PB.3, pin PD.6, dan pin PD.7 dihubungkan ke rangkaian *driver motor L298*
6. Port C dihubungkan ke rangkaian LCD
7. Pin PD.0, pin PD.1, pin PD.3, pin PD.4, dan pin PD.5 dihubungkan ke *push button*

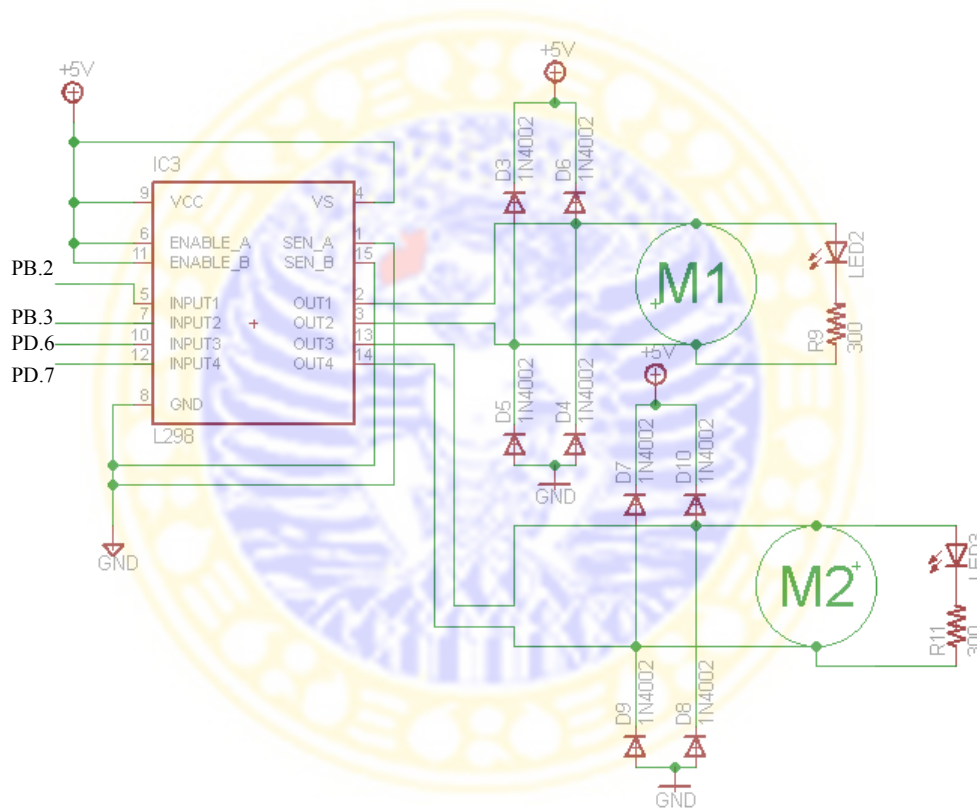
Gambar 3.7 merupakan rangkaian minimum sistem AVR ATmega8535 dari skema rangkaian keseluruhan dari alat *syringe pump*.



Gambar 3.7 Rangkaian minimum sistem AVR ATmega8535

3.3.2.2.3 Rangkaian *Driver Motor* L298

Pin VCC dan GND pada *driver motor* L298 dihubungkan dengan VCC dan *ground* pada *power supply* yang mengalirkan tegangan masukan sebesar 5V. Pin *current sensing* pada *driver motor* L298 dihubungkan ke *ground*. Pin *enable* pada *driver motor* dihubungkan ke tegangan 5V untuk menjalankan motor. Skema rangkaian pada *driver motor* L298 dapat dilihat pada Gambar 3.8.

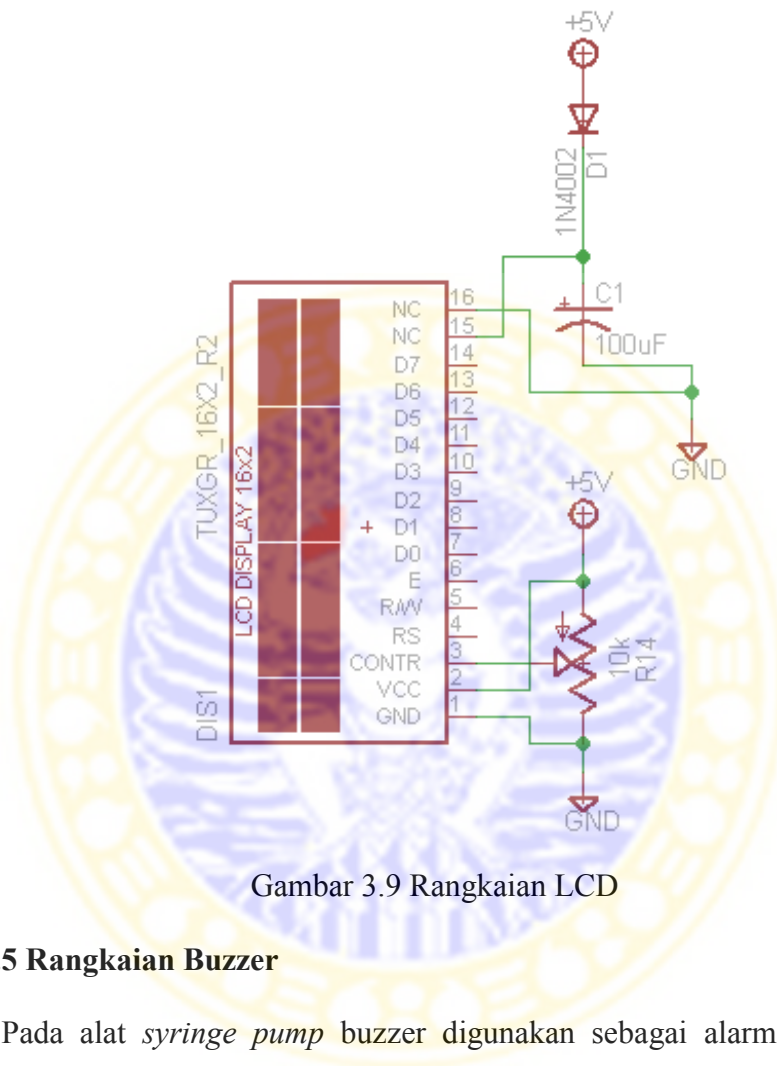


Gambar 3.8 Rangkaian *driver motor* L298

3.3.2.2.4 Rangkaian LCD

LCD difungsikan untuk menampilkan tulisan berupa angka atau huruf. Pada rancang bangun alat *syringe pump* ini LCD digunakan untuk menampilkan jumlah volume cairan obat dan *flow rate*, serta dapat menampilkan sisa volume

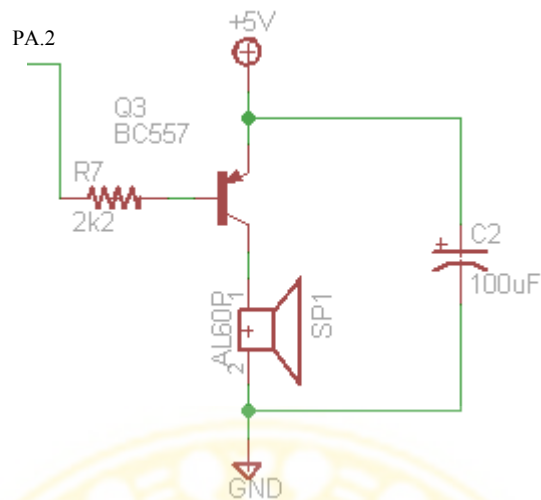
pada suntikan ketika proses injeksi sedang berlangsung. Skema rangkaian LCD dapat dilihat pada Gambar 3.9.



Gambar 3.9 Rangkaian LCD

3.3.2.2.5 Rangkaian Buzzer

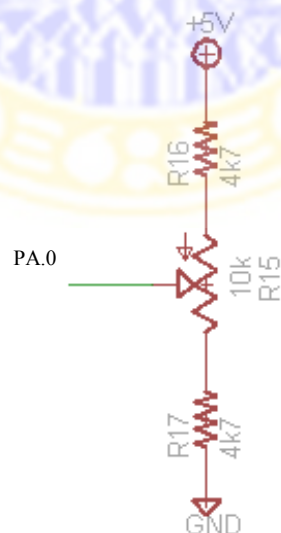
Pada alat *syringe pump* buzzer digunakan sebagai alarm penanda tiga keadaan. Pertama untuk mendeteksi *nearly empty* yaitu ketika volume cairan obat dalam suntikan mendekati habis, kedua untuk mendeteksi terjadinya oklusi, ketiga untuk mengindikasikan bahwa volume cairan obat telah habis diinjeksikan. Rangkaian buzzer pada penelitian ini menggunakan transistor BC557 yang berfungsi sebagai saklar. Skema rangkaian buzzer ditunjukkan pada Gambar 3.10.



Gambar 3.10 Rangkaian buzzer

3.3.2.2.6 Rangkaian Potensio Geser

Indikasi untuk mendeteksi *nearly empty* digunakan sensor potensio geser. Nilai hambatan pada potensio geser akan berubah seiring dengan gerak translasi dari *plunger* alat suntik. Skema rangkaian potensio geser ditunjukkan pada Gambar 3.11.

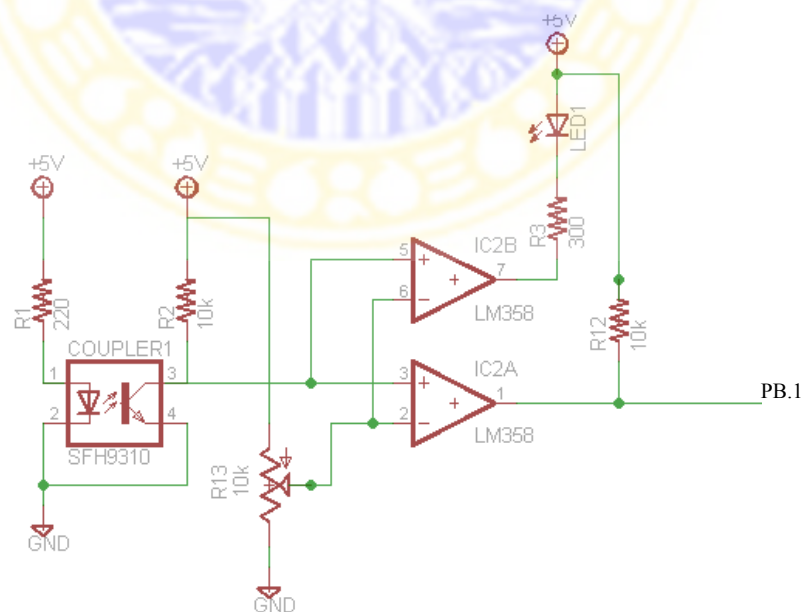


Gambar 3.11 Rangkaian potensio geser

3.3.2.2.7 Rangkaian Optocoupler dan Komparator

Optocoupler terdiri dari bagian *transmitter* dan *receiver*. Prinsip kerja optocoupler adalah ketika ada benda yang berada di antara celah sensornya, maka cahaya dari bagian *transmitter* tidak dapat diterima oleh bagian *receiver*, sehingga akan menghasilkan tegangan keluaran yang nilainya mendekati VCC. Begitu pula sebaliknya, jika tidak ada benda di antara celah sensor, maka akan menghasilkan tegangan keluaran yang nilainya mendekati 0 V.

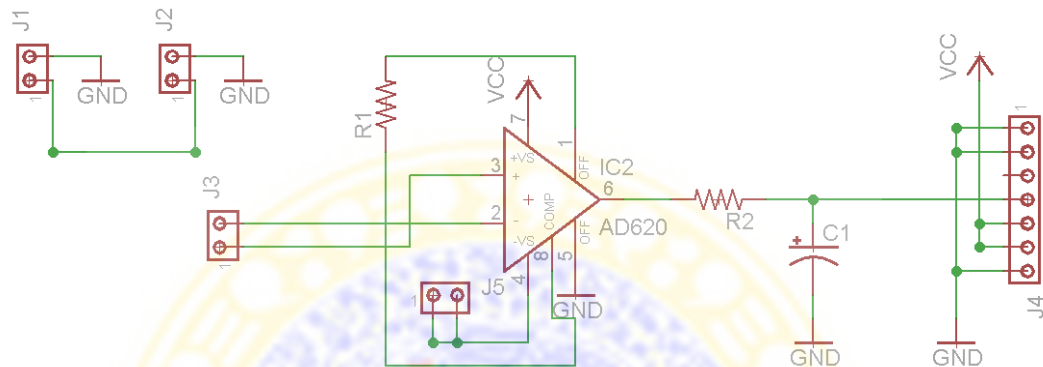
Tegangan keluaran dari optocoupler pada saat celah sensor terhalang benda atau tidak seringkali tidak tetap. Oleh karena itu, pada *output* rangkaian optocoupler perlu dihubungkan dengan rangkaian komparator LM358 sehingga tegangan keluarannya hanya dua nilai yang dapat dipastikan dalam kondisi terhalang atau tidak terhalang benda. Skema rangkaian optocoupler dan komparator dapat dilihat pada Gambar 3.12.



Gambar 3.12 Rangkaian optocoupler dan komparator

3.3.2.2.8 Rangkaian *Load Cell*

Indikasi untuk mendeteksi oklusi digunakan sensor *load cell*. Rangkaian *load cell* pada penelitian ini dilengkapi dengan *instrumentation amplifier*. Skema rangkaian *load cell* ditunjukkan pada Gambar 3.13.



Gambar 3.13 Rangkaian *load cell*

Keluaran tegangan dari *load cell* memiliki orde kecil yaitu kisaran milivolt sehingga diperlukan penguatan dengan menggunakan rangkaian *instrumentation amplifier* yang konfigurasinya telah terdapat pada IC AD620. Hasil penguatan tegangan dari rangkaian *instrumentation amplifier* ditentukan berdasarkan besarnya nilai resistor R_1 yang dihubungkan pada pin 1 dan 8 (pin R_G , *Resistor Gain*) dari IC AD620.

Rangkaian *load cell* ini dilengkapi dengan rangkaian *low pass filter* untuk menghilangkan *noise* yang akan timbul dari *load cell* yang digunakan. Konfigurasi rangkaian *low pass filter* berdasarkan Gambar 3.13 yaitu menggunakan resistor R_2 dan kapasitor C_1 . Frekuensi *cutoff* yang dihasilkan dari rangkaian *low pass filter* ditunjukkan oleh Persamaan (3.4).

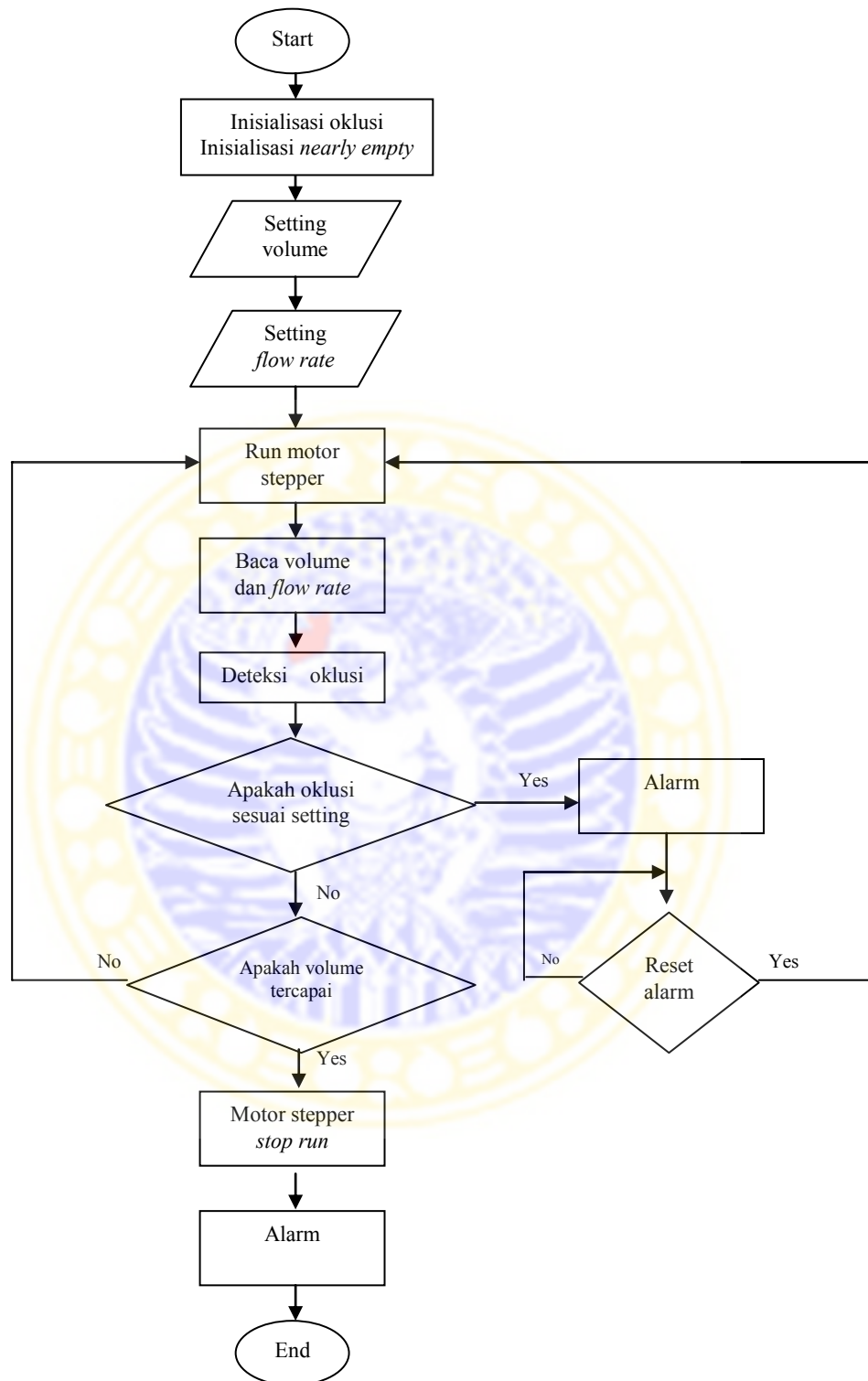
$$f_c = \frac{1}{2 \pi R C} \dots\dots\dots(3.4)$$

3.3.2.3 Perancangan Perangkat Lunak (*Software*)

Pada perancangan *software* untuk mikrokontroler AVR ATmega8535 menggunakan *software compiler* Code Vision AVR. Perancangan *software* meliputi inisialisasi volume dan *flow rate*, proses pengaturan motor stepper, pendeteksian oklusi, pendeteksian volume, dan sistem alarm.

Syringe pump dihidupkan dengan menekan tombol ON, kemudian memasukkan data jumlah volume obat yang akan diinjeksikan kepada pasien, lalu memasukkan *setting flow rate*. Jumlah volume obat antara 1 ml sampai 50 ml. *Flow rate* penginjeksian obat antara 1 ml/jam sampai 50 ml/jam. Motor stepper lalu berjalan sesuai dengan *input* data volume dan *flow rate*. Selama motor stepper berjalan, sensor *strain gauge* mendeteksi adanya oklusi. Jika oklusi mencapai nilai yang ditentukan maka motor stepper akan berhenti, lalu mikrokontroler akan membunyikan alarm. Reset alarm dilakukan sehingga motor stepper kembali bekerja.

Jika tidak terjadi oklusi atau oklusi belum mencapai nilai yang ditentukan, maka *counter* pada mikrokontroler akan mendeteksi volume cairan obat yang tersisa. Jika volume cairan obat belum mencapai nilai *nearly empty* atau belum habis, maka motor stepper akan terus berjalan. Namun, jika volume cairan obat telah mencapai nilai *nearly empty* atau volume cairan obat telah habis, maka motor stepper akan berhenti berjalan dan alarm akan berbunyi. Adapun *flowchart* pada rancangan *software syringe pump* ditunjukkan pada Gambar 3.14.



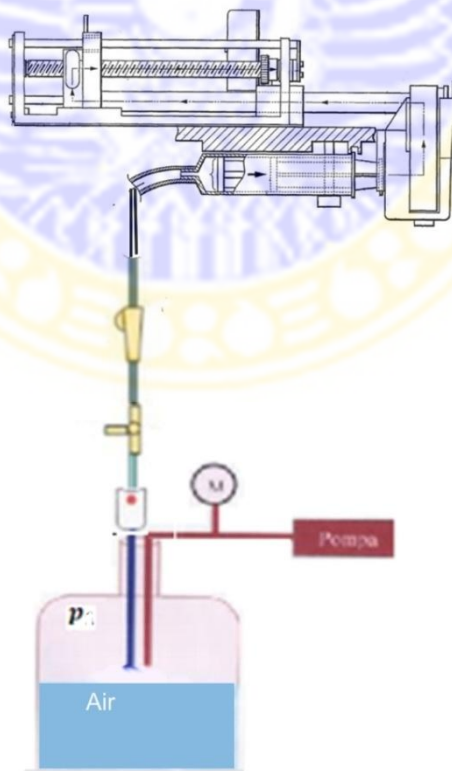
Gambar 3.14 *Flowchart* perangkat lunak (*software*) syringe pump

3.3.3 Tahap Pengujian

Pada tahap ini dilakukan pengujian mekanik, *hardware*, dan *software*. Pada pengujian *hardware syringe pump* dilakukan pada setiap rangkaian pendukung *syringe pump*. Setelah *hardware*, dan *software* selesai dikerjakan, maka dilakukan uji kinerja alat antara lain kalibrasi motor stepper, pengujian volume, pengujian *flow rate*, pengujian waktu, dan pengujian tekanan.

1. Kalibrasi motor stepper dilakukan untuk mengetahui jumlah *count* atau cacahan yang dicacah oleh mikrokontroler dalam menghasilkan injeksi volume sebesar 1 ml. Jumlah *count* ini didapatkan dari putaran piringan optocoupler. Alat yang diperlukan untuk pengujian ini yaitu *stopwatch* dan gelas ukur 10 ml. *Stopwatch* yang digunakan memiliki skala terkecil 0,01 sekon, sedangkan gelas ukur yang digunakan memiliki skala terkecil 0,2 ml. Volume 1 ml diperoleh berdasarkan analisis perbandingan *gear* (roda gigi) yang digunakan dalam sistem mekanik alat *syringe pump*. Nilai kecepatan rpm motor dibandingkan dengan kecepatan rpm *gear* yang bersinggungan sehingga akan diketahui jumlah *count* yang diperlukan untuk menghasilkan 1 ml.
2. Pengujian volume adalah pengujian yang dilakukan untuk mengetahui ketepatan penginjeksian volume cairan obat. Pengujian ini diperlukan gelas ukur 10 ml dengan skala terkecil 0,2 ml.
3. Pengujian waktu adalah pengujian yang dilakukan untuk mengetahui ketepatan waktu dalam penginjeksian cairan obat. Alat yang diperlukan untuk pengujian ini yaitu *stopwatch* dengan skala terkecil 0,01 sekon.

4. Pengujian *flow rate* bertujuan untuk mengetahui ketepatan laju alat *syringe pump* dalam menginjeksikan cairan obat. Nilai *flow rate* (ml/jam) diperoleh dari hasil perhingan volume (ml) dengan waktu (jam). Alat yang diperlukan untuk pengujian ini yaitu *stopwatch* dengan skala terkecil 0,01 sekon dan gelas ukur 10 ml dengan skala terkecil 0,2 ml.
5. Pengujian tekanan bertujuan untuk mengetahui tegangan keluaran dari rangkaian *load cell* terhadap tekanan yang dikondisikan pada nilai-nilai tertentu. Alat yang diperlukan untuk pengujian ini yaitu tensimeter dengan skala terkecil 2 mmHg dan tabung pemodelan pembuluh darah intravena. Pengujian tekanan dengan tabung pemodelan intravena ditunjukkan pada Gambar 3.15.



Gambar 3.15 Pengujian tekanan dengan tabung pemodelan intravena

3.3.4 Tahap Pengambilan Data

Pengambilan data dilakukan untuk menguji seberapa besar kinerja alat serta untuk mengetahui hasil dari kerja alat tersebut. Berdasarkan hasil perhitungan perbandingan *gear* dengan jumlah *count* yang telah dilakukan dalam proses kalibrasi motor stepper, akan diperoleh jumlah *count* untuk volume 1 ml. Data tersebut diolah ke program lalu dilakukan data perhitungan alat untuk besaran waktu, volume, dan *flow rate*. Sedangkan pada pengambilan data tegangan yang terukur pada nilai tekanan yang telah ditentukan digunakan tabung pemodelan pembuluh darah vena serta tensimeter.

3.3.5 Analisis Data

Validasi metode analitik dilakukan untuk mengetahui kelayakan dari metode yang digunakan. Parameter yang digunakan antara lain presisi dan akurasi.

3.3.5.1 Presisi (Ketelitian)

Presisi merupakan derajat keterulangan (*reproductibility*) dari suatu metode analisis. Hasil yang diperoleh dengan pengukuran yang berulang pada kondisi yang sama dapat dinyatakan dengan presisi. Presisi ditentukan dengan menghitung simpangan baku (standar deviasi/SD) dan koefisien variasi (KV) nilai volume cairan dan *flow rate* dengan Persamaan (3.5) dan (3.6).

$$SD = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n-1}} \dots\dots\dots(3.5)$$

$$KV = \frac{SD}{\bar{x}} \times 100\% \dots\dots\dots(3.6)$$

Keterangan :

- SD : standar deviasi
- KV : koefisien variasi
- x_i : volume pada masing-masing pengukuran
- $\bar{x}$: volume rata-rata
- n : jumlah uji yang dilakukan

3.3.5.2 Akurasi (*Recovery*)

Recovery menyatakan seberapa dekat hasil analisis terhadap volume cairan, *flow rate*, dan tekanan sebenarnya. Harga *recovery* atau persen perolehan kembali dapat ditentukan dengan membandingkan nilai hasil uji volume, *flow rate*, dan tekanan dengan nilai yang sebenarnya (berupa *input* pada alat *syringe pump*). Harga persen *recovery* atau persen perolehan kembali (R) dapat dihitung dengan Persamaan (3.7) dan (3.8).

$$Error(\%) = \left| \frac{N_{sp} - N_s}{N_s} \right| \times 100\% \dots \dots \dots (3.7)$$

$$Akurasi = 100\% - Error(\%) \dots \dots \dots (3.8)$$

dengan ketentuan N_{sp} = nilai hasil pengukuran dan N_s = nilai *input* pada *syringe pump*.

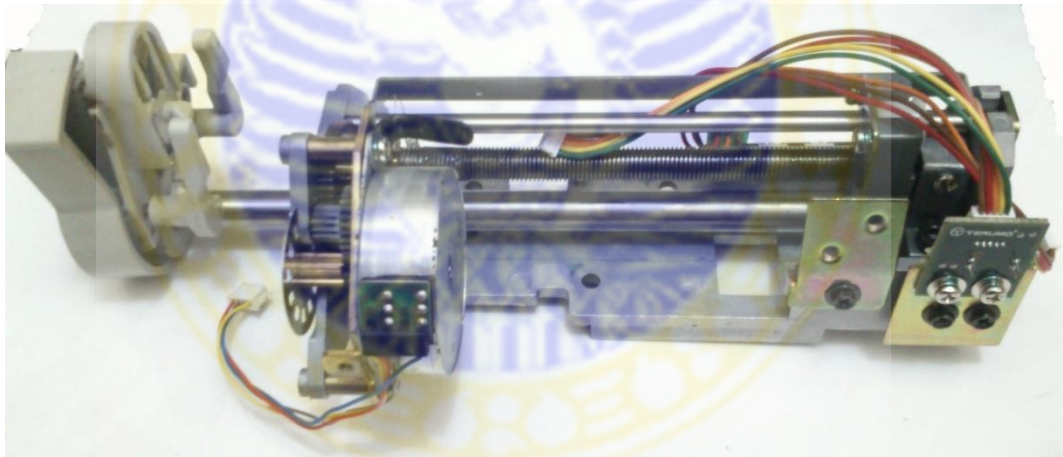
BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Perancangan Alat

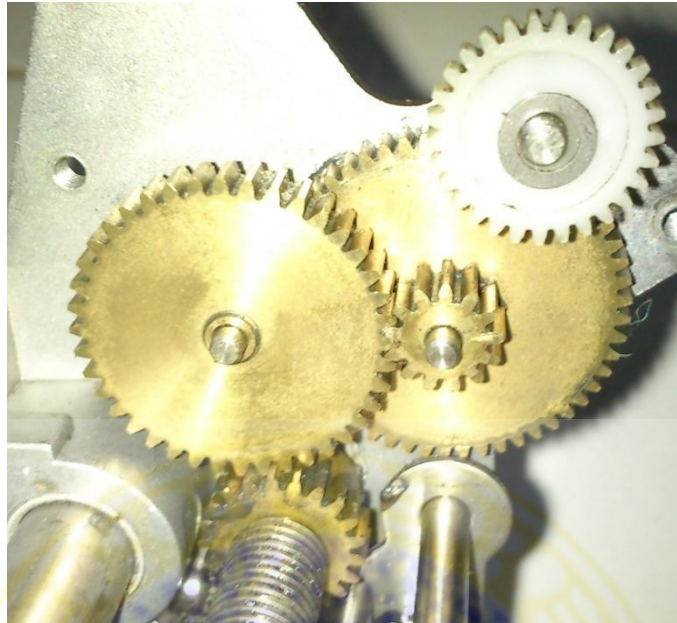
4.1.1 Analisis Mekanik

Perangkat mekanik sangat berpengaruh terhadap keberhasilan dibuatnya alat *syringe pump*. Perangkat mekanik yang sesuai akan mendukung *hardware* dan *software* sehingga alat *syringe pump* sesuai dengan yang diharapkan. Pada penelitian ini, digunakan perangkat mekanik dari *syringe pump* merek Terumo TE-331. Adapun bentuk mekanik alat *syringe pump* ditunjukkan pada Gambar 4.1.



Gambar 4.1 Perangkat mekanik alat *syringe pump*

Perangkat mekanik pada Gambar 4.1 tersusun atas 7 buah *gear* yang saling bersinggungan satu dengan yang lain. Pada *gear* pertama langsung dihubungkan dengan motor stepper sehingga kelajuan motor stepper (rpm) sama dengan kelajuan *gear* pertama. Pada *gear* nomor 7 berhubungan dengan *feed screw* yang secara langsung akan menggerakkan alat suntik. Susunan *gear* yang terdapat pada alat *syringe pump* ditunjukkan pada Gambar 4.2.



Gambar 4.2 Susunan gear pada perangkat mekanik

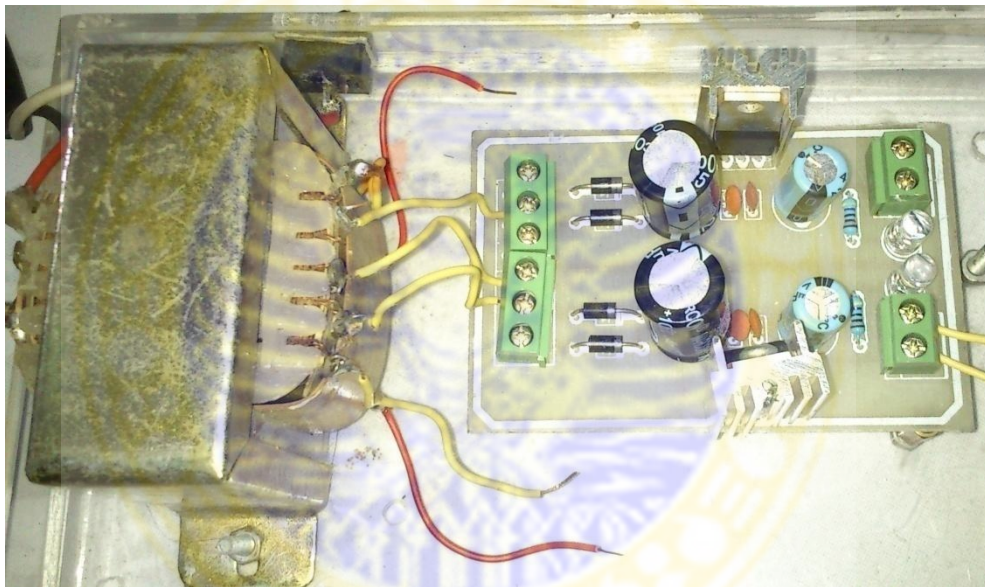
Berdasarkan hubungan *gear* pada perangkat mekanik alat *syringe pump* maka dapat diperoleh perbandingan kecepatan motor (rpm) dengan kecepatan *gear* yang berhubungan langsung dengan *feed screw*. Persamaan perbandingan kecepatan ini diperlukan untuk kalibrasi motor stepper sehingga diperoleh jumlah *count* untuk menghasilkan nilai volume sebesar 1 ml.

4.1.2 Perangkat Keras (*Hardware*)

Perangkat keras yang berhasil dibuat dalam penelitian ini adalah alat *syringe pump* berbasis mikrokontroler ATmega8535 dilengkapi dengan detektor oklusi yang mampu menginjeksikan dosis volume cairan obat dengan *flow rate* tertentu. Adapun perangkat keras ini terdiri dari rangkaian catu daya, rangkaian *driver motor* L298, rangkaian LCD, rangkaian buzzer, rangkaian potensiometer geser, rangkaian optocoupler dan komparator, rangkaian *load cell*, dan rangkaian minimum sistem AVR ATmega8535.

4.1.2.1 Rangkaian Catu Daya

Rangkaian catu daya yang telah dibuat mampu menghasilkan tegangan keluaran stabil +5,06 V dan -5,06 V. Pada rangkaian catu daya menggunakan trafo yang berfungsi untuk menurunkan tegangan, dioda untuk mengubah arus AC menjadi arus DC, IC 7805 yang berfungsi untuk menstabilkan tegangan keluaran menjadi +5 V, dan IC 7905 yang berfungsi untuk menstabilkan tegangan keluaran menjadi -5 V. Hasil pembuatan rangkaian catu daya ditunjukkan pada Gambar 4.3.



Gambar 4.3 Rangkaian catu daya

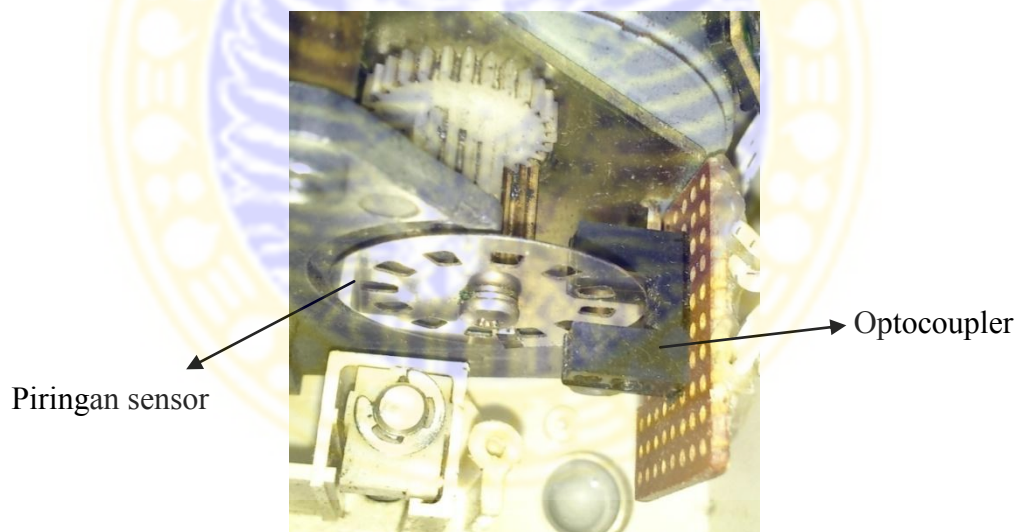
4.1.2.2 Rangkaian *Driver Motor* L298

Rangkaian *driver motor* digunakan untuk memberikan tegangan atau arus yang memadai agar motor dapat bekerja dengan baik. L298 merupakan jenis IC *driver motor* yang dapat mengendalikan arah putaran dan kecepatan motor stepper. *Driver motor* L298 mendapatkan tegangan masukan dari catu daya sebesar +5 V. Jenis motor yang digunakan pada penelitian adalah motor stepper bipolar ($7,5^0/step$) yang memiliki empat masukan dimana terdapat 2 kabel yang saling

berpasangan. Setiap pasangan kabel dihubungkan dengan rangkaian *diode bridge* yang berfungsi sebagai penyearah arus. Hasil rangkaian *driver motor* dapat dilihat pada Gambar 4.5.

4.1.2.3 Rangkaian Optocoupler dan Komparator

Optocoupler berfungsi sebagai sensor untuk mengetahui jumlah volume cairan obat yang telah diinjeksi. Pemasangan optocoupler dilengkapi dengan piringan sensor yang dipasang dengan *gear box* motor sehingga gerakan motor akan menyebabkan piringan sensor mengalami rotasi. Optocoupler akan mendeteksi piringan sensor yang terdapat celah atau tidak bercelah. Gambar 4.4 merupakan hasil pemasangan optocoupler pada piringan sensor.



Gambar 4.4 Pemasangan optocoupler pada piringan sensor

Tegangan keluaran dari kolektor fototransistor optocoupler pada saat di antara bagian *transmitter* dan *receiver* terhalangi atau tidak terhalangi oleh piringan sensor memiliki nilai tegangan yang berubah-ubah. Oleh karena itu, pada rangkaian optocoupler ditambah dengan rangkaian komparator menggunakan IC LM358 untuk mempertegas tegangan keluaran sebelum diolah oleh

mikrokontroler. Tegangan keluaran dari optocoupler dihubungkan ke masukan *non-inverting* (+) *operational amplifier* dalam LM358. Pada masukan *inverting* (-) dihubungkan dengan potensiometer sebagai tegangan referensi.

Rangkaian komparator akan membandingkan tegangan masukan pada masukan *non-inverting* (+) dengan masukan *inverting* (-). Tegangan keluaran dari komparator akan bernilai $+V_{\text{supply}}$ jika tegangan pada *non-inverting* (+) lebih besar daripada tegangan *inverting* (-). Sebaliknya, tegangan keluaran komparator akan bernilai 0 V jika tegangan pada *inverting* (-) bernilai lebih besar daripada tegangan *non-inverting* (+). Tegangan keluaran dari komparator selanjutnya masuk pada Pin PB.1 mikrokontroler AVR ATmega8535 yang berfungsi sebagai *counter*. Hasil pembuatan rangkaian optocoupler dan komparator dapat dilihat pada Gambar 4.5.

4.1.2.4 Rangkaian LCD

Pada penelitian ini digunakan LCD dot matriks dengan karakter 2 x 16, kaki-kakinya berjumlah 16 pin, dilengkapi dioda 1N4002 untuk menyearahkan tegangan masukan 5 V, serta resistor variabel untuk memberi tegangan kontras pada matriks LCD. LCD digunakan untuk menampilkan kondisi *syringe pump* ON atau OFF, menampilkan jumlah volume cairan obat (ml) dan *flow rate* (ml/jam) penginjeksian obat, dan dapat menampilkan sisa volume obat pada alat suntik ketika proses injeksi sedang berlangsung. Hasil rangkaian LCD dapat dilihat pada Gambar 4.5.

4.1.2.5 Rangkaian Buzzer

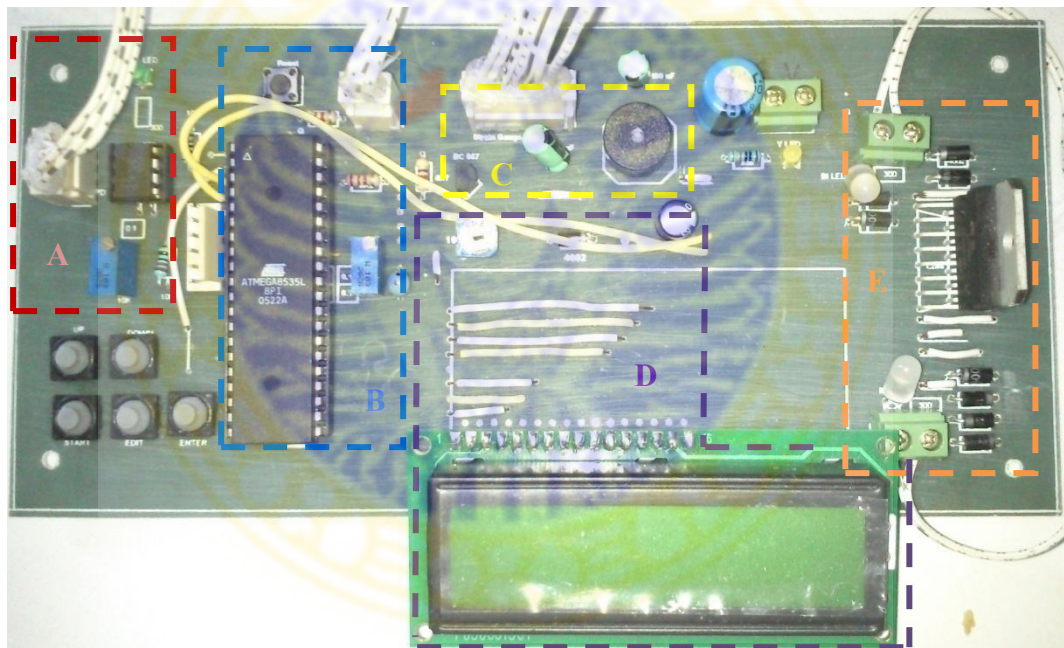
Pada alat *syringe pump* buzzer digunakan sebagai alarm penanda tiga keadaan. Pertama untuk mendeteksi *nearly empty* yaitu ketika volume cairan obat dalam suntikan mendekati habis (± 1 ml), kedua untuk mendeteksi terjadinya oklusi, ketiga untuk mengindikasikan bahwa volume cairan obat telah habis diinjeksikan. Rangkaian buzzer dilengkapi dengan komponen transistor PNP BC557 yang berfungsi sebagai saklar.

Penggunaan transistor sebagai saklar artinya mengoperasikan transistor pada salah satu kondisi yaitu saturasi atau *cut off*. Pada transistor PNP kondisi *cut off* adalah kondisi transistor dimana arus basis sama dengan nol ($I_B = 0$) dan arus pada emitor (I_E) sama dengan nol, sedangkan saturasi adalah kondisi transistor dimana arus basis adalah maksimal, arus emitor adalah maksimal, dan tegangan emitor-kolektor adalah minimal. Jadi, pada saat basis mendapatkan tegangan dari mikrokontroler akan menyebabkan transistor dalam kondisi *cut off* dan terminal emitor-kolektor terputus seperti saklar terbuka, akibatnya arus tidak akan mengalir dari emitor ke kolektor karena arus pada basis sama dengan nol. Pada kondisi ini arus kolektor sama dengan nol.

Sebaliknya, jika terminal basis tidak diberi tegangan akan menyebabkan transistor dalam kondisi saturasi seolah-olah terminal emitor-kolektor terhubung singkat seperti halnya saklar tertutup, akibatnya arus akan mengalir dari emitor ke kolektor. Hasil rangkaian buzzer dapat dilihat pada Gambar 4.5.

4.1.2.6 Rangkaian Minimum Sistem AVR ATmega8535

Rangkaian minimum sistem AVR ATmega8535 yang telah dirancang ini telah mampu mengolah data yang dimasukkan melalui *push button* untuk masukan nilai volume dan *flow rate* pada alat *syringe pump*. Rangkaian ini juga berfungsi sebagai pengatur untuk masing-masing rangkaian penyusun *syringe pump* seperti *driver motor* L298, LCD, buzzer, potensio geser, optocoupler dan komparator, dan *load cell*. Rangkaian minimum sistem dan rangkaian penyusun *syringe pump* ditunjukkan pada Gambar 4.5.



Keterangan :

A = Rangkaian optocoupler dan komparator

D = Rangkaian LCD

B = Rangkaian minimum sistem

E = Rangkaian *driver motor*

C = Rangkaian buzzer

Gambar 4.5 Rangkaian *driver motor*, optocoupler dan komparator, buzzer, LCD, dan minimum sistem alat *syringe pump*

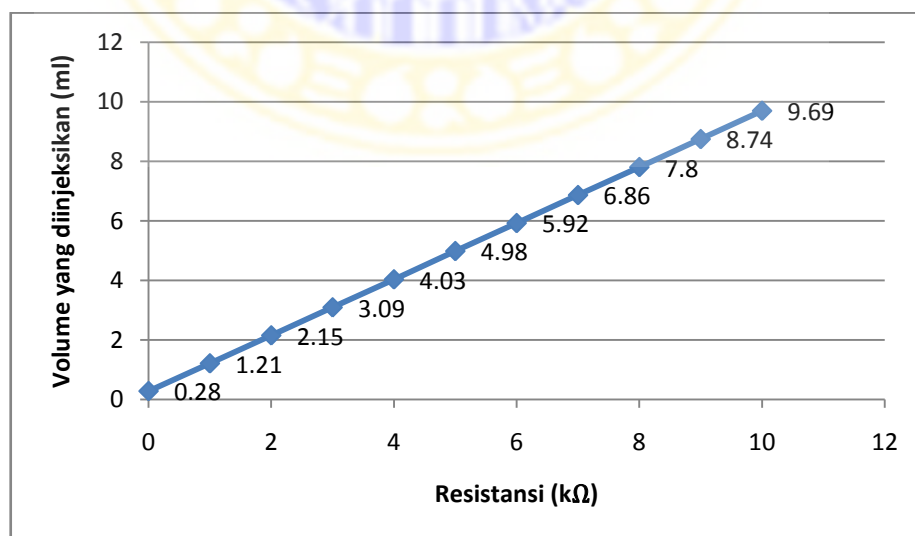
4.1.2.7 Rangkaian Potensio Geser

Rangkaian potensio geser digunakan untuk mendeteksi *nearly empty*, yaitu indikasi bahwa volume cairan obat dalam alat suntik mendekati habis (± 1 ml). Sensor potensio geser yang digunakan dapat dilihat pada Gambar 4.6.



Gambar 4.6 Sensor potensio geser

Nilai hambatan pada potensio geser akan berubah seiring dengan gerak translasi dari *plunger* alat suntik. Hubungan linieritas antara hambatan potensio geser dengan gerak translasi untuk *nearly empty* ditunjukkan pada Gambar 4.7.

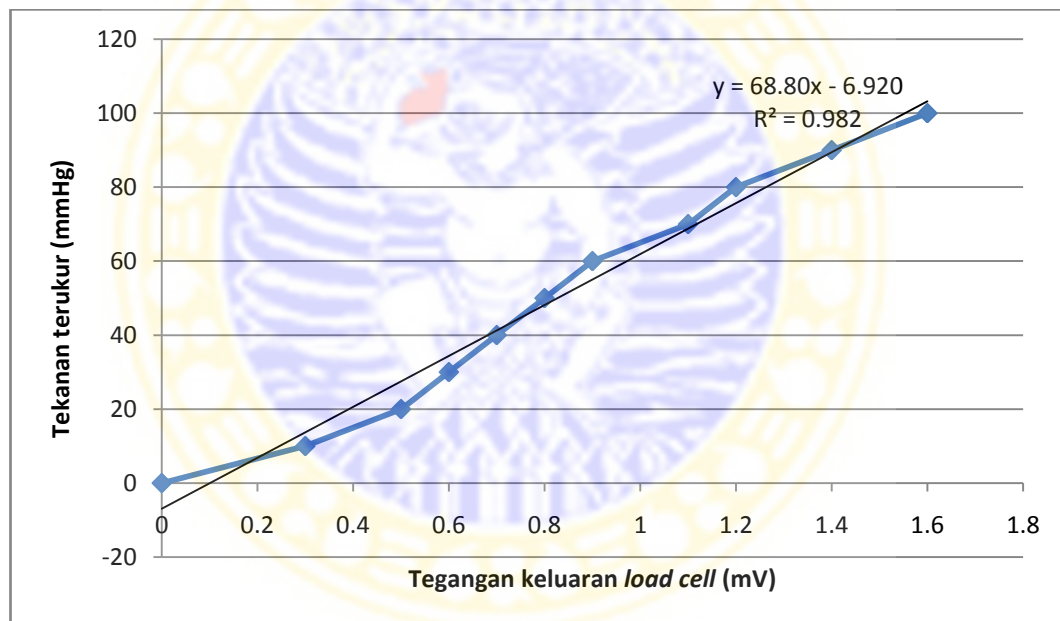


Gambar 4.7 Hubungan linieritas volume dengan resistansi potensio geser

Berdasarkan Gambar 4.7 dapat dilihat bahwa semakin mendekati volume habis pada alat suntik, maka nilai resistansi potensio geser semakin besar. Sebaliknya, semakin jauh dari nilai volume *nearly empty*, maka nilai hambatan dan tegangan keluaran dari potensiometer semakin kecil.

4.1.2.8 Rangkaian Load Cell

Rangkaian *load cell* digunakan untuk mendeteksi terjadinya oklusi pada tekanan 100 mmHg. Hubungan linieritas antara tekanan yang diukur dengan tegangan keluaran *load cell* ditunjukkan pada Gambar 4.8.



Gambar 4.8 Hubungan linieritas tegangan keluaran *load cell* dengan tekanan

Berdasarkan Gambar 4.8 dapat dilihat bahwa semakin tinggi nilai tekanan yang terukur, maka nilai tegangan keluaran *load cell* semakin besar. Nilai tegangan keluaran dari *load cell* sangat kecil (orde mV) sehingga diperlukan rangkaian penguat. Oleh karena itu, rangkaian *load cell* dilengkapi dengan *instrumentation amplifier*. Keluaran tegangan dari *load cell* dikuatkan dengan

menggunakan rangkaian *instrumentation amplifier* yang konfigurasinya telah terdapat pada IC AD620. Hasil penguatan tegangan dari rangkaian *instrumentation amplifier* ditentukan berdasarkan besarnya nilai resistor yang dihubungkan pada pin 1 dan 8 (pin R_G , *Resistor Gain*) dari IC AD620. Pada rangkaian *instrumentation amplifier* digunakan *gain* sebesar 727 kali, sehingga nilai resistor R_G yang diperlukan dapat dihitung menggunakan Persamaan (2.19).

$$727 = 1 + \frac{49,4 \text{ k}\Omega}{R_G}$$

$$R_G = 68,044 \text{ }\Omega$$

Sinyal keluaran *load cell* dari rangkaian *instrumentation amplifier* masih terdapat *noise* sehingga diperlukan rangkaian filter untuk menghilangkan *noise* tersebut. Pada penelitian ini filter yang digunakan adalah *low pass filter* dengan nilai $R = 1 \text{ k}\Omega$ dan $C = 10 \text{ }\mu\text{F}$. Filter ini memiliki nilai frekuensi *cutoff* yang dihitung berdasarkan Persamaan (3.4) yaitu :

$$f_c = \frac{1}{2 (3,14) (1 \text{ k}\Omega) (10 \text{ }\mu\text{F})}$$

$$f_c = 15,92 \text{ Hz}$$

Setelah tegangan keluaran *load cell* dikuatkan melalui rangkaian *instrumentation amplifier*, kemudian *noise* dihilangkan dengan rangkaian *low pass filter*, maka diperoleh hasil tegangan keluaran pada Tabel 4.1.

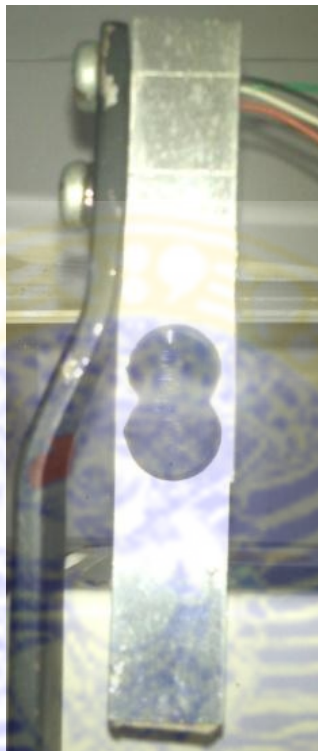
Tabel 4.1. Hasil pengukuran tegangan keluaran rangkaian *load cell*

P(mmHg)	0	10	20	30	40	50	60	70	80	90	100
V_a (mV)	0,0	0,3	0,5	0,6	0,7	0,8	0,9	1,1	1,2	1,4	1,6
V_b (V)	0,00	0,22	0,37	0,44	0,50	0,59	0,65	0,80	0,88	1,02	1,16

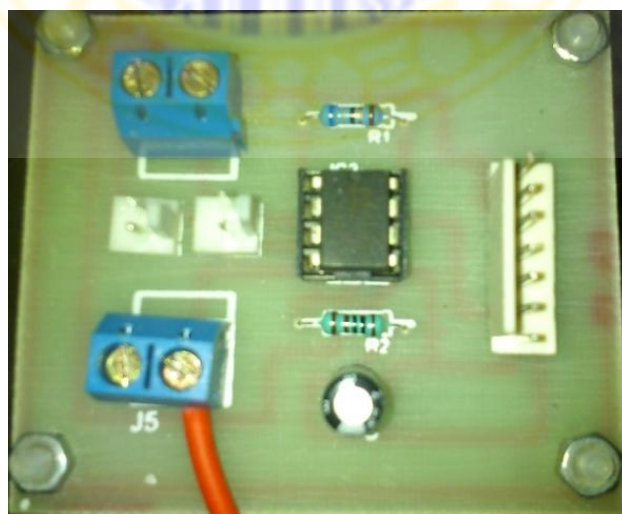
Keterangan : V_a = tegangan keluaran *load cell*

V_b = tegangan keluaran setelah dikuatkan dan difilter

Bentuk fisik *load cell* 3 kg ditunjukkan pada Gambar 4.9, sedangkan hasil rangkaian dari *instrumentation amplifier* dan *low pass filter* ditunjukkan pada Gambar 4.10.



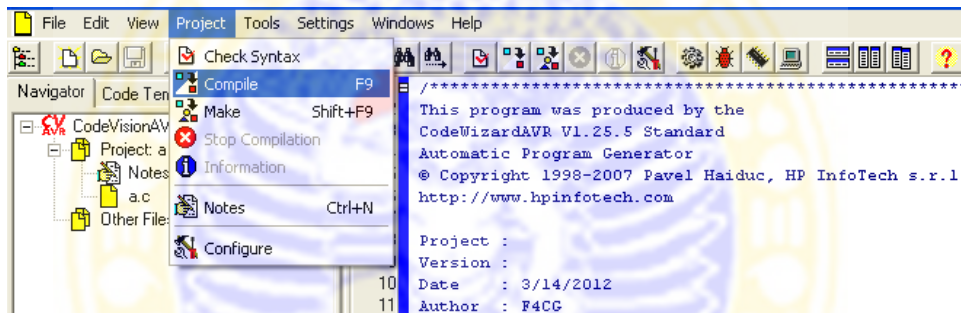
Gambar 4.9 Bentuk fisik *load cell* 3 kg



Gambar 4.10 Hasil rangkaian *instrumentation amplifier* dan *low pass filter*

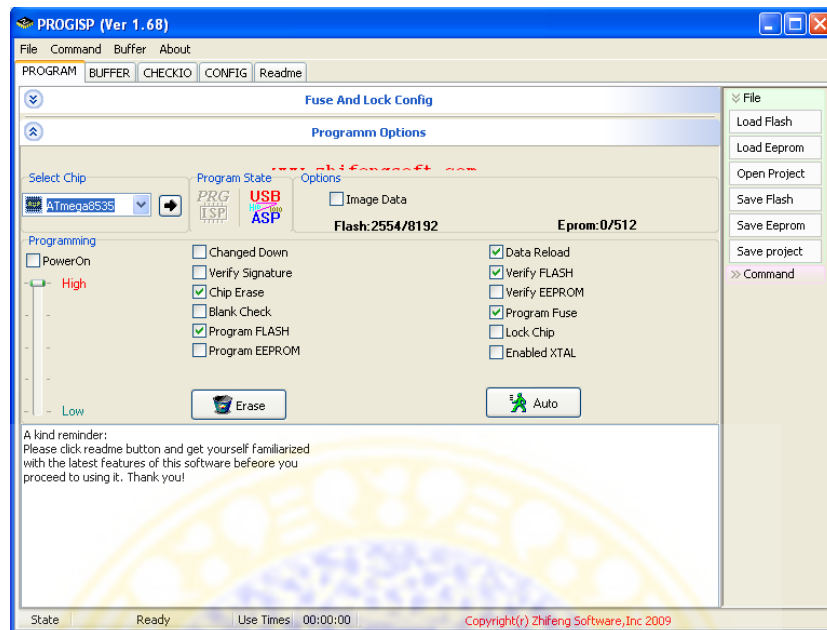
4.1.3 Perangkat Lunak (*Software*)

Perangkat lunak (*software*) berupa program *syringe pump* yang ditulis dengan bahasa C melalui CodeVisionAVR. Selanjutnya dilakukan pengisian ke IC mikrokontroler melalui *downloader software* PROGISP (Ver 1.68) menggunakan downloader USB dengan kabel USB. *Software* yang telah dibuat pada CodeVisionAVR sebelumnya dilakukan *compile* terlebih dahulu untuk mengetahui apakah pada program masih terdapat *error* atau tidak. Gambar 4.11 merupakan tampilan proses *compile* pada *software* CodeVisionAVR.



Gambar 4.11 Proses *compile* program pada CodeVisionAVR

Jika pada proses *compile* program tidak terdapat *error*, maka program tersebut disimpan. Selanjutnya untuk proses *download* program ke mikrokontroler digunakan *software* PROGISP (Ver 1.68). Caranya adalah membuka *file* program yang telah disimpan sebelumnya (*file* dengan format .hex) dengan klik *Load Flash*, lalu klik tombol *Auto*. Tombol *Auto* berfungsi untuk menghapus program yang sebelumnya telah terdapat pada mikrokontroler selanjutnya diisi dengan program baru. Tampilan *downloader software* PROGISP (Ver 1.68) ditunjukkan pada Gambar 4.12.



Gambar 4.12 Tampilan PROGISP (Ver 1.68) *software downloader*

Perangkat lunak (*software*) yang telah berhasil dibuat pada penelitian ini meliputi program *push button up-down*, *setting volume*, *setting flow rate*, program untuk menjalankan motor stepper, program detektor *nearly empty*, program detektor oklusi, dan program tampilan LCD.

4.1.3.1 Program *Push Button Up-Down*

Program *push button up-down* digunakan untuk memberikan nilai *input* pada variabel *volume* dan *flow rate* yang akan tampil pada layar LCD. Adapun *listing* programnya adalah sebagai berikut.

```
if(up==0) // Jika tombol up ditekan
{
    delay_ms(100); // Memanggil delay dari library delay
    if(status==0) // Jika variabel status bernilai 0
    { volume++; }
```

```

if(status==1) // Jika variabel status bernilai 1

    { flowrate++; } }

if(down==0) // Jika tombol down ditekan

{

    delay_ms(100); // Memanggil delay dari library delay

    if(status==0) // Jika variabel status bernilai 0

        { volume--; }

    if(status==1) // Jika variabel status bernilai 1

        { flowrate--; } }

```

4.1.3.2 Program *Setting Volume*

Program *setting* variabel volume adalah dengan mengubah variabel volume (ml) menjadi *count*. Pada proses kalibrasi volume yang telah dilakukan diperoleh hasil yaitu setiap 1 ml volume cairan setara dengan 848 *step* dimana 1 *count* sebanding dengan 4 *step*, sehingga $848 \text{ step} = 212 \text{ count}$. Perhitungan *setting* variabel volume adalah :

$$1 \text{ ml} = 212 \text{ count}$$

$$\text{Jadi, volume} = X \text{ ml} \times 212$$

Listing program untuk *setting* volume adalah sebagai berikut :

```
hasil_volume=volume*212; // Rumus untuk perhitungan counter
```

4.1.3.3 Program *Setting Flow Rate*

Program *setting flow rate* diperlukan untuk mengatur kecepatan motor stepper saat proses injeksi berlangsung. Nilai variabel *time_delay* yang akan

dimasukkan pada program motor stepper diperoleh dari perhitungan volume (ml) dibagi dengan *flow rate* (ml/jam).

```
t= (3600000*volume/flowrate); // Rumus untuk mencari waktu kemudian
```

dikonversi ke satuan milidetik

```
ts= t/hasil_volume; // Rumus untuk mencari waktu dalam 1 putaran
```

counter

```
t_tunda= ts/16; // Rumus untuk mencari waktu tunda dalam motor stepper
```

4.1.3.4 Program Menjalankan Motor Stepper

Pada saat *push button* ditekan tombol *start*, maka mikrokontroler akan mengontrol motor agar berada dalam kondisi *ON*. Kecepatan motor stepper diatur sesuai dengan *input* volume dan *flow rate* sehingga akan diperoleh variabel *time_delay*. Adapun *listing* programnya adalah :

```
while (data<=hasil_volume) // Selama nilai data kurang dari sama dengan
```

hasil_volume

```
{
```

```
PORTB.3=1;
```

```
PORTD.7=0;
```

```
PORTB.2=0;
```

```
PORTD.6=0;
```

```
delay_ms(t_tunda);
```

```
PORTB.3=0;
```

```
PORTD.7=1;
```

```
PORTB.2=0;
```



```

PORTD.6=0;

delay_ms(t_tunda);

PORTB.3=0;

PORTD.7=0;

PORTB.2=1;

PORTD.6=0;

delay_ms(t_tunda);

PORTB.3=0;

PORTD.7=0;

PORTB.2=0;

PORTD.6=1;

delay_ms(t_tunda);
}

```

Motor stepper yang digunakan pada alat *syringe pump* berputar secara diskrit, tidak secara *continuous*. Hal ini dikarenakan prinsip kerja dari motor stepper bergerak secara per *step* dan pada program menggunakan perintah *delay*. Perintah *delay* ini berfungsi untuk memberikan waktu tunda sehingga waktu penginjeksian yang diinginkan dapat tercapai.

4.1.3.5 Program Deteksi *Nearly Empty*

Program deteksi *nearly empty* digunakan untuk mengetahui ketika jumlah cairan obat dalam *syringe pump* sudah hampir habis. Buzzer akan menyala ketika sisa volume cairan dalam alat suntik sebesar 1 ml. Adapun *listing* programnya adalah sebagai berikut.

```

NEARLY= read_adc(0); // Membaca ADC dari channel 0

if (NEARLY>=501)

{

    buzzer=0; // Buzzer menyala

    volume_tampung=tampil_vol; // Variabel volume_tampung =

                                tampil_vol

    awal=2; // Nilai variabel awal adalah 2

    goto exit_stopmotor; // Keluar ke kondisi exit_stopmotor

                                ditempatkan

}

```

4.1.3.6 Program Deteksi Oklusi

Program deteksi oklusi digunakan untuk mengetahui ketika terjadi oklusi pada saat proses injeksi sedang berlangsung. Adapun *listing* programnya adalah sebagai berikut.

```

if (OKLUSI>=530)

{

    bunyi=1; // Nilai variabel bunyi adalah 1

    goto exit; // Keluar ke kondisi exit ditempatkan

}

```

4.1.3.7 Program Tampilan LCD

Program tampilan LCD digunakan untuk menampilkan hasil nilai volume dan *flow rate*, serta sisa volume yang terdapat pada alat suntik pada saat proses

injeksi sedang berlangsung. Adapun listing program tampilan awal adalah sebagai berikut.

```

lcd_clear(); // Menghapus tampilan LCD

lcd_gotoxy(0,0); // Menempatkan tulisan pada posisi kolom 0 baris 0

lcd_putsf(" Syringe Pump "); // Menampilkan string Syringe Pump

lcd_gotoxy(0,1); // Menempatkan tulisan pada posisi kolom 0 baris 1

lcd_putsf("Teknobiomedik-UA"); // Menampilkan string

delay_ms(100); // Memanggil delay dari library delay

```

Listing program secara keseluruhan pada tahap pembuatan perangkat lunak (*software*) alat *syringe pump* dapat dilihat pada Lampiran 3.

4.2 Hasil Pengujian Alat dan Analisis Data

Pada penelitian yang telah dilakukan diperoleh hasil pengujian, antara lain pengujian kalibrasi motor stepper, pengujian pengujian volume, pengujian waktu, pengujian *flow rate*, dan pengujian tekanan.

4.2.1 Kalibrasi Motor Stepper

Kalibrasi motor stepper dilakukan untuk mengetahui jumlah *count* yang diperoleh dari putaran piringan optocoupler dalam menghasilkan injeksi volume sebesar 1 ml. Volume 1 ml diperoleh dari perhitungan pergeseran *feed screw* yang sama dengan pergeseran alat suntik. Pergeseran *feed screw* untuk volume 1 ml ini diperoleh berdasarkan analisis perbandingan *gear* (roda gigi) yang digunakan dalam sistem mekanik alat *syringe pump*.

Berdasarkan hasil perhitungan diperoleh volume 1 ml sama dengan pergeseran *feed screw* sebesar 1,413 mm. Volume 1 ml sebanding dengan banyaknya gerak *step* motor stepper sebesar 848 *step*. Data ini dimasukkan ke program *counter* agar menghasilkan nilai volume sesuai dengan volume *input*. Hasil dari pengujian ini dapat dilihat pada Tabel 4.2.

Tabel 4.2. Hasil pengujian *count* untuk volume 1 ml

No.	Volume <i>input</i> (ml)	Volume terukur (ml)	KV (%)
1.	1	1,00	0
2.	1	1,00	0
3.	1	1,00	0
4.	1	1,00	0
5.	1	1,00	0
6.	1	1,00	0
7.	1	1,00	0
8.	1	1,00	0
9.	1	1,00	0
10.	1	1,00	0

Presisi ditentukan dengan menghitung nilai koefisien variasi (KV). Semakin kecil nilai KV maka data tersebut semakin presisi. Berdasarkan hasil pengujian pada Tabel 4.2 dapat dilihat bahwa data variabel volume untuk alat *syringe pump* memiliki nilai presisi 100% dengan ketelitian gelas ukur 0,2 ml.

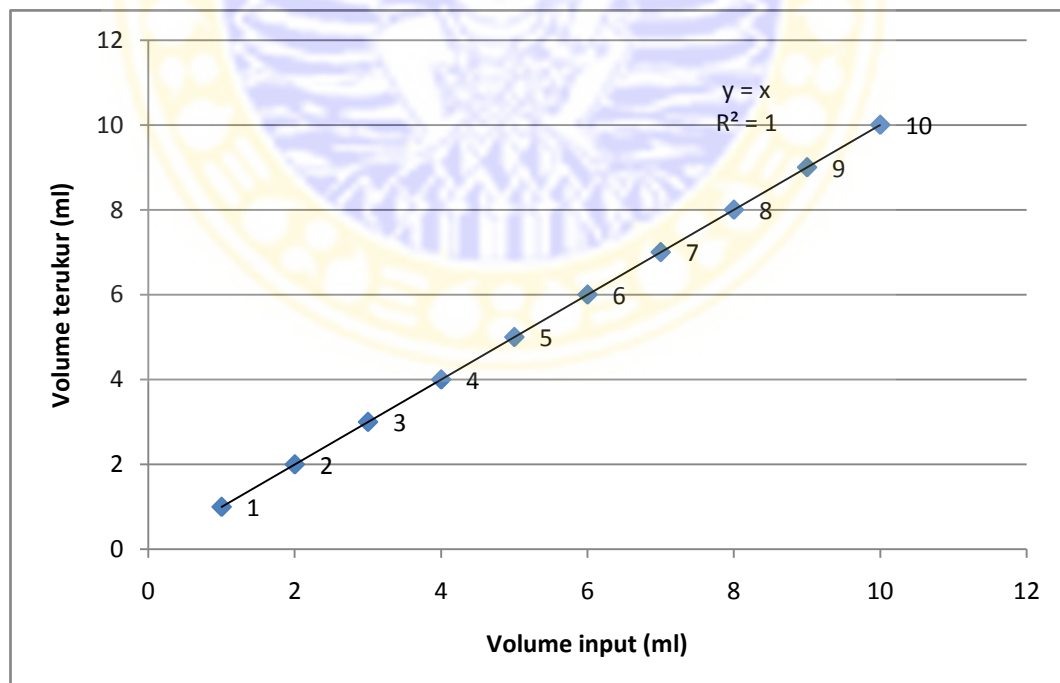
4.2.2 Hasil Pengujian Volume

Setelah dilakukan kalibrasi motor stepper diperoleh bahwa untuk menghasilkan volume sebesar 1 ml dibutuhkan 848 *step* dari motor stepper. Selanjutnya dilakukan pengujian volume dari alat *syringe pump* untuk mengetahui ketepatan alat *syringe pump* dalam menginjeksikan volume cairan obat. Hasil dari pengujian ini dapat dilihat pada Tabel 4.3.

Tabel 4.3. Hasil pengujian volume

No.	Volume <i>input</i> (ml)	Volume terukur (ml)	% Akurasi (volume <i>input</i> / volume terukur)
1.	1	1,00	100%
2.	2	2,00	100%
3.	3	3,00	100%
4.	4	4,00	100%
5.	5	5,00	100%
6.	6	6,00	100%
7.	7	7,00	100%
8.	8	8,00	100%
9.	9	9,00	100%
10.	10	10,00	100%

Berdasarkan hasil pengujian volume pada Tabel 4.3, dibuatlah grafik linieritas antara volume yang terukur dengan volume yang diinginkan sehingga diperoleh persamaan linieritasnya. Grafik persamaan linieritas volume dapat dilihat pada Gambar 4.13.



Gambar 4.13 Grafik linieritas volume

Pada Gambar 4.13 diperoleh persamaan linieritas untuk volume yaitu $y = x$ yang berarti bahwa nilai volume *input* selalu sama dengan volume terukur. Berdasarkan Tabel 4.3 terlihat bahwa variabel volume alat *syringe pump* memiliki tingkat akurasi 100% dengan ketelitian gelas ukur 0,2 ml.

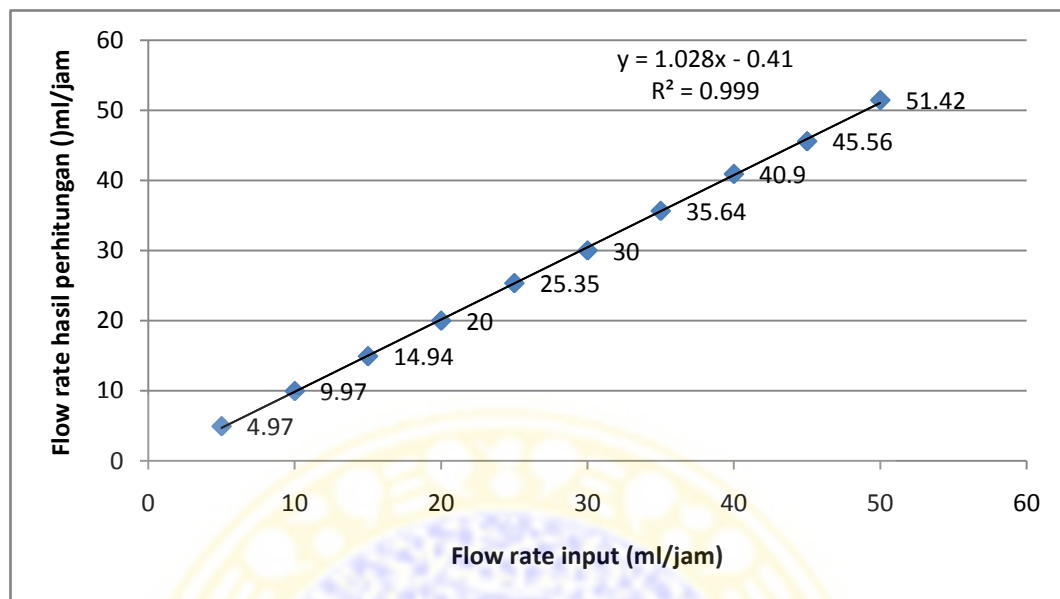
4.2.3 Hasil Pengujian Waktu dan *Flow Rate*

Pengujian waktu dan *flow rate* bertujuan untuk mengetahui ketepatan alat *syringe pump* dalam menginjeksikan cairan obat sesuai dengan laju alir yang diinginkan. Pada pengujian ini menggunakan variasi nilai *flow rate* untuk proses injeksi dengan volume tetap 1 ml. Hasil dari pengujian ini dapat dilihat pada Tabel 4.4.

Tabel 4.4. Hasil pengujian volume, waktu, dan *flow rate*

No.	<i>Flow rate</i> (ml/jam)	Volume input (ml)	Waktu pengukuran (menit:detik)	Volume yang dihasilkan (ml)	<i>Flow rate</i> (volume / waktu) (ml/jam)
1.	5	1	12 : 04	1	4,97
2.	10	1	06 : 01	1	9,97
3.	15	1	04 : 01	1	14,94
4.	20	1	03 : 00	1	20,00
5.	25	1	02 : 22	1	25,35
6.	30	1	02 : 00	1	30,00
7.	35	1	01 : 41	1	35,64
8.	40	1	01 : 28	1	40,90
9.	45	1	01 : 19	1	45,56
10.	50	1	01 : 10	1	51,42

Berdasarkan hasil pengujian volume pada Tabel 4.4, dibuatlah grafik linieritas antara *flow rate* hasil perhitungan dengan *flow rate* yang diinginkan sehingga diperoleh persamaan linieritasnya. Grafik persamaan linieritas *flow rate* dapat dilihat pada Gambar 4.14.

Gambar 4.14 Grafik linieritas *flow rate*

Pada Gambar 4.14 diperoleh persamaan linearitas $y = 1,028x - 0,41$. Oleh karena persamaan tersebut kurang linear ($y \neq 1$), maka dihitung nilai kesalahannya yang ditunjukkan pada Tabel 4.5.

Tabel 4.5. Data persentase kesalahan dari *flow rate*

No.	Flow rate (ml/jam)	Flow rate (volume / waktu) (ml/jam)	Kesalahan (%)
1.	5	4,97	0,60
2.	10	9,97	0,30
3.	15	14,94	0,40
4.	20	20,00	0,00
5.	25	25,35	1,40
6.	30	30,00	0,00
7.	35	35,64	1,80
8.	40	40,90	2,25
9.	45	45,56	1,24
10.	50	51,42	2,84
Rata-rata kesalahan :			1,08

Tingkat ketepatan (akurasi) *syringe pump* dalam menentukan nilai *flow rate* dihitung melalui persamaan :

$$\begin{aligned}\text{Ketepatan alat} &= 100\% - \text{persentase kesalahan} \\ &= 100\% - 1,08\% \\ &= 98,92\%\end{aligned}$$

Jadi, tingkat akurasi variabel *flow rate* pada alat *syringe pump* adalah sebesar 98,92% dan tingkat presisi 99,88% (perhitungan di Lampiran 8). Adanya persentase error pada variabel *flow rate* disebabkan oleh berbagai faktor antara lain pengukuran waktu manual yang tidak tepat dan gesekan yang terjadi antara *gear* penyusun mekanik *syringe pump*.

4.2.4 Hasil Pengujian Tekanan

Pengujian tekanan bertujuan untuk mengetahui tegangan keluaran dari sensor *load cell* pada tekanan 100 mmHg. Selanjutnya dilakukan pengujian tekanan dari alat *syringe pump* untuk mengetahui ketepatan alat *syringe pump* dalam mendeteksi tekanan 100 mmHg. Hasil dari pengujiannya dapat dilihat pada Tabel 4.6.

Tabel 4.6. Hasil pengujian pada tekanan 100 mmHg

No.	Tekanan terukur (mmHg)	Kesalahan (%)
1.	106	6,00
2.	100	0,00
3.	84	16,00
4.	104	4,00
5.	88	12,00
6.	89	11,00
7.	91	9,00
8.	90	10,00
9.	87	13,00
10.	97	3,00
Rata-rata kesalahan :		8,40

Tingkat ketepatan (akurasi) *syringe pump* dalam menentukan nilai tekanan dihitung melalui persamaan :

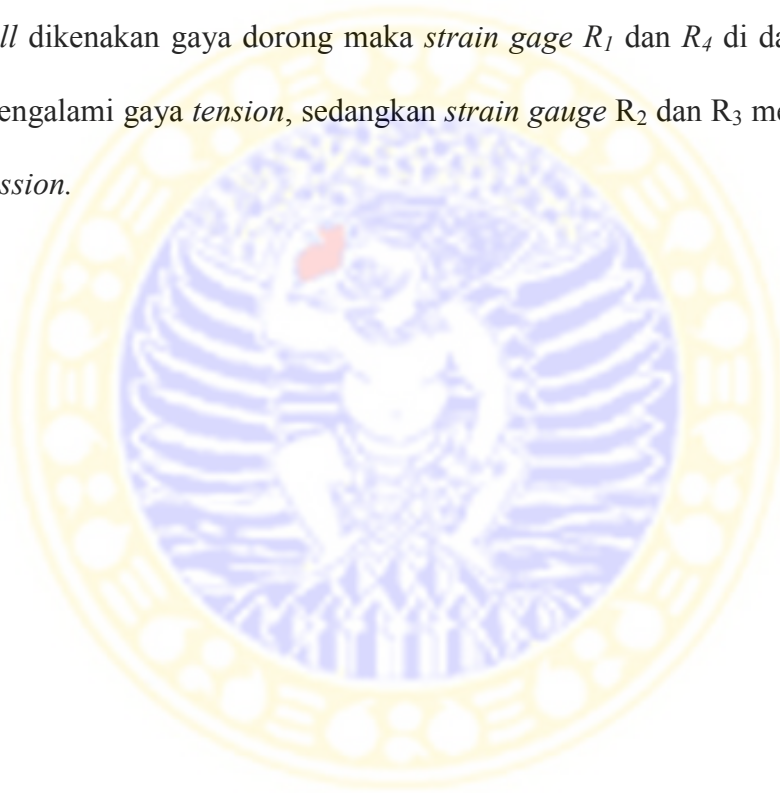
$$\begin{aligned}\text{Ketepatan alat} &= 100\% - \text{persentase kesalahan} \\ &= 100\% - 8,40\% \\ &= 91,60\%\end{aligned}$$

Jadi, tingkat akurasi variabel tekanan pada alat *syringe pump* adalah sebesar 91,60% dengan persentase kesalahan 8,40%. Permasalahan yang terjadi pada deteksi tekanan dengan *load cell* adalah nilai tegangan yang selalu berubah pada parameter nilai tekanan yang sama walaupun nilai tegangannya cenderung mengalami kenaikan dari tekanan rendah hingga tekanan tinggi. Hal ini disebabkan karena pengaruh mekanik dimana *load cell* akan mengalami vibrasi pada saat *syringe pump* dihidupkan. Vibrasi ini berasal dari motor stepper yang bekerja. Adapun hasil tegangan keluaran sensor *load cell* terhadap tekanan dapat dilihat di Lampiran 9.

Berdasarkan Tabel di Lampiran 9 dapat dilihat bahwa fungsi tekanan terhadap tegangan adalah berbanding lurus. Semakin besar nilai tekanan, maka tegangan yang terukur semakin meningkat. Namun, hasil tegangan keluaran sensor *load cell* selalu berubah pada uji yang berbeda dengan parameter nilai tekanan yang sama. Hasil tegangan keluaran *load cell* tidak akan presisi dengan nilai parameter beban atau tekanan yang sama jika sistem mekaniknya terdapat vibrasi yang mempengaruhi mekanik *load cell* (DS Europe srl, 1998).

Arah gaya yang diberikan terhadap sensor *load cell* akan mempengaruhi tanda hasil keluaran tegangan *load cell*. Berdasarkan Tabel di Lampiran 9 dapat

dilihat bahwa gaya dorong akan menghasilkan tegangan keluaran positif, sedangkan gaya tarik akan menghasilkan tegangan keluaran negatif. Hal ini terkait dengan hubungan persamaan antara tegangan keluaran *load cell* dengan gaya yang diberikan pada Persamaan (2.12). Jika pada *load cell* dikenakan gaya tarik maka *strain gauge* R_1 dan R_4 di dalam *load cell* akan mengalami gaya *compression*, sedangkan *strain gauge* R_2 dan R_3 mengalami gaya *tension*. Sebaliknya, jika pada *load cell* dikenakan gaya dorong maka *strain gage* R_1 dan R_4 di dalam *load cell* akan mengalami gaya *tension*, sedangkan *strain gauge* R_2 dan R_3 mengalami gaya *compression*.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisis data dan pembahasan yang dilakukan dalam penelitian ini dapat diperoleh kesimpulan sebagai berikut :

1. *Syringe pump* berbasis mikrokontroler ATmega8535 telah dibuat dan dapat bekerja dengan baik. *Syringe pump* ini dapat menginjeksikan cairan obat dengan tingkat keakuratan yang tinggi pada volume mulai dari 1 ml hingga 50 ml dan pada *flow rate* 1 ml/jam hingga 50 ml/jam. Tingkat akurasi variabel *flow rate* pada alat *syringe pump* adalah sebesar 98,92% dengan nilai presisi 99,88%.
2. *Syringe pump* yang telah dibuat belum mampu secara spesifik mendeteksi terjadinya oklusi pada nilai tekanan 100 mmHg diakibatkan karena adanya vibrasi pada *load cell*. Rata-rata persentase kesalahan sebesar 8,40% dengan tingkat akurasi sebesar 91,60% untuk tekanan.

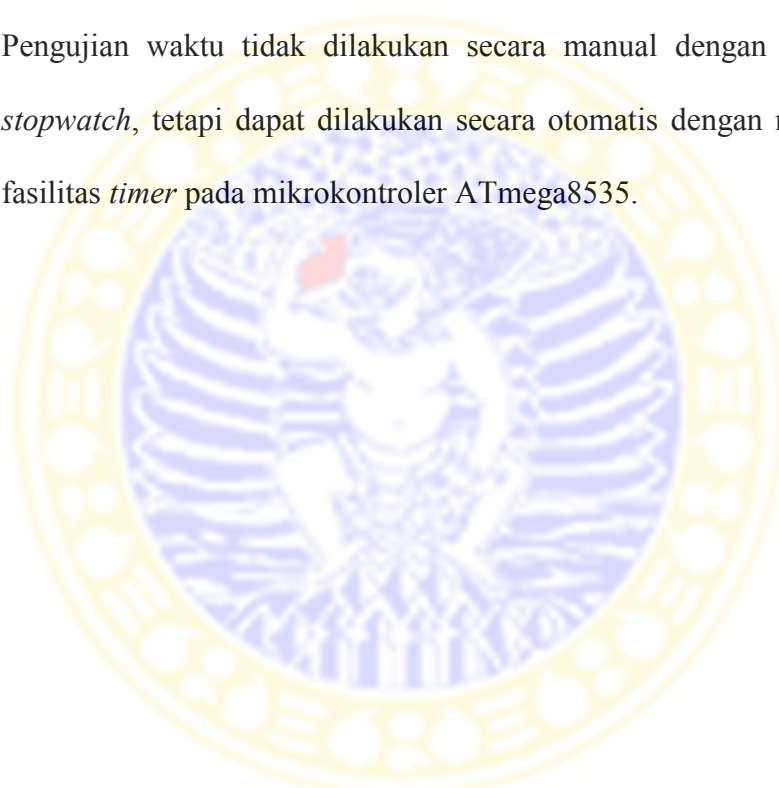
5.1 Saran

Berikut ini adalah beberapa saran yang dapat dipertimbangkan untuk penyempurnaan penelitian lebih lanjut :

1. Dalam pengembangan berikutnya diharapkan perancangan mekanik pada sistem deteksi oklusi dapat menghilangkan vibrasi akibat gerakan motor

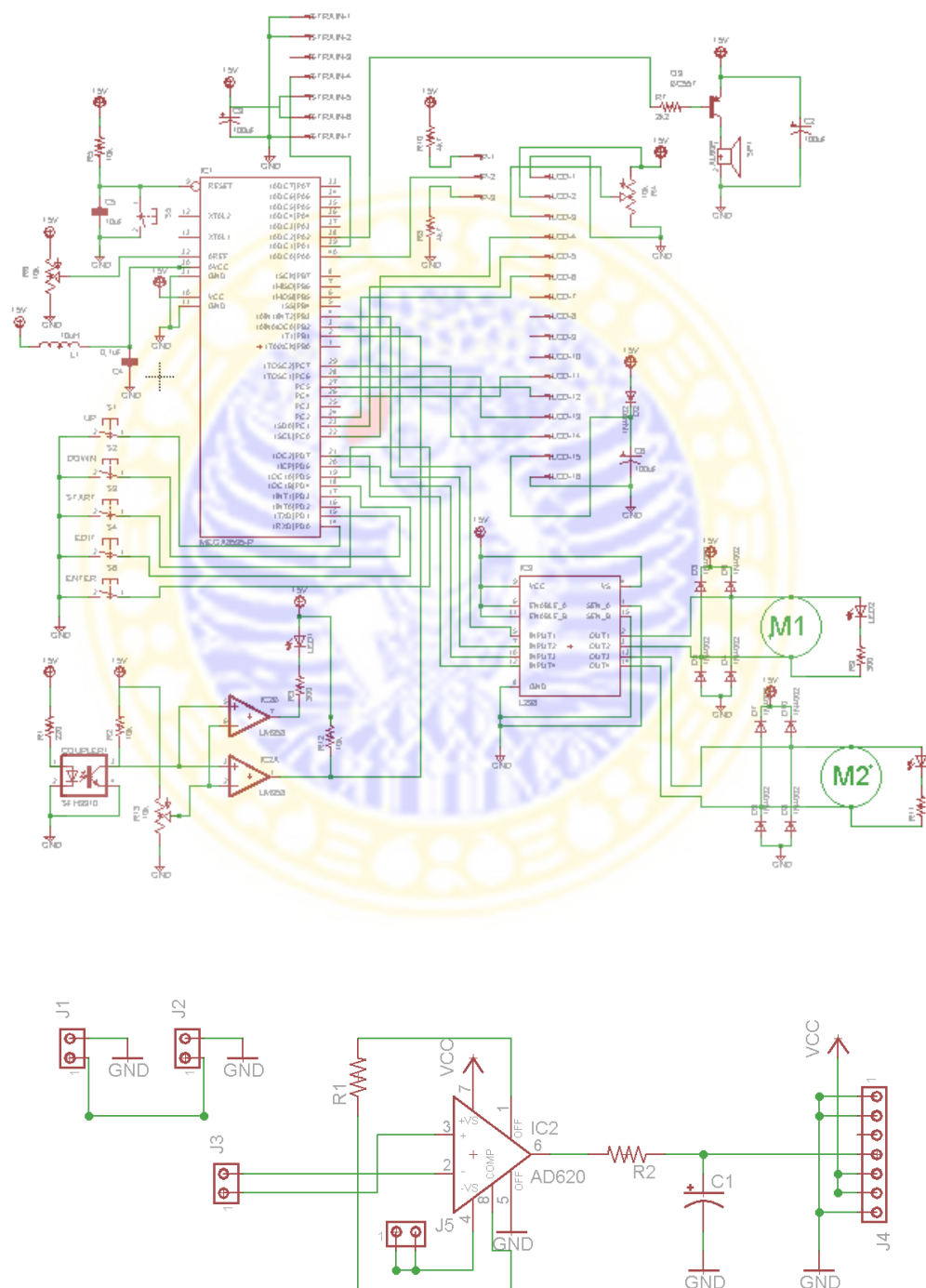
stepper yang akan mempengaruhi kinerja *load cell* dengan cara memberi sistem peredam vibrasi pada perangkat mekaniknya.

2. Ukuran alat suntik yang digunakan tidak hanya 50 ml tetapi juga mampu melakukan injeksi dengan alat suntik ukuran 10, 20, dan 30 ml dengan menambahkan sensor potensio geser yang diletakkan secara vertikal pada mekanik penjepit alat suntik.
3. Pengujian waktu tidak dilakukan secara manual dengan menggunakan *stopwatch*, tetapi dapat dilakukan secara otomatis dengan memanfaatkan fasilitas *timer* pada mikrokontroler ATmega8535.



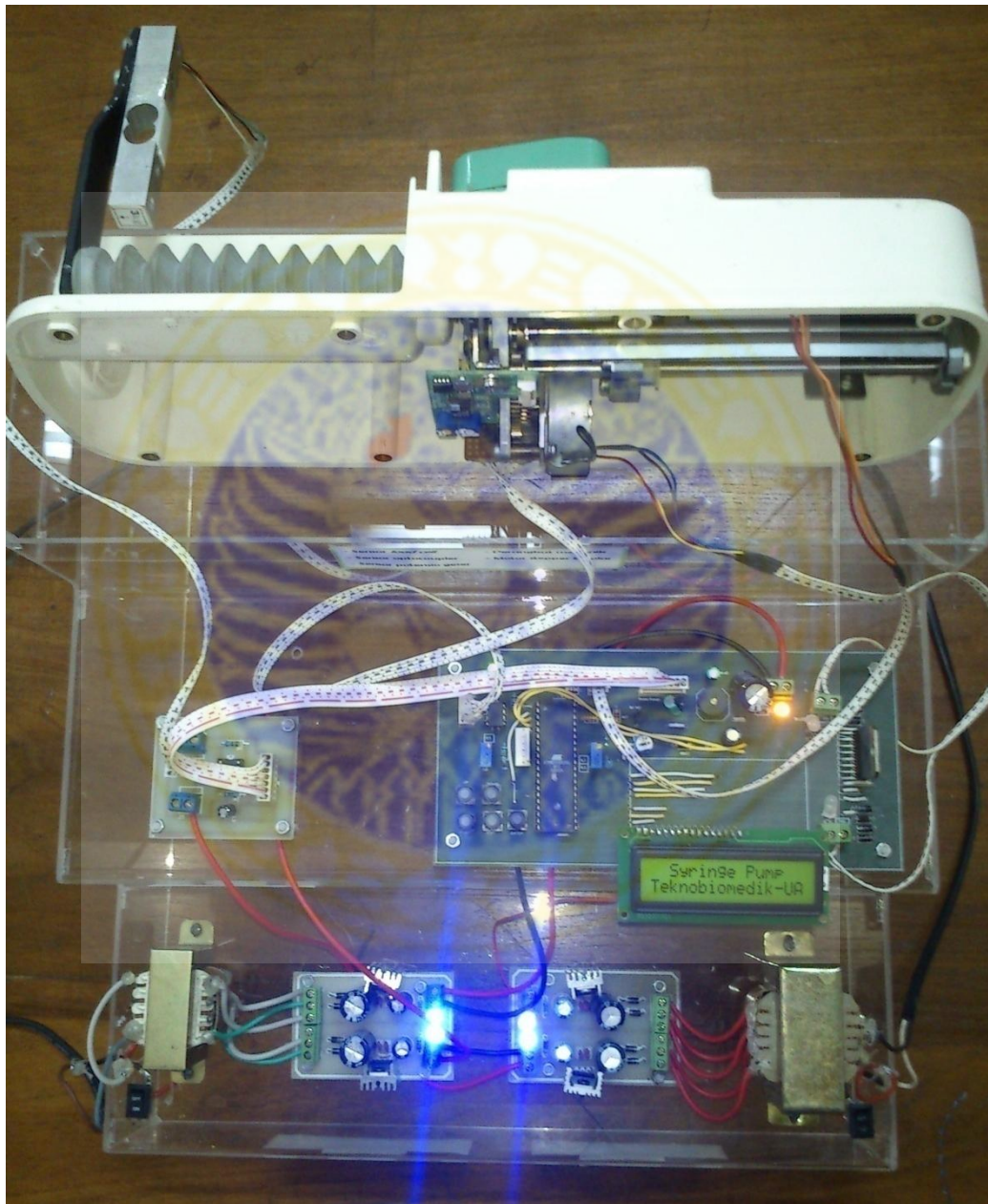
Lampiran 1

Skematik Rangkaian Syringe Pump



Lampiran 2

Hasil Alat *Syringe Pump*



Lampiran 3

Listing Program

/******

This program was produced by the

CodeWizardAVR V1.25.5 Standard

Automatic Program Generator

© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.com>

Project :

Version :

Date : 3/14/2012

Author : F4CG

Company : F4CG

Comments:

Chip type : ATmega8535

Program type : Application

Clock frequency : 4.000000 MHz

Memory model : Small

External SRAM size : 0

Data Stack size : 128

*****/


```

// temp[6] adalah variabel array untuk menampilkan string di LCD

// temp2[6] adalah variabel array untuk menampilkan string di LCD

unsigned char temp[6], temp2[6], temp3[6];


// data adalah variabel untuk menghitung counter

// volume adalah variabel volume input awal yang bernilai tetap

// hasil_volume adalah variabel hasil perkalian variabel data dan volume

// flowrate adalah variabel untuk input flowrate yang menentukan kecepatan

// motor

// status adalah variabel untuk memilih menu input volume atau input flowrate

// tampilkan_panah adalah variabel untuk menampilkan karakter string panah

// tampil_vol adalah variabel untuk menampilkan volume yang tersisa selama

// proses berjalan pada LCD

// pilih_edit adalah variabel untuk memilih tampilan menu panah volume dan

// flowrate atau sebagai kondisi pause

// volume_tampung adalah nilai untuk volume yang belum diinjeksikan pada saat

// kondisi pause

// awal adalah variabel untuk membedakan fungsi push button start sebagai start

// volume dari awal atau start setelah pause

int data, volume, hasil_volume, flowrate, status, tampilkan_panah, tampil_vol

int pilih_edit, volume_tampung, awal, i, bunyi, bunyi2, bunyi3;


// t adalah variabel untuk perhitungan waktu yang bernilai sekon

```

```
// ts adalah variabel untuk perhitungan waktu dalam satu nilai counter

// t_tunda adalah variabel sebagai input ke delay motor stepper bipolar

float t,ts,t_tunda ;

#include <mega8535.h>

#include <stdlib.h>

#include <delay.h>

// Alphanumeric LCD Module functions

#asm

    .equ __lcd_port=0x15 ;PORTC

#endasm

#include <lcd.h>

#define ADC_VREF_TYPE 0x00

// Read the AD conversion result

unsigned int read_adc(unsigned char adc_input)

{

    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);

    // Delay needed for the stabilization of the ADC input voltage

    delay_us(10);

    // Start the AD conversion
```

```

ADCSRA|=0x40;

// Wait for the AD conversion to complete

while ((ADCSRA & 0x10)==0);

ADCSRA|=0x10;

return ADCW;

}

```

```

#define start PIND.3

#define enter PIND.5

#define up PIND.0

#define down PIND.1

#define edit PIND.4

#define buzzer PORTA.2

// Declare your global variables here

void main(void)

{

// Declare your local variables here

int NEARLY, OKLUSI;

// Input/Output Ports initialization

// Port A initialization

```

```
// Func7=In Func6=Out Func5=Out Func4=In Func3=In Func2=Out Func1=In
```

```
//Func0=In
```

```
// State7=T State6=0 State5=0 State4=T State3=T State2=1 State1=T State0=T
```

```
PORTA=0x04;
```

```
DDRA=0x64;
```

```
// Port B initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
```

```
//Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTB=0x00;
```

```
DDRB=0x00;
```

```
// Port C initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
```

```
//Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTC=0x00;
```

```
DDRC=0x00;
```

```
// Port D initialization
```

```
// Func7=Out Func6=Out Func5=In Func4=In Func3=In Func2=In Func1=In
```

```
//Func0=In
```

```
// State7=0 State6=0 State5=P State4=P State3=P State2=T State1=P State0=P

PORTD=0x3B;

DDRD=0xC0;

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: Timer 0 Stopped

// Mode: Normal top=FFh

// OC0 output: Disconnected

TCCR0=0x00;

TCNT0=0x00;

OCR0=0x00;

// Timer/Counter 1 initialization

// Clock source: T1 pin Rising Edge

// Mode: Normal top=FFFFh

// OC1A output: Discon.

// OC1B output: Discon.

// Noise Canceler: Off

// Input Capture on Falling Edge

// Timer 1 Overflow Interrupt: Off

// Input Capture Interrupt: Off

// Compare A Match Interrupt: Off
```



```
// Compare B Match Interrupt: Off
```

```
TCCR1A=0x00;
```

```
TCCR1B=0x07;
```

```
TCNT1H=0x00;
```

```
TCNT1L=0x00;
```

```
ICR1H=0x00;
```

```
ICR1L=0x00;
```

```
OCR1AH=0x00;
```

```
OCR1AL=0x00;
```

```
OCR1BH=0x00;
```

```
OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: Timer 2 Stopped
```

```
// Mode: Normal top=FFh
```

```
// OC2 output: Disconnected
```

```
ASSR=0x00;
```

```
TCCR2=0x00;
```

```
TCNT2=0x00;
```

```
OCR2=0x00;
```

```
// External Interrupt(s) initialization
```

```
// INT0: Off

// INT1: Off

// INT2: Off

MCUCR=0x00;

MCUCSR=0x00;


// Timer(s)/Counter(s) Interrupt(s) initialization

TIMSK=0x00;


// Analog Comparator initialization

// Analog Comparator: Off

// Analog Comparator Input Capture by Timer/Counter 1: Off

ACSR=0x80;

SFIOR=0x00;


// ADC initialization

// ADC Clock frequency: 500.000 kHz

// ADC Voltage Reference: AVCC pin

// ADC High Speed Mode: Off

// ADC Auto Trigger Source: None

ADMUX=ADC_VREF_TYPE & 0xff;

ADCSRA=0x81;

SFIOR&=0xEF;
```

```

// LCD module initialization

lcd_init(16);

awal=0; // Untuk inisialisasi variabel awal bernilai 0
status=0; // Untuk inisialisasi variabel status bernilai 0
tampilkan_panah=0; // Untuk inisialisasi variabel tampilkan_panah bernilai 0
pilih_edit=0; // Untuk inisialisasi variabel pilih_edit bernilai 0
data=0; // Untuk inisialisasi variabel data bernilai 0
bunyi=0;
bunyi2=0;
bunyi3=0;
while (1) //Tampilan awal
{
    lcd_clear(); // Menghapus tampilan LCD
    lcd_gotoxy(0,0); // Menempatkan tulisan pada posisi kolom 0 baris 0
    lcd_putsf(" Syringe Pump "); // Menampilkan string Syringe Pump
    lcd_gotoxy(0,1); // Menempatkan tulisan pada posisi kolom 0 baris 1
    lcd_putsf("Teknobiomedik-UA"); // Menampilkan string Teknobiomedik-UA
    delay_ms(100); // Memanggil delay dari library delay

    if(enter==0) // Jika tombol enter ditekan maka masuk menu input Volume &
                //Flowrate

```

```

{
volume=0; // Inisialisasi volume awal bernilai 0

delay_ms(50); // Memanggil delay dari library delay

while(enter==1) // Selama tombol enter tidak ditekan lagi
{
    delay_ms(100); // Memanggil delay dari library delay
    lcd_clear(); // Menghapus tampilan LCD
    lcd_gotoxy(1,0); // Menempatkan tulisan pada posisi kolom 1 baris 0
    lcd_putsf("Volume : "); // Menampilkan string Volume :

    //Inisialisasi nilai volume untuk kondisi awal atau kondisi di pause
    if (volume_tampung!=0) // Jika kondisi adalah pause maka terdapat
        //nilai volume_tampung
        {
            itoa(volume_tampung,temp); // Konversi integer to string
            lcd_gotoxy(10,0); // Menempatkan tulisan pada posisi kolom 10
            //baris 0

            lcd_puts(temp); // Menampilkan string dari integer variabel

            //volume_tampung

            tampil_vol=volume_tampung; // Deklarasi variabel tampil_vol

            //sama dengan nilai volume_tampung
        }
}

```

```

else // Jika kondisi awal

{

    itoa(volume,temp); // Konversi integer to string

    lcd_gotoxy(10,0); // Menempatkan tulisan pada posisi kolom 10

                        //baris 0

    lcd_puts(temp); // Menampilkan string dari integer variabel

                        //volume

    tampil_vol=volume; // Deklarasi variabel tampil_vol sama

                        //dengan nilai volume

}

lcd_gotoxy(13,0); // Menempatkan tulisan pada posisi kolom 13 baris 0

lcd_putsf("ml"); // Menampilkan string ml

lcd_gotoxy(1,1); // Menempatkan tulisan pada posisi kolom 1 baris 1

lcd_putsf("Flow : "); // Menampilkan string Flow :

itoa(flowrate,temp2); // Konversi integer to string

lcd_gotoxy(8,1); // Menempatkan tulisan pada posisi kolom 8 baris 1

lcd_puts(temp2); // Menampilkan string dari integer variabel flowrate

lcd_gotoxy(11,1); // Menempatkan tulisan pada posisi kolom 11 baris 1

lcd_putsf("ml/h"); // Menampilkan string ml/h :

if(tampilkan_panah==1) // Jika kondisi tampilkan_panah bernilai 1

{

```



```

    lcd_gotoxy(0,1); // Menempatkan pada posisi kolom 0 baris 1

    lcd_putsf(">"); // Menampilkan string >

}

else // Jika kondisi tampilkan_panah selain nilai 1

{

    lcd_gotoxy(0,0); // Menempatkan tulisan posisi kolom 0 baris 0

    lcd_putsf(">"); // Menampilkan string >

}

if(bunyi==1) // Jika tombol edit ditekan

{

    for (i=0;i<=100;i++)

    {

        buzzer=0; // Buzzer menyala

        if(edit==0) // Jika tombol edit ditekan

        {

            buzzer=1;

            volume_tampung=tampil_vol; // Variabel

                                //volume_tampung = tampil_vol

            awal=2; // Nilai variabel awal adalah 2

            bunyi=0;

            goto exit;

```

```
//goto exit_steptomotor;

}

delay_ms(200);

if(edit==0) // Jika tombol edit ditekan
{
    buzzer=1;
    volume_tampung=tampil_vol;
    awal=2; // Nilai variabel awal adalah 2
    bunyi=0
    goto exit;
}
buzzer=1;

if(edit==0) // Jika tombol edit ditekan
{
    buzzer=1;

    volume_tampung=tampil_vol;

    awal=2; // Nilai variabel awal adalah 2

    bunyi=0;

    goto exit;

}

delay_ms(200);
```

```

        if(edit==0) // Jika tombol edit ditekan
        {
            buzzer=1;

            volume_tampung=tampil_vol;

            awal=2; // Nilai variabel awal adalah 2

            bunyi=0;

            goto exit;
        }
    }

    if(edit==0) // Jika tombol edit ditekan
    {
        buzzer=1; // Buzzer tidak berbunyi

        if(pilih_edit==0) // Jika kondisi pilih_edit bernilai 1
        {
            if (status==0) // Jika kondisi status bernilai 0
            {

                status=1; // Variabel status bernilai 1

                tampilkan_panah=1;

                goto exit_edit; // Keluar ke kondisi exit_edit

                //ditempatkan

            }

            if (status==1) // Jika kondisi status bernilai 1

```

```

    {
        status=0; // Variabel status bernilai 0

        tampilkan_panah=0; // Variabel tampilkan_panah
                               //bernilai 0

        goto exit_edit; // Keluar ke kondisi exit_edit
                               //ditempatkan
    }
}

exit_edit: // Kondisi exit_edit

if(up==0) // Jika tombol up ditekan
{
    delay_ms(100); // Memanggil delay dari library delay

    if(status==0) // Jika variabel status bernilai 0
    {
        volume+=5; // Volume bertambah 1
    }

    if(status==1) // Jika variabel status bernilai 1
    {
        flowrate+=10; // Flowrate bertambah 1
    }
}
}

```

```

if(down==0) // Jika tombol down ditekan
{
    delay_ms(100); // Memanggil delay dari library delay
    if(status==0) // Jika variabel status bernilai 0
    {
        volume--; // Volume berkurang 1
    }

    if(status==1) // Jika variabel status bernilai 1
    {
        flowrate--; // Flowrate berkurang 1
    }
}

if(enter==0) // Jika tombol enter ditekan
{
    delay_ms(100); // Memanggil delay dari library delay
    goto exit; // Keluar ke kondisi exit ditempatkan
}

if(start==0) // Jika tombol start ditekan
{
    delay_ms(100); // Memanggil delay dari library delay

```



```

if(awal != 0) // Jika nilai variabel awal tidak sama dengan 0
{
    pilih_edit=1; // Variabel pilih_edit bernilai 1
}
else
{
    pilih_edit=1; // Saat motor jalan, siap untuk di pause
    hasil_volume=volume*283; // Rumus untuk perhitungan
                                //counter
}

t=(3600000*volume/flowrate); // Rumus untuk mencari waktu
                                //kemudian dikonversi ke satuan milidetik
ts= t/hasil_volume; // Rumus untuk mencari waktu dalam 1
                                //putaran counter
t_tunda= ts/16; // Rumus untuk mencari waktu tunda dalam
                                //motor stepper

while (data<=hasil_volume) // Selama nilai data kurang dari sama
                                //dengan hasil_volume
{
    // Program untuk menjalankan motor stepper bipolar

    PORTB.3=1;

```

```
PORTD.7=0;  
  
PORTB.2=0;  
  
PORTD.6=0;  
  
delay_ms(t_tunda);
```

```
PORTB.3=0;  
  
PORTD.7=1;  
  
PORTB.2=0;  
  
PORTD.6=0;  
  
delay_ms(t_tunda);
```

```
PORTB.3=0;  
  
PORTD.7=0;  
  
PORTB.2=1;  
  
PORTD.6=0;  
  
delay_ms(t_tunda);
```

```
PORTB.3=0;  
  
PORTD.7=0;  
  
PORTB.2=0;  
  
PORTD.6=1;  
  
delay_ms(t_tunda);  
  
data=TCNT1; // Hasil counter (TCNT1) dipindah ke data
```

```
NEARLY= read_adc(0); // Membaca ADC dari channel 0
```

```
OKLUSI= read_adc(1); // Membaca ADC dari channel 1
```

```
if (NEARLY>=750)
```

```
{
```

```
    buzzer=0; // Buzzer menyala
```

```
    volume_tampung=tampil_vol; // Variabel
```

```
                                //volume_tampung = tampil_vol
```

```
    awal=2; // Nilai variabel awal adalah 2
```

```
    goto exit_stopmotor; // Keluar ke kondisi
```

```
                                //exit_stopmotor ditempatkan
```

```
}
```

```
if (OKLUSI>=510)
```

```
{
```

```
    bunyi=1; // Nilai variabel bunyi adalah 1
```

```
    goto exit; // Keluar ke kondisi exit ditempatkan
```

```
}
```

```
if (data % 212==0 && data>5)
```

```
{
```

```
    tampil_vol--; // Tampilan jumlah volume berkurang 1
```

```
    itoa(tampil_vol,temp); //konversi integer to string
```

```

    lcd_gotoxy(10,0); // Menempatkan tulisan pada posisi
                        //kolom 10 baris 0

    lcd_puts(temp); // Menampilkan string dari integer
                        //variabel tampil_vol

    if (tampil_vol==0)
    {
        volume=0; // Nilai volume habis diinjeksikan
    }

    if(edit==0)
    {
        if(pilih_edit==1)
        {
            volume_tampung=tampil_vol; // Nilai variabel
                        //volume_tampung = tampil_vol

            awal=2; // Nilai variabel awal adalah 2

            goto exit_stopmotor; // Keluar ke kondisi
                        //exit_stopmotor ditempatkan

        }
    }

    if (data>=hasil_volume)

```

```

{
    volume=0; // Nilai volume habis diinjeksikan
    volume_tampung=0; // Nilai volume_tampung habis
                //diinjeksikan
    pilih_edit=0; // Variabel edit bernilai 0
    buzzer=0; // Buzzer berbunyi
    goto exit_stopmotor; // Keluar ke kondisi
                //exit_stopmotor ditempatkan
}
}
    exit_stopmotor: // Kondisi exit_stopmotor
}
    exit: // Kondisi exit
}
};
}
}

```


Lampiran 8

Data Hasil Perhitungan Presisi Variabel *Flow Rate*Tabel Pengujian *Flow Rate* 10 ml/jam

No.	<i>Flow rate</i> (ml/jam)	Volume input (ml)	Waktu pengukuran (menit:detik)	Volume yang dihasilkan (ml)	<i>Flow rate</i> (volume / waktu) (ml/jam)	$(x_i - \bar{x})^2$
1.	10	1	6 : 01,53	1,00	9,9577	$3,844 \times 10^{-5}$
2.	10	1	6 : 01,06	1,00	9,9707	$4,624 \times 10^{-5}$
3.	10	1	6 : 01,40	1,00	9,9613	$6,760 \times 10^{-6}$
4.	10	1	6 : 01,10	1,00	9,9695	$3,136 \times 10^{-5}$
5.	10	1	6 : 01,09	1,00	9,9698	$3,481 \times 10^{-5}$
6.	10	1	6 : 01,60	1,00	9,9558	$6,561 \times 10^{-5}$
7.	10	1	6 : 01,28	1,00	9,9645	$3,600 \times 10^{-7}$
8.	10	1	6 : 01,43	1,00	9,9605	$1,156 \times 10^{-5}$
9.	10	1	6 : 01,38	1,00	9,9618	$4,410 \times 10^{-6}$
10.	10	1	6 : 01,18	1,00	9,9673	$1,156 \times 10^{-5}$
Jumlah					99,6389	$2,511 \times 10^{-4}$

$$\bar{x} = \frac{\sum x_i}{n} = \frac{99,6389}{10} = 9,9639$$

Perhitungan standart deviasi :

$$SD = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}}$$

$$= \sqrt{\frac{2,511 \times 10^{-4}}{10-1}} = \sqrt{2,7901 \times 10^{-5}} = 0,005282$$

Perhitungan koefisien variasi :

$$KV = \frac{SD}{\bar{x}} \times 100\% = \frac{0,005282}{9,9639} \times 100\% = 0,053\%$$

Tabel Pengujian *Flow Rate* 20 ml/jam

No.	<i>Flow rate</i> (ml/jam)	Volume input (ml)	Waktu pengukuran (menit:detik)	Volume yang dihasilkan (ml)	<i>Flow rate</i> (volume / waktu) (ml/jam)	$(x_i - \bar{x})^2$
1.	20	1	3 : 00,12	1,00	19,9868	$9,216 \times 10^{-5}$
2.	20	1	3 : 00,21	1,00	19,9768	$3,842 \times 10^{-4}$
3.	20	1	3 : 00,17	1,00	19,9812	$2,310 \times 10^{-4}$
4.	20	1	3 : 00,29	1,00	19,9677	$8,237 \times 10^{-4}$
5.	20	1	3 : 00,15	1,00	19,9832	$1,742 \times 10^{-4}$
6.	20	1	3 : 00,32	1,00	19,9645	$1,018 \times 10^{-3}$
7.	20	1	3 : 00,17	1,00	19,9812	$2,310 \times 10^{-4}$
8.	20	1	3 : 00,22	1,00	19,9756	$4,326 \times 10^{-4}$
9.	20	1	3 : 00,40	1,00	19,9557	$1,657 \times 10^{-3}$
10.	20	1	3 : 00,08	1,00	19,9912	$2,704 \times 10^{-5}$
Jumlah					199,9639	$5,070 \times 10^{-3}$

$$\bar{x} = \frac{\sum x_i}{n} = \frac{199,9639}{10} = 19,9964$$

Perhitungan standart deviasi :

$$\begin{aligned}
 SD &= \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}} \\
 &= \sqrt{\frac{5,070 \times 10^{-3}}{10-1}} \\
 &= \sqrt{5,6335 \times 10^{-4}} = 0,023735
 \end{aligned}$$

Perhitungan koefisien variasi :

$$\begin{aligned}
 KV &= \frac{SD}{\bar{x}} \times 100\% \\
 &= \frac{0,023735}{19,9964} \times 100\% = 0,119\%
 \end{aligned}$$

Tabel Pengujian *Flow Rate* 30 ml/jam

No.	<i>Flow rate</i> (ml/jam)	Volume input (ml)	Waktu pengukuran (menit:detik)	Volume yang dihasilkan (ml)	<i>Flow rate</i> (volume / waktu) (ml/jam)	$(x_i - \bar{x})^2$
1.	30	1	1 : 59,96	1,00	30,0100	$1,513 \times 10^{-4}$
2.	30	1	1 : 59,83	1,00	30,0426	$4,121 \times 10^{-4}$
3.	30	1	2 : 00,03	1,00	29,9925	$8,880 \times 10^{-4}$
4.	30	1	1 : 59,99	1,00	30,0025	$3,920 \times 10^{-4}$
5.	30	1	1 : 59,92	1,00	30,0200	$5,290 \times 10^{-6}$
6.	30	1	1 : 59,81	1,00	30,0476	$6,401 \times 10^{-4}$
7.	30	1	1 : 59,77	1,00	30,0576	$1,246 \times 10^{-3}$
8.	30	1	2 : 00,03	1,00	29,9925	$8,880 \times 10^{-3}$
9.	30	1	1 : 59,93	1,00	30,0175	$2,304 \times 10^{-5}$
10.	30	1	1 : 59,84	1,00	30,0400	$3,133 \times 10^{-4}$
Jumlah					300,2228	$4,959 \times 10^{-3}$

$$\bar{x} = \frac{\sum x_i}{n} = \frac{300,2228}{10} = 30,0223$$

Perhitungan standart deviasi :

$$\begin{aligned}
 SD &= \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}} \\
 &= \sqrt{\frac{4,959 \times 10^{-3}}{10-1}} \\
 &= \sqrt{5,5103 \times 10^{-4}} = 0,023474
 \end{aligned}$$

Perhitungan koefisien variasi :

$$\begin{aligned}
 KV &= \frac{SD}{\bar{x}} \times 100\% \\
 &= \frac{0,023474}{30,0223} \times 100\% = 0,078\%
 \end{aligned}$$

Tabel Pengujian *Flow Rate* 40 ml/jam

No.	<i>Flow rate</i> (ml/jam)	Volume input (ml)	Waktu pengukuran (menit:detik)	Volume yang dihasilkan (ml)	<i>Flow rate</i> (volume / waktu) (ml/jam)	$(x_i - \bar{x})^2$
1.	40	1	1 : 28,10	1,00	40,8627	$8,575 \times 10^{-3}$
2.	40	1	1 : 27,76	1,00	41,0210	$4,316 \times 10^{-3}$
3.	40	1	1 : 27,84	1,00	40,9836	$8,009 \times 10^{-4}$
4.	40	1	1 : 27,89	1,00	40,9603	$2,500 \times 10^{-5}$
5.	40	1	1 : 27,98	1,00	40,9184	$1,362 \times 10^{-3}$
6.	40	1	1 : 27,83	1,00	40,9883	$1,089 \times 10^{-3}$
7.	40	1	1 : 28,08	1,00	40,8719	$6,956 \times 10^{-3}$
8.	40	1	1 : 27,76	1,00	41,0210	$4,316 \times 10^{-3}$
9.	40	1	1 : 27,83	1,00	40,9883	$1,089 \times 10^{-3}$
10.	40	1	1 : 27,94	1,00	40,9370	$3,345 \times 10^{-4}$
Jumlah					409,5525	$2,886 \times 10^{-2}$

$$\bar{x} = \frac{\sum x_i}{n} = \frac{409,5525}{10} = 40,9553$$

Perhitungan standart deviasi :

$$\begin{aligned}
 SD &= \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}} \\
 &= \sqrt{\frac{2,886 \times 10^{-2}}{10-1}} \\
 &= \sqrt{3,2071 \times 10^{-3}} = 0,056631
 \end{aligned}$$

Perhitungan koefisien variasi :

$$\begin{aligned}
 KV &= \frac{SD}{\bar{x}} \times 100\% \\
 &= \frac{0,056631}{40,9553} \times 100\% = 0,138\%
 \end{aligned}$$

Tabel Pengujian *Flow Rate* 50 ml/jam

No.	<i>Flow rate</i> (ml/jam)	Volume input (ml)	Waktu pengukuran (menit:detik)	Volume yang dihasilkan (ml)	<i>Flow rate</i> (volume / waktu) (ml/jam)	$(x_i - \bar{x})^2$
1.	50	1	1 : 09,89	1,00	51,5095	$1,980 \times 10^{-2}$
2.	50	1	1 : 09,56	1,00	51,7539	$1,075 \times 10^{-2}$
3.	50	1	1 : 09,90	1,00	51,5021	$2,193 \times 10^{-2}$
4.	50	1	1 : 09,51	1,00	51,7911	$1,985 \times 10^{-2}$
5.	50	1	1 : 09,83	1,00	51,5538	$9,293 \times 10^{-3}$
6.	50	1	1 : 09,86	1,00	51,5316	$1,407 \times 10^{-2}$
7.	50	1	1 : 09,47	1,00	51,8209	$2,914 \times 10^{-2}$
8.	50	1	1 : 09,73	1,00	51,6277	$5,063 \times 10^{-4}$
9.	50	1	1 : 09,57	1,00	51,7464	$9,254 \times 10^{-3}$
10.	50	1	1 : 09,68	1,00	51,6647	$2,103 \times 10^{-4}$
Jumlah					516,5017	$1,348 \times 10^{-1}$

$$\bar{x} = \frac{\sum x_i}{n} = \frac{516,5017}{10} = 51,6502$$

Perhitungan standart deviasi :

$$\begin{aligned}
 SD &= \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}} \\
 &= \sqrt{\frac{1,348 \times 10^{-1}}{10-1}} \\
 &= \sqrt{1,4978 \times 10^{-2}} = 0,122386
 \end{aligned}$$

Perhitungan koefisien variasi :

$$\begin{aligned}
 KV &= \frac{SD}{\bar{x}} \times 100\% \\
 &= \frac{0,122386}{51,6502} \times 100\% = 0,237\%
 \end{aligned}$$

Perhitungan rata-rata koefisien variasi untuk variabel *flow rate* :

$$KV = \frac{0,053\% + 0,119\% + 0,078\% + 0,138\% + 0,237\%}{5}$$
$$= 0,125\%$$

Perhitungan nilai presisi untuk variabel *flow rate* :

$$\text{Nilai presisi } flow\ rate = 100\% - 0,125\% = 99,88\%$$



Lampiran 9

Data Tegangan Keluaran *Load Cell* Terhadap Tekanan

Hasil uji berulang tegangan keluaran sensor *load cell* (mV) terhadap tekanan (mmHg) dengan gaya dorong

Uji ke-	Tekanan (mmHg)										
	0	10	20	30	40	50	60	70	80	90	100
1.	0,1 mV	0,3 mV	0,4 mV	0,5 mV	0,6 mV	0,8 mV	0,9 mV	1,0 mV	1,2 mV	1,4 mV	1,6 mV
2.	0,0 mV	0,3 mV	0,5 mV	0,6 mV	0,7 mV	0,8 mV	0,9 mV	1,1 mV	1,2 mV	1,4 mV	1,6 mV
3.	0,0 mV	0,3 mV	0,5 mV	0,6 mV	0,7 mV	0,9 mV	1,0 mV	1,1 mV	1,2 mV	1,4 mV	1,6 mV
4.	0,1 mV	0,4 mV	0,5 mV	0,6 mV	0,8 mV	0,9 mV	1,1 mV	1,2 mV	1,3 mV	1,4 mV	1,6 mV
5.	0,1 mV	0,4 mV	0,6 mV	0,7 mV	0,8 mV	1,0 mV	1,1 mV	1,2 mV	1,3 mV	1,5 mV	1,6 mV
6.	0,0 mV	0,4 mV	0,6 mV	0,7 mV	0,8 mV	1,0 mV	1,1 mV	1,2 mV	1,3 mV	1,5 mV	1,7 mV
7.	0,1 mV	0,4 mV	0,6 mV	0,7 mV	0,9 mV	1,1 mV	1,2 mV	1,2 mV	1,3 mV	1,5 mV	1,6 mV
8.	0,2 mV	0,5 mV	0,6 mV	0,7 mV	0,8 mV	1,0 mV	1,2 mV	1,3 mV	1,3 mV	1,5 mV	1,6 mV
9.	0,0 mV	0,5 mV	0,6 mV	0,7 mV	0,9 mV	1,1 mV	1,2 mV	1,3 mV	1,4 mV	1,5 mV	1,7 mV
10.	0,1 mV	0,5 mV	0,7 mV	0,8 mV	0,9 mV	1,1 mV	1,3 mV	1,3 mV	1,4 mV	1,5 mV	1,7 mV

Hasil berulang tegangan keluaran sensor *load cell* (mV) terhadap tekanan (mmHg) dengan gaya tarik

Uji ke-	Tekanan (mmHg)										
	100	90	80	70	60	50	40	30	20	10	0
1.	-0,1 mV	-0,2 mV	-0,4 mV	-0,5 mV	-0,6 mV	-0,8 mV	-0,8 mV	-1,0 mV	-1,2 mV	-1,3 mV	-1,4 mV
2.	-0,1 mV	-0,3 mV	-0,5 mV	-0,7 mV	-0,9 mV	-1,0 mV	-1,2 mV	-1,3 mV	-1,4 mV	-1,4 mV	-1,5 mV
3.	-0,1 mV	-0,3 mV	-0,5 mV	-0,7 mV	-0,9 mV	-1,0 mV	-1,2 mV	-1,3 mV	-1,4 mV	-1,5 mV	-1,5 mV
4.	-0,1 mV	-0,3 mV	-0,5 mV	-0,7 mV	-0,9 mV	-1,1 mV	-1,2 mV	-1,3 mV	-1,4 mV	-1,4 mV	-1,5 mV
5.	-0,1 mV	-0,3 mV	-0,5 mV	-0,8 mV	-1,0 mV	-1,1 mV	-1,3 mV	-1,3 mV	-1,4 mV	-1,4 mV	-1,5 mV
6.	-0,1 mV	-0,3 mV	-0,5 mV	-0,7 mV	-1,0 mV	-1,2 mV	-1,2 mV	-1,3 mV	-1,4 mV	-1,5 mV	-1,5 mV
7.	-0,1 mV	-0,3 mV	-0,5 mV	-0,7 mV	-0,9 mV	-1,2 mV	-1,2 mV	-1,3 mV	-1,4 mV	-1,5 mV	-1,5 mV
8.	-0,1 mV	-0,4 mV	-0,5 mV	-0,7 mV	-1,0 mV	-1,2 mV	-1,3 mV	-1,3 mV	-1,4 mV	-1,5 mV	-1,5 mV
9.	-0,1 mV	-0,4 mV	-0,6 mV	-0,8 mV	-0,9 mV	-1,1 mV	-1,3 mV	-1,3 mV	-1,4 mV	-1,5 mV	-1,5 mV
10.	-0,1 mV	-0,4 mV	-0,5 mV	-0,7 mV	-1,0 mV	-1,2 mV	-1,3 mV	-1,4 mV	-1,4 mV	-1,5 mV	-1,5 mV

