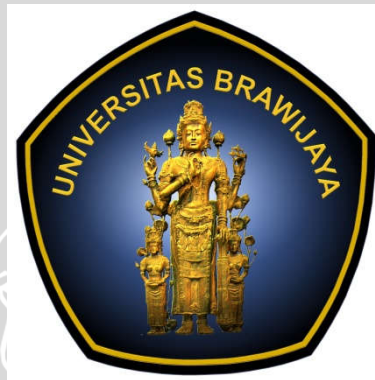


**PENGONTROLAN KECEPATAN MOTOR DC PADA *PERISTALTIC*
PUMP SEBAGAI PENGATUR *FLOWRATE* CAIRAN**

SKRIPSI
TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Ditujukan untuk memenuhi persyaratan
Memperoleh gelar Sarjana Teknik



Disusun oleh:
TAUFIQ NOR AHMAD
NIM. 135060309111014 -63

KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
JURUSAN TEKNIK ELEKTRO
MALANG
2016

LEMBAR PENGESAHAN

PENGONTROLAN KECEPATAN MOTOR DC PADA *PERISTALTIC PUMP* SEBAGAI PENGATUR *FLOWRATE* CAIRAN

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Ditujukan untuk memenuhi persyaratan
Memperoleh gelar Sarjana Teknik



TAUFIQ NOR AHMAD
NIM. 135060309111014 – 63

Skripsi ini telah direvisi dan disetujui oleh dosen pembimbing
Pada tanggal 08 Agustus 2016

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Ir. Bambang Siswoyo., M.T.
NIP. 19630104 198701 1 001

Ir. Purwanto, M.T.
NIP. 19540424 198601 1 001

Mengetahui
Ketua Jurusan/Ketua Program Studi

M. Aziz Muslim, S.T., M.T., Ph.D.
NIP. 19741203 200012 1 001

JUDUL SKRIPSI :

PENGONTROLAN KECEPATAN MOTOR DC PADA *PERISTALTIC PUMP* SEBAGAI
PENGATUR *FLOWRATE* CAIRAN

Nama Mahasiswa : Taufiq Nor Ahmad

NIM : 135060309111014

Program Studi : Teknik Elektro

Minat : Teknik Kontrol

KOMISI PEMBIMBING :

Ketua : Dr. Ir. Bambang Siswoyo., M.T.

Anggota : Ir. Purwanto, M.T.

TIM DOSEN PENGUJI :

Dosen Penguji 1 : Goegoes Dwi N., S.T., M.T.

Dosen Penguji 2 : M. Aziz Muslim, S.T., M.T., Ph.D.

Dosen Penguji 3 : Dipl. Ing. Ir. Moch Rusli

Tanggal Ujian : 04 Agustus 2016

SK Penguji : 962/ UN10.6/ SK/ 2016

PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas dalam naskah Skripsi ini adalah asli dari pemikiran saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi ini dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2, dan pasal 70)

Malang, 15 Juni 2016

Mahasiswa,
Materai Rp. 6000,-

Taufiq Nor Ahmad
135060309111014



UNIVERSITAS BRAWIJAYA

*Teriring Ucapan Terima Kasih kepada:
Ayahanda dan Ibunda tercinta*



UNIVERSITAS BRAWIJAYA



PENGANTAR

Alhamdulillah, segala puji hanya bagi Allah Subhanahu Wa Ta'ala, Rabb alam semesta. Dialah Allah, Tuhan Yang Maha Esa, Yang Maha Pengasih lagi Maha Penyayang. Sebaik-baiknya Penolong, dan Sebaik-baiknya Pelindung. Shalawat serta salam kepada Nabi Muhammad Rasulullah Shallallahu Alaihi Wa Salam, Sang pembawa kabar gembira dan sebaik-baiknya suri tauladan bagi yang mengharap Rahmat dan Hidayah-Nya.

Sungguh hanya melalui Pertolongan dan Perlindungan Allah SWT semata sehingga penulis dapat menyelesaikan skripsi yang berjudul “Pengontrolan Kecepatan Motor DC Pada *Peristaltic Pump* Sebagai Pengatur *Flowrate* Cairan”. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.

Penulis menyadari bahwa penyusunan skripsi ini tidak lepas dari bantuan, bimbingan serta dorongan dari berbagai pihak. Dengan seizin Allah SWT, di kesempatan yang baik ini, dengan segala ketulusan dan kerendahan hati penulis menyampaikan terima kasih kepada :

- Kedua orang tua tercinta, Bapak H. Suryani dan Ibu Hj. Siti Tasriyah yang senantiasa mendoakan, memberikan nasihat, motivasi, perhatian, inspirasi dan kesabaran.
- Bapak M. Aziz Muslim, S.T.,M.T.,Ph.D sebagai Ketua Jurusan Teknik Elektro Universitas Brawijaya.
- Bapak Hadi Suyono, S.T.,M.T.,Ph.D selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya.
- Bapak Ir. Purwanto, M.T. selaku KKDK Teknik Kontrol sekaligus Dosen Pembimbing II atas memberikan banyak waktu, ilmu, masukan, inspirasi dan kesabaran dalam membimbing.
- Bapak Dr. Ir. Bambang Siswojo, M.T. selaku Dosen Pembimbing I atas memberikan banyak waktu, ilmu, masukan, inspirasi dan kesabaran dalam membimbing.
- Bapak Ir. Wijono, M.T., Ph.D selaku Dosen Pembimbing Akademik yang telah meluangkan waktu dalam memberi bimbingan akademik.

- Bapak Ibu Dosen, karyawan, staf recording dan RBTE atas segala bantuan dan kemudahan.
- Saudara-saudara khususnya Ade, Akbar, Andy, Asfari, Bambang, Dimas, Deaz, Ebay, Fatah, Imam, Okky, Praja, Rafi, Ridho, Rio, Surya, dan Wahyu atas rasa kebersamaan, kepedulian, waktu dan tenaga yang diberikan selama penulis menyelesaikan skripsi.
- Teristimewa untuk SAP angkatan 2011, 2012, serta 2013
- Teman Teman dari angkatan 2008, 2010, 2011 dan 2012 Elektro UB yang tidak mungkin dicantumkan satu-persatu, atas segala dukungan, bantuan dan doanya.
- Para Anggota Kost Semanggi Barat 5E
- Teman-teman yang tidak bisa saya sebutkan satu-persatu, yang telah membantu semangat serta doa sehingga penelitian ini bisa diselesaikan.

Pada akhirnya, penulis menyadari bahwa skripsi ini masih belum sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun. Penulis berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi serta bagi masyarakat.

Malang, Juni 2016

Penulis

DAFTAR ISI

PENGANTAR.....	i
DAFTAR ISI	iii
DAFTAR GAMBAR.....	vi
DAFTAR TABEL	vii
RINGKASAN.....	viii
<i>SUMMARY</i>	ix
BAB I.....	1
PENDAHULUAN	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah	2
1.4. Tujuan	3
1.5. Sistematika Penulisan	3
BAB II	5
DASAR TEORI.....	5
2.1. Motor DC.....	5
2.2. Menentukan Parameter Motor DC.....	7
2.3. Sensor Hall Effect.....	9
2.4. Kontroler.....	11
2.4.1. Kontroler Proporsional.....	12
2.4.2. Kontroler <i>Integral</i>	13
2.4.3. Kontroler Differensial.....	14
2.4.4. Kontroler Proporsional <i>Integral</i>	14
2.4.5. Kontroler Proporsional <i>Differensial</i> (PD)	15
2.4.6. Kontroler Proporsional <i>IntegralDifferensial</i> (PID)	15
2.4.7. <i>Hand Tuning</i> Kontroler PID	16
2.5. Arduino Uno	17
2.5.1. Daya.....	18
2.5.2. Memori	18
2.5.3. <i>Input Ouput</i>	18
2.6. Pulse Width Modulation (PWM).....	19



BAB III.....	21
METODOLOGI PENELITIAN.....	21
3.1. Studi Literatur.....	21
3.2. Spesifikasi alat.....	21
3.3. Spesifikasi Kontrol.....	22
3.4. Perancangan dan Realisasi Pembuatan Alat.....	22
3.4.1. Perancangan Perangkat Keras dan Realisasi Pembuatan Alat.....	22
3.4.2. Perancangan dan Perhitungan Komponen yang akan Digunakan.....	22
3.4.3. Perancangan Perangkat Lunak.....	22
3.5. Pengujian Alat.....	23
3.5.1. Pengujian Tiap Blok.....	23
3.5.2. Pengujian Keseluruhan Sistem.....	23
3.6. Pengambilan Kesimpulan.....	23
BAB IV.....	25
PERANCANGAN DAN PEMBUATAN ALAT.....	25
4.1. Diagram Blok Sistem.....	25
4.2. Prinsip Kerja Alat.....	26
4.3. Perancangan Perangkat Keras.....	26
4.3.1. Mekanik Peristaltic Pump.....	26
4.3.2. Spesifikasi <i>Peristaltic Pump</i>	27
4.3.3. Sensor Hall Effect.....	29
4.3.4. Perancangan Driver Motor DC.....	29
4.3.5. Rangkaian Frequency to Voltage.....	30
4.3.6. <i>Liquid Crystal Display</i> (LCD).....	32
4.4. Identifikasi Motor DC.....	33
4.4.1. Menentukan Parameter Motor DC.....	34
4.4.2. Menentukan Fungsi Alih Motor DC.....	35
4.4.3. Penentuan Nilai Penguatan Kontroler.....	36
4.5. Perancangan Perangkat lunak.....	37
BAB V.....	39
PENGUJIAN DAN ANALISIS SISTEM.....	39
5.1. Pengujian Sensor Hall Effect.....	39
5.1.1. Peralatan yang Digunakan.....	39
5.1.2. Langkah Pengujian.....	40

5.1.3.	Hasil Pengujian.....	40
5.2.	Pengujian <i>Driver</i> Motor DC	42
5.2.1.	Langkah pengujian	42
5.2.1	Hasil pengujian.....	43
5.3.	Pengujian Motor DC	44
5.3.1.	Peralatan yang Digunakan.....	45
5.3.2.	Prosedur Pengujian Kecepatan Motor DC	45
5.3.3.	Hasil Pengujian Kecepatan Motor DC	45
5.4.	Simulasi Matlab	46
5.4.1.	Peralatan yang Digunakan.....	46
5.4.2.	Prosedur Percobaan	46
5.4.3.	Hasil Simulasi Matlab	47
5.5.	Pengujian Blok Kontroler	48
5.5.1	Pengujian Tanpa Kontroler.....	48
5.5.2	Pengujian Dengan Kontroler	49
5.6.	Pengujian Keseluruhan Sistem	49
5.6.1	Peralatan yang Digunakan.....	49
5.6.2.	Langkah Percobaan.....	50
5.6.3	Hasil Pengujian.....	50
BAB VI	54
6.1.	Kesimpulan	54
6.2.	Saran	54
DAFTAR PUSTAKA	55
LAMPIRAN I	56
LAMPIRAN II	58
LAMPIRAN III	70

DAFTAR GAMBAR

Gambar 2.1 Cara Kerja Motor DC	6
Gambar 2.3 Diagram Sensor <i>Hall Effect</i>	10
Gambar 2.4 Gelombang Sinyal Kotak Sensor <i>Hall Effect</i>	11
Gambar 2.5. Diagram Blok Kontroler Proporsional	12
Gambar 2.6. Diagram Blok Kontroler <i>Integral</i>	13
Gambar 2.7. Diagram Blok Kontroler <i>Differensial</i>	14
Gambar 2.8. Diagram Blok Kontroler PID	16
Gambar 2.9. Hubungan Fungsi Waktu Antara Sinyal Keluaran dan Sinyal Masukan Kontroler PID	16
Gambar 2.10. <i>Arduino Uno</i> Tampak Depan	18
Gambar 2.11. Gelombang kotak yang memiliki y_{max} , y_{min} dan D	20
Gambar 4.1 Diagram blok sistem	25
Gambar 4.2 Bentuk Keseluruhan <i>Peristaltic Pump</i>	27
Gambar 4.3 Diagram blok pengaturan <i>flow rate</i>	29
Gambar 4.4 Bentuk Sensor Hall Effect	29
Gambar 4.5 Koneksi Modul EMS 30A H-Bridge	30
Gambar 4.8 LCD ke Arduino Uno	32
Gambar 4.9 Rangkaian Wiring LCD ke Arduino Uno	33
Gambar 4.10 Motor DC dengan modul <i>peristaltic pump</i>	33
Gambar 4.11 Program parameter motor DC pada Matlab	35
Gambar 4.12 Fungsi alih dan pole motor DC	35
Gambar 4.13 Respon motor DC dari hasil simulasi Matlab	36
Gambar 4.14 <i>Flowchart</i> Perangkat Lunak	37
Gambar 5.1 Diagram Blok Pengujian Sensor	40
Gambar 5.2 Grafik Perbandingan Keluaran Sensor	41
Gambar 5.3 Diagram Blok Pengujian <i>Driver</i> Motor DC	43
Gambar 5.4 Grafik Hubungan Tegangan dengan <i>PWM</i>	44
Gambar 5.5 Diagram blok pengujian kecepatan motor	44
Gambar 5.6 Grafik hubungan antara tegangan dengan kecepatan	46
Gambar 5.7 Diagram blok simulasi Matlab	47
Gambar 5.8 Grafik simulasi respon sistem menggunakan Matlab	47
Gambar 5.9 Grafik Pengujian tanpa Kontroler dengan <i>Setpoint</i> 4000 RPM	48
Gambar 5.11 Grafik Pengujian dengan Kontroler <i>Setpoint</i> 4000 RPM	49
Gambar 5.12 Grafik Pengujian Sistem dengan Beban Air	51
Gambar 5.13 Grafik Pengujian Sistem dengan Beban Premium	51
Gambar 5.14 Grafik Pengujian Sistem dengan Beban Pelumas Mesin Motor	52
Gambar 5.15 Respon Sistem Terhadap Ketiga Cairan	53

DAFTAR TABEL

Tabel 5.1. Hasil Pengujian Sensor Hall Effect.....	41
Tabel 5.2 Hasil Pengujian <i>Driver</i> Motor DC	43
Tabel 5.3 Data <i>Duty Cycle</i> dengan Tegangan dan Kecepatan Motor.....	45



RINGKASAN

Taufiq Nor Ahmad, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Juni 2016, Pengontrolan Kecepatan Motor DC Pada *Peristaltic Pump* Sebagai Pengatur *Flowrate* Cairan, Dosen Pembimbing: Dr., Ir. Bambang Siswojo MT. dan Ir. Purwanto MT.

Pengaturan *flowrate* cairan pada *peristaltic pump* yang dilakukan dalam penelitian ini bertujuan untuk merancang dan membuat sistem kontrol yang diterapkan pada *peristaltic pump* supaya dapat menjaga kecepatan motor DC tetap konstan saat memompa beberapa macam cairan, apabila saat diberi beban cairan dengan tingkat kekentalan yang berbeda, pompa dapat memompa dengan kecepatan sama, sehingga lama waktu memompa tiap jenis cairan relatif sama, serta nantinya diharapkan dapat mempermudah pengaturan *flowrate* cairan.

Pada penelitian ini, 3 jenis cairan dipilih dari sekian banyak jenis cairan yang ada, dikarenakan sudah mewakili 3 tingkat kekentalan berbeda. Cairan tersebut yaitu air mineral mewakili jenis cairan dengan kekentalan paling rendah, premium mewakili jenis cairan dengan tingkat kekentalan sedang, dan pelumas mesin motor mewakili jenis cairan paling kental. Hasil Identifikasi sistem dengan menggunakan MATLAB 2016a didapatkan fungsi alih sistem
$$\frac{3.641}{0.003602 s^2 + 0.6827 s + 15.83}$$

Menggunakan Arduino Uno sebagai pengontrol utama, kemudian sensor *hall effect* sebagai sensor kecepatan yang diberi kontroler berupa PID. Untuk mencari parameter PID menggunakan metode *hand tuning*.

Dari hasil penelitian respon sistem untuk mencapai *steady* tidak melebihi 1 detik. nilai parameter $K_p=85$, $K_i=7$, $K_d=4$, dan Ess (*error steady state*)% untuk beban air mineral sebesar 0,1%, untuk beban bahan baku premium sebesar 0,13% untuk beban pelumas mesin motor sebesar 0,53%. Selain itu juga didapat persen M_p (*maximum overshoot*) untuk beban air mineral sebesar 4,9% untuk beban premium sebesar 5,15% untuk beban pelumas mesin motor sebesar 5,75%.

Kata Kunci : *Peristaltic pump*, Kontroler PID, Arduino Uno, Motor DC.

SUMMARY

Taufiq Nor AhmadDepartement of Electrical Engineering, Faculty of Engineering, Universitas Brawijaya, Juni 2016, **Controlling DC Motor Speed In peristaltic Pump As the Liquid Flow Rate Regulator**, Supervisor: Dr., Ir. Bambang Siswojo MT. dan Ir. Purwanto MT.

Flowrate setting of fluid in the peristaltic pump performed in this research is to design and create a control system can applied to peristaltic pump in order to keep the DC motor speed remains constant when pumping some sort of liquid, to load fluid with different viscosity level, the pump can pumping at the same speed, so the length of time each type of fluid pumping relatively similar, and is expected to facilitate fluid flowrate settings. In this research types of fluids selected from the many types of fluids because it represents three different viscosity levels. The liquid is mineral water represents most types of liquids with a low viscosity, premium represents the type of fluid with medium viscosity level, and lubricating the engine represents the most viscous type of liquid. Results Identification system by using MATLAB 2016a obtained 3,641 system transfer function
$$\frac{3.641}{0.003602 s^2 + 0.6827 s + 15.83}$$

Using Arduino Uno as the main controller, then hall effect sensor as speed sensor given PID controller. To search for PID parameter using hand tuning method. The result of research shows that to achieve steady state response didn't over 1 second. parameter values $K_p = 85$, $K_i = 7$, $K_d = 4$, and percentage E_{ss} (error steady state) for mineral water is 0.1%, for a premium is 0.13% for motorcycle engine lubricant is 0, 53%. It also obtained percent M_p (maximum overshoot) for mineral water load of 4.9% for premium of 5.15% for motorcycle engine lubricant load of 5.75%.

Keyword: Peristaltic Pump, PID Controller, Arduino Uno, DC Motor.



BAB I

PENDAHULUAN

1.1. Latar Belakang

Pompa pada awalnya diwujudkan sebagai kincir air atau kincir angin untuk memompa dan mengalirkan air. Namun seiring perkembangan yang ada, definisi dari pompa tidak luput ikut mengalami perkembangan. Sekarang, pompa dapat diartikan dengan mesin atau seperangkat alat penggerak untuk memindahkan cairan, gas, maupun lumpur dari satu tempat ke tempat lain dengan tindakan mekanis. Kebutuhan akan pompa untuk saat ini sudah sangat beragam, sehingga muncul bermacam-macam karakteristik pompa yang mempunyai banyak bentuk dan ukuran. Salah satu penggunaan pompa adalah untuk titrasi pada berbagai bidang yang berhubungan dengan pengukuran *flow rate* cairan, seperti pada bidang pertanian, lab kimia, maupun bidang kedokteran.

Keakuratan pengukuran, sterilisasi, serta pengaturan *flow rate* cairan menjadi permasalahan tersendiri, sehingga membutuhkan karakteristik pompa yang mampu memberikan solusi dari permasalahan tersebut. *Peristaltic Pump* merupakan salah satu pompa dengan karakteristik yang dibutuhkan, dimana jenis pompa ini mempunyai keunggulan dalam hal pengaturan kecepatan serta keakuratan volume yang diinginkan. Gerakan *peristaltic* juga merupakan kelebihan tersendiri dari pompa jenis ini. Gerak *peristaltic* merupakan gerak seperti pada saluran pencernaan, terjadi gaya menarik, mengalirkan dan menahan cairan. Kebanyakan dimensi dari pompa ini yang tidak terlalu besar sehingga lebih fleksibel digunakan dari satu tempat ke tempat lain.

Pada penulisan ini akan dibahas bagaimana merancang pengaturan kecepatan motor DC pada *Peristaltic Pump*. Penulisan kali ini penulis menggunakan *Peristaltic Pump* yang terdiri dari berberapa bagian utama yaitu, motor, tabung fleksibel, dan mekanikal. Jenis motor yang digunakan dalam pompa ini menggunakan jenis motor DC, sehingga dapat melakukan pengaturan serta pengontrolan kecepatan pada motor tersebut. Bagian tabung fleksibel pada *peristaltic pump* selain sebagai tempat mengalirkan cairan, tabung fleksibel juga dapat berfungsi untuk melindungi agar cairan tetap *steril*.

Mekanikal pada *peristaltic pump* merupakan pendukung dari kinerja motor, bertindak untuk menekan tabung fleksibel, sehingga putaran yang terjadi akan menimbulkan gerakan *peristaltic*. Pada penerapan kinerja pompa dengan berbagai jenis macam cairan, faktor utama adalah pengaturan kecepatan motor DC, sehingga waktu dan *flow rate* yang diinginkan sesuai dengan *set point*. Maka dari itulah dibutuhkan pengaturan motor DC secara tepat, agar dapat menjaga *flow rate* tetap konstan saat digunakan dengan berbagai jenis cairan.

Penggunaan sensor *hall effect* beserta kontroler PID (Proporsional Integral Diferensial) menawarkan solusi dalam mengatasi permasalahan pengaturan kecepatan di atas. Saat pompa bekerja dengan jenis cairan yang berbeda, sehingga dapat menyebabkan kecepatan motor DC pada pompa tidak konstan. Kecepatan motor DC tersebut akan dikendalikan secara otomatis menggunakan metode PID setelah mendapatkan *feedback* dari sensor *hall effect*. Karena keunggulan dari kontroler PID adalah memiliki respon halus dan cepat, serta merupakan kontroler berumpan balik yang sederhana dan dapat memberikan performa kontrol yang optimal.

1.2. Rumusan Masalah

Berdasarkan permasalahan yang telah diuraikan pada latar belakang di atas, dapat disusun rumusan masalah sebagai berikut :

- 1) Bagaimana merancang serta membuat sistem pengendalian *Flow Rate* pada *Peristaltic Pump*.
- 2) Bagaimana merancang dan membuat *hardware* dan *software* sistem pengendalian *Flow Rate* pada *Peristaltic Pump*.

1.3. Batasan Masalah

Dengan mengacu pada permasalahan yang telah dirumuskan, maka hal-hal yang berkaitan dengan alat akan diberi batasan sebagai berikut :

- 1) *Arduino Uno* berbasis mikrokontroler ATmega 328 sebagai pusat pengendali sistem.
- 2) Pembahasan akan ditekankan pada unjuk kerja sensor, pengukuran nilai *error*, serta *output*. Kinerja *driver* dan rangkaian elektronika tidak dibahas mendalam.

- 3) Pembahasan ditekankan pada penggunaan kontroler PID pada sistem.
- 4) Tegangan motor DC 12V sedangkan kecepatan motor DC diatur pada 4000 RPM.
- 5) Sensor yang digunakan adalah sensor *hall effect*.
- 6) Cairan yang dipergunakan untuk pengujian berupa Air Mineral, Bahan Bakar Premium, dan Pelumas Mesin Motor.

1.4. Tujuan

Tujuan dari penelitian ini adalah merancang agar *output* pada *peristaltic pump* tetap konstan terhadap perubahan jenis cairan menggunakan kontroler PID (Proporsional Integral Diferensial).

1.5. Sistematika Penulisan

Agar penyusunan laporan skripsi ini dapat mencapai sasaran dan tidak menyimpang dari judul yang telah ditentukan, maka diperlukan sistematika pembahasan yang jelas. Pembahasan dalam skripsi ini secara garis besar sistematika penulisan dalam skripsi ini adalah sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan, dan sistematika penulisan.

BAB II Tinjauan Pustaka

Membahas teori-teori yang mendukung dalam perancangan dan pembuatan alat.

BAB III Metodologi Penelitian

Berisi tentang metode-metode yang dipakai dalam melakukan perancangan, pengujian dan analisis data.

BAB IV Perancangan dan Pembuatan Alat

Perancangan dan perealisasiian alat yang meliputi spesifikasi, perencanaan diagram blok, prinsip kerja, dan realisasi alat.

BAB V Pengujian dan Analisis

Memuat aspek pengujian meliputi penjelasan tentang cara pengujian dan hasil pengujian. Aspek analisis meliputi penilaian atau komentar terhadap hasil-hasil pengujian. Pengujian dan analisis ini terhadap alat yang telah direalisasikan berdasarkan masing-masing blok dan sistem secara keseluruhan.

BAB VI Kesimpulan dan Saran

Memuat intisari hasil pengujian dan menjawab rumusan masalah serta memberikan rekomendasi untuk perbaikan kualitas penelitian di masa akan datang.



BAB II

DASAR TEORI

Agar dapat mempermudah dalam pemahaman cara kerja rangkaian ataupun dasar perancangan alat pada judul skripsi pengontrolan motor DC pada *peristaltic pump* sebagai pengatur *flow rate* cairan ini. Maka, diperlukan penjelasan dan uraian teori penunjang yang digunakan dalam penulisan kali ini. Teori penunjang yang akan dijelaskan dalam bab ini adalah sebagai berikut:

- ✓ Motor DC
- ✓ Sensor *Hall Effect*
- ✓ Kontroler PID (Proporsional Integral Diferensial)
- ✓ *Arduino Uno*
- ✓ PWM (*Pulse Width Modulation*)

2.1. Motor DC

Prinsip kerja motor DC sesuai dengan hukum kemagnetan Lorentz, yaitu membangkitkan fungsi magnet pada suatu konduktor berarus dalam medan magnet sehingga timbul ggl induksi. Setiap arus yang mengalir melalui sebuah konduktor akan menimbulkan medan magnet. Arah medan magnet dapat ditentukan dengan kaidah tangan kiri. Kaidah tangan kiri untuk motor menunjukkan arah arus yang mengalir didalam sebuah konduktor yang berada dalam medan magnet. Jari tengah menunjukkan arah arus yang mengalir pada konduktor, jari telunjuk menunjukkan arah medan magnet dan ibu jari menunjukkan arah medan putar. Adapun besarnya gaya (f) yang bekerja pada konduktor tersebut dapat dirumuskan dengan (Soemarwanto,1999):

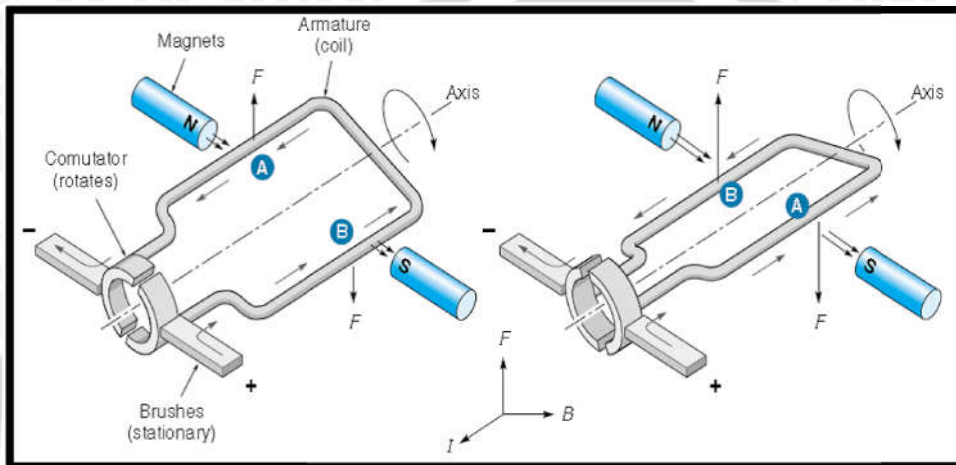
$$F = B \cdot I \cdot L \text{ (Newton)}$$

Dimana :

B = kerapatan fluks magnet (weber)

L = panjang konduktor (meter)

I = arus listrik (ampere)



Gambar 2.1 Cara Kerja Motor DC
Sumber : Kilian, 1996

Gambar 2.1 adalah ilustrasi cara kerja motor DC yang mempunyai satu lilit kawat a dan b berada didalam medan magnet. Lilitan ini dapat berputar dengan bebas, lilitan ini biasa disebut dengan jangkar (*armature*).

Pada jangkar diberikan arus yang berasal dari sumber yang terhubung dengan sikat (*brushes*). Sikat-sikat ini terpasang pada sebuah cincin yang terbelah dua, yang disebut cincin belah (*commutator*). Adapun tujuan dari konstruksi ini adalah agar lilitan kawat dapat berputar apabila ada arus listrik yang melewatinya.

Pada kawat yang berada dikanan arus mengalir dari depan kebelakang. Pada kawat yang berada di bagian kiri, arus mengalir dari belakang kedepan kawat a dan b secara bergantian berada di kiri dan kanan. Karena itu arah arus di a dan arah arus di b selalu bersifat bolak-balik. Pembalikan arah arus itu terjadi pada saat lilitan kawat melintasi posisi vertikal.

Bagian *commutator* berfungsi sebagai penyearah mekanik. Fluksi magnet yang ditimbulkan magnet permanen disebut medan magnet motor. Dalam gambar 2.1 arah *fluks*

magnetic adalah dari kiri ke kanan. Adapun gaya yang bekerja pada penghantar b adalah ke atas, sementara gaya yang bekerja pada penghantar a adalah ke bawah. Gaya-gaya yang bekerja sama kuatnya, sehingga terdapat kopel yang bekerja pada kawat sehingga lilitan jangkar dapat berputar. Setelah berputar 180° arah arus berbalik, pada saat itu penghantar a dan b bertukar tempat. Akibatnya arah gerak putaran tidak berubah.

2.2. Menentukan Parameter Motor DC

Untuk menentukan fungsi alih dari motor DC metode yang digunakan adalah pemodelan fisik, metode pemodelan fisik adalah identifikasi paramater fisik dan elektrik pada Motor DC. Parameter yang digunakan dalam penentuan fungsi alih adalah resistansi, induktansi, konstanta motor, koefisien vricsi dan Momen inersia.

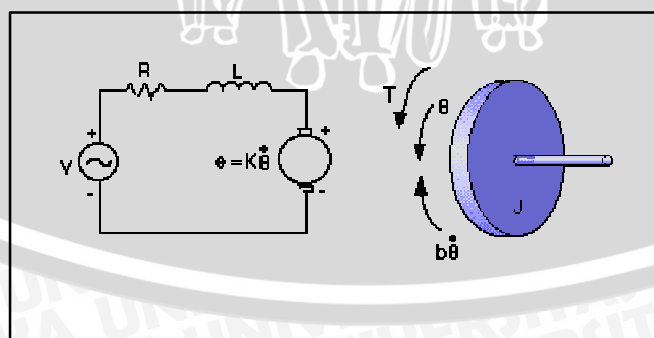
Karakteristik motor DC adalah nilai torsi sebanding dengan arus jangkari (*i*) dengan faktor konstanta (*K_t*), seperti yang ditunjukkan pada persamaan 2.1

$$T = K_T i \dots\dots\dots (2.1)$$

Back-emf (*e*) adalah jika sebuah konduktor listrik memotong garis medan magnet maka timbul gaya gerak listrik pada konduktor.

$$e = K_e \dots\dots\dots (2.2)$$

Dalam SI unit, torsi motor dengan *back-emf* adalah sebanding maka nilai dari *K_t* = *K_e*, oleh karena itu untuk mempermudah dapat digunakan *K* untuk mewakili dari nilai *K_t* dan *K_e*.



Gambar 2.2 Rangkaian motor DC

Dari gambar 2.2 maka dapat diuraikan menggunakan hukum Newton II dan Hukum tegangan Kirchoff.

$$\Sigma F = ma$$

$$K_i \quad b = J$$

$$J + b = K_i \dots\dots\dots (2.3)$$

$$\Sigma V = 0$$

$$L \frac{di}{dt} + R_i + K \quad V = 0$$

$$L \frac{di}{dt} + R_i = V \quad K \dots\dots\dots (2.4)$$

Dari persamaan (2.3) dan (2.4) dapat diturunkan menggunakan transformasi laplace maka didapatkan persamaan:

$$s(Js + b)I(s) = KI(s) \dots\dots\dots (2.5)$$

$$(Ls + R)I(s) = V(s) \quad Ks(s) \dots\dots\dots (2.6)$$

Untuk mendapatkan fungsi alih maka $I(s)$ dihilangkan pada persamaan (2.5) dan (2.6), dimana kecepatan rotasi dianggap sebagai keluaran dan masukan dianggap sebagai masukan.

$$P(s) = \frac{(s)}{V(s)} = \frac{K \quad rad/s}{(Js+b)(Ls+R)+K^2} \frac{1}{V} \dots\dots\dots (2.7)$$

Apabila persamaan fungsi alih sudah didapatkan, untuk nilai resistansi (R) dan nilai induktansi (L) dapat dicari dengan menggunakan alat ukur. Selanjutnya untuk mencari nilai dari *back-emf* didapatkan dengan persamaan:

$$e = V \quad IR \dots\dots\dots (2.8)$$

Untuk mencari nilai dari K didapatkan dengan mengukur kecepatan motor ω (dalam *rad/s*) pada saat diberi sumber tegangan (V) dari persamaan (2.2) dan (2.8). Sehingga didapatkan persamaan:

$$K = \frac{e}{\omega} \dots\dots\dots (2.9)$$

Selanjutnya untuk mengestimasi nilai koefisien dari nilai friksi (b) motor menggunakan persamaan (2.3). Motor dinyalakan hingga mencapai kecepatan konstan sehingga $\frac{d\theta}{dt} = 0$ atau bisa diasumsikan bahwa nilai percepatannya adalah 0.

$$b = \frac{Ki}{\omega} \dots\dots\dots (2.10)$$

Yang terakhir adalah menentukan nilai dari momen inersia dari motor. Motor dinyalakan sampai mencapai kecepatan yang konstan kemudian tidak diberi sumber tegangan atau $V=0$, sehingga nilai arus sama dengan 0.

$$J + b = 0 \dots\dots\dots (2.11)$$

Pada persamaan (4.11) diperoleh dari dengan cara pada saat kecepatan *steady state* sebagai awal kecepatan dalam persamaan.

$$\omega = \frac{Te}{B} e^{-\left(\frac{B}{\tau}\right)t} \dots\dots\dots (2.12)$$

Ketika nilai (t) sama dengan $\tau = \frac{J}{B}$ adalah waktu konstan pada saat di beri sumber tegangan (V). Kemudian (V) dilepas hingga kondisi motor berhenti. Karena nilai b sudah diketahui dari persamaan (2.10), maka akan didapatkan nilai momen inersia dari motor DC.

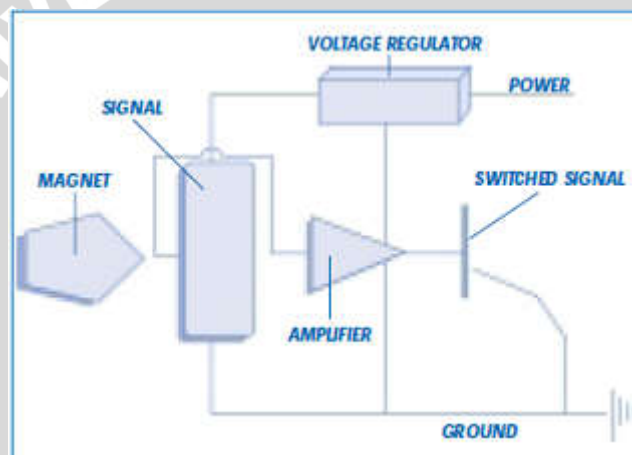
2.3. Sensor Hall Effect

Sensor hall effect yang digunakan oleh banyak distributor sistem pengapian untuk memicu sirkuit primer (*fire the coil*) dan memantau mesin RPM.

Sensor hall effect sering dikenal dengan sebutan "switch" daripada sensor tegangan, karena sinyal yang dihasilkan adalah sinyal digital (on-off). Tidak seperti sensor magnetik yang menghasilkan arus bolak-balik (AC) sinyal tegangan yang bervariasi dengan kecepatan, *Sensor hall effect* menghasilkan sinyal tegangan konstan yang bisa berubah tiba-tiba dari tegangan maksimum hampir nol dan kembali lagi terlepas dari kecepatan motor DC. Hal ini menghasilkan gelombang sinyal keluaran kotak yang mudah digunakan oleh komputer untuk tujuan pewaktuan.

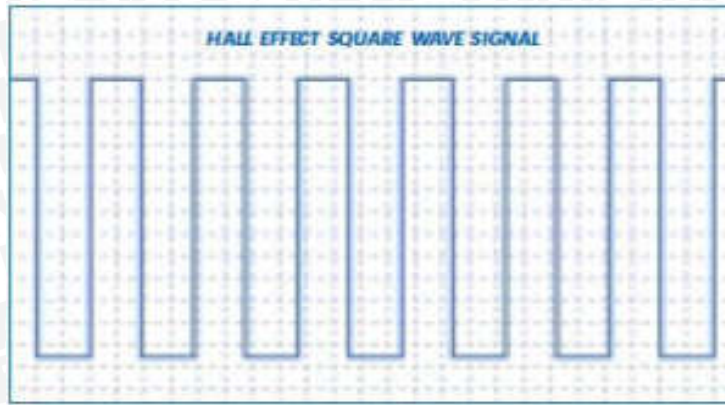
Tipikal *Sensor hall effect* memiliki tiga kabel atau terminal yaitu *ground*, suplai atau tegangan referensi, dan sinyal *output*. Untuk menghasilkan sinyal *output*,

Sensor hall effect harus dipasok dengan tegangan referensi dari sumber (yang mungkin 5 sampai 12 volt tergantung pada aplikasi). Tegangan suplai diperlukan untuk menciptakan efek beralih yang berlangsung di dalam sensor. Prinsip kerja pada *Sensor hall effect* bermula pada tahun 1879 ketika Edwin H. Hall, seorang ilmuwan Amerika, menemukan sebuah fenomena listrik baru. Ketika saya menerapkan arus listrik untuk sepotong logam dimasukkan antara dua magnet, menciptakan tegangan sekunder dalam logam di sudut kanan dengan tegangan yang diberikan. Penemuan ini tidak memiliki banyak penggunaan praktis pada saat itu, tetapi dimasa depan ternyata insinyur akan perlu untuk membuat perangkat *switching* yang mampu menghasilkan sinyal tegangan kotak yang efisien. *Hall Effect* diadaptasikan sehingga tegangan akan berubah ketika sebuah chip silikon ditempatkan di sudut kanan medan magnet.



Gambar 2.3 Diagram Sensor *Hall Effect*
Sumber : Krisnanda, 2014

Ketika pisau logam melewati medan magnet celah udara dan diantara *chip* silikon, blok medan magnet dan menyebabkan tegangan output chip turun menjadi nol secara tiba-tiba. Dengan sirkuit tambahan, sensor dapat dibuat untuk melakukan hal yang berlawanan yaitu untuk menghasilkan sinyal tegangan ketika pisau Memasuki medan magnet. Sirkuit tambahan mengatur tegangan suplai ke *chip* dan menguatkan tegangan output. Dalam system pengapian otomotif, pisau rana (*shutter blade*) yang dipasang pada poros distributor, rotor, crankshaft pulley, atau camshaft sehingga sensor dapat menghasilkan memicu dan/ atau sinyal posisi sebagaimana crankshaft berputar. Dalam beberapa aplikasi, lekukan dalam katrol, gigi-gigi atau bahkan tumbol magnet berputar bertujuan yang sama seperti pisau rana untuk mengganggu jendela magnet sensor dan saklar.



Gambar 2.4 Gelombang Sinyal Kotak Sensor *Hall Effect*
Sumber : Krisnanda, 2014

2.4. Kontroler

Keberadaan kontroler pada sebuah sistem kontrol mempunyai kontribusi yang besar perilaku suatu sistem. Hal itu disebabkan oleh karena tidak dapat diubahnya suatu komponen penyusun pada sistem rangkaian tersebut. Ini dapat diartikan, karakteristik plant harus diterima apa adanya, sehingga perubahan perilaku suatu sistem hanya dapat dilakukan oleh suatu sub sistem, yaitu kontroler.

Salah satu tugas dari komponen disuatu kontroler adalah mengurangi sinyal kesalahan (*error*), yaitu perbedaan antara sinyal *input* dan *output*. Hal ini sesuai dengan tujuan sistem kontrol yaitu mendapatkan sinyal *output* yang sesuai dengan sinyal *input*. Semakin cepat reaksi sistem mengikuti sinyal *output* dan semakin kecil *error* yang terjadi, maka semakin baik kinerja sistem kontrol yang diterapkan.

Apabila terjadi perbedaan yang cukup signifikan antara sinyal input dengan sinyal output relative cukup besar, maka kontroler yang baik seharusnya mampu menganalisis perbedaan yang terjadi, hal ini dapat dikatakan kinerja sistem sesuai dengan yang diharapkan.

1. Pengendalian dengan *loop* terbuka

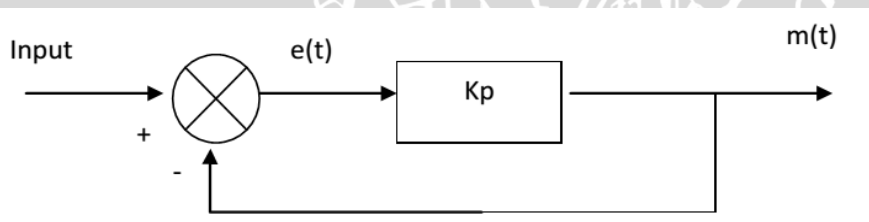
Sistem kontrol *loop* terbuka adalah sistem kontrol yang keluarannya tidak berpengaruh pada aksi pengontrolan. Jadi pada sistem kontrol *loop* terbuka, keluaran tidak diukur atau diumpan balik untuk dibandingkan dengan masukan.

2. Pengendalian dengan *loop* tertutup

Sistem kontrol *loop* tertutup adalah sistem kontrol yang keluarannya mempunyai pengaruh langsung pada aksi pengontrolan. Disebut juga sistem kontrol yang menggunakan umpan balik untuk memperkecil kesalahan sistem.

2.4.1. Kontroler Proporsional

Pengontrol proporsional memiliki keluaran yang sebanding atau proporsional dengan besarnya sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya). Secara lebih sederhana dapat dikatakan bahwa keluaran pengontrol proporsional merupakan perkalian antara konstanta proporsional dengan masukannya. Perubahan pada sinyal masukan akan segera menyebabkan sistem secara langsung mengeluarkan *output* sinyal sebesar konstanta pengalinya. Gambar 2.5 menunjukkan blok diagram yang menggambarkan hubungan antara *input* (besaran referensi yang diinginkan), besaran *actual* dengan besaran keluaran kontroler proporsional, dan besaran kesalahan (*error*). Sinyal kesalahan (*error*) merupakan selisih antara besaran setting dengan besaran aktualnya.



Gambar 2.5. Diagram Blok Kontroler Proporsional
Sumber: Ogata, K., 1995:058

Pada pengendali proporsional hubungan antara keluaran dan kontroler $m(t)$ dan sinyal kesalahan $e(t)$ adalah:

$$m(t) = K_p \times e(t) \dots\dots\dots (2.13)$$

Dengan K_p adalah penguatan proporsional. Keluaran $m(t)$ hanya tergantung pada K_p dan *error*, Ciri-ciri pengontrol proporsional harus diperhatikan ketika pengontrol tersebut diterapkan pada suatu sistem. Secara eksperimen, pengguna pengontrol proporsional harus memperhatikan ketentuan-ketentuan berikut ini:

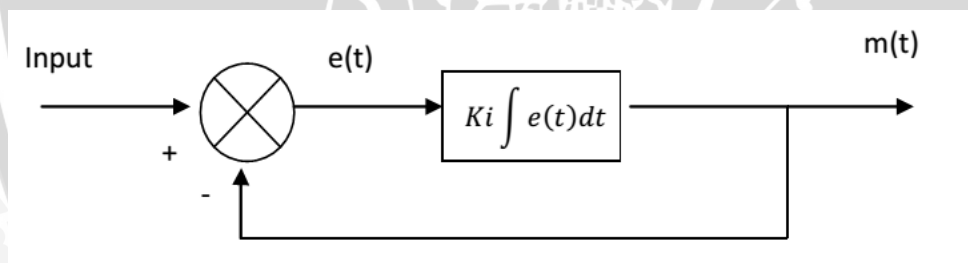
1. kalau nilai K_p kecil, pengontrol proporsional hanya mampu melakukan koreksi kesalahan yang kecil, sehingga akan menghasilkan respon sistem yang lambat.

2. kalau nilai K_p dinaikan, respon sistem menunjukkan semakin cepat mencapai set point dan keadaan stabil.
3. namun jika nilai K_p diperbesar sehingga mencapai harga yang berlebihan, akan mengakibatkan sistem bekerja tidak stabil, atau respon sistem akan berisolasi.

2.4.2. Kontroler *Integral*

Kontroler *integral* berfungsi mengurangi kesalahan keadaan mantap yang dihasilkan pada kontroler proporsional sebelumnya. Kalau sebuah plant tidak memiliki unsure integrator ($1/s$), kontroler proporsional tidak akan mampu menjamin keluaran sistem dengan kesalahan keadaan mantapnya nol.

Kontroler *integral* memiliki karakteristik seperti halnya sebuah *integral*. Keluaran kontroler sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal kesalahan. Keluaran kontroler ini merupakan jumlahan yang terus menerus dari perubahan masukannya. Kalau sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Gambar 2.6 menunjukkan blok diagram kontroler *integral*.



Gambar 2.6. Diagram Blok Kontroler *Integral*.
Sumber: Ogata, K., 1995:058

Nilai keluaran kontroler $m(t)$ sebanding dengan *integral* sinyal kesalahan $e(t)$, sehingga:

$$\frac{dm(t)}{dt} = K_i \cdot e(t) \dots\dots\dots (2.14)$$

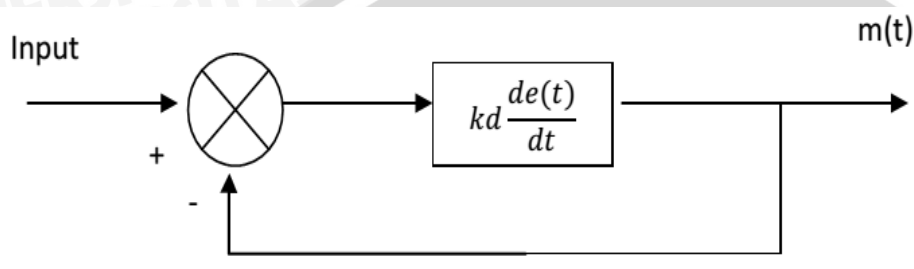
$$m(i) = K_i \int_0^t e(t) dt \dots\dots\dots (2.15)$$

Dengan K_i adalah konstanta *integral*. Jika sinyal kesalahan $e(t) = 0$, maka laju perubahan sinyal kendali *integral* $\frac{dm(t)}{dt} = 0$ atau sinyal keluaran kendalai akan tetap berada pada nilai yang dicapai sebelumnya. Aksi kontrol *integral* digunakan untuk menghilangkan

kesalahan posisi dalam keadaan mantap (*errorsteady state*) tanpa memperhitungkan kecepatan respon.

2.4.3. Kontroler Differensial

Kontroler *differensial* memiliki sifat seperti halnya suatu operasi derivative. Perubahan yang mendadak pada masukan kontroler, akan mengakibatkan perubahan yang sangat besar dan cepat. Gambar 2.7 menunjukkan blok diagram pada kontroler *differensial*.



Gambar 2.7.Diagram Blok Kontroler *Differensial*
Sumber : Ogata 1995:058.

Nilai keluaran kontroler $m(t)$ sebanding laju sinyal kesalahan $\frac{de(t)}{dt}$. Hubungan ini dapat ditulis sebagai:

$$m(t) = Kd \frac{de(t)}{dt} \dots\dots\dots (2.16)$$

Kontroler *differensial* akan memberikan sinyal kendali keluaran $m(t) = 0$, untuk sinyal kesalahan $e(t)$ yang konstan sehingga kontroler *differensial* tidak mempengaruhi keadaan mantap. Kontroler *differensial* digunakan untuk memperbaiki atau mempercepat respon *transien* sebuah sistem serta dapat meredam osilasi.

Berdasarkan karakteristik kontroler tersebut, kontroler *differensial* umumnya dipakai untuk mempercepat respon awal suatu sistem, tetapi tidak memperkecil kesalahan pada keadaan stabilnya. Kerja kontroler differensial hanyalah efek dari lingkup yang sempit, yaitu periode peralihan. Oleh sebab itu kontroler *differensial* tidak bisa digunakan tanpa ada kontroler lain

2.4.4. Kontroler Proporsional *Integral*

Aksi kontrolnya dinyatakan dalam persamaan:

$$m(t) = K_p \cdot e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \dots \dots \dots (2.17)$$

Kontroler ini menghasilkan sinyal kesalahan $\int e(t) dt$ kemudian ditambahkan dengan sinyal $e(t)$.

2.4.5. Kontroler Proporsional *Differensial* (PD)

Aksi kontrolnya dinyatakan dalam persamaan:

$$m(t) = K_p \cdot e(t) + K_p \cdot T_d \frac{de(t)}{dt} \dots \dots \dots (2.18)$$

Kontroler PD selalu mengukur kemiringan (slope) sinyal kesalahan $\frac{de(t)}{dt}$ dan memperkirakan akan besarnya *overshoot* yang akan terjadi serta memberikan koreksi sebelum terjadi lewatannya sehingga diperoleh maksimum *overshoot* yang kecil.

Jika kesalahan keadaan mantap tidak berubah terhadap waktu maka turunannya terhadap waktu sama dengan nol, sehingga kontroler PD tidak mempunyai pengaruh terhadap kesalahan keadaan mantap, tetapi jika terdapat perubahan kesalahan, kontroler PD digunakan untuk memperbaiki suatu sistem pengendalian yang tanggapan peralihannya mempunyai maksimum *overshoot* yang berlebihan tanpa memperhitungkan kecepatan responnya.

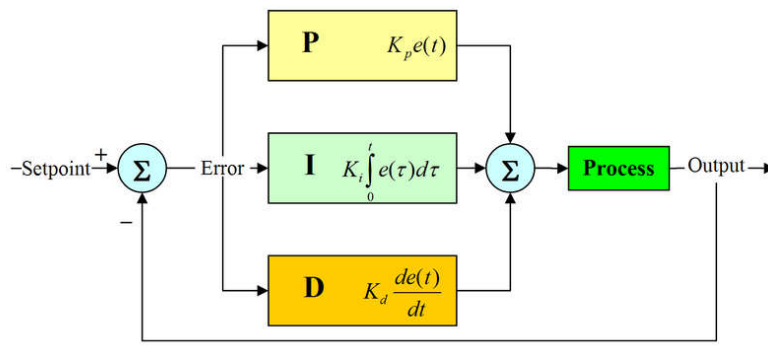
2.4.6. Kontroler Proporsional *IntegralDifferensial* (PID)

Setiap kekurangan dan kelebihan dari masing-masing kontroler P, I dan D dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi kontroler proporsional *integraldifferensial* (PID). Elemen-elemen kontroler P, I dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan *offset* dan menghasilkan perubahan awal yang besar (Gunterus, 1994, 8-10). Kontroler PID memiliki diagram kendali seperti yang ditunjukkan dalam gambar 2.8.

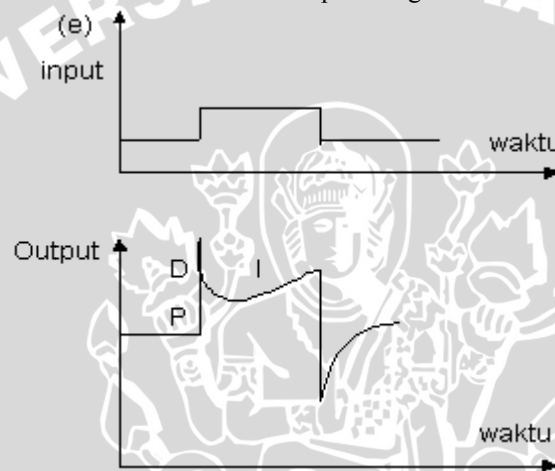
Aksi kontrolnya dinyatakan sebagai berikut:

$$m(t) = K_p \cdot e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p \cdot T_d \frac{de(t)}{dt} \dots \dots \dots (2.19)$$

Jenis kontroler ini digunakan untuk memperbaiki kecepatan respon, mencegah terjadinya kesalahan keadaan mantap serta mempertahankan kestabilan.



Gambar 2.8. Diagram Blok Kontroler PID
Sumber : Wikipedia.org



Gambar 2.9. Hubungan Fungsi Waktu Antara Sinyal Keluaran dan Sinyal Masukan Kontroler PID
Sumber : (Gunterus, 1994, 8-11).

Keluaran kontroler PID merupakan penjumlahan dari keluaran kontroler proporsional, *integral* dan *differensial*. Gambar 2.9 menunjukkan hubungan tersebut. karakteristik kontroler PID sangat dipengaruhi oleh kontribusi besar dari ketiga parameter P, I dan D. penyetelan konstanta K_p , T_i dan T_d akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat diatur lebih menonjol dibanding yang lain. Konstanta yang menonjol itulah akan memberikan kontribusi pengaruh pada respon sistem secara keseluruhan.

2.4.7. Hand Tuning Kontroler PID

Kontroler PID dapat di *tuning* dalam beberapa cara, antara lain ZieglerNichols *tuning*, *loop shaping*, metode analitis, optimisasi, *pole placement*, *auto tuning* dan *hand tuning* (Smith, 1979; Astrom & Hagglund, 1995). Pada skripsi ini

digunakan cara *hand tuning* untuk menentukan besar K_P , K_i , dan K_d . Hal ini dilakukan karena ada kendala pada pengaturan motor. Kendala tersebut adalah tidak dapat melihat respons motor secara langsung karena tidak digunakannya sensor untuk mengukur kecepatan motor saat sistem berjalan. Selain itu tidak adanya model matematis dari motor membuat cara analitis sulit untuk dilakukan.

Menurut Smith (1979), untuk melakukan *hand tuning* prosedur yang dilakukan adalah sebagai berikut:

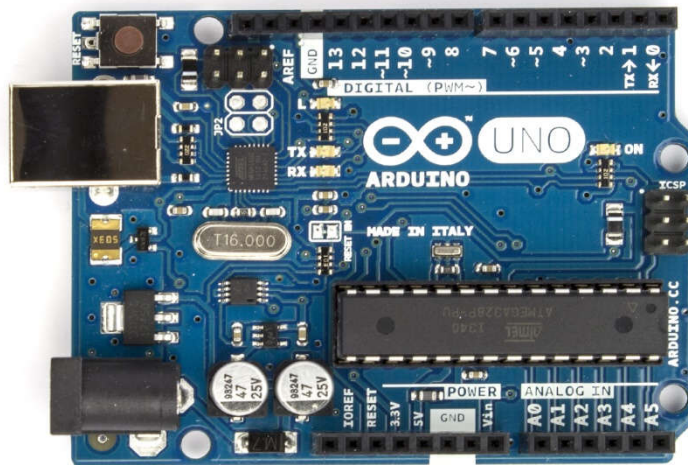
1. Melepaskan kontroler integral dan deferensial dengan memberikan nilai $K_i = 0$ dan $K_d = 0$.
2. Mengatur nilai K_p hingga didapatkan respons yang diinginkan, dengan mengabaikan offset dari setpoint.
3. Dengan terus menaikkan nilai K_p , nilai dari K_d dinaikkan untuk mengurangi overshoot yang terjadi.
4. Naikkan nilai K_i untuk mengurangi offset.

Keuntungan dari *hand tuning* adalah prosedur di atas dapat dilakukan dengan segera, *online* dan dapat melihat dengan cepat respons sistem setelah perubahan K_p , K_i dan K_d . Kerugian dari cara ini adalah kesulitan untuk melihat apakah *setting* akhir dari kontroler merupakan nilai optimal atau tidak (Jantzen, 2001).

2.5. Arduino Uno

Arduino Uno adalah board mikrokontroler berbasis ATmega328. Memiliki 14 pin *input* dari *output digital* dimana 6 pin *input* tersebut dapat digunakan sebagai *output* PWM dan 6 pin *input analog*, 16 MHz *osilator* kristal, koneksi USB, jack *power*, ICSP header, dan tombol *reset*. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan *Board Arduino Uno* ke komputer dengan menggunakan kabel USB atau listrik dengan AC yang ke adaptor-DC atau baterai untuk menjalankannya.

Uno berbeda dengan semua board sebelumnya dalam hal koneksi USB-to-serial yaitu menggunakan fitur Atmega8U2 yang diprogram sebagai *konverter* USB-to-serial berbeda dengan board sebelumnya yang menggunakan chip FTDI driver USB-to-serial.



Gambar 2.10. ArduinoUno Tampak Depan

Sumber : <https://www.arduino.cc/en/Main/ArduinoBoardUno>

Nama “Uno” berarti satu dalam bahasa Italia, untuk menandai peluncuran *Arduino* 1.0. *Uno* dan versi 1.0 akan menjadi versi referensi dari *Arduino*. *Uno* adalah yang terbaru dalam serangkaian board USB *Arduino*.

2.5.1. Daya

ArduinoUno dapat diaktifkan melalui koneksi USB atau dengan catu daya *eksternal* (otomatis). *Eksternal* (non-USB) daya dapat berasal baik dari AC - ke adaptor DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan plug jack pusat positif ukuran 2.1mm konektor power. Ujung kepala dari baterai dapat dimasukkan kedalam Gnd dan Vin pin header dari konektor power. Kisaran kebutuhan daya yang disarankan untuk board *Uno* adalah 7 sampai dengan 12 volt, jika diberi daya kurang dari 7 volt kemungkinan pin 5v *Uno* dapat beroperasi tetapi tidak stabil kemudian jikadiberi daya lebih dari 12V, regulator tegangan bisa panas dan dapat merusak *boardUno*.

2.5.2. Memori

ATmega328 memiliki 32 KB (dengan 0,5 KB digunakan untuk bootloader), 2 KB dari SRAM dan 1 KB EEPROM.

2.5.3. Input Ouput

Masing-masing dari 14 pin *digital* di *Uno* dapat digunakan sebagai *input* atau *output*, dengan menggunakan fungsi *pinMode* (), *digitalWrite* (), dan *digitalRead* (),

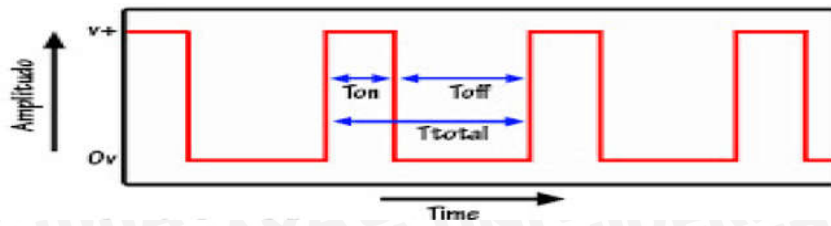
beroperasi dengan daya 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki internal *pull-up* resistor (secara *default* terputus) dari 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus:

- a. Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. Pin ini dihubungkan ke pin yang berkaitan dengan chip Serial ATmega8U2 USB-to-TTL.
- b. *Eksternal* menyela: 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu *interrupt* pada nilai yang rendah, dengan batasan tepi naik atau turun, atau perubahan nilai. PWM: 3, 5, 6, 9, 10, dan 11. Menyediakan *output* PWM 8-bit dengan fungsi *analogWrite ()*.
- c. SPI: 10 (SS), 11 (Mosi), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan SPI *library*.
- d. LED: 13. Ada built-in LED terhubung ke pin *digital* 13. Ketika pin bernilai nilai HIGH, LED on, ketika pin bernilai LOW, LED off.
- e. *Uno* memiliki 6 masukan *analog*, berlabel A0 sampai dengan A5, yang masing-masing menyediakan 10 bit dengan resolusi (yaitu 1024 nilai yang berbeda). Selain itu, beberapa pin memiliki fungsi khusus:
- f. I2C: A4 (SDA) dan A5 (SCL). Dukungan I2C (TWI) komunikasi menggunakan perpustakaan Wire. i. Aref. Tegangan referensi (0 sampai 5V saja) untuk *inputanalog*. Digunakan dengan fungsi *analog Reference ()*.
- g. *Reset*. Bawa baris ini LOW untuk me-*reset* mikrokontroler.

2.6. Pulse Width Modulation (PWM)

PWM (*Pulse Width Modulation*) digunakan untuk mengatur kecepatan dari motor DC. Dimana kecepatan motor DC tergantung pada besarnya *duty cycle* yang diberikan pada motor DC tersebut.

Pada sinyal PWM, frekuensi sinyal *konstan* sedangkan *duty cycle* bervariasi dari 0%-10%. Dengan mengatur *duty cycle* akan diperoleh keluaran yang diinginkan. Sinyal PWM (*Pulse Width Modulation*) secara umum dapat dilihat dalam gambar 2.11 berikut:



Gambar 2.11. Gelombang kotak yang memiliki y_{max} , y_{min} dan D
 Sumber : Heri Andrianto, 2008 : 137

$$T_{total} = T_{on} + T_{off}$$

$$D = \frac{T_{on}}{T} \times 100(\%) \dots\dots\dots (2.20)$$

Dengan:

T_{on} = Periode logika tinggi

T = Periode keseluruhan

D = *Duty cycle* (lamanya pulsa tinggi dalam satu periode)

$$V_{dc} = Duty\ cycle \times V_{cc}(V) \dots\dots\dots (2.21)$$

Sedangkan *frekuensi* sinyal dapat ditentukan dengan rumus berikut:

$$f_{On} = \frac{f_{clk}}{N.256} (Hz) \dots\dots\dots (2.22)$$



BAB III

METODOLOGI PENELITIAN

Penyusunan proposal ini didasarkan pada masalah yang bersifat aplikatif, yaitu perencanaan dan perealisasiian alat agar dapat bekerja sesuai dengan yang direncanakan dengan mengacu pada rumusan masalah. Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang dirancang adalah penentuan studi literatur, spesifikasi alat, perancangan dan realisasi pembuatan alat, pengujian alat, dan pengambilan kesimpulan:

3.1. Studi Literatur

Studi literatur dilakukan agar dapat memahami komponen penyusun sistem. Studi literatur yang dilakukan tentang:

1. Sensor *Hall Effect*
2. Kontroler PID
3. *ArduinoUno* Berbasis Atmega 328
4. Motor *DC*
5. *Peristaltic Pump*

3.2. Spesifikasi alat

Spesifikasi alat secara keseluruhan ditetapkan terlebih dahulu sebagai acuan dalam perancangan selanjutnya. Spesifikasi alat yang direncanakan adalah sebagai berikut:

1. Sensor yang digunakan adalah sensor *hall effect* beserta rangkaian pengkondisi sinyal.
2. Kontroler yang digunakan adalah kontroler PID (*Proporsional, Integral, dan Deferenisial*).
3. Menggunakan *ArduinoUno* sebagai pusat kendali.
4. Modul *Peristaltic Pump* sebagai *plant*.

5. Sistem penggerak menggunakan motor DC. Pemrograman *software* program *Arduino* ERW 1.0.5.

3.3. Spesifikasi Kontrol

1. *Setpoint* pada penelitian ini diatur pada kecepatan 4000 RPM.
2. *Time steady* kurang dari 2 detik.
3. *Error steady state* (%ESS) tidak melebihi 2%.
4. *Maximum Overshot* (MP) kurang dari 5%.

3.4. Perancangan dan Realisasi Pembuatan Alat

3.4.1. Perancangan Perangkat Keras dan Realisasi Pembuatan Alat

- a. Perancangan diagram blok sistem.
- b. Perancangan sistem kontrol *flowrate* cairan dengan mengontrol kecepatan motor.
- c. Merakit perangkat keras (hardware) untuk masing-masing blok.

3.4.2. Perancangan dan Perhitungan Komponen yang akan Digunakan

Setelah perangkat keras telah selesai dibuat, langkah selanjutnya adalah perancangan perangkat lunak guna mengendalikan dan mengatur kerja alat. Desain dan parameter yang telah dirancang kemudian diterapkan pada *Arduino Uno* dengan menggunakan *software Arduino ERW 1.0.5*.

3.4.3. Perancangan Perangkat Lunak

Untuk perancangan perangkat lunak hal pertama yang dilakukan adalah mengetahui karakteristik motor DC dengan cara mencari fungsi alih dari motor DC tersebut. Perancangan perangkat lunak dilakukan setelah mengetahui nilai parameter Proporsional (P), Integral (I) dan Deferenensial (D). Perancangan dimulai dari pembuatan flowchart, kemudian penulisan listing code.

3.5. Pengujian Alat

Pengujian alat dilakukan untuk memastikan bahwa sistem ini berjalan sesuai yang direncanakan. Pengujian alat meliputi pengujian perangkat keras maupun perangkat lunak yang dilakukan baik secara per blok rangkaian maupun keseluruhan sistem.

3.5.1 Pengujian Tiap Blok

Pengujian per blok dilakukan dengan tujuan untuk menyesuaikan nilai masukan dan nilai keluaran tiap-tiap blok sesuai dengan perancangan yang dilakukan sebelumnya.

3.5.2. Pengujian Keseluruhan Sistem

Pengujian sistem secara keseluruhan dilakukan dengan tujuan untuk mengetahui kerja alat setelah perangkat keras dan perangkat lunak diintegrasikan bersama.

3.6. Pengambilan Kesimpulan

Kesimpulan diambil berdasarkan hasil data yang diperoleh dari pengujian keseluruhan. Jika hasil yang didapatkan telah sesuai dengan yang sudah direncanakan, maka alat tersebut telah berhasil memenuhi harapan dan untuk penyempurnaan yang lebih mendalam, perlu adanya pengembangan lebih lanjut.



BAB IV

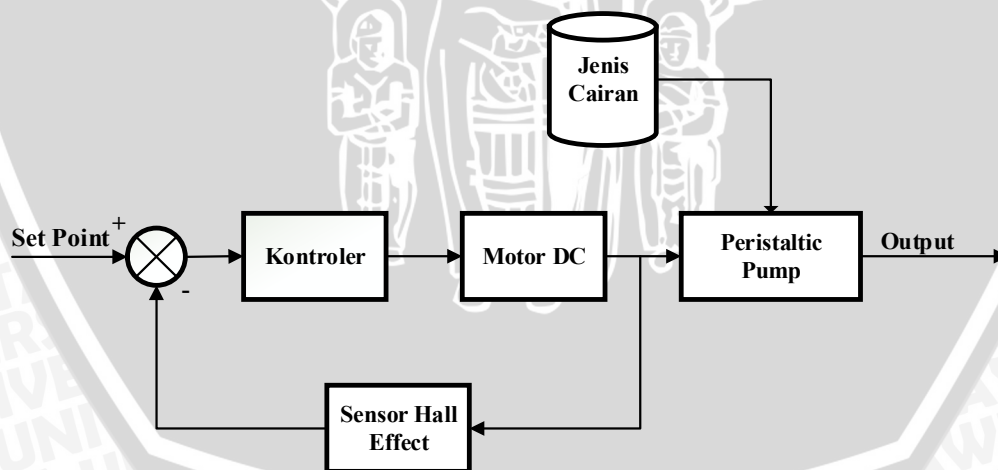
PERANCANGAN DAN PEMBUATAN ALAT

Perancangan dan pembuatan alat dilakukan secara bertahap, sehingga akan memudahkan dalam analisa pada setiap blok maupun secara keseluruhan, dimulai dengan pembuatan diagram blok pada Gambar 4.1. Sedangkan pembuatan alat bertujuan untuk menghasilkan semua perangkat pendukung, maupun alat secara keseluruhan. Secara garis besar terdapat dua bagian perangkat yaitu:

- Perancangan perangkat keras (*hardware*)
- Perangan perangkat lunak (*software*).

4.1. Diagram Blok Sistem

Pada perancangan sistem ini diperlukan blok diagram sistem untuk menjelaskan sebagian besar sistem kerja *peristaltic pump* dan diharapkan sistem kerjanya sesuai dengan yang direncanakan. Blok diagram dapat dilihat pada Gambar 4.1



Gambar 4.1 Diagram blok sistem

Berdasarkan diagram blok dalam Gambar 4.1, dapat dijelaskan secara umum mengenai bagian-bagian yang menyusun keseluruhan sistem alat ini. Keterangan bagian-bagian tersebut adalah sebagai berikut:

- *Setpoint* berupa *flow rate* yang diinginkan dalam *ml/menit*
- Sensor yang dipakai adalah sensor *hall effect* dengan keluaran berupa pulsa, selanjutnya diubah menjadi tegangan menggunakan konverter *frequency to voltage*.
- Motor DC sebagai penggerak poros *plant* berupa *peristaltic pump*.
- *Output* dari sensor kemudian dikurangkan dengan *setpoint* sehingga mikrokontroler mampu mengkompensasi *error* yang terjadi dengan menggunakan kontroler PID.

4.2. Prinsip Kerja Alat

Peristaltic pump ini akan berputar setelah penggerak motor DC diberi sumber tegangan, sistem dari alat ini akan bekerja setelah memasukan input berupa *flow rate* yang telah ditentukan terlebih dahulu. Saat tombol *start* diaktifkan, maka seluruh sistem akan aktif. *Peristaltic pump* ini menggunakan satu buah motor DC sebagai penggerak.

Peristaltic pump selain bertindak sebagai *plant*, *peristaltic pump* juga berfungsi sebagai pengatur *flow rate* yang diinginkan. Pengaturan *flow rate* dilakukan dengan cara memperhitungkan *flow rate* pipa yang menjadi jalur cairan pada *peristaltic pump* tersebut. Dengan demikian, diperlukan data fisik *peristaltic pump* berupa diameter pipa, reaksi putaran, serta RPM motor. Sehingga diperoleh hubungan antara kecepatan motor dan kecepatan aliran.

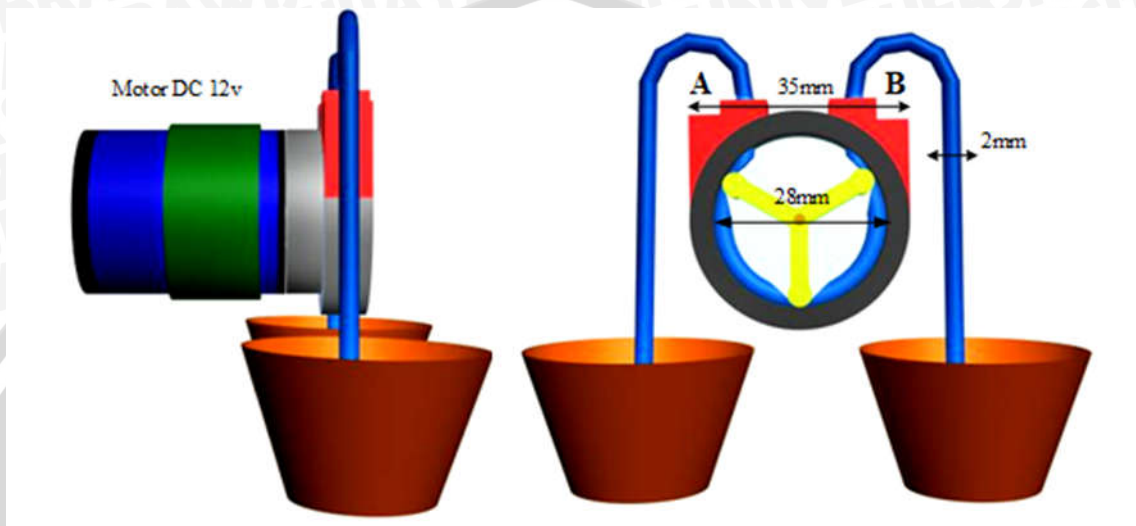
4.3. Perancangan Perangkat Keras

Berdasarkan diagram blok perancangan alat yang telah disusun, perancangan perangkat keras terdiri dari modul *peristaltic pump*, motor DC dan *driver* motor.

4.3.1. Mekanik Peristaltic Pump

1. Penggerak *peristaltic pump* menggunakan sebuah motor DC.
2. Menggunakan sebuah sensor *hall effect* sebagai sensor kecepatan yang dipasang pada motor DC.

3. *ArduinoUno* sebagai pusat kendali.
4. LCD diletakan di depan *peristaltic pump* yang berfungsi untuk menampilkan konstanta-konstanta yang akan ditentukan.
5. Catu tegangan untuk motor DC menggunakan tegangan 24V dari power suply.



Gambar 4.2 Bentuk Keseluruhan *Peristaltic Pump*

Keterangan gambar 4.2:

- *Peristaltic pump* berada di tengah.
- Motor DC diletakan dibelakang.
- Terdapat 2 gelas ukur cairan sebagai sumber cairan dan penampung cairan .

4.3.2 Spesifikasi *Peristaltic Pump*

Spesifikasi *peristaltic pump*:

1. Diameter luar pipa aliran *fluida* pada *peristaltic pump* berukuran 3,5mm.
2. Diameter dalam pipa aliran *fluida* pada *peristaltic pump* berukuran 2mm
3. Diameter satu putaran pada *peristaltic pump* berukuran 2,8cm
4. Motor DC dapat bekerja pada tegangan berkisar antara 12 Volt, sedangkan kecepatanputaran motor DC sebesar 4000 rpm
5. Rasio perbandingan putaran motor DC dengan *peristaltic pump* 1 : 10, jadi setiap 10 putaran motor DC sama dengan 1 putaran *peristaltic pump*.

Nilai *flowrate* dapat dirumuskan sebagai berikut:

$$\text{Flowrate (Fr)} = \frac{\text{RPM}}{10} \cdot V_{LAB} \cdot V_p$$

Dimana:

Fr : Nilai aliran cairan setiap menit

V_p : Volume pipa *peristaltic* setiap sentimeter

V_{LAB} : Volume pipa *peristaltic* dalam satu putaran

Volume pipa aliran *fluida* pada setiap sentimeter pipa didapatkan dengan perhitungan matematika dan menghasilkan nilai sebagai berikut :

$$V_p = \pi r^2 \text{ ml/cm} \dots\dots\dots (4.1)$$

$$V_p = 3,14 \cdot 0,1^2 \text{ ml/cm} \dots\dots\dots (4.2)$$

$$V_p = 0,0314 \text{ ml/cm} \dots\dots\dots (4.3)$$

Panjang pipa untuk setiap satu putaran *peristaltic pump* atau 360° adalah:

$$LAB = 2,8 \text{ cm}$$

Sehingga nilai volume aliran *fluida* dalam satu putaran *peristaltic pump* diperoleh dengan perhitungan sebagai berikut:

$$V_{LAB} = V_p \cdot LAB \dots\dots\dots (4.4)$$

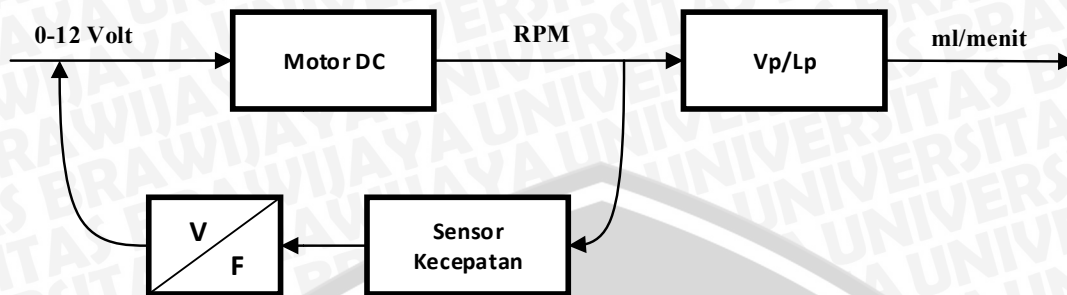
$$V_{LAB} = 0,0314 \cdot 2,8 = 0,08792 \text{ ml/360}^\circ \dots\dots\dots (4.5)$$

Dengan demikian diperoleh hubungan antara kecepatan motor dalam (RPM) dengan *flow rate* cairan *ml/menit* sebagai berikut:

$$Fr = \frac{\text{rpm}}{10} \cdot V_{LAB} \cdot V_p \text{ ml/menit} \dots\dots\dots (4.6)$$

$$Fr = \frac{4000}{10} \cdot 0,0314 \cdot 2,8 = 35,168 \text{ ml/menit} \dots\dots\dots (4.7)$$

Pengaturan *flow rate* selanjutnya dapat diatur dengan mengatur putaran motor DC secara konstan.



Gambar 4.3 Diagram blok pengaturan *flow rate*

Dengan demikian, akan diperoleh hubungan keterkaitan antara pengaturan kecepatan kestabilan putaran motor dengan pengaturan kestabilan aliran.

4.3.3. Sensor Hall Effect

Seperti yang dijelaskan diatas, sensor *hall effect* yang digunakan pada perancangan alat ini terpasang menjadi satu pada motor DC, bekerja sebagai pengolah sinyal yang dihasilkan dari magnet/ reluktor yang berputar. Sinyal yang dihasilkan kemudian diolah oleh rangkaian *Frequency to Voltage* untuk menghasilkan keluaran berupa sinyal analog.



Gambar 4.4 Bentuk Sensor Hall Effect

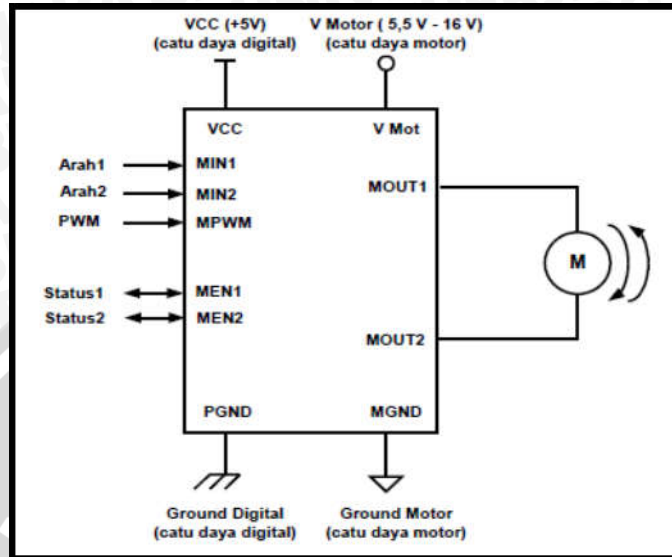
4.3.4. Perancangan Driver Motor DC

Modul pengendali motor DC yang digunakan adalah modul EMS 30A *H-Bridge*. Secara garis besar, fungsi modul pengendali motor ini adalah untuk mengendalikan arah

dan kecepatan putaran motor DC sesuai instruksi kendali dari Arduino Uno ERW 1.0.5.

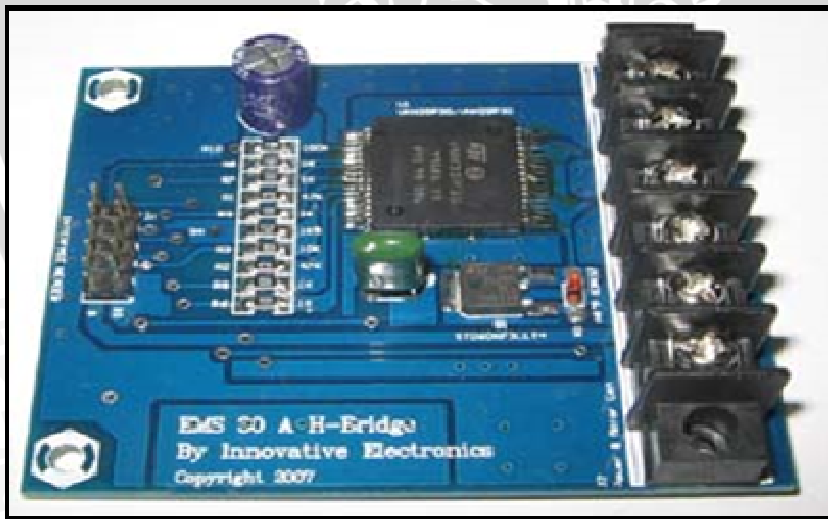
Gambar koneksi modul pengendali motor DC *EMS30A H-Bridge* ditunjukkan dalam

Gambar 4.5



Gambar 4.5 Koneksi Modul EMS 30A H-Bridge
Sumber: *EMS 30A H-Bridge datasheet*

Rangkaian *driver* ini sanggup bekerja dengan tegangan maksimal 40 volt, serta kapasitas arus maksimum yang dapat dilewatkan pada modul ini sebesar 30 ampere. Gambar modul rangkaian *driver* EMS 30A H-Bridge dapat dilihat pada Gambar 4.5.

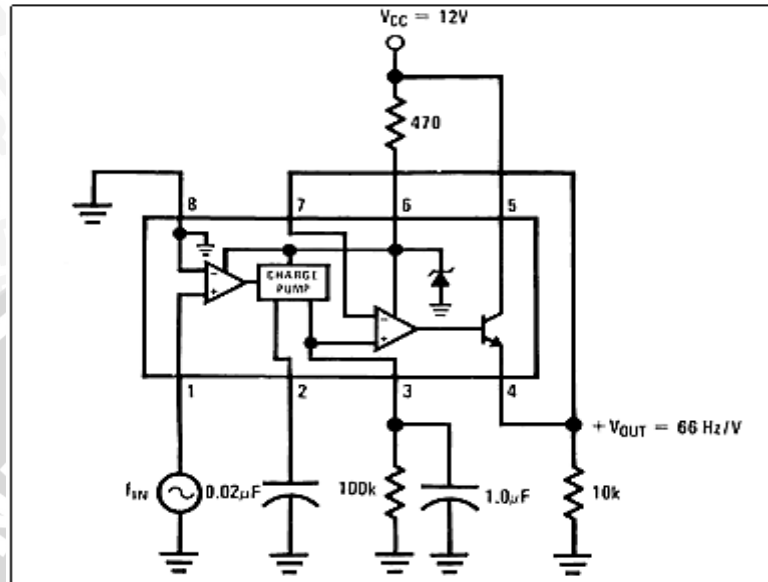


Gambar 4.6 Modul Rangkaian *Driver* EMS 30A H-Bridge

4.3.5. Rangkaian Frequency to Voltage

Rangkaian *Frequency to Voltage* digunakan untuk mengubah sinyal digital dari keluaran sensor *hall effect* yang diproses menjadi sinyal analog agar dapat dimasukkan

pada board Arduino Uno. Rangkaian ini menggunakan *integrated circuit* (IC) LM2917 yang memiliki tegangan kerja +12 volt DC hingga +24 volt DC. Skema rangkaianannya dapat ditunjukkan pada Gambar 4.7.



Gambar 4.7 Skematik Rangkaian *Frequency to Voltage*
Sumber Datasheet LM2917

Berdasarkan pada *datasheet* dengan rangkaian seperti pada gambar 4.7, maka tegangan keluaran dapat dihitung dengan menggunakan persamaan:

$$V_{out} = \frac{f_{in}}{66} \dots \dots \dots (4-7)$$

Atau secara umum perhitungan yang digunakan pada rangkaian yang menggunakan IC LM2907/2917 dapat dijelaskan pada persamaan berikut.

$$V_{out} = V_{in} \times V_{cc} \times R_1 \times C_1 \dots \dots \dots (4-8)$$

Dimana:

f_{in} = Frekuensi sinyal input (Hz)

V_{cc} = Tegangan sumber yang digunakan (volt)

R_1 = Resistor pada pin 3 IC LM2917 (Ohm)

C_1 = Kapasitor pada pin 2 IC LM2917 (Farad)

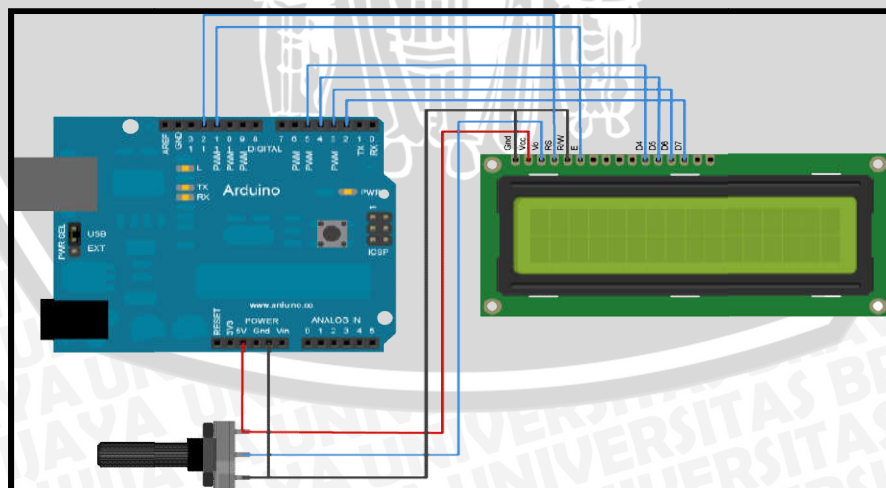
4.3.6. *Liquid Crystal Display (LCD)*

LCD yang digunakan dalam perancangan kali ini adalah Hitachi HD44780 dan di dalam sistem ini berfungsi sebagai penampil atau penunjuk waktu dan error yang ditunjukkan. Digunakannya LCD dalam perancangan ini diperlukan karena apabila hasil atau output waktu dan error ditunjukkan pada serial monitor yang ada pada Arduino Uno, untuk pengambilan data sebagai bahan analisis perancangan akan menjadi lebih sulit.

LCD Hitachi dapat dikontrol dalam dua mode : 4-bit atau 8-bit. Mode 4-bit membutuhkan tujuh I/O pin Arduino Uno, sedangkan mode 8-bit membutuhkan sebelas I/O pin Arduino Uno. Untuk rangkaian LCD, diperlukan beberapa hardware sebagai penunjang agar LCD dapat berfungsi dengan baik, yaitu :

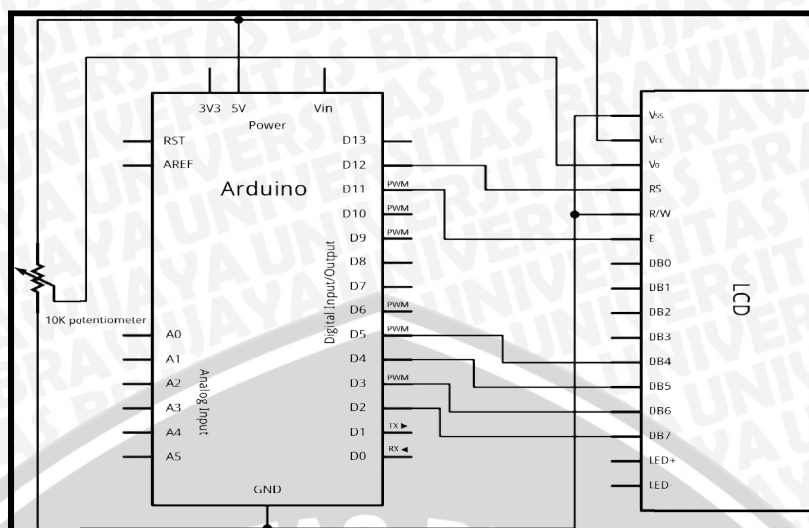
- Arduino Uno
- LCD screen (yang kompatibel dengan Hitachi HD44780)
- Pin header untuk LCD *display*
- 10k potensiometer
- *Project board*
- Kabel *male to female*

Dengan pin header disolder terlebih dulu pada LCD, kemudian dengan menggunakan kabel male to female dihubungkan ke Arduino Uno seperti yang ditunjukkan pada Gambar 4.8 di bawah ini.



Gambar 4.8 LCD ke Arduino Uno

Sumber: www.arduino.cc/en/Tutorial/LiquidCrystal



Gambar 4.9Rangkaian Wiring LCD ke Arduino Uno
Sumber: www.arduino.cc/en/Tutorial/LiquidCrystal

4.4 Identifikasi Motor DC

Pada pengujian identifikasi driver motor DC menggunakan Matlab ini akan didapatkan fungsi alih. Fungsi alih sendiri digunakan untuk mengetahui karakteristik dari motor DC yang digunakan. Setelah melakukan pengujian rangkaian driver DC maka didapatkan data dari hasil pengujian tersebut. Data hasil pengujian kemudian disimulasikan menggunakan Matlab. Pengujian alat ini menggunakan Motor DC dengan RPM 5000 dan bekerja pada tegangan 12 V. Motor DC yang digunakan ditunjukkan pada gambar 4.10.



Gambar 4.10Motor DC dengan Modul *Peristaltic Pump*

4.4.1. Menentukan Parameter Motor DC

Untuk mendapatkan fungsi alih dari motor DC diperlukan beberapa parameter data hasil pengujian. Data yang diperoleh dari pengujian fisik motor DC antara lain:

- $V = 2 \text{ volt}$
- $i = 0,25 \text{ A}$
- $\omega = 0,46 \text{ rad/s}$
- $R = 1,3 \Omega$
- $L = 0,007 \text{ H}$
- $T_m = 0,26 \text{ s}$

Kemudian hasil dari pengujian fisik motor DC didapatkan beberapa parameter pada persamaan (2.8) sampai (2.10) sebagai berikut:

Nilai dari *back - emf* didapatkan menggunakan persamaan (2.8) dengan memasukkan nilai parameter diatas dan menghasilkan nilai sebagai berikut :

$$e = V - i \times R \dots\dots\dots (4.9)$$

$$e = 2 - 0,25 \times 1,3 = 1,675 \text{ volt} \dots\dots\dots (4.10)$$

Nilai dari konstanta motor dapat diperoleh dari persamaan (2.9), setelah memasukan nilai parameter yang diperoleh diatas akan didapatkan:

$$K = e : \omega = 3,641 \dots\dots\dots (4.11)$$

$$K = 1,675 : 0,46 = 3,641 \dots\dots\dots (4.12)$$

Nilai koefisien *friction* dapat diperoleh dari persamaan (2.10), setelah memasukan nilai parameter yang diperoleh diatas akan didapatkan:

$$b = K \times i : \omega = 0,181 \dots\dots\dots (4.13)$$

$$b = 3,641 \times 0,25 : 0,46 = 1,978 \dots\dots\dots (4.14)$$

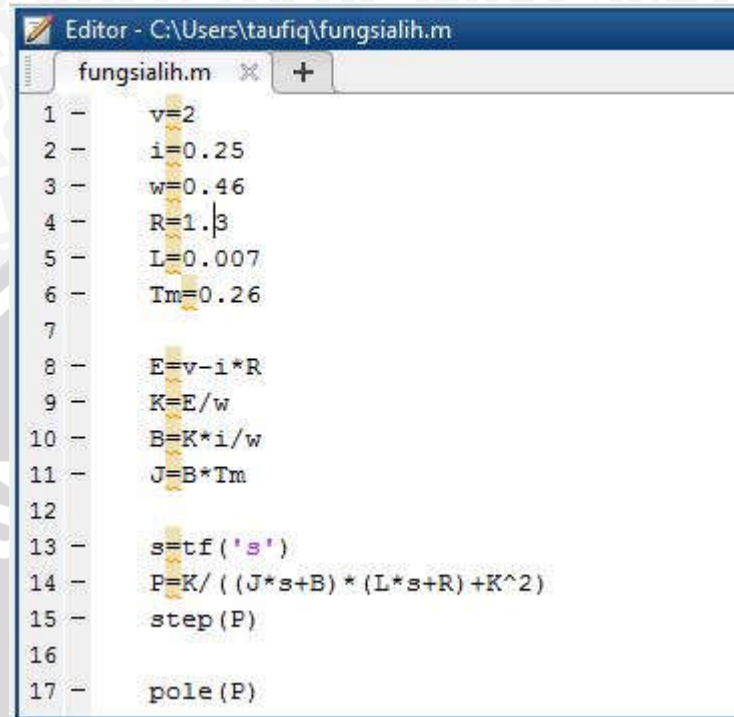
Dari nilai-nilai diatas akan didapatkan nilai momen inersia yang dihitung sebagai berikut ini:

$$J = b \times T_m = 0,0941 \dots\dots\dots (4.15)$$

$$J = 1,978 \times 0,26 = 0,514 \dots\dots\dots (4.16)$$

4.4.2. Menentukan Fungsi Alih Motor DC

Setelah mendapatkan parameter motor DC, kemudian akan ditentukan fungsi alih dengan cara memasukan ke dalam simulasi menggunakan Matlab. Dengan memasukan paramter motor DC kedalam program Matlab seperti Gambar 4.11



```

Editor - C:\Users\taufiq\fungisialih.m
fungisialih.m
1 - v=2
2 - i=0.25
3 - w=0.46
4 - R=1.3
5 - L=0.007
6 - Tm=0.26
7
8 - E=v-i*R
9 - K=E/w
10 - B=K*i/w
11 - J=B*Tm
12
13 - s=tf('s')
14 - P=K/((J*s+B)*(L*s+R)+K^2)
15 - step(P)
16
17 - pole(P)

```

Gambar 4.11 Program parameter motor DC pada Matlab

Kemudian pada Matlab akan mensimulasikan untuk mendapatkan fungsi alih dari hasil paramater motor DC dan menunjukkan pole dari fungsi alih. Hasil dari fungsi alih dan pole ditunjukkan pada gambar 4.12. Setelah itu akan di plot grafik respon dari karakteristik motor DC yang ditunjukkan pada Gambar 4.13.

```

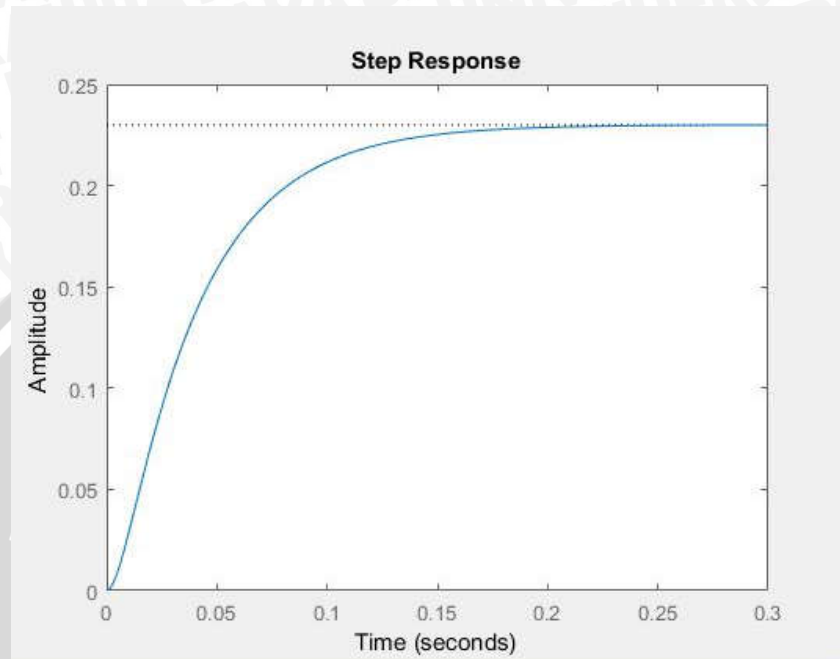
P =
      3.641
-----
0.003602 s^2 + 0.6827 s + 15.83
Continuous-time transfer function.

ans =
-162.5127
-27.0478

```

Gambar 4.12 Fungsi alih dan pole motor DC

Setelah mendapatkan fungsi alih kemudian diberi masukan step pada simulasi Matlab bisa dilihat grafik respon dari motor DC *time constan* dari motor 0,25 detik pada gambar 4.13



Gambar 4.13 Respon motor DC dari hasil simulasi Matlab

4.4.3. Penentuan Nilai Penguatan Kontroler

Perancangan kontroler PID untuk kecepatan motor DC, sehingga sistem dapat bekerja dengan yang diharapkan. Untuk mencari nilai dari PID, menggunakan fungsi alih dari motor. Nilai penguatan kontroler (K_p , K_i , K_d) ditentukan dengan menggunakan metode *hand tuning* untuk mendapatkan hasil respon sesuai dengan yang diinginkan.

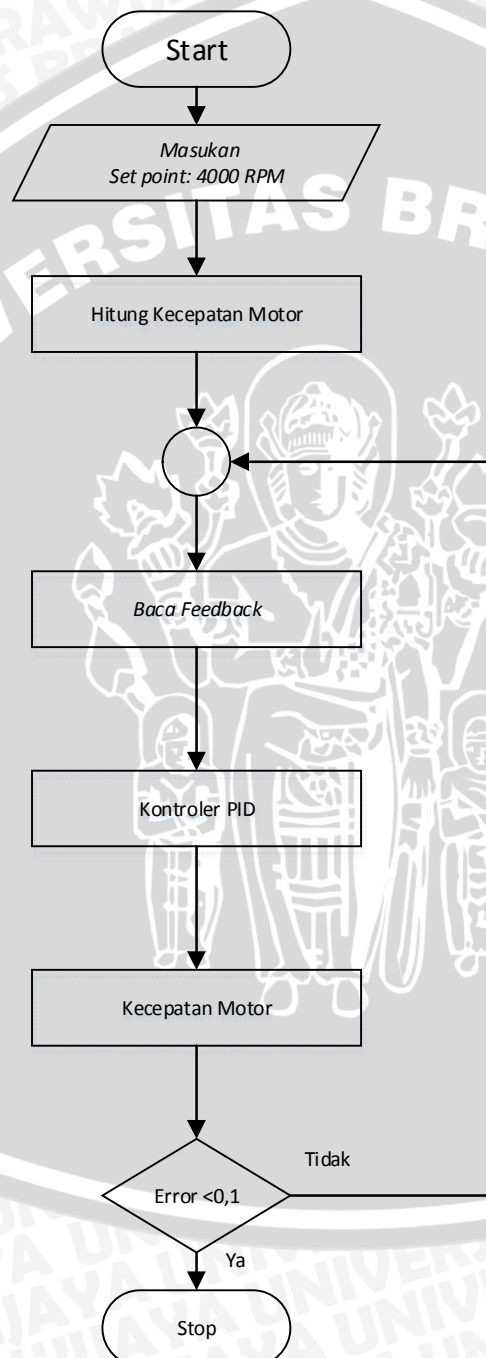
Hand tuning adalah proses yang dilakukan untuk mendapatkan hasil kontroler yang optimal dengan cara suatu percobaan. Inti dari *handtuning* adalah menentukan nilai dari tiga buah parameter yang terdapat pada kontroler PID yaitu konstanta proporsional (K_p), konstanta integral (K_i) dan konstanta diferensial (K_d).

Proses *tuning* parameter PID ini dilakukan dengan cara mengatur nilai K_p hingga didapatkan respon sistem yang mendekati *setpoint* 4000 RPM. Pengambilan data dilakukan dengan pembacaan sensor yang masuk dalam serial monitor Arduino Uno.

Berdasarkan hasil tuning ketiga parameter K_p , K_i , dan K_d dengan menggunakan metode *Hand Tuning*, maka dapat ditentukan parameter penguatan kontroler yang akan digunakan pada sistem yaitu $K_p = 85$, $K_i = 7$, dan $K_d = 4$.

4.5. Perancangan Perangkat Lunak

Perancangan perangkat lunak pada skripsi ini menggunakan bahasa pemrograman C++ dengan menggunakan *software* CodeVision AVR. *Tuning* kontroler PID adalah dengan menggunakan metode *hand tuning*. *Flowchart* perancangan perangkat lunak ditunjukkan dalam Gambar 4.14.



Gambar 4.14 Flowchart Perangkat Lunak



BAB V

PENGUJIAN DAN ANALISIS SISTEM

Tujuan pengujian sistem ini adalah untuk menentukan apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perancangan. Pengujian pada sistem ini meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk menemukan letak kesalahan dan mempermudah analisa pada sistem apabila alat tidak bekerja sesuai dengan perancangan. Pengujian dibagi menjadi beberapa bagian, yaitu:

1. Pengujian data sensor *hall effect*.
2. Pengujian *driver* motor DC
3. Pengujian motor DC.
4. Simulasi Matlab.
5. Pengujian keseluruhan sistem meliputi pengujian pompa terhadap beberapa cairan yang berbeda.

5.1. Pengujian Sensor Hall Effect

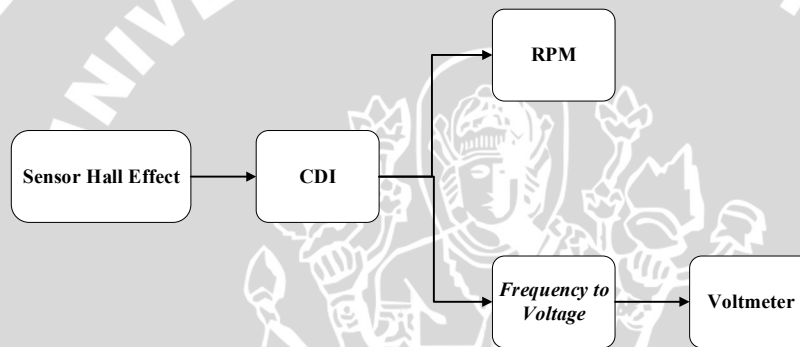
Pengujian ini dilakukan bertujuan untuk mengetahui tingkat kelinieran dari sensor hall effect terhadap perubahan kecepatan motor DC.

5.1.1 Peralatan yang Digunakan

- Sensor *Hall effect*/ CDI tester
- RPM meter
- UBEC
- Catu daya 12 V
- Rangkaian *Frequency to Voltage*
- *Voltmeter*

5.1.2. Langkah Pengujian

- Merangkai peralatan seperti pada gambar 5.1
- Hubungkan *source power*
- Tentukan nilai RPM menggunakan CDI tester
- Ukur nilai tegangan yang keluar dari rangkaian *frequency to voltage*
- Catat tegangan keluaran yang ditampilkan *voltmeter*
- Kembali ulangi langkah ke-3 dengan nilai RPM yang berbeda sampai batas RPM yang telah ditentukan.



Gambar 5.1 Diagram Blok Pengujian Sensor

5.1.3. Hasil Pengujian

Bedasarkan rumus perhitungan pada bab IV, tegangan keluaran yang dihasilkan dari rangkaian *frequency to voltage* dapat ditulis dalam persamaan berikut:

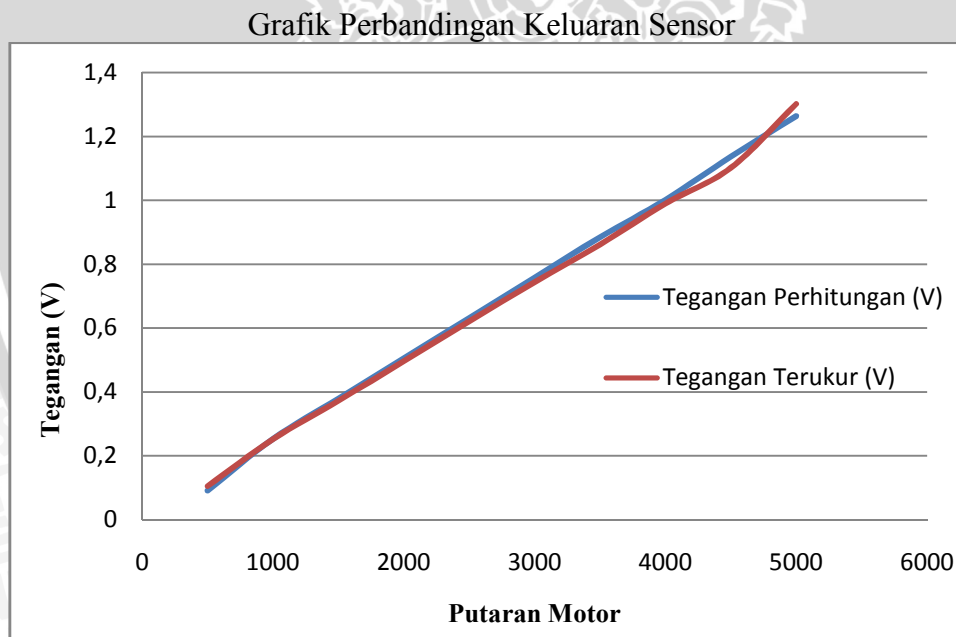
$$V_{out} = \frac{f_{in}}{66} \dots \dots \dots (5.1)$$

Data hasil pengujian sensor ditunjukkan dalam Tabel 5.1 berikut

Tabel 5.1. Hasil Pengujian Sensor Hall Effect

RPM	Tegangan Perhitungan (V)	Tegangan Terukur (V)	Error
500	0,091	0,105	1,55
1000	0,253	0,252	0,21
1500	0,379	0,372	1,79
2000	0,505	0,496	1,79
2500	0,631	0,621	1,63
3000	0,758	0,743	1,92
3500	0,884	0,861	2,58
4000	1,001	0,989	2,09
4500	1,136	1,101	2,67
5000	1,263	1,301	3,04

Grafik hubungan antara tegangan dengan RPM digambarkan pada gambar 5.2.

**Gambar 5.2** Grafik Perbandingan Keluaran Sensor

Dari hasil pengujian yang dilakukan, sensor dapat bekerja dengan maksimal dan terlihat kelinieran yang baik sehingga ideal untuk digunakan sebagai pendeteksi kecepatan putaran pada motor DC.

5.2. Pengujian *Driver Motor DC*

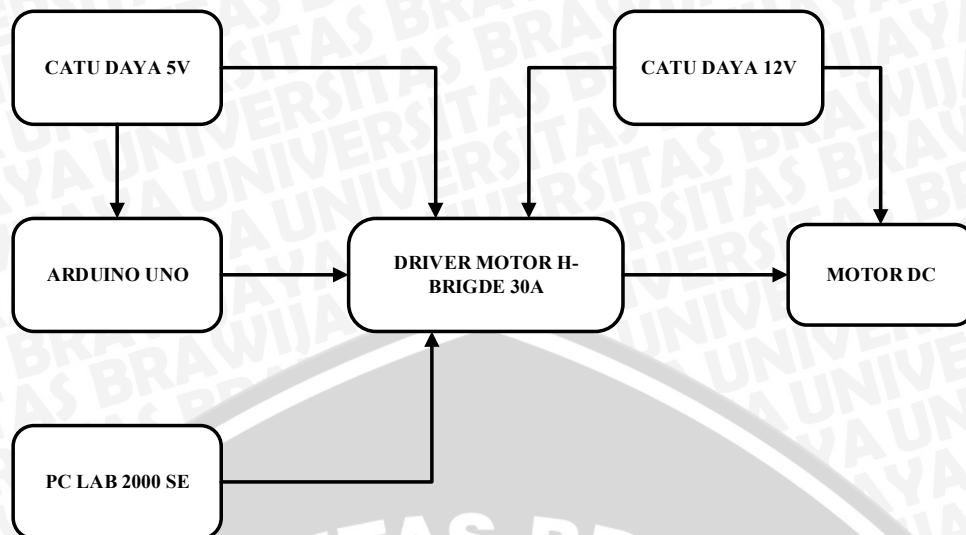
Pengujian *driver motor Direct Current (DC)* ini bertujuan untuk mengetahui perbandingan *output driver motor* jika dibandingkan dengan masukan, kemudian dapat diketahui juga hubungan keluaran *Pulse Width Modulation (PWM)* dengan tegangan yang dibutuhkan untuk motor *Direct Current (DC)*.

Peralatan yang digunakan

- Modul Arduino Uno
- *Driver motor*
- Motor DC
- Catu daya 5V dan 12V
- *Hardware dan software Pulse Width Modulation (PWM)*

5.2.1. Langkah pengujian

1. Merangkai peralatan seperti Gambar 5.3.
2. Mengunduh program yang mengeluarkan data berupa *Pulse Width Modulation (PWM)* dengan keluaran 10-250 dengan interval setiap 10 *Pulse Width Modulation (PWM)* pada *software arduino ERW 1.0.5*.
3. Jalankan *software PC Lab 2000SE*.
4. Pilih mode osiloskop lalu pilih *Run* untuk menjalankan osiloskop.
5. Mengamati sinyal kontrol berupa *duty cycle* pada osiloskop untuk setiap pengujian mikrokontroler dan *driver*.

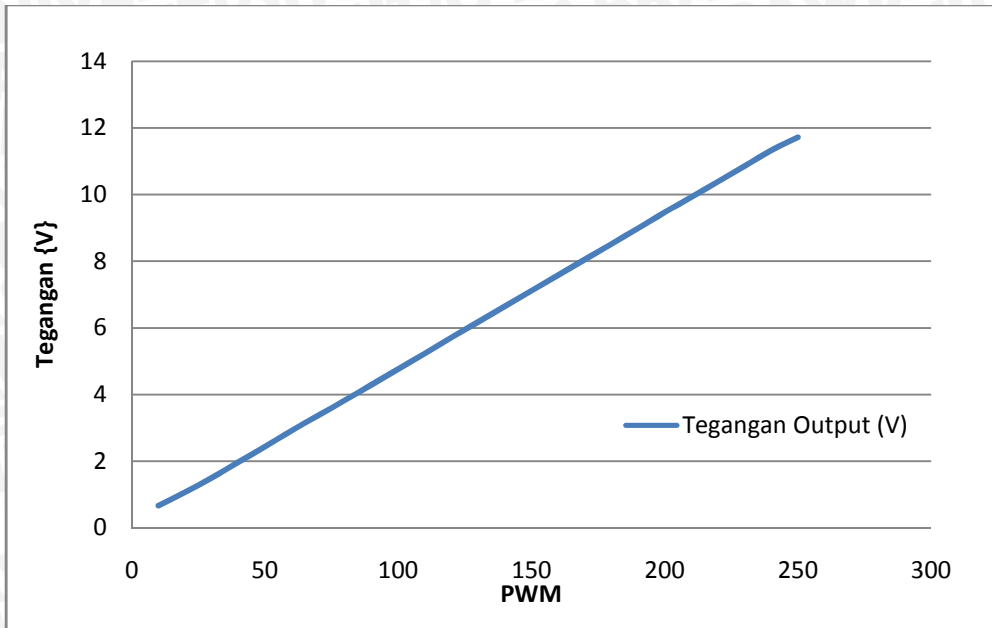


Gambar 5.3 Diagram Blok Pengujian *Driver Motor DC*

5.2.1 Hasil pengujian

Tabel 5.2 Hasil Pengujian *Driver Motor DC*

DRIVER MOTOR		
PWM	Duty Cycle Driver	Tegangan Driver (V)
10	5,52	0,6624
20	8,9	1,068
30	12,5	1,5
40	16,4	1,968
50	20,3	2,436
60	24,3	2,916
70	28,1	3,372
80	31,9	3,828
90	35,8	4,296
100	39,7	4,764
110	43,6	5,232
120	47,6	5,712
130	51,5	6,18
140	55,4	6,648
150	59,3	7,116
160	63,2	7,584
170	67,1	8,052
180	71	8,52
190	74,9	8,988
200	78,9	9,469
210	82,7	9,924
220	86,6	10,392
230	90,5	10,86
240	94,4	11,328
250	97,7	11,724

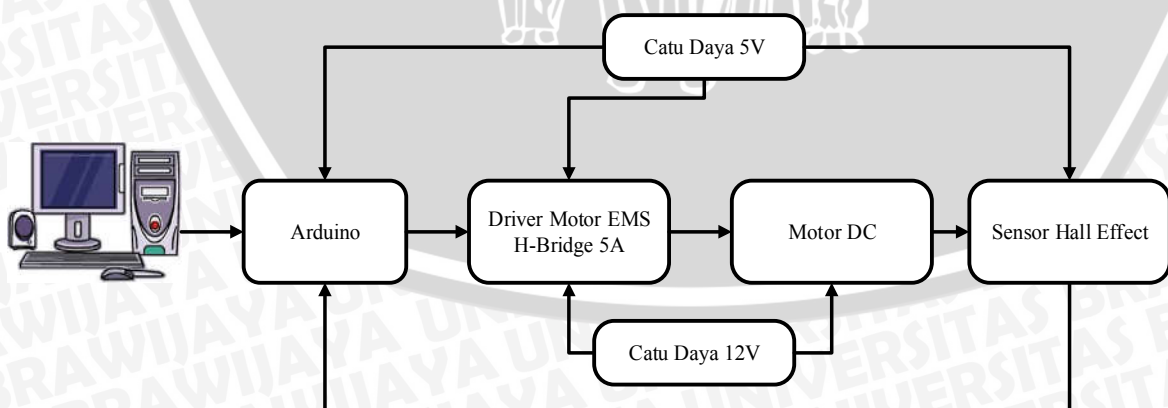


Gambar 5.4 Grafik Hubungan Tegangan dengan *PWM*

Berdasarkan hasil pengujian pada tabel dapat diketahui bahwa tegangan yang keluar berubah-ubah sesuai dengan perubahan *Pulse Width Modulation (PWM)*, dengan demikian dapat disimpulkan *driver* dapat berfungsi dengan baik.

5.3. Pengujian Motor DC

Pada pengujian motor DC bertujuan untuk mengetahui hubungan antara perubahan tegangan terhadap kecepatan motor. Langkah-langkah percobaan pengujian dilakukan seperti pada diagram blok pengujian kecepatan motor DC yang ditunjukkan seperti pada gambar 5.5.



Gambar 5.5 Diagram blok pengujian kecepatan motor

5.3.1. Peralatan yang Digunakan

1. Catu Daya 5v dan 12V
2. Motor DC
3. *Driver* Motor DC EMS H-Bridge 5A
4. Program dan *Software* Arduino
5. Rangkaian sensor hall effect

5.3.2. Prosedur Pengujian Kecepatan Motor DC

Prosedur pengujian dilakukan dengan menghubungkan motor DC dengan power supply, kemudian poros motor DC dihubungkan dengan sensor hall effect, setelah itu tegangan dari power supply diubah-ubah mulai 0-12 V dengan kenaikan 0,5 V, catat hasil setiap perubahan kecepatan motor yang ditunjukkan di tachometer.

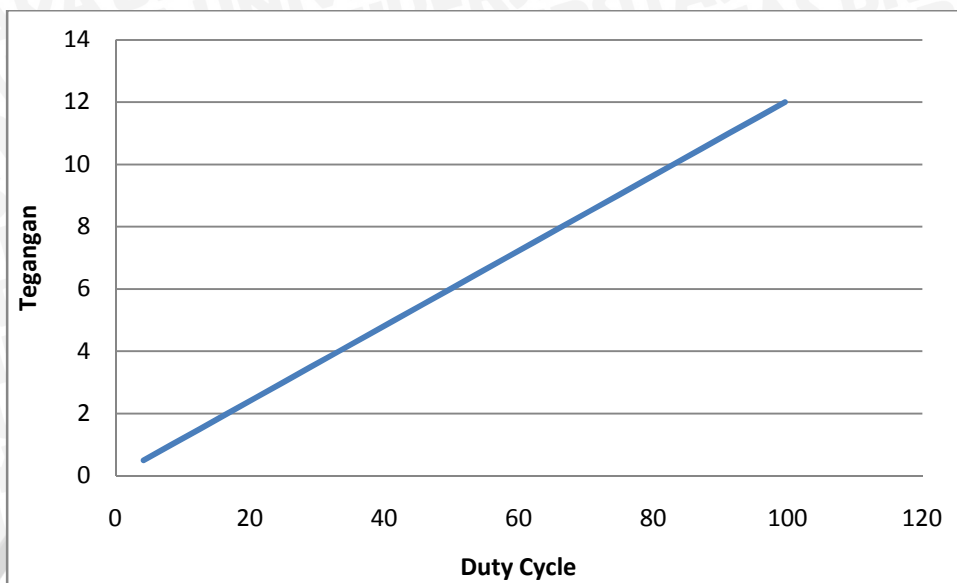
5.3.3. Hasil Pengujian Kecepatan Motor DC

Hasil pengujian kecepatan motor terhadap tegangan tertentu yang diperoleh melalui beberapa kali percobaan ditunjukkan pada tabel 5.3

Tabel 5.3 Data *Duty Cycle* dengan Tegangan dan Kecepatan Motor

Duty Cycle Driver	Tegangan (V)	Kecepatan Motor (RPM)
4,15	0,5	203
8,3	1	412
12,45	1,5	619
16,6	2	828
20,75	2,5	1033
24,9	3	1239
29,05	3,5	1445
33,2	4	1657
37,35	4,5	1862
41,5	5	2073
45,65	5,5	2280
49,8	6	2496
53,95	6,5	2700
58,1	7	2902
62,25	7,5	3116
66,4	8	3317
70,55	8,5	3531
74,7	9	3739
78,85	9,5	3947
83	10	4153
87,15	10,5	4357
91,3	11	4551

95,45	11,5	4732
99,6	12	4976



Gambar 5.6 Grafik hubungan antara *Duty Cycle* dengan Tegangan

5.4. Simulasi Matlab

Pada simulasi menggunakan *software* Matlab ini bertujuan untuk menerapkan algoritma yang telah ditetapkan sebelumnya dan untuk mengetahui performasi yang ideal secara simulasi dan dapat dijadikan tolok ukur dengan pengujian secara keseluruhan yang akan dilakukan selanjutnya.

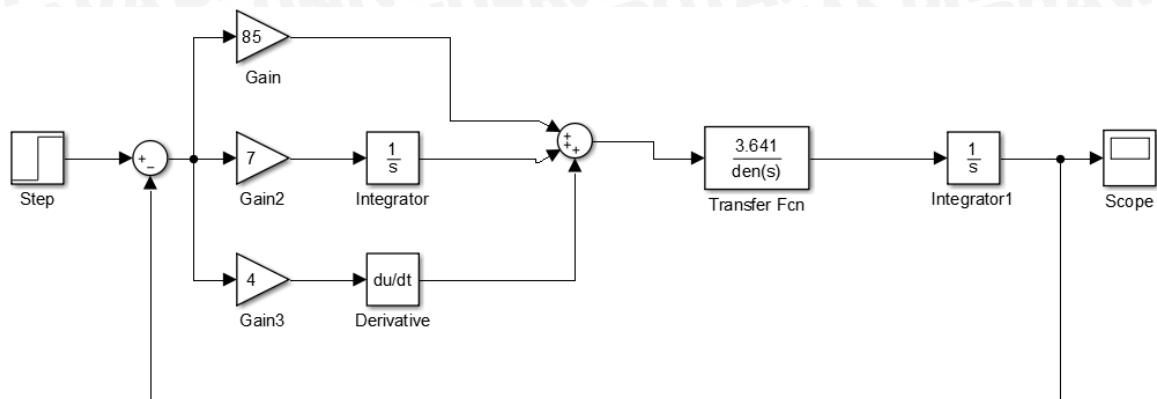
5.4.1. Peralatan yang Digunakan

1. Program Matlab dengan *toolbox simulink*.

5.4.2. Prosedur Percobaan

1. Menyusun diagram blok pada Matlab.
2. Memasukan nilai kontroler P, I dan D.
3. Memasukan fungsi alih motor DC yang telah didapatkan.
4. Menampilkan hasil dari simulasi melalui scope.
5. Menggunakan sinyal masukan step.

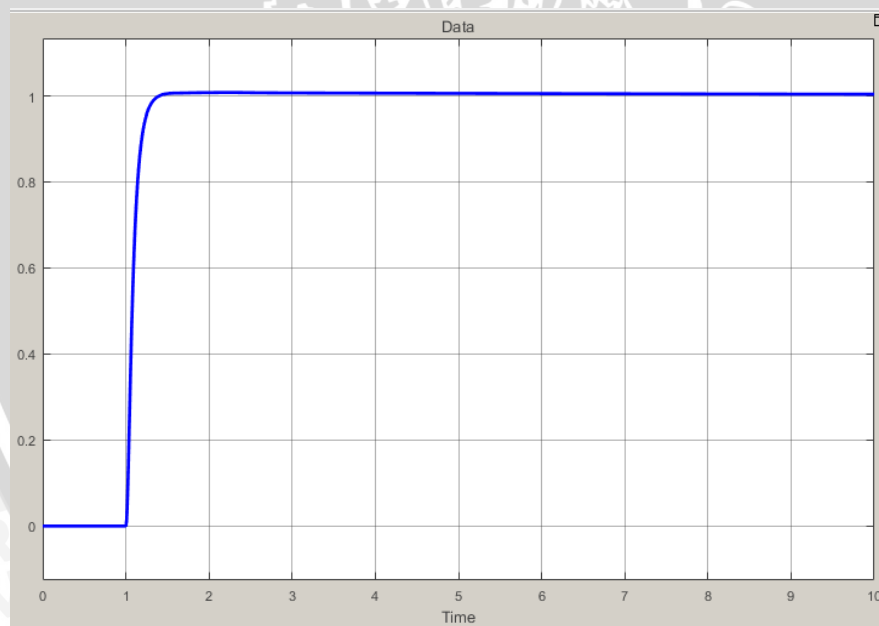
Setelah melakukan prosedur diatas maka akan didapatkan gambar seperti yang ditunjukkan gambar 5.7



Gambar 5.7 Diagram blok simulasi Matlab

5.4.3. Hasil Simulasi Matlab

Hasil simulasi sistem menggunakan Matlab ditunjukkan pada gambar 5.8



Gambar 5.8 Grafik simulasi respon sistem menggunakan Matlab

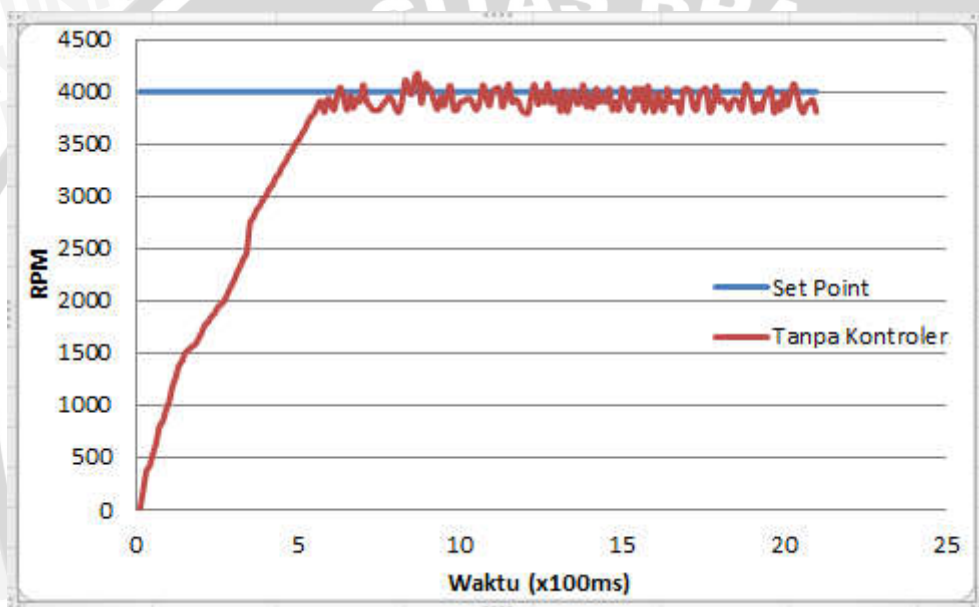
Dari diagram blok simulasi matlab pada gambar 5.7, didapatkan grafik respon pada gambar 5.8. Hasil simulasi pada matlab menghasilkan respon cukup baik.

5.5. Pengujian Blok Kontroler

Pengujian ini bertujuan untuk mengetahui bagaimana perbedaan respon sistem terhadap *setpoint* ketika menggunakan kontroler dan tanpa kontroler, sehingga dapat ditentukan bahwa sistem memerlukan sebuah sistem pengendalian.

5.5.1 Pengujian Tanpa Kontroler

Pengujian ini masih tetap dengan peralatan dan prosedur yang sama dengan pengujian motor DC, akan tetapi nilai parameter K_p , K_i , dan K_d tidak dimasukkan.



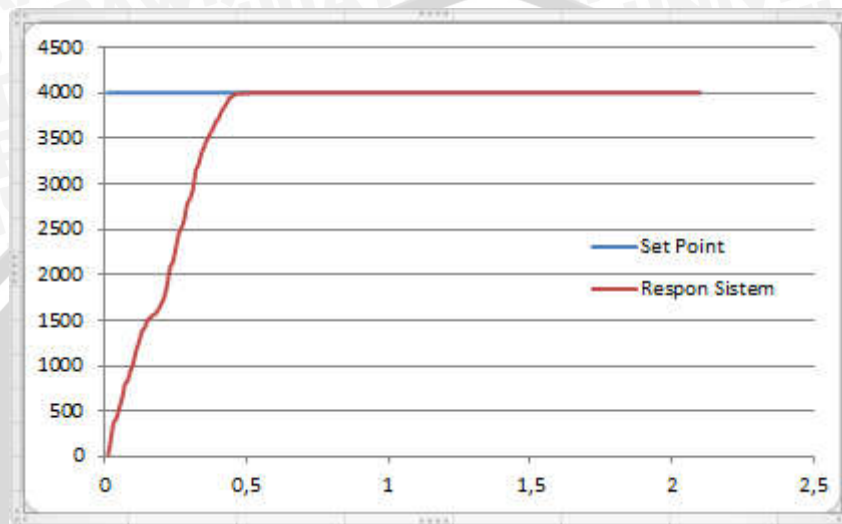
Gambar 5.9 Grafik Pengujian tanpa Kontroler dengan *Setpoint* 4000 RPM

Respon sistem saat tanpa kontroler menunjukkan bahwa sebagian besar kecepatan motor tidak dapat mencapai nilai *setpoint* yang sudah ditentukan yaitu 4000 RPM, seperti yang ditunjukkan pada gambar 5.9. Maka agar dapat mencapai *setpoint* perlu ditambahkan kontroler untuk mengendalikan kecepatan motor.

$$\begin{aligned} \%Ess &= \frac{3461,6-4000}{4000} \times 100\% \\ &= 13,46\% \end{aligned}$$

5.5.2 Pengujian Dengan Kontroler

Pengujian ini bertujuan untuk mengetahui respon sistem ketika diberi kontroler. Dengan alat dan langkah prosedur pengujian yang sama seperti pada pengujian motor dilakukan pengambilan data respon kecepatan motor ketika menggunakan kontroler.



Gambar 5.11 Grafik Pengujian dengan Kontroler *Setpoint* 4000 RPM

Pada gambar 5.11 terlihat bahwa apabila sistem menggunakan kontroler maka kecepatan dapat mencapai nilai *setpoint* yang ditentukan yaitu 4000 RPM. Dengan nilai parameter $K_p=85$, $K_i=7$, dan $K_d=4$ sehingga respon yang dihasilkan dapat mencapai keadaan yang diinginkan yaitu *setpoint* 4000 RPM.

5.6. Pengujian Keseluruhan Sistem

Pengujian ini bertujuan untuk mengetahui bagaimana kinerja sistem secara keseluruhan dan mengamati respon kontroler terhadap *setpoint* ketika mendapat beban beberapa cairan. Untuk pengujian dibutuhkan cairan dengan tingkat kekentalan yang berbeda. Pengujian menggunakan 3 jenis cairan yaitu Air sebagai beban 1, bahan bakar premium sebagai beban 2, pelumas mesin motor sebagai beban 3.

5.6.1 Peralatan yang Digunakan

1. Catu Daya 5v dan 12V
2. Motor DC
3. *Driver* Motor DC EMS H-Bridge 5A

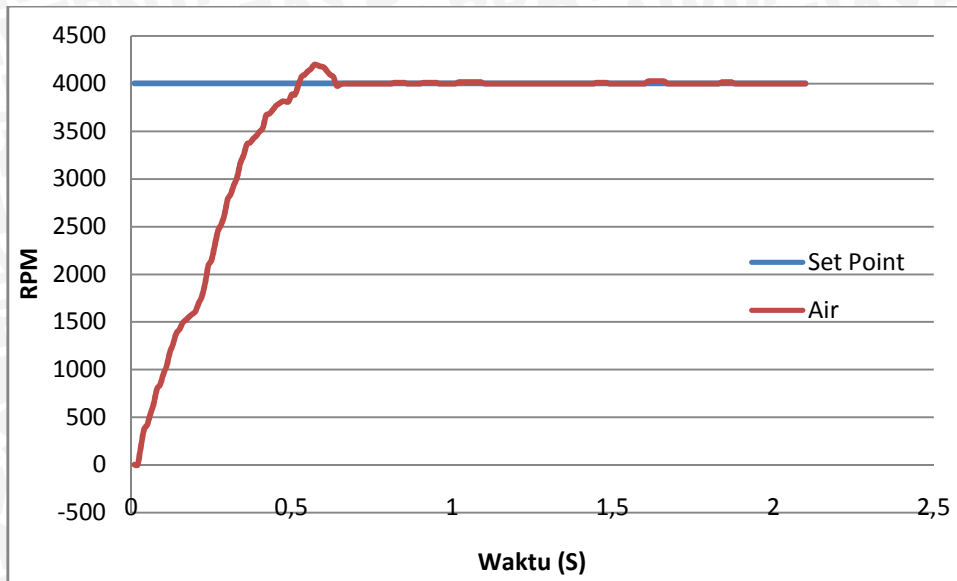
4. Program dan *Software* Arduino
5. Rangkaian sensor hall effect

5.6.2. Langkah Percobaan

1. Merangkai rangkaian seperti yang ditunjukkan pada gambar 5.5.
2. Mengunduh program dengan nilai $k_p = 85$, $k_i = 7$, $k_d = 4$ dan *setpoint* pada 4000 RPM pada *software* Arduino ERW 1.0.5.
3. Menyiapkan cairan yang akan digunakan sebagai gangguan.
4. Menampilkan keluaran nilai kecepatan RPM pada serial monitor *software* Arduino.
5. Mengamati hasil keluaran nilai RPM kemudian membuat grafik kecepatanputaran terhadap waktu.
6. Ulangi langkah percobaan dengan menggunakan cairan yang berbeda

5.6.3 Hasil Pengujian

Setelah melakukan prosedur pengujian, berikut ini grafik hasil dari pengujian dengan beban air dapat dilihat pada gambar 5.12.



Gambar 5.12 Grafik Pengujian Sistem dengan Beban Air

Pada grafik gambar 5.12, terjadi % error steady state sebesar:

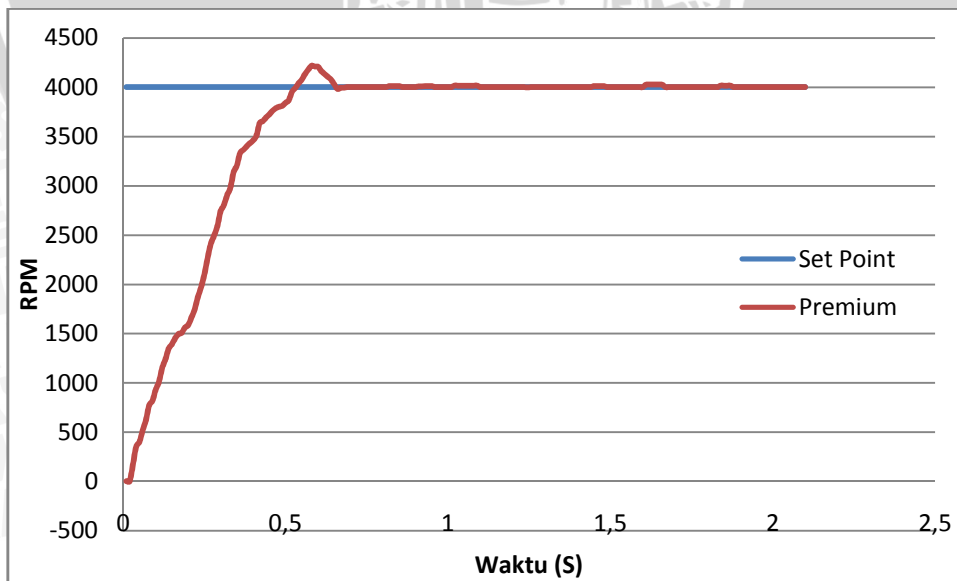
$$\%Ess = \frac{3995,92-4000}{4000} \times 100\%$$

$$= 0,1\%$$

$$\%Mp = \frac{4198-4000}{4000} \times 100\%$$

$$= 4,9\%$$

Grafik pengujian dengan menggunakan beban premium.

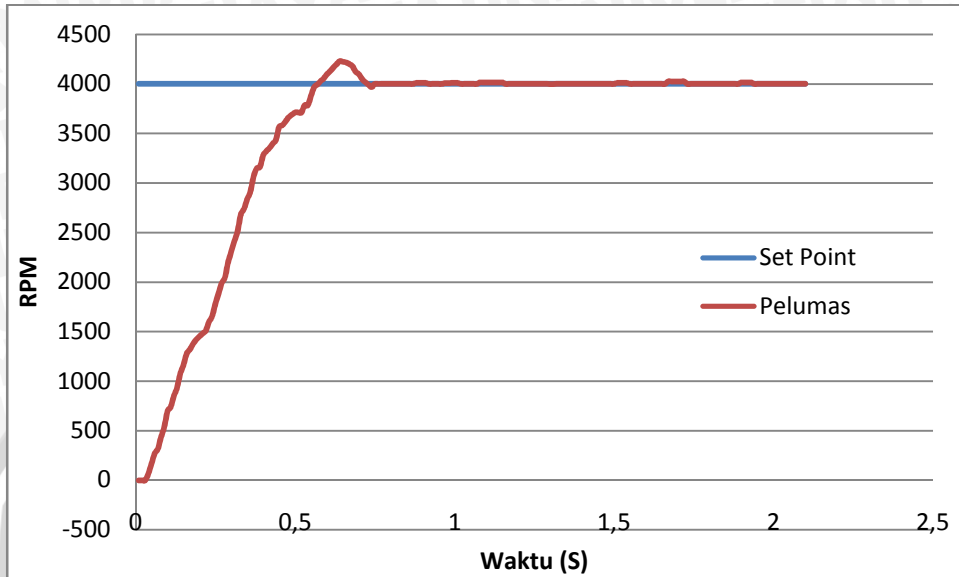


Gambar 5.13 Grafik Pengujian Sistem dengan Beban Premium.

Pada grafik gambar 5.13, terjadi % error steady state sebesar:



$$\begin{aligned}\%Ess &= \frac{3994,45-4000}{4000} \times 100\% \\ &= 0,13\% \\ \%Mp &= \frac{4206-4000}{4000} \times 100\% \\ &= 5,15\%\end{aligned}$$

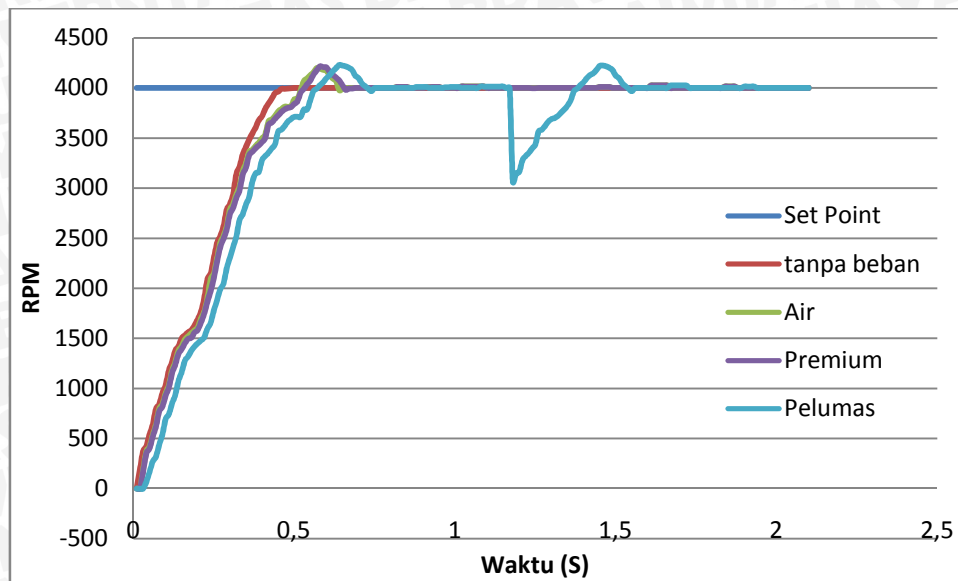


Gambar 5.14 Grafik Pengujian Sistem dengan Beban Pelumas Mesin Motor.

Pada grafik gambar 14, terjadi *% error steady state* sebesar:

$$\begin{aligned}\%Ess &= \frac{3978,586-4000}{4000} \times 100\% \\ &= 0,53\% \\ \%Mp &= \frac{4230-4000}{4000} \times 100\% \\ &= 5,75\%\end{aligned}$$

Berikut gambar 5.15 menunjukkan perbandingan grafik dari hasil pengujian dengan 3 jenis cairan yang berbeda.



Gambar 5.15 Respon Sistem Terhadap Ketiga Cairan

Dari grafik hasil pengujian keseluruhan pada gambar 5.15, dengan *set point* 4000 RPM, dapat dilihat respon sistem terhadap beban dari beberapa macam cairan. Terjadi perbedaan respon sistem untuk tiap jenis cairan, dikarenakan masa jenis setiap cairan berbeda, akan tetapi respon sistem mencapai setpoint saat mendapat beban cukup stabil. Meskipun *maximum overshoot* melebihi dari spesifikasi yang diberikan, *error system* keseluruhan tidak melebihi 5%. Ketika sistem mendapatkan gangguan, respon sistem untuk dapat kembali ke *setpoint* cukup baik.

BAB VI PENUTUP

6.1. Kesimpulan

Dari perancangan, pengujian dan pengamatan yang telah dilakukan pada penelitian sistem pengaturan kecepatan motor DC pada *peristaltic pump* dapat diambil kesimpulan sebagai berikut :

- 1) Dengan menggunakan kontroler PID metode *hand tuning*, *peristaltic pump* telah mampu mempertahankan kecepatan motor dengan beban yang berbeda-beda sesuai dengan set point yang sudah ditetapkan yaitu sebesar 4000 RPM.
- 2) Berdasarkan data respons sistem yang diperoleh dari pengujian dengan menggunakan metode *hand tuning*, didapatkan nilai $K_p = 85$, $K_i = 7$ dan $K_D = 4$.
- 3) Dari grafik hasil pengujian keseluruhan, dapat dilihat respon sistem terhadap beban dari beberapa macam cairan. Respon sistem untuk mencapai setpoint saat mendapat beban cukup baik, *error* sistem tidak melebihi 5% . Dengan begitu dapat dikatakan sistem kontrol pada perancangan ini telah bekerja dengan baik.

6.2. Saran

Dalam perancangan dan pembuatan alat ini masih terdapat kelemahan. Agar tercapainya penyempurnaan kinerja alat dan pengembangan lebih lanjut disarankan.

- Disarankan untuk penelitian selanjutnya agar mencoba menggunakan metode kontroler yang lain untuk mencapai respon yang lebih cepat dari sebelumnya.
- Disarankan untuk menggunakan motor dengan torsi yang lebih besar agar mampu memompa cairan dengan kekentalan yang lebih berat.
- Untuk penelitian lebih lanjut disarankan agar mencoba berbagai jenis cairan yang lebih variatif.

DAFTAR PUSTAKA

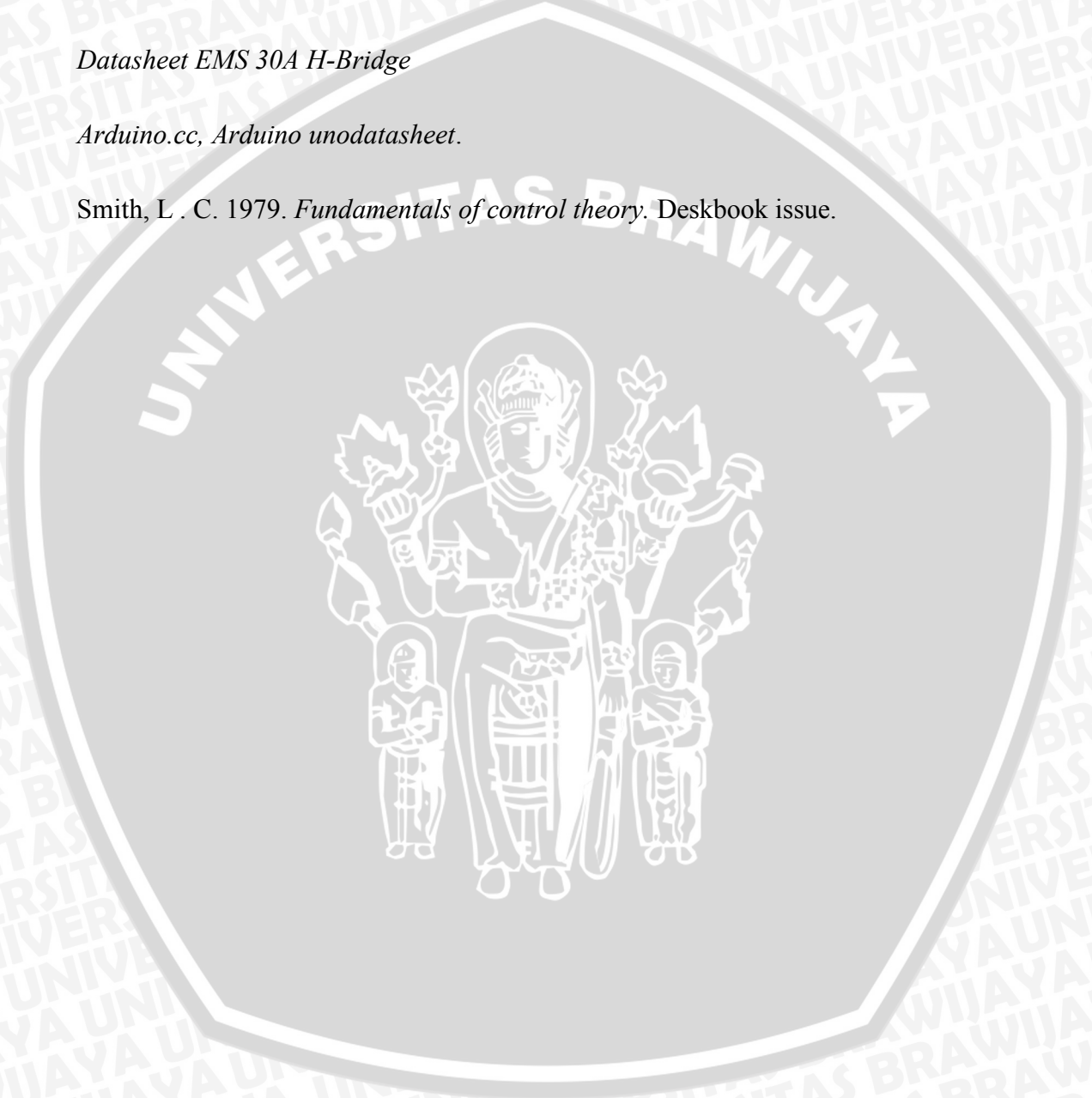
Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik (Sistem Pengaturan)*. Jakarta: Erlangga.

Amstrom, K. J., & Haggund, Tore. 1995 *PID Controllers: Theory, Design and Tuning*. Instrument Society of America: Research Triangle Park.

Datasheet EMS 30A H-Bridge

Arduino.cc, Arduino unodatasheet.

Smith, L . C. 1979. *Fundamentals of control theory*. Deskbook issue.



LAMPIRAN I

Foto Alat



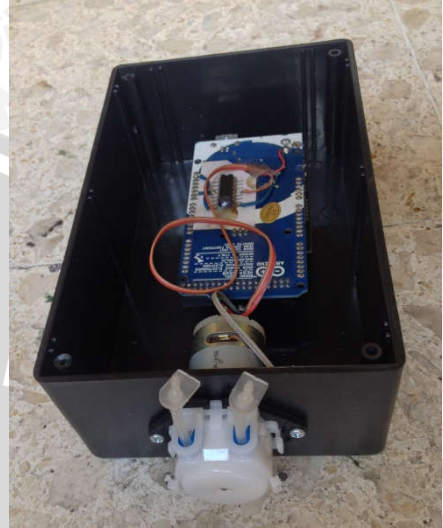
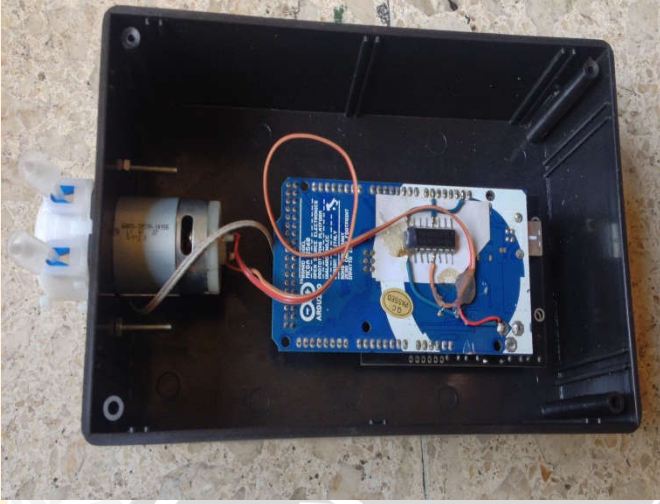
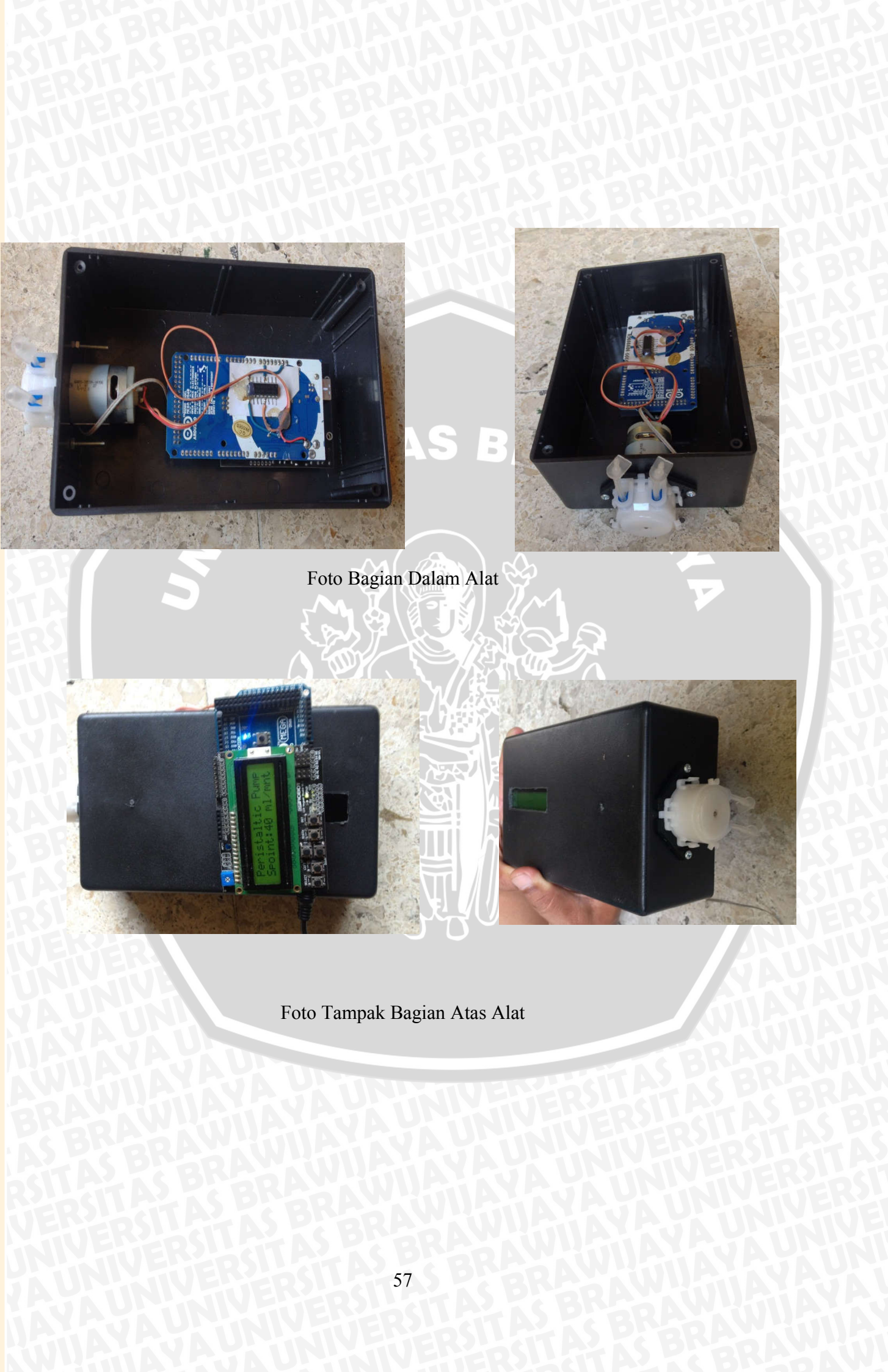


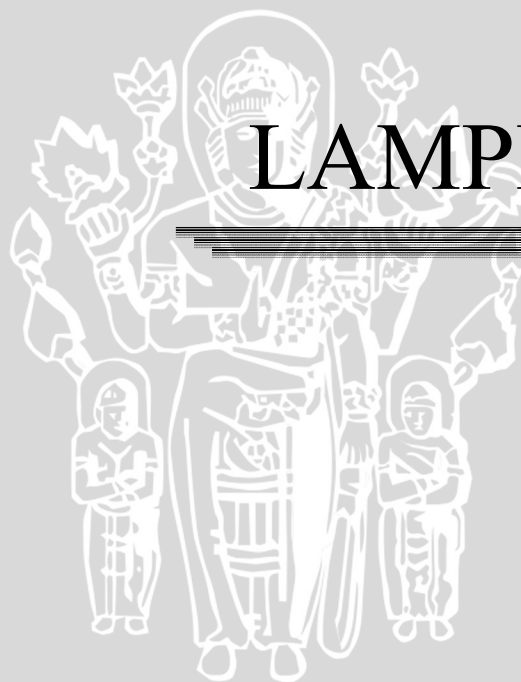
Foto Bagian Dalam Alat



Foto Tampak Bagian Atas Alat



UNIVERSITAS BRAWIJAYA

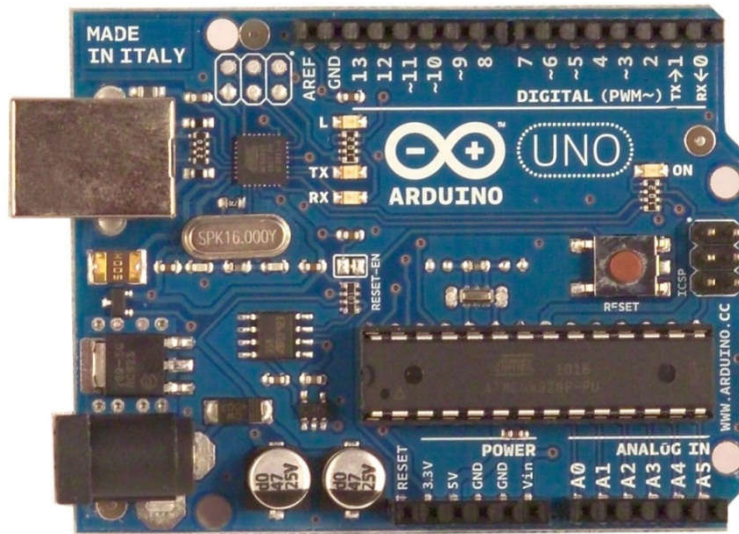


LAMPIRAN II

Data Sheet



Arduino UNO



Product Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.1 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V



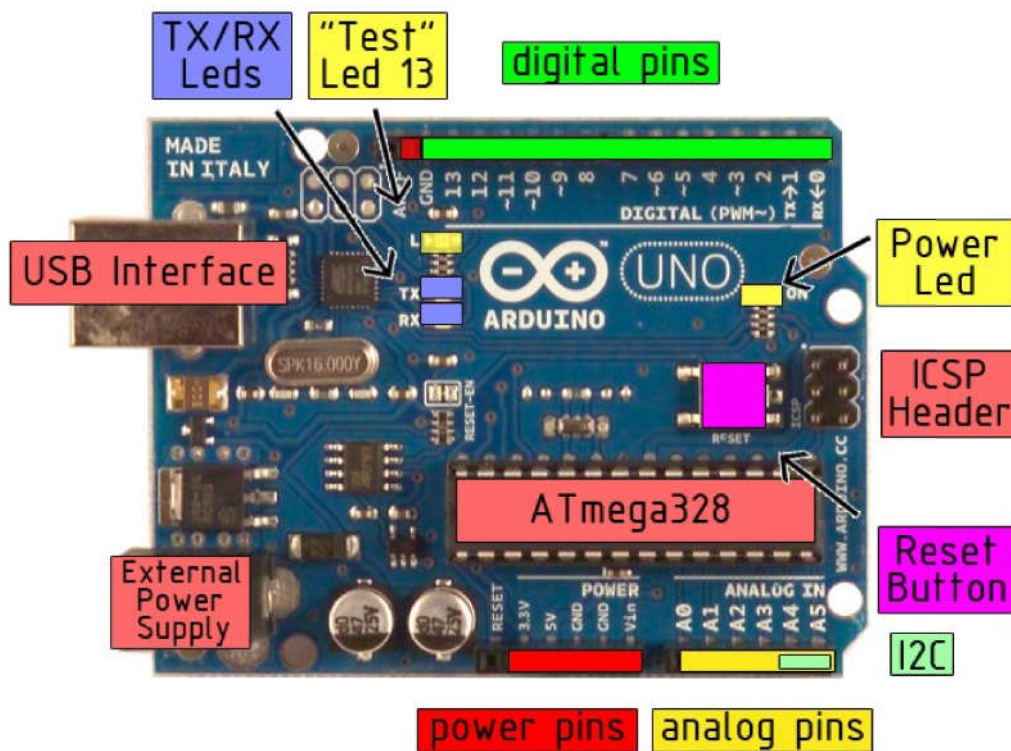
radiospares

RADIONICS



Input	6-
Voltage(limits)	20V
DigitalI/OPins	14 (of which 6 provide PWMoutput)
AnalogInputPins	6
DC Current per I/OPin	40
mA DC Current for 3.3VPin	50
mA	
Flash Memory	32 KB of which 0.5 KBusedby bootloader
SRAM	2KB
EEPROM	1KB
Clock Speed	16MHz

the board



Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50mA.
- **GND.** Ground pins.

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2KB of SRAM and 1KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output



radiospares

RADIONICS



Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **I²C: 4 (SDA) and 5 (SCL).** Support I²C (TWI) communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and Atmega328 ports](#).

**radiospares****RADIONICS**

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual COM port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

**radiospares****RADIONICS**

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).



radiospares

RADIONICS



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

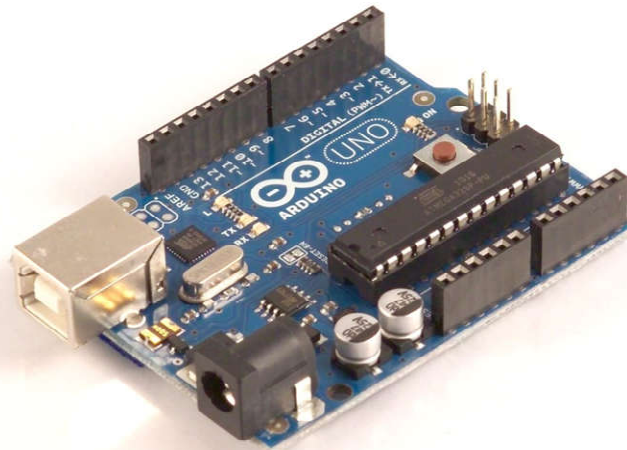
The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra

layer of protection. If more than 500mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics



radiospares

RADIONICS



The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>



radiospares

RADIONICS



Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>Arduin
o-
0017>Examples>Digital>
Blink**

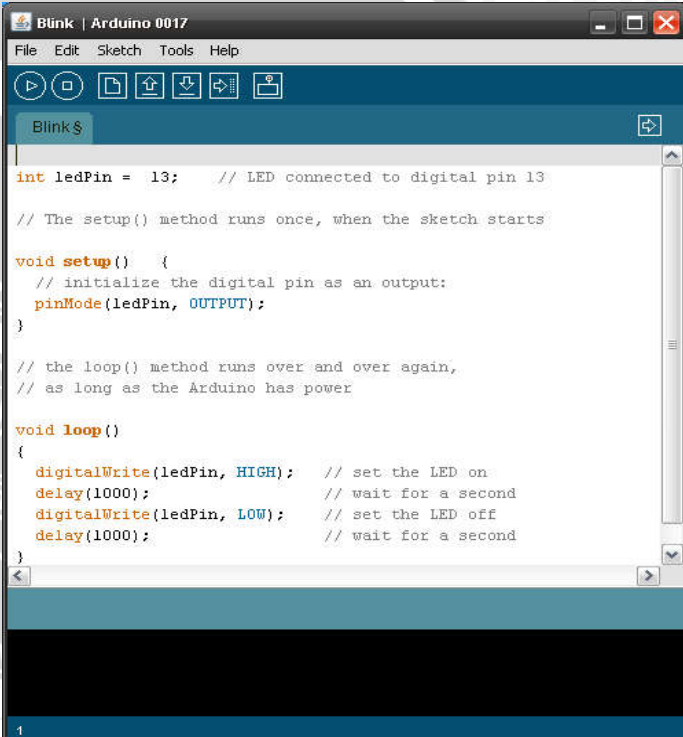
Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select

Now you have to go to

Tools>SerialPort

and select the right serial port, the one arduino is attached to.



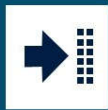
```

Blink | Arduino 0017
File Edit Sketch Tools Help
Blink
int ledPin = 13; // LED connected to digital pin 13
// The setup() method runs once, when the sketch starts
void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}
// the loop() method runs over and over again,
// as long as the Arduino has power
void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000); // wait for a second
}
1

```



Done compiling.



Upload

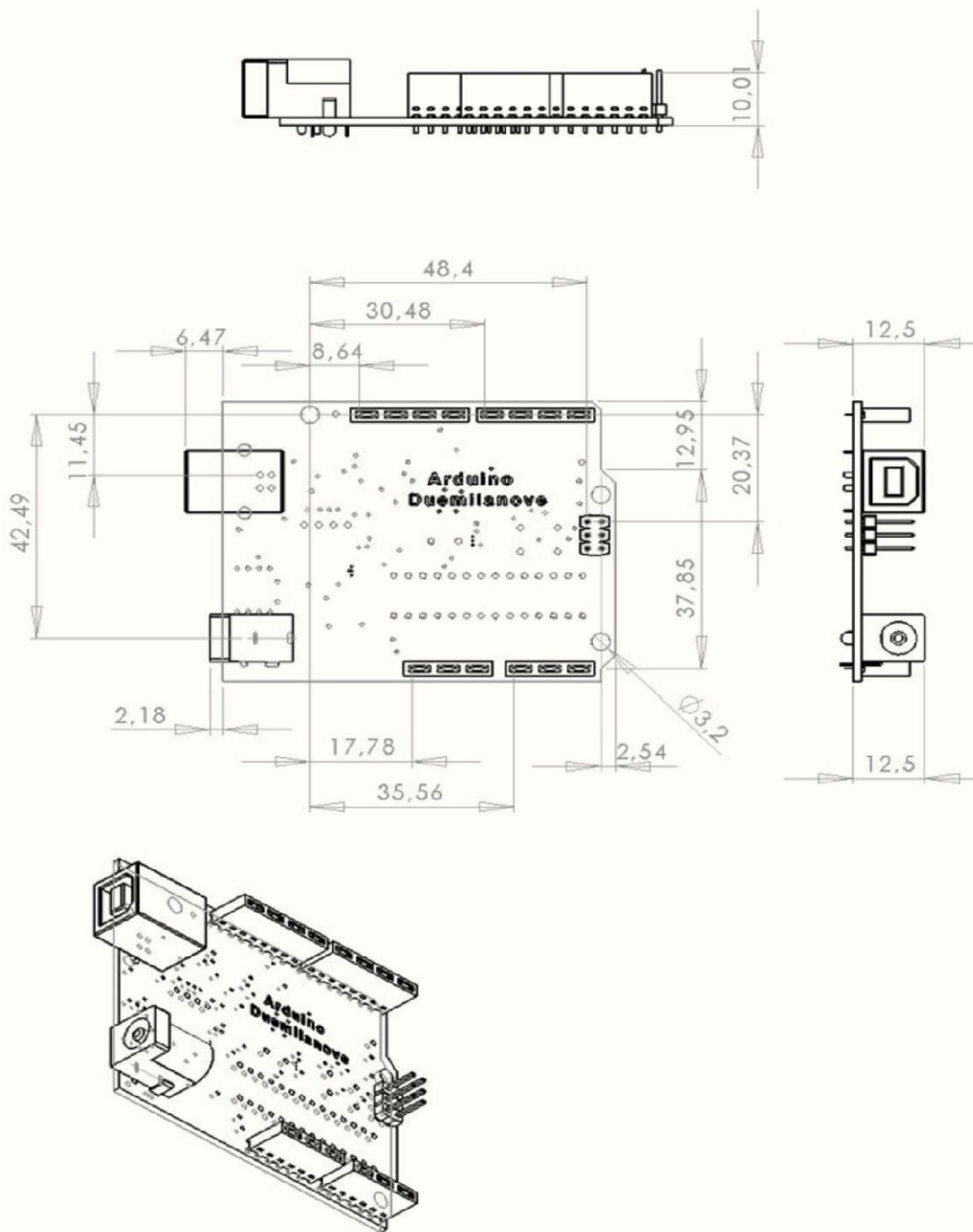


TX RX Flashing



Blinking Led!

Dimensioned Drawing



radiospares

RADIONICS



Terms & Conditions



1. Warranties

- 1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.
- 1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replace them. Replaced products shall be warranted for a new full warranty period.
- 1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
- 1.4 Customer agrees that prior to using any system that includes the producer's products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.
- 1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.
- 1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification



The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses incurred to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.





UNIVERSITAS BRAWIJAYA

LAMPIRAN III

Listing Program

/*.....*/

*

* NAMA : Taufiq Nor Ahmad

* NIM :135060309111014

* JUDUL :PENGONTROLAN MOTOR DC PADA PERISTALTIC PUMP
SEBAGAI PENGATUR FLOWRATE CAIRAN

*

/*.....*/

#include <PID_v1.h>

#include <LiquidCrystal.h>

double Setpoint;

double Input, Output;

double Kp = 85;

double Ki = 7;

double Kd = 4;

int dcMotor = 9;

long fb;

// Inisialisasi PID

PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);



```
// Inisialisasi pin LCD
```

```
LiquidCrystal lcd( 8, 9, 4, 5, 6, 7 );
```

```
int feedback()
```

```
{
```

```
    // baca feedback (RPM) adc kanal 0
```

```
    fb = analogRead(0);
```

```
    fb = (fb/1024)*5000;
```

```
    // konversi ke flowrate
```

```
    fb = (fb/10)*0.08792;
```

```
    lcd.setCursor(0, 1);
```

```
    lcd.print("Spnt: ");
```

```
    lcd.print(Setpoint);
```

```
    lcd.print(" ml/m");
```

```
}
```

```
void setup() {
```

```
    // set pid secara otomatis
```

```
    myPID.SetMode(AUTOMATIC);
```

```
    // setpoint dalam ml/menit
```

```
    Setpoint = 35.17;
```

```
}
```

```
void loop() {
```




```
// baca feedback  
feedback();  
myPID.Compute();  
analogWrite(dcMotor,Output);  
delay(50);  
}
```

