

repository.ub.ac.id

**METODE *ROOT LOCUS* UNTUK Mencari Parameter PID
PADA Pengendalian Posisi *Stamping Rod* Berbasis
Pneumatic Menggunakan Arduino Uno**

SKRIPSI

**Diajukan untuk Memenuhi Persyaratan
Memperoleh Gelar Sarjana Teknik**

UNIVERSITAS BRAWIJAYA



Disusun oleh :

**ADE AMRUCHLY YANA
NIM. 105060301111013 - 63**

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
JURUSAN TEKNIK ELEKTRO
MALANG
2014**



LEMBAR PERSETUJUAN

**METODE *ROOT LOCUS* UNTUK Mencari Parameter PID Pada
Pengendalian Posisi *Stamping Rod* Berbasis *Pneumatic*
Menggunakan Arduino Uno**

SKRIPSI

**Diajukan untuk Memenuhi Persyaratan
Memperoleh Gelar Sarjana Teknik**



Disusun oleh:

**ADE AMRUCHLY YANA
NIM. 105060301111013 - 63**

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

**M. Aziz Muslim, ST., MT., Ph.D.
NIP. 19741203 200012 1 001**

**Dr. Ir. Erni Yudaningtyas, MT.
NIP. 19650913 199002 2 001**

LEMBAR PENGESAHAN

**METODE *ROOT LOCUS* UNTUK Mencari Parameter PID pada
Pengendalian Posisi *Stamping Rod* Berbasis *Pneumatic*
Menggunakan Arduino Uno**

SKRIPSI

Disusun oleh:

ADE AMRUCHLY YANA
10506030111013 - 63

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 11 Juni 2014

MAJELIS PENGUJI

Rahmadwati, ST., MT., Ph.D.
NIP. 19771102 200604 2 003

Ir. Purwanto, MT.
NIP. 19540424 198601 1 001

Ir. Retnowati, MT.
NIP. 19511224 198203 2 001

Mengetahui
Ketua Jurusan Teknik Elektro

M. Aziz Muslim, ST., MT., Ph.D.
NIP. 19741203 200012 1 001

KATA PENGANTAR

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

Sungguh hanya melalui Pertolongan dan Perlindungan Allah SWT semata sehingga saya dapat menyelesaikan skripsi ini. Dengan seizin Allah SWT, di kesempatan yang baik ini saya ingin menghaturkan rasa terima kasih dan penghargaan yang sebesar besarnya atas bantuan sehingga terselesainya skripsi ini kepada:

- Orang tua tercinta, Bapak Suryono dan Ibu Siti Aminah Hariyati yang senantiasa memberi kasih sayang, dukungan, dan doa yang tiada akhir.
- Kakak tercinta Iradat Dwi Rahmawan, yang selalu memberi nasihat dan adik Putri Wijayanti yang selalu memberi dukungan serta Mbahibu yang tiada henti memberikan do'a, dan alm. Suryo Ardyansah semoga bangga melihat penerusmu berjuang di sini.
- Bapak Aziz Muslim, ST., MT., Ph.D selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya sekaligus sebagai dosen pembimbing yang telah banyak memberikan masukan, nasehat, pengarahan, motivasi, dan saran hingga selesainya skripsi ini.
- Bapak Ir. Purwanto, MT selaku KKDK Teknik Kontrol yang memberikan pengarahan, bimbingan dan motivasinya.
- Ibu Dr. Erni Yudaningtyas, ST., MT. selaku Ka. Lab Sistem Kontrol sekaligus sebagai dosen pembimbing atas segala kesempatan, nasehat, pengarahan, motivasi, saran dan masukan yang telah diberikan.
- Laboratorium Sistem Kontrol Teknik Elektro Universitas Brawijaya atas segala alat serta sarana dan prasarana yang dimanfaatkan penulis dalam melakukan penelitian.
- Keluarga besar angkatan 2010 "MAGNET'10" atas doa, semangat, dan kesempatan menjadi keluarga tanpa ikatan darah yang terbaik.
- Keluarga besar teman-teman laboratorium sistem kontrol Ayu, Hamu, Dito, Sendok, Khai, Dina, Neta, Ika, Hakiki, Tesu, Samid, Rozi, Emon, Zai, Azri, Mirza, mas Dharmo, Salmy, Adeck beserta mbak Eka Desiana, terima kasih telah memberikan banyak bantuan serta pengetahuan dalam belajar dan berkreasi.

- Workshop HME, Feby, Ayu serta kadiv-kadiv Dito, Abu, Hamu, Dewi, Safti, Shanty, Vicky dan Erwan serta seluruh teman-teman WS HME terima kasih telah memberi berbagai pengalaman, pengetahuan dan canda tawa.
- Teman-teman tim robot KRI, KRCI, KRSBI dan KRSI yang telah membantu dalam peminjaman alat serta sarana dan prasarana.
- Teman-teman seperjuangan kuliah ngopi Dito, Hamu, Dayat, Rara, Anas, Ika, Betmen, Sendok, Firman yang telah memberikan banyak inspirasi, canda tawa, saudara tanpa ikatan darah yang istimewa.
- Semua pihak yang telah memberikan serta dukungan baik secara langsung maupun tidak langsung atas penyusunan skripsi ini.

Sekiranya Allah SWT mencatat amalan ikhlas kami dan semua pihak yang turut membantu sehingga skripsi ini terselesaikan. Akhirnya, kami menyadari bahwa skripsi ini masih jauh dari sempurna namun semoga skripsi ini dapat memberikan manfaat bagi kita semua. Allahumma Amin.



Malang, 29 Mei 2014

Penulis

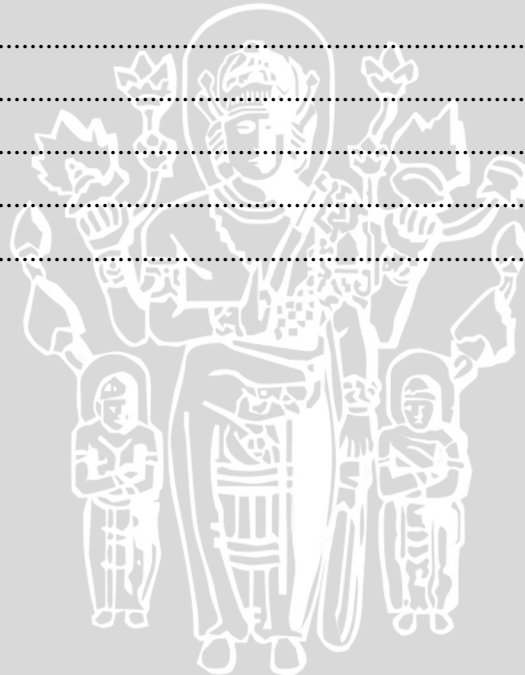


DAFTAR ISI

LEMBAR PERSETUJUAN	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR GAMBAR	viii
DAFTAR TABEL	x
ABSTRAK	xi
BABI PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA	5
2.1 <i>Stamping Rod</i>	5
2.2 <i>Sistem Pneumatic</i>	6
2.2.1 <i>Rotary Vane Compressor</i>	7
2.2.2 <i>Buffer Accumulator</i>	8
2.2.3 <i>Air-Filter Regulator</i>	8
2.2.4 <i>Pressure Regulator</i>	9
2.2.5 <i>Tubing, Fitting, dan Air Coupler</i>	10
2.2.6 <i>Silinder Double Acting</i>	11
2.3 <i>Electro-Pneumatic Regulator ITV3052-013B</i>	12
2.4 Sensor PING)))	14
2.5 <i>V/I Converter ISO-U1-P3-O1</i>	16
2.6 Kontroler	16
2.6.1 Kontroler Proporsional	17
2.6.2 Kontroler Integral	17
2.6.3 Kontroler Diferensial	18
2.6.4 Kontroler Proporsional Integral Diferensial	19
2.7 Metode Perancangan PID Menggunakan <i>Root Locus</i>	19
2.8 Arduino Uno	21
2.8.1 Daya	22
2.8.2 Memori	22

2.8.3	<i>Input dan Output</i>	22
2.8.4	Komunikasi	23
2.9	Konveyor.....	24
BAB III METODE PENELITIAN		25
3.1	Perancangan Sistem	25
3.2	Realisasi Pembuatan Sistem.	25
3.3	Pengujian dan Analisis Data	26
3.4	Pengambilan Kesimpulan dan Saran	27
BAB IV PERANCANGAN DAN PEMBUATAN ALAT		28
4.1	Perancangan Sistem	28
4.2	Blok Diagram Sistem	28
4.3.	Prinsip Kerja Sistem	29
4.4	Perancangan Perangkat Keras	30
4.4.1	Spesifikasi Alat.	30
4.4.2	Kompresor Hydrovane HV-01	31
4.4.3	<i>Buffer Accumulator</i>	32
4.4.4	<i>Air Filter</i>	32
4.4.5	<i>Electro-Pneumatic Regulator</i> ITV3051-013B	32
4.4.6	<i>Pressure Regulator</i>	33
4.4.7	<i>Silinder Double Acting</i>	34
4.4.8	<i>V/I Converter</i> ISO U1-P3-O1	36
4.4.9	Digital Analog <i>Converter</i> (DAC) PCF 8591	37
4.4.10	<i>Liquid Crystal Display</i> (LCD)	38
4.4.11	Sensor PING))	39
4.4.12	Konveyor	40
4.5	Perancangan Kontroler.....	41
4.6	Modul Arduino Uno	44
4.7	Perancangan Perangkat Lunak	46
BAB V PENGUJIAN DAN ANALISIS		47
5.1	Pengujian Digital Analog <i>Converter</i> PCF 8591.	47
5.1.1	Peralatan Pengujian	47
5.1.2	Prosedur Pengujian	47
5.1.3	Hasil Pengujian	47
5.2	Pengujian <i>V/I Converter</i>	49
5.2.1	Peralatan Pengujian	49

5.2.2	Prosedur Pengujian	49
5.2.3	Hasil Pengujian	50
5.3	Pengujian <i>Electro-Pneumatic Regulator</i>	51
5.3.1	Peralatan Pengujian	51
5.3.2	Prosedur Pengujian	51
5.3.3	Hasil Pengujian	52
5.4	Pengujian Sensor PING)))	54
5.4.1	Peralatan Pengujian	54
5.4.2	Prosedur Pengujian	54
5.4.3	Hasil Pengujian	55
5.5	Pengujian Sistem Secara Keseluruhan	56
5.5.1	Peralatan Pengujian	60
5.5.2	Prosedur Pengujian	60
5.5.3	Hasil Pengujian	61
BAB VI PENUTUP		66
6.1	Kesimpulan	66
6.2	Saran	66
DAFTAR PUSTAKA		67



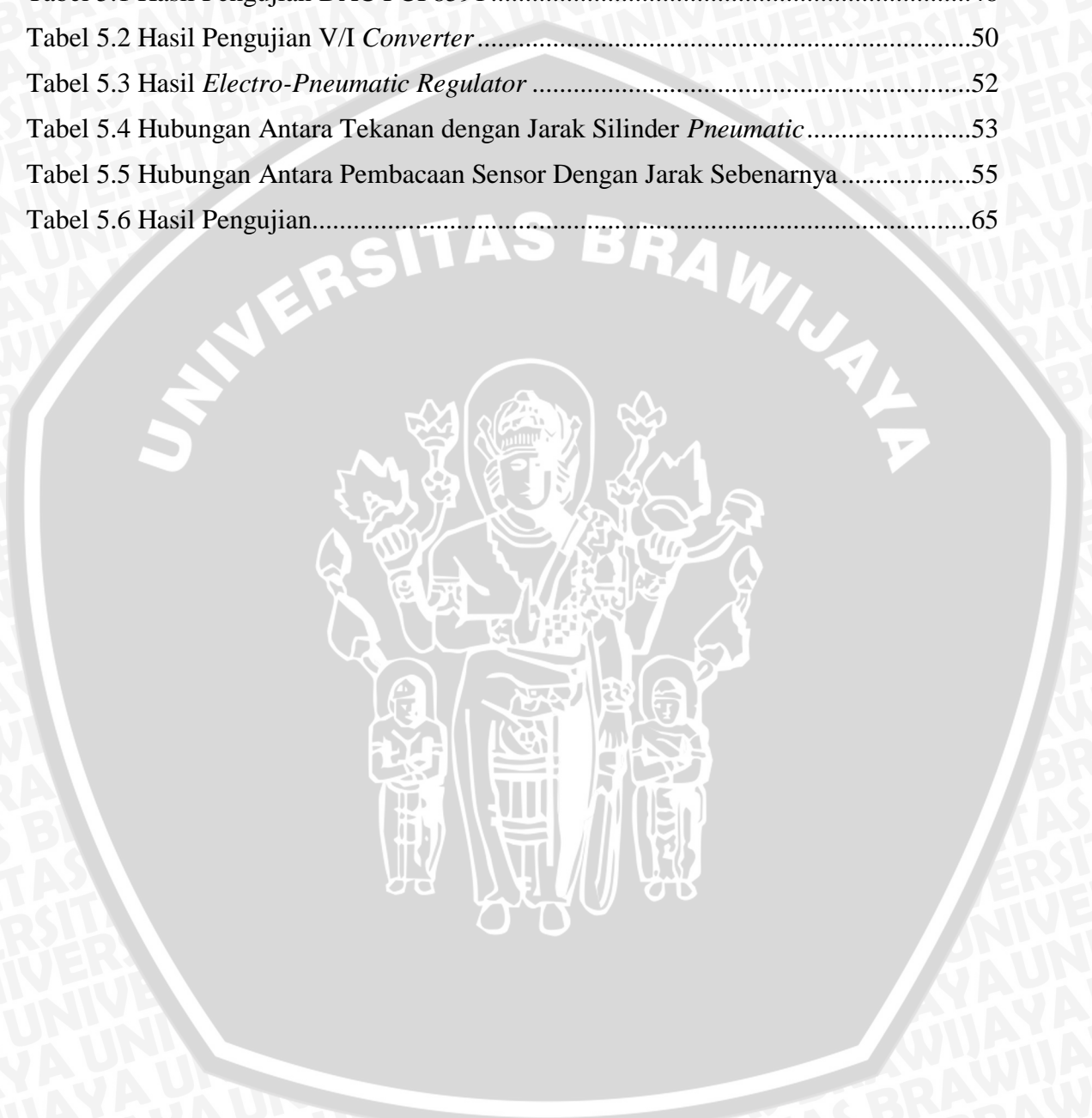
DAFTAR GAMBAR

Gambar 2.1	Diagram Alir Sistem dan Elemen-elemennya.....	7
Gambar 2.2	<i>Rotary Vane Compressor</i>	7
Gambar 2.3	<i>Air Filter-Regulator</i>	9
Gambar 2.4	Komponen <i>Pressure Regulator</i>	10
Gambar 2.5	<i>Tubing Pneumatic</i>	10
Gambar 2.6	<i>Fitting Pneumatic</i>	11
Gambar 2.7	<i>Air Coupler</i>	11
Gambar 2.8	Silinder Pneumatik.....	12
Gambar 2.9	Skematik ITV3052-013B.....	13
Gambar 2.10	Rangkaian Kabel TV3501-013B	13
Gambar 2.11	Sensor Ultrasonik PING))).....	14
Gambar 2.12	Diagram Waktu Sensor PING))).....	14
Gambar 2.13	Prinsip Kerja Sensor PING))).....	15
Gambar 2.14	IC ISO-U1-P3-O1.....	16
Gambar 2.15	Diagram Blok Kontroler Proporsional	17
Gambar 2.16	Diagram Blok Kontroler Integral	18
Gambar 2.17	Diagram Blok Kontroler Diferensial	18
Gambar 2.18	Diagram Blok Kontroler PID	19
Gambar 2.19	Sistem Kendali <i>Loop</i> Tertutup.....	19
Gambar 2.20	Arduino Uno	21
Gambar 2.21	Konveyor	24
Gambar 4.1	Diagram Blok Sistem.....	28
Gambar 4.2	Skema Keseluruhan Sistem.....	31
Gambar 4.3	<i>Buffer Accumulator</i>	32
Gambar 4.4	<i>Electro-Pneumatic Regulator</i> ITV3051-013B	33
Gambar 4.5	Rangkaian <i>Wiring</i> ITV 3051-013B.....	33
Gambar 4.6	<i>Pressure Regulator</i>	34
Gambar 4.7	Silinder <i>Double Acting</i> SMC CDM2b25-300.....	34
Gambar 4.8	Analogi Silinder <i>Double Acting</i>	35
Gambar 4.9	Rangkaian <i>Wiring</i> ISO U1-P3-01	36
Gambar 4.10	IC PCF8591	37
Gambar 4.11	Rangkaian PCF8591	37
Gambar 4.12	Program DAC PCF8591 ke Arduino Uno.....	38

Gambar 4.13 IC LCD ke Arduino Uno	39
Gambar 4.14 Rangkaian <i>Wiring</i> LCD ke Arduino Uno	39
Gambar 4.15 Konveyor	41
Gambar 4.16 Grafik Respon Tanpa Kontroler	41
Gambar 4.17 Penulisan Program <i>Root Locus</i> pada MATLAB 2010	42
Gambar 4.18 Letak <i>Pole</i> pada Diagram <i>Root Locus</i>	42
Gambar 4.19 Desain Sistem Arduino Uno	44
Gambar 4.20 Modul Arduino Uno	45
Gambar 4.21 <i>Flowchart</i> Program.....	46
Gambar 5.1 Program PCF8591	48
Gambar 5.2 Grafik Hubungan PWM dan Tegangan	49
Gambar 5.3 Program PCF8591	50
Gambar 5.4 Grafik Hubungan Tegangan dan Arus	51
Gambar 5.5 Program PCF8591	52
Gambar 5.6 Grafik Hubungan Tekanan dan Arus	53
Gambar 5.7 Grafik Hubungan Tekanan dan Jarak.....	54
Gambar 5.8 Grafik Hubungan Pembacaan Sensor dan Jarak Sebenarnya	56
Gambar 5.9 Grafik Respon Sistem dengan Nilai $K_p= 0,2002$, $K_i= 0,1$, $K_d= 0,1018$...	57
Gambar 5.10 Grafik Respon Sistem dengan Nilai $K_p= 2,3166$, $K_i= 0,5$, $K_d= 1,0182$...	57
Gambar 5.11 Grafik Respon Sistem dengan Nilai $K_p= 3,0256$, $K_i= 1,5$, $K_d= 1,5274$...	58
Gambar 5.12 Grafik Respon Sistem dengan Nilai $K_p= 5,0438$, $K_i= 2,5$, $K_d= 2,5456$...	58
Gambar 5.13 Grafik Respon Sistem dengan Nilai $K_p= 10,089$, $K_i= 5$, $K_d= 5,0912$	59
Gambar 5.14 Grafik Respon Sistem dengan $K_p= 2.3166$, $K_i= 0.5$ $K_d= 1.0182$ pada MATLAB 2010	60
Gambar 5.15 Rangkaian Pengujian <i>Setpoint</i> Berbeda	61
Gambar 5.16 Hasil Respon dengan <i>Setpoint</i> 10cm	61
Gambar 5.17 Hasil Respon dengan <i>Setpoint</i> 15cm	62
Gambar 5.18 Hasil Respon dengan <i>Setpoint</i> 20cm	63
Gambar 5.19 Hasil Respon dengan <i>Setpoint</i> 25cm	64

DAFTAR TABEL

Tabel 4.1 Parameter PID dengan $s = -0,991$	44
Tabel 4.2 Fungsi Pin Arduino Uno.....	45
Tabel 5.1 Hasil Pengujian DAC PCF8591	48
Tabel 5.2 Hasil Pengujian V/I Converter	50
Tabel 5.3 Hasil <i>Electro-Pneumatic Regulator</i>	52
Tabel 5.4 Hubungan Antara Tekanan dengan Jarak Silinder <i>Pneumatic</i>	53
Tabel 5.5 Hubungan Antara Pembacaan Sensor Dengan Jarak Sebenarnya.....	55
Tabel 5.6 Hasil Pengujian.....	65



ABSTRAK

Ade Amruchly Yana, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, Juni 2014, Metode *Root Locus* untuk Mencari Parameter PID pada Pengendalian Posisi *Stamping Rod* Berbasis *Pneumatic* Menggunakan Arduino Uno, Dosen Pembimbing: M. Aziz Muslim, ST., MT., Ph.D, Dr. Ir. Erni Yudaningtyas, MT.

Industri pada zaman sekarang ini, segalanya dituntut serba cepat untuk memenuhi kebutuhan konsumen. Salah satunya proses *stamping* pada kemasan produk. Dibutuhkan keefektifan dan keakurasian dalam pengerjaannya. Dengan semakin tingginya permintaan, dibutuhkan sebuah sistem pneumatik untuk *stamping rod* yang bisa digunakan pada berbagai macam produk dengan ukuran berbeda secara otomatis. Salah satu solusi untuk memecahkannya ialah membuat sistem pneumatik yang dikendalikan secara otomatis oleh PID menggunakan kontroler Arduino Uno. Skripsi ini menggunakan metode *root locus* untuk menentukan parameter PID-nya.

Hasil perancangan parameter PID menggunakan metode *root locus* didapatkan bahwa terdapat akar berada disebelah kiri bidang s , sehingga respon yang didapat bisa stabil. Dengan nilai *pole* $s = -0,991$ didapatkan parameter terbaik $K_p=2.3166$, $K_i=0.5$ dan $K_d=1.0182$. Berdasarkan hasil pengujian dengan 4 macam *setpoint* yang berbeda-beda (10cm, 15cm, 20cm, dan 25cm) diperoleh *settling time* (ts) rata-rata sebesar 1,75 detik dan *error* rata-rata sebesar 0,892%, di bawah toleransi *error steady state* 5%.

Kata kunci: *Root locus*, PID, Arduino Uno, *stamping rod*, *pneumatic*.

BAB I PENDAHULUAN

1.1. Latar Belakang

Saat ini teknologi otomatis telah berkembang pesat. Dalam dunia industri, hampir semua perusahaan menerapkan otomatisasi pada alat-alatnya untuk menunjang proses produksi yang cepat, efisien, dan tidak membutuhkan banyak tenaga kerja. Berkat kemajuan teknologi kini udara juga dapat dimanfaatkan dalam berbagai proses produksi.

Pneumatik ([bahasa Yunani](#): *pneumatikos*) berasal dari kata dasar "pneu" yang berarti [udara](#) tekan dan "matik" yang berarti [ilmu](#) atau hal-hal yang berhubungan dengan sesuatu, sehingga arti lengkap pneumatik adalah ilmu/hal-hal yang berhubungan dengan udara bertekanan (Wikipedia, 2013). Sistem pneumatik adalah teknologi yang memanfaatkan udara terkompresi untuk menggerakkan aktuator ataupun sebagai pengganti sinyal kontrol. Dikarenakan menggunakan udara sebagai medianya, sistem pneumatik mempunyai suhu yang rendah. Sistem pneumatik sederhana terdiri dari kompresor, *valve*, dan *tubing* (Zen, 2013).

Tekanan udara merupakan tenaga yang bekerja untuk menggerakkan massa udara dalam setiap satuan luas tertentu. Udara tekan atau udara terkompresi banyak digunakan pada sistem pneumatik pada industri dikarenakan mampu bekerja dalam waktu yang lama. Udara sebagai fluida kerja pada sistem pneumatik memiliki karakteristik khusus, antara lain: jumlahnya tak terbatas, mencari tekanan yang lebih rendah, memberi tekanan yang besarnya sama ke segala arah, mempunyai berat, dan tidak mempunyai bentuk (Palinggi, 2006).

Sistem pneumatik mempunyai berbagai keunggulan, yaitu: udara mudah diperoleh, udara bertekanan mudah dipindahkan melalui sistem perpipaan, udara bertekanan tidak banyak dipengaruhi temperatur, udara bertekanan bersih dan tidak dapat terbakar, dan dapat menghasilkan gerakan yang cepat. Namun, sistem pneumatik juga mempunyai beberapa kekurangan, antara lain: daya yang dihasilkan relatif kecil, menimbulkan suara berisik, dan biaya instalasi awal yang mahal. Beberapa contoh aplikasi sistem pneumatik adalah pada: pengecapan (*stamping*), pengisian (*filling*), pencetakan (*embossing*), pengepressan, *material conveying*, dan lain-lain.

Pada sebagian industri, industri makanan khususnya, tanggal produksi dan tanggal kadaluarsa wajib dicantumkan pada kemasan produk. Pemberian cap tanggal atau *stamping* akan sangat merepotkan apabila dilakukan secara manual oleh karena itu umumnya digunakan *stamping rod*. *Stamping rod* yang posisinya dapat dikontrol selain memudahkan proses *stamping* dari sisi biaya juga lebih murah dibanding menggunakan sistem *stamping rod* konvensional, karena untuk ukuran kemasan yang berbeda hanya dibutuhkan satu *stamping rod*.

Pada penelitian sebelumnya (Zen, 2013), kontroler yang digunakan adalah *Distributed Control System* (DCS). Respon keluarannya dapat mencapai *setpoint* dan tidak terdapat *error steady state*, namun *settling time* yang dibutuhkan cukup lama mencapai 900 detik. Selain itu, terdapat juga penelitian (Budi, 2013) yang menggunakan Arduino Uno sebagai kontrolernya dan menggunakan metode *hand tuning* untuk menentukan parameter PID. *Settling time* yang dibutuhkan cukup cepat yaitu 2 detik dan tidak terdapat *error steady state*, namun tidak dapat digunakan dengan *setpoint* yang berbeda-beda. Sistem pneumatik ini pada dasarnya memang terdapat *disturbance* berupa sambungan antar saluran udara serta angin yang bertekanan pada silindernya.

PID adalah kontroler yang merupakan gabungan dari kontroler proposional, kontroler integral dan kontroler diferensial. Gabungan dari ketiga kontroler ini diharapkan agar mendapat keluaran sistem yang mempunyai *settling time* yang cepat, tidak ada *error*, tidak ada *overshoot* dan dapat digunakan pada *setpoint* yang berbeda-beda.

Pada skripsi ini akan dibuat suatu miniatur pengendalian posisi *stamping rod* menggunakan Proporsional Integral Diferensial (PID) dan dilengkapi dengan metode *root locus* untuk menentukan parameternya.

Root locus merupakan metode yang menunjukkan kestabilan sistem berupa grafik. Grafik *root locus* dipengaruhi oleh penambahan *pole* dan *zero* pada fungsi alih. Dari metode *root locus* ini akan muncul parameter PID.

1.2. Rumusan Masalah

1. Bagaimana menemukan parameter PID menggunakan metode *root locus* untuk pengendalian *stamping rod* berbasis *pneumatic* menggunakan Arduino Uno?

2. Bagaimana merancang *hardware* dan *software* sistem pengendalian posisi *stamping rod* berbasis *pneumatic* menggunakan Arduino Uno?

1.3. Batasan Masalah

1. Sistem pneumatik yang digunakan adalah miniatur.
2. Pembahasan ditekankan pengendalian posisi *stamping rod* pada sistem dan perhitungan *root locus* untuk mengetahui parameter PID.
3. Pengaturan yang dilakukan adalah pengaturan tekanan udara melalui pengaturan arus yang mengalir ke *stamping rod*.
4. Tekanan udara yang digunakan berkisar 0-3 bar.

1.4. Tujuan

1. Merancang parameter PID dengan metode *root locus* agar dapat digunakan pada *setpoint* yang berbeda-beda sehingga sistem dapat berjalan efisien.
2. Merancang pengendalian posisi *stamping rod* berbasis *pneumatic* menggunakan Arduino Uno.

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 adalah sebagai berikut:

BAB I Pendahuluan

Menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

BAB II Tinjauan Pustaka

Menjelaskan teori dasar yang berisi penjelasan tentang teori *stamping rod*, sistem *pneumatic*, *rotary vane compressor*, *buffer accumulator*, *air-filter regulator*, *pressure regulator*, *tubing*, *fitting*, *air coupler*, *silinder double acting*, *electro-pneumatic regulator*, sensor (PING)), *converter ISO-U1-P3-01*, kontroler PID, *root locus*, Arduino Uno dan konveyor.

BAB III Metode Penelitian

Menjelaskan tentang metodologi penelitian yang terdiri atas perancangan alat, pembuatan alat, pengujian alat, serta pengambilan kesimpulan dan saran.

BAB IV Perancangan dan Pembuatan Alat

Menjelaskan tentang perancangan dan pembuatan alat yang meliputi prinsip kerja alat, perancangan perangkat keras dan perangkat lunak.

BAB V Pengujian dan Analisis

Menjelaskan tentang pengujian alat dan analisa yang meliputi pengujian bagian blok sistem secara keseluruhan.

BAB VI Penutup

Menjelaskan tentang pengambilan kesimpulan sesuai dengan hasil perealisasiian dan pengujian alat sesuai dengan tujuan dan rumusan masalah, serta pemberian saran untuk pengembangan.



BAB II

TINJAUAN PUSTAKA

Dalam memudahkan untuk memahami cara kerja rangkaian maupun dasar-dasar perencanaan dari sistem yang dibuat, maka perlu adanya penjelasan dan uraian mengenai teori penunjang yang digunakan dalam penulisan tugas akhir ini. Teori penunjang yang akan dijelaskan dalam bab ini adalah:

- *Stamping rod*
- Sistem pneumatik
- *Electro-pneumatic regulator*
- Sensor PING)))
- *V/I Converter*
- Kontroler
- Arduino Uno
- Konveyor

2.1 *Stamping Rod*

Pada kemasan berbagai produk pencantuman tanggal produksi dan tanggal kadaluarsa umum dilakukan. Semakin tua umur produk tentunya berpengaruh terhadap kualitas produk dalam kemasan tersebut oleh tanggal produksi dan tanggal kadaluarsa perlu diketahui konsumen. Umumnya pada pabrik-pabrik proses *stamping* sudah dilakukan menggunakan silinder sebagai lengan mekanik yang bergerak secara linier untuk mencapai objek yang diberi cap (Zen, 2013).

Silinder banyak digunakan sebagai aktuator pada pabrik baik skala kecil maupun besar. Pada umumnya, silinder tersebut bergerak menggunakan sistem pneumatik. Keunggulan sistem ini adalah aktuator dapat bergerak dengan cepat tanpa menimbulkan panas. Salah satu aplikasinya yaitu pada *stamping rod*. *Stamping rod* adalah peralatan mekanis yang bergerak secara translasi dan berfungsi untuk menandai tanggal kadaluarsa pada suatu produk, umumnya produk makanan dan minuman.

Dalam satu pabrik produk yang dihasilkan tidak hanya satu ukuran. Hal ini dapat menimbulkan permasalahan mengingat silinder pneumatik yang digunakan pada *stamping rod* umumnya hanya bisa bergerak maju secara penuh atau mundur

secara penuh, tidak dapat berhenti di tengah, sehingga untuk melakukan *stamping* pada dua kemasan produk yang berbeda ukuran harus digunakan dua silinder dengan ukuran *stroke* yang berbeda. Dengan alasan tersebut maka dirancanglah pengendalian posisi silinder agar dapat mengoptimalkan penggunaan silinder. Prinsipnya dengan cara menabrakkan udara bertekanan pada kedua sisi silinder *double acting*, sehingga panjang lengan silinder yang keluar dapat diatur sesuai dengan *setpoint*.

2.2 Sistem Pneumatik

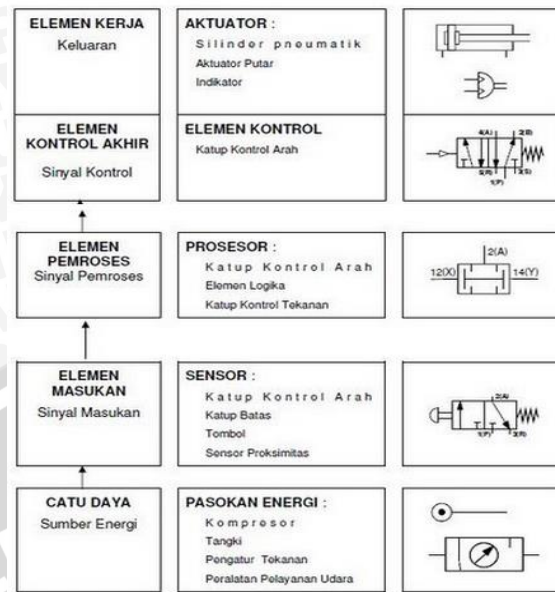
Pneumatik adalah teknologi yang berkenaan dengan penggunaan udara tekan untuk menghasilkan gerak mekanik. Sistem pneumatik biasanya banyak digunakan di industri, hal ini dikarenakan udara terkompresi yang digunakan untuk menggerakkan alat mekanik dapat menghasilkan daya gerak yang lebih murah, lebih aman, lebih fleksibel, dan lebih handal dibanding sejumlah motor elektrik dan aktuator.

Dalam suatu sistem kontrol pneumatik terdapat arsitektur dan bagian-bagian yang menyangkut fungsi kerja alat tersebut. Perancangan sistem kontrol pneumatik mengacu pada diagram alir sistem.

Tata letak komponen pneumatik yang akan digunakan harus disesuaikan dengan diagram alir sistem, yaitu sebuah sinyal harus mulai mengalir dari bawah menuju ke atas dari gambar diagram tersebut.

Diagram alir sistem dan elemen-elemennya ditunjukkan dalam Gambar

2.1.



Gambar 2.1 Diagram Alir Sistem dan Elemen-elemennya

Sumber : <http://monangcidakkal.blogspot.com/2013/04/sistem-pneumatik.html>

2.2.1 Rotary Vane Compressor

Kompresor adalah alat yang digunakan untuk menaikkan tekanan suatu gas. Tekanan gas dapat dinaikkan dengan mengurangi volumenya. Ketika volume suatu gas dikurangi, tekanannya akan naik. Komponen inilah yang memberi *supply* udara tekan pada sistem pneumatik dan menjaga tekanan udara agar sistem tetap berada pada tekanan kerjanya.

Kompresor secara garis besar terdiri dari dua tipe, yaitu *positive displacement* dan *dynamic*. Dari dua tipe tersebut bercabang lagi menjadi beberapa tipe. Salah satunya adalah *rotary vane compressor* yang tampak seperti dalam Gambar 2.2.



Gambar 2.2 Rotary Vane Compressor

Sumber : <http://nuvair.com>

Rotary vane compressor terdiri dari sebuah rotor dengan beberapa baling-baling yang terpasang secara radial pada *slot* di rotor. Ketika rotor berputar terjadi penyempitan volume yang mengakibatkan tekanan udara semakin besar. Kompresor jenis ini menghasilkan panas karena rotor berputar secara terus menerus, oleh karena itu ketika kompresor aktif oli terus disirkulasikan di bagian rotor untuk menurunkan suhu.

2.2.2 *Buffer Accumulator*

Untuk meminimalisir resiko kehilangan tekanan udara secara drastis ketika kompresor terjadi gangguan atau kegagalan digunakan *buffer accumulator*. Udara tekan dari kompresor terlebih dahulu ditampung di *buffer accumulator*, dengan begitu apabila tekanan dari kompresor tiba-tiba turun, perangkat pneumatik tidak langsung kehilangan tekanan udara karena masih mendapat suplai dari *buffer accumulator*.

2.2.3 *Air Filter - Regulator*

Air filter adalah alat yang digunakan untuk menyaring partikel kotoran yang keluar dari kompresor. Dengan begitu udara yang keluar bebas dari partikel kotoran sehingga tidak merusak komponen pneumatik lainnya. Lama kelamaan pada *filter* akan timbul air akibat dehumidifikasi pada udara yang masuk pada *filter*. Air yang timbul harus dibuang secara berkala agar proses dehumidifikasi tetap optimal. Dehumidifikasi adalah proses pemisahan udara dari partikel air. Tercampurnya udara dengan air dapat menyebabkan korosi pada komponen pneumatik.

Sedangkan *regulator* adalah alat yang digunakan untuk mengatur besar tekanan udara yang keluar dengan cara memutar tuas. Biasanya *regulator* dan *air filter* digabungkan menjadi satu alat, sering juga disebut *air filter-regulator* (AFR). Bentuk umumnya seperti yang terlihat dalam Gambar 2.3.



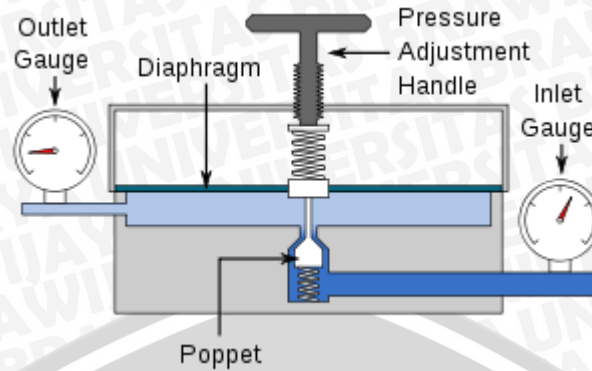
Gambar 2.3 Air Filter-Regulator
Sumber : Waller D, 2002

2.2.4 Pressure Regulator

Pressure regulator adalah *valve* yang memotong aliran udara pada tekanan tertentu. *Regulator* digunakan untuk mengurangi tekanan udara. Tekanan udara harus dibatasi karena faktor keamanan atau faktor prinsip kerja aplikasi yang digunakan.

Pressure regulator terdiri dari elemen pembatas, elemen beban, dan elemen pengukur. Elemen pembatas adalah *valve* yang digunakan untuk membatasi tekanan udara, bisa berupa *globe valve*, *butterfly valve*, *poppet valve*, atau tipe *valve* lain yang berfungsi sama. Elemen beban memberi tekanan yang diinginkan pada elemen pembatas. Elemen beban biasanya terdiri dari diafragma yang dikombinasikan dengan per. Elemen pengukur berfungsi mengukur aliran udara yang masuk dan keluar. Biasanya untuk elemen pengukur digunakan diafragma (Budi, 2013).

Dalam Gambar 2.4, *diaphragm* diberi tekanan untuk mengatur *poppet valve* dengan tujuan mengatur tekanan. Dengan mengatur sekrup (*pressure adjustment handle*), tekanan pada *diaphragm* dapat ditambah. Sehingga diperlukan tekanan udara yang lebih besar pada ruangan atas untuk mengembalikan *diaphragm* kembali ke posisi awal. Dengan cara ini tekanan udara yang keluar dapat diatur.



Gambar 2.4 Komponen *Pressure Regulator*

sumber : <http://en.wikipedia.org/wiki/File:Single-stage-regulator.svg>

2.2.5 *Tubing, Fitting, dan Air Coupler*

Tubing adalah silinder berongga terbuat dari plastik yang didisain untuk memindahkan fluida dari satu tempat ke tempat lain. Umumnya *tubing* yang digunakan pada sistem pneumatik terbuat dari bahan yang fleksibel dan memiliki beberapa ukuran diameter seperti yang terlihat dalam Gambar 2.5. Pemilihan diameter *tubing* tergantung dengan *fitting* yang digunakan pada sistem tersebut.



Gambar 2.5 *Tubing Pneumatik*

Fitting adalah komponen yang digunakan untuk menghubungkan suatu komponen pneumatik dengan *tubing* ataupun antara *tubing* dengan *tubing*. Dengan pemasangan *tubing* ke *fitting* yang tepat tidak akan terjadi kebocoran gas. Umumnya *fitting* mempunyai ukuran diameter 4mm, 6mm, dan 8mm. Semakin besar ukuran *fitting* dan *tubing* yang digunakan, semakin besar pula aliran udara yang dilewatkan. Fungsi dan kegunaan *fitting* bervariasi tergantung dengan bentuknya. Macam-macam bentuk *fitting* seperti yang terlihat dalam Gambar 2.6.



Gambar 2.6 Fitting Pneumatik

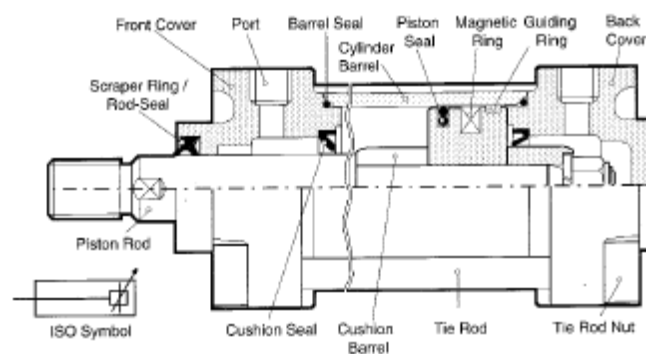
Air coupler adalah komponen yang berfungsi seperti *fitting*, yaitu untuk mencegah adanya kebocoran pada tiap sambungan. Perbedaan *fitting* dengan *air coupler* adalah *fitting* bersambungan langsung dengan *tubing*, sedangkan *air coupler* hanya dapat bersambungan dengan sesama *air coupler*. *Air coupler male* hanya dapat tersambung dengan *air coupler female*. Bentuk *air coupler male* dan *female* seperti yang terlihat dalam Gambar 2.7. Umumnya ukuran *air coupler* berukuran lebih besar dibanding *fitting*. *Air coupler* biasanya digunakan pada aplikasi yang menggunakan tekanan udara yang besar, contohnya pada kompresor.



Gambar 2.7 Air Coupler

2.2.6 Silinder Double Acting

Silinder pneumatik atau sering juga disebut *air cylinder*, adalah peralatan mekanik yang memanfaatkan kekuatan dari udara bertekanan untuk menghasilkan gerakan maju mundur secara linier. Gaya dari udara bertekanan yang masuk menggerakkan *piston* dalam silinder sehingga *piston rod* atau yang biasa juga disebut *stroke*, akan ikut bergerak ke arah yang tekanan udaranya lebih rendah. *Piston rod* itulah yang dimanfaatkan dalam berbagai aplikasi. Bagian-bagian silinder *double acting* terlihat dalam Gambar 2.8.



Gambar 2.8 Silinder Pneumatik

Sumber : SMC Pneubook

Biasanya silinder pneumatik lebih banyak disukai penggunaannya karena lebih tidak berisik bila dibandingkan motor dan tidak membutuhkan ruang banyak untuk menyimpan udara.

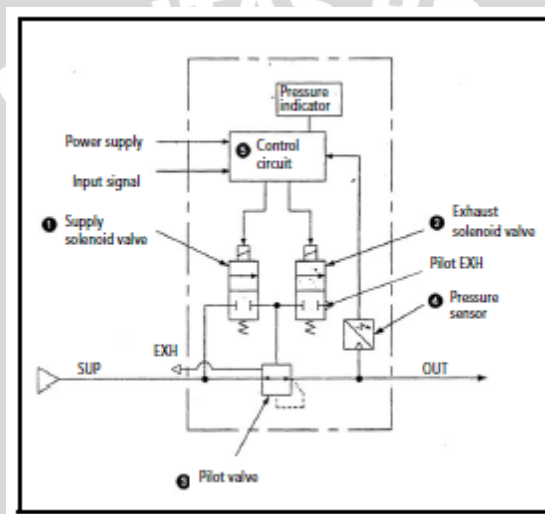
Karena fluida yang digunakan adalah udara, kebocoran pada silinder pneumatik tidak akan menetes dan mengkontaminasi lingkungan sekitar silinder, sehingga silinder pneumatik dapat digunakan pada ruangan yang harus bebas dari kontaminasi.

Silinder *double acting* adalah silinder pneumatik yang menggunakan tekanan udara untuk bergerak maju dan mundur (Zen, 2103). Berbeda dengan silinder *single acting* yang salah satu gerak maju atau mundurnya menggunakan pegas. Pada silinder *double acting* terdapat dua lubang tempat udara masuk, satu untuk maju dan satu untuk mundur. Panjang *stroke* yang digunakan bervariasi, tergantung dengan pemanfaatan silinder.

2.3 *Electro-Pneumatic Regulator ITV3051-013B*

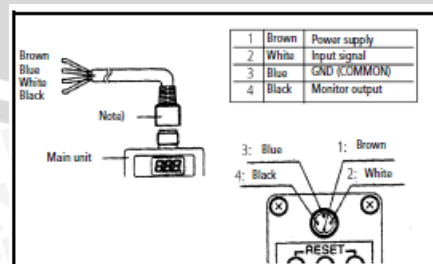
Alat ini digunakan untuk mengatur tekanan udara berdasarkan arus yang masuk. Tentunya dibutuhkan *supply* udara bertekanan. Udara yang masuk dan keluar akan berbeda tekanannya tergantung sinyal arus pada *input*.

Skema *electro-pneumatic regulator* seperti yang terlihat dalam Gambar 2.9. Prinsip kerjanya, ketika diberi sinyal masukan *air supply solenoid valve* terbuka sehingga sebagian udara bertekanan masuk melewati *pilot valve*. *Pilot valve* membuka *valve* utama sehingga menyebabkan sebagian *supply* tekanan ke *port output*. Sensor tekanan dalam *electro-pneumatic regulator* akan memberi *feedback* tekanan udara yang keluar ke sirkuit kontrol. Sirkuit kontrol akan mengolah sinyal *input* dan *output* sehingga keluaran yang dihasilkan proporsional terhadap sinyal *input*. Terdapat *exhaust solenoid valve* yang akan melepaskan sebagian udara ke atmosfer apabila tekanannya berlebih.



Gambar 2.9 Skematik ITV3051-013B

Electro-pneumatic regulator mempunyai 4 kabel yang terletak pada bagian atas, pembagiannya seperti yang terlihat dalam Gambar 2.10. *Electro-pneumatic regulator* membutuhkan catu daya sebesar 24 VDC dan menggunakan sinyal *input* berupa arus 4-20mA DC. Juga terdapat kabel untuk memonitor tekanan dengan keluaran 1-5VDC. Terdapat satu display yang berisikan informasi besar tekanan udara keluaran dalam satuan MPa.



Gambar 2.10 Rangkaian Kabel ITV3051-013B

2.4 Sensor PING)))

Sensor jarak ultrasonik PING))) adalah sensor 40 KHz produksi parallax yang banyak digunakan untuk aplikasi atau kontes robot cerdas untuk mendeteksi jarak suatu objek. Gambar fisik sensor PING))) ditunjukkan dalam Gambar 2.11



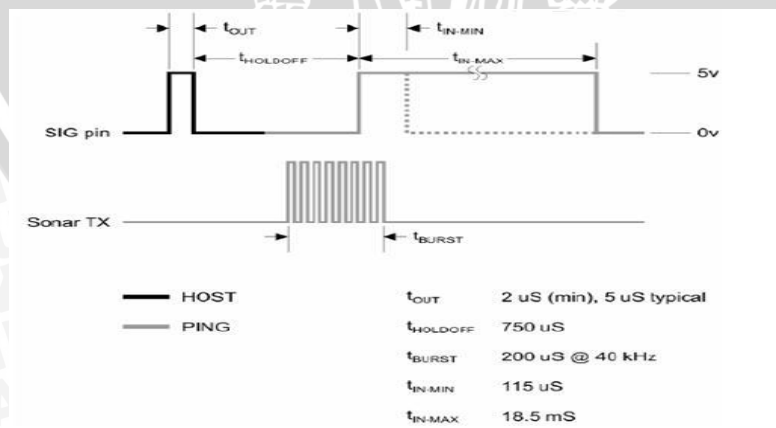
Gambar 2.11 Sensor Ultrasonik PING)))

(Sumber : lab.binus.ac.id)

Sensor PING))) mendeteksi jarak objek dengan cara memancarkan gelombang ultrasonik (40 KHz) selama $t = 200 \mu\text{s}$ kemudian mendeteksi pantulannya. Sensor PING))) memancarkan gelombang ultrasonik sesuai dengan kontrol dari mikrokontroler pengendali (pulsa trigger dengan t_{out} min $2 \mu\text{s}$). Diagram waktu sensor ini ditunjukkan dalam Gambar 2.12

Spesifikasi sensor ini:

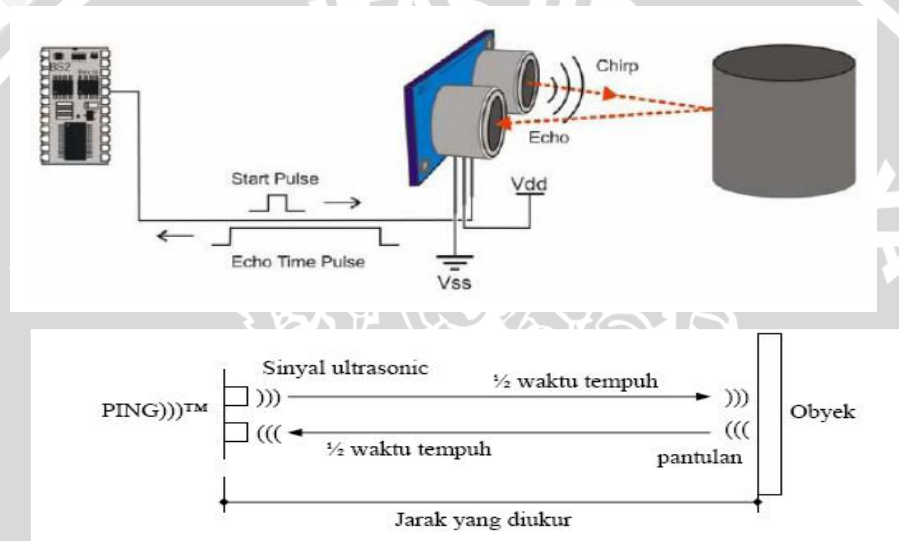
- Kisaran pengukuran 3cm-3m.
- Input trigger –positive* TTL pulse, $2\mu\text{s}$ min., $5\mu\text{s}$ tipikal.
- Echo hold off* $750\mu\text{s}$ dari *fall of trigger pulse*.
- Delay before next measurement* $200\mu\text{s}$.
- Burst indicator* LED menampilkan aktifitas sensor.



Gambar 2.12 Diagram Waktu Sensor PING)))

(Sumber : <http://fahmizaleeits.wordpress.com/category/sensor/>)

Pada dasarnya sensor PING))) terdiri dari sebuah *chip* pembangkit sinyal 40KHz, sebuah speaker ultrasonik dan sebuah *microphone* ultrasonik. *Speaker* ultrasonik mengubah sinyal 40 KHz menjadi suara sementara *microphone* ultrasonik berfungsi untuk mendeteksi pantulan suaranya. Prinsip kerja ditunjukkan dalam Gambar 2.13. Sensor PING))) mendeteksi jarak obyek dengan cara memancarkan gelombang ultrasonik (40 kHz) selama t_{BURST} (200 μ s) kemudian mendeteksi pantulannya. Sensor PING))) memancarkan gelombang ultrasonik sesuai dengan kontrol dari mikrokontroler pengendali (pulsa *trigger* dengan t_{OUT} min. 2 μ s) (Fajar, 2013).



Gambar 2.13 Prinsip Kerja Sensor PING)))

(Sumber : <http://kuliahandifajar.com/sensor-ultrasonic/>)

Gelombang ultrasonik ini melalui udara dengan kecepatan 344 meter per detik, mengenai obyek dan memantul kembali ke sensor. PING))) mengeluarkan pulsa *output high* pada pin SIG setelah memancarkan gelombang ultrasonik dan setelah gelombang pantulan terdeteksi PING))) akan membuat *output low* pada pin SIG. Lebar pulsa *High* (t_{IN}) akan sesuai dengan lama waktu tempuh gelombang ultrasonik untuk dua kali jarak ukur dengan obyek (Fajar, 2013). Maka jarak yang diukur adalah

$$S = \frac{(t_{IN} \times V)}{2} \tag{2.1}$$

Keterangan:

- S = Jarak antara sensor ultrasonik dengan objek yang dideteksi
 V = Cepat rambat gelombang ultrasonik di udara (344 m/s)
 t_{IN} = Selisih waktu pemancaran dan penerimaan pantulan gelombang.

2.5 VII Converter ISO-U1-P3-O1

ISO-U1-P3-O1 adalah *isolated amplifier* untuk mengubah sinyal analog yang berupa tegangan menjadi arus. IC ini umum digunakan dalam aplikasi akuisisi data analog. ISO-U1-P3-O1 dapat mengubah sinyal masukan berupa tegangan 0-5V menjadi sinyal keluaran berupa arus 4-20mA. IC ini dicatu dengan tegangan 5V DC dan mempunyai 10 pin seperti yang tampak dalam Gambar 2.14. Di antaranya terdapat pin yang disambungkan dengan potensiometer, fungsinya sebagai pengatur *zero* dan *gain* sehingga bisa dilakukan kalibrasi.



Gambar 2.14 IC ISO-U1-P3-O1

2.6 Kontroler

Sistem pengendalian dirancang untuk melakukan dan menyelesaikan tugas tertentu. Syarat utama sistem pengendalian adalah harus stabil. Di samping kestabilan mutlak, maka sistem harus memiliki kestabilan secara relatif, yakni tolak ukur kualitas kestabilan sistem dengan menganalisis sampai sejauh mana batas-batas kestabilan sistem tersebut jika dikenai gangguan (Ogata K.,1997). Selain itu analisis juga dilakukan untuk mengetahui bagaimana kecepatan sistem dalam merespon *input* dan peredamannya terhadap adanya lonjakan (*over shoot*).

Suatu sistem dikatakan stabil, jika diberi gangguan maka sistem tersebut akan kembali ke keadaan *steady state* yaitu *output* berada dalam keadaan tetap seperti tidak ada gangguan. Sistem dikatakan tidak stabil, jika *output* berosilasi terus menerus ketika dikenai suatu gangguan. Karena suatu sistem pengendalian biasanya melibatkan penyimpanan energi maka *output* sistem ketika diberi suatu *input*, tidak dapat mengikuti *input* secara serentak, tapi menunjukkan respons transien berupa suatu osilasi teredam sebelum mencapai *steady state*.

Dalam sistem pengendalian terdapat 2 macam *loop*:

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 diumpam balikan 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.

2.6.1 Kontroler Proporsional

Untuk kontroler dengan aksi kontrol proporsional, hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan penggerak $e(t)$ adalah:

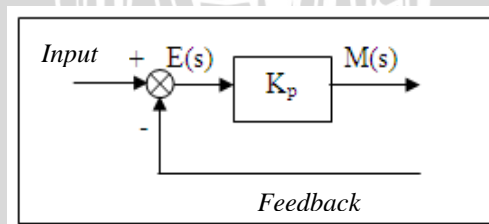
$$m(t) = K_p e(t) \quad (2.2)$$

atau, dalam besaran transformasi Laplace,

$$\frac{M(s)}{E(s)} = K_p \quad (2.3)$$

Di mana K_p adalah kepekaan proporsional atau penguatan.

Apapun wujud mekanisme yang sebenarnya dan apapun bentuk daya penggerakannya, kontroler proporsional pada dasarnya merupakan penguat dengan penguatan yang dapat diatur (Ogata K.,1997).



Gambar 2.15 Diagram Blok Kontroler Proporsional

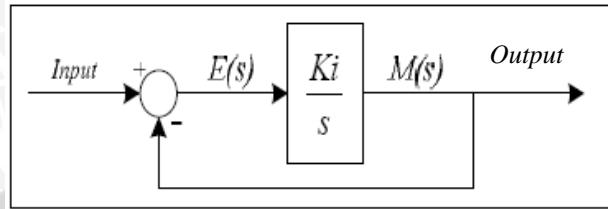
(Sumber: Ogata K., 1997)

2.6.2 Kontroler Integral

Pada kontroler dengan aksi integral, harga keluaran kontroler $m(t)$ diubah dengan laju yang sebanding dengan sinyal kesalahan penggerak $e(t)$. Jadi,

$$\frac{dm(t)}{dt} = K_i e(t) \quad (2.4)$$

Jika harga $e(t)$ diduakalikan, maka harga $m(t)$ berubah dengan laju perubahan menjadi dua kali semula. Jika kesalahan penggerak nol, maka harga $m(t)$ tetap stasioner. Aksi kontrol integral seringkali disebut kontrol *reset* (Ogata K.,1997).



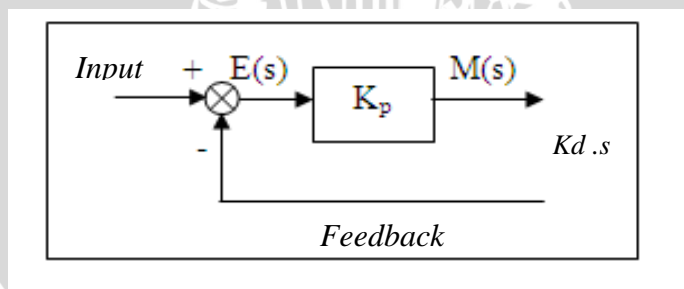
Gambar 2.16 Diagram Blok Kontroler Integral

(Sumber: Ogata K., 1997)

2.6.3 Kontroler Diferensial

Kontroler ini digunakan untuk memperbaiki atau mempercepat respons transien sebuah sistem kontrol dengan cara memperbesar *phase lead* terhadap penguatan kontrol dan mengurangi *phase lag* penguatan tersebut (Ogata K.,1997). Kontroler diferensial tidak dapat mengeluarkan *output* bila tidak ada perubahan *input*, selain itu kontroler differensial tidak dapat digunakan untuk proses yang mengandung *noise*. Hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan penggerak $e(t)$ adalah:

$$\frac{M(s)}{E(s)} = Kd \cdot s \tag{2.5}$$



Gambar 2.17 Diagram Blok Kontroler Diferensial

(Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997)

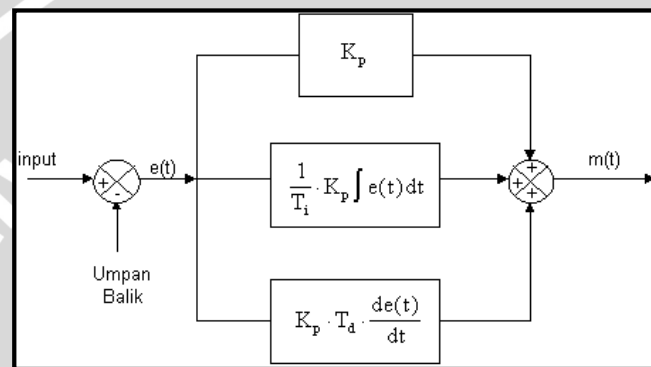
2.6.4 Kontroler Proporsional Integral Diferensial (PID)

Gabungan aksi kontrol proporsional, integral, dan diferensial mempunyai keunggulan dapat saling menutupi kekurangan dan kelebihan dari masing-masing kontroler. Persamaan kontroler PID ini dapat dinyatakan sebagai berikut:

$$m(t) = K_p \cdot e(t) + \frac{K_p}{T_i} \cdot e(t)dt + K_p \cdot T_d \frac{de(t)}{dt} \quad (2.6)$$

Dalam transformasi Laplace dinyatakan sebagai berikut:

$$\frac{M(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) \quad (2.7)$$



Gambar 2.18 Diagram Blok Kontroler PID

(Sumber: Ogata K., 1997)

2.7 Metode Perancangan Kontroler Proporsional Integral Diferensial (PID) Menggunakan *Root Locus*.

Rancangan sistem kendali loop tertutup menggunakan *root locus* memungkinkan untuk mengatur sekurang-kurangnya beberapa letak *pole* sistem *loop* tertutup sehingga dapat mengatur tanggapan transien pada tingkat tertentu dan pengaruhnya terhadap tanggapan keadaan mantap (Philips, 1996:209). Sistem kendali *loop* tertutup ditunjukkan dalam Gambar 2.19 berikut:



Gambar 2.19 Sistem Kendali Loop Tertutup

(Sumber: Ogata K., 1997)

Untuk sistem tersebut, persamaan karakteristik diberikan persamaan 2.8

$$1 + Gc(s)Gp(s) = 0 \quad (2.8)$$

Misalkan diinginkan lokus akar melalui $s = s_1$, maka hasil dari persamaan ditunjukkan pada persamaan 2.9.

$$\begin{aligned} Gc(s_1)Gp(s_1) &= -1 \\ Gc(s_1)|Gp(s_1)|e^{j\psi} &= 1e^{j\pi} \end{aligned} \quad (2.9)$$

Fungsi alih kontroler PID setelah ditransformasi laplace dinyatakan oleh persamaan 2.10.

$$Gc(s) = Kp + \frac{Ki}{s} + Kd s \quad (2.10)$$

Perhitungan dari persamaan 2.9 ditunjukkan pada persamaan 2.11,

$$Gc(s_1) = \frac{1}{|Gp(s_1)|} e^{j(\pi - \psi)} \quad (2.11)$$

Substitusi persamaan 2.11 pada persamaan 2.9 didapatkan persamaan 2.12,

$$Kd s_1^2 + Kp s_1 + Ki = \frac{e^{j(\pi - \psi)}}{|Gp(s_1)|} \quad (2.12)$$

Dengan

$$s_1 = |s_1|e^{j\beta} \quad (2.13)$$

Hasil dari substitusi persamaan 2.13 ke persamaan 2.12, didapatkan pada persamaan 2.14.

$$\begin{aligned} Kd |s_1|^2 (\cos 2\beta + j \sin 2\beta) + Kp |s_1| (\cos \beta + j \sin \beta) + Ki \\ = \frac{|s_1|}{|Gp(s_1)|} [\cos(\beta + \pi - \psi) + j \sin(\beta + \pi - \psi)] \end{aligned} \quad (2.14)$$

Menyamakan real dengan real dan imajiner dengan imajiner, didapatkan hasil pada persamaan 2.15

$$\begin{bmatrix} |s_1|^2 & |s_1| \cos \beta \\ |s_1|^2 & |s_1| \sin \beta \end{bmatrix} \begin{bmatrix} Kd \\ Kp \end{bmatrix} = \begin{bmatrix} \frac{|s_1|}{Gp(s_1)} \cos(\beta + \pi + \psi) - Ki \\ \frac{|s_1|}{Gp(s_1)} \sin(\beta + \pi + \psi) \end{bmatrix} \quad (2.15)$$

Atau dapat ditunjukkan pada persamaan 2.16.

$$\begin{bmatrix} |s_1|^2 & |s_1| \cos \beta \\ |s_1|^2 & |s_1| \sin \beta \end{bmatrix} \begin{bmatrix} Kd \\ Kp \end{bmatrix} = \begin{bmatrix} -\frac{|s_1|}{Gp(s_1)} \cos(\psi - \beta) - Ki \\ \frac{|s_1|}{Gp(s_1)} \sin(\psi - \beta) \end{bmatrix} \quad (2.16)$$

Dari persamaan dapat dilihat bahwa untuk perancangan kontroler PID, satu dari tiga penguatan K_p , K_i , K_d , harus ditentukan dahulu. Sedangkan untuk perancangan PI atau PD, penguatan yang sesuai pada persamaan dibuat sama dengan nol.

2.8 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 driveir *USB-to-serial*. Bentuk Arduino Uno ditunjukkan dalam gambar 2.20.



Gambar 2.20 Arduino Uno

(Sumber : Arduino Uno datasheet)

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, dan sebagai model referensi untuk *platform* Arduino, untuk perbandingan dengan versi sebelumnya.

2.8.1 Daya

Arduino Uno dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. 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 ke dalam *Gnd* dan *V_{in} 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 jika diberi daya lebih dari 12V, regulator tegangan bisa panas dan dapat merusak *board* Uno.

Pin listrik pada Arduino Uno adalah sebagai berikut:

1. V_{IN} adalah tegangan masukan kepada board Arduino ketika itu menggunakan sumber daya eksternal (sebagai pengganti dari 5 Volt koneksi USB atau sumber daya lainnya).
2. 5V adalah catu daya digunakan untuk daya mikrokontroler dan komponen lainnya.
3. 3v3 adalah sebuah pasokan 3,3 Volt dihasilkan oleh *regulator on-board*.
4. *GND* adalah sebuah *ground pin*.

2.8.2 Memori

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

2.8.3 Input dan Output

Masing-masing dari 14 pin digital pada Arduino 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:

1. *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.
2. *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.
3. *PWM*: 3, 5, 6, 9, 10, dan 11. Menyediakan output PWM 8-bit dengan fungsi *analogWrite* ().
4. *SPI*: 10 (SS), 11 (Mosi), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan *SPI library*.
5. *LED*: 13. Ada *built-in* LED terhubung ke pin digital 13. Ketika pin bernilai nilai HIGH, LED on, ketika pin bernilai LOW, LED *off*.

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 :

1. *I2C*: A4 (*SDA*) dan A5 (*SCL*). Dukungan I2C (TWI) komunikasi menggunakan perpustakaan Wire.
2. *Aref*. Tegangan referensi (0 sampai 5V saja) untuk input analog. Digunakan dengan fungsi *analogReference* ().
3. *Reset*. Bawa baris ini LOW untuk me-*reset* mikrokontroler.

2.8.4 Komunikasi

Arduino memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. ATmega328 menyediakan UART TTL (5V) untuk komunikasi serial, yang tersedia di pin digital 0 (RX) dan 1 (TX). Sebuah ATmega8U2 sebagai saluran komunikasi serial melalui USB dan sebagai *port virtual com* untuk perangkat lunak pada komputer. Firmware '8 U2 menggunakan *driver* USB standar COM, dan tidak ada *driver* eksternal yang diperlukan. Namun, pada *Windows* diperlukan sebuah file inf. Perangkat lunak Arduino terdapat monitor serial yang memungkinkan digunakan memonitor data tekstual sederhana yang akan dikirim ke atau dari *board* Arduino. LED RX dan TX di *layar* akan berkedip ketika data sedang dikirim melalui *chip* USB-to-serial

dengan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

Sebuah *Software Serial library* memungkinkan untuk berkomunikasi secara serial pada salah satu pin digital pada *board* Uno. ATmega328 juga mendukung I2C (TWI) dan komunikasi SPI.

2.9 Konveyor

Konveyor adalah suatu sistem mekanik yang mempunyai fungsi memindahkan barang dari satu tempat ke tempat yang lain. Konveyor banyak dipakai di industri untuk transportasi barang yang jumlahnya sangat banyak dan berkelanjutan. Dalam kondisi tertentu, konveyor banyak dipakai karena mempunyai nilai ekonomis dibanding transportasi berat seperti truk dan mobil pengangkut.

Konveyor membuat proses produksi lebih mudah dan efektif. Konveyor dapat memobilisasi barang dalam jumlah banyak dan kontinyu dari satu tempat ke tempat lain. Perpindahan tempat tersebut harus mempunyai lokasi yang tetap agar sistem conveyor mempunyai nilai ekonomis.



Gambar 2.21 Belt Conveyor

Sumber : <http://www.indonetwork.co.id/saranatekniksemarang/4778248/flat-belt-belt-conveyor-tebal-1mm-2mm-3mm-5mm-pvc-putih.htm>

BAB III

METODE PENELITIAN

Kajian dalam skripsi ini merupakan penelitian yang bersifat aplikatif, yaitu merancang suatu sistem *stamping rod* berbasis *pneumatic* yang dikendalikan oleh Arduino Uno dengan menggunakan kontroler Proporsional Integral Diferensial yang bertujuan agar dapat menampilkan performansi sistem sesuai dengan yang direncanakan.

Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang akan dibuat adalah sebagai berikut:

3.1. Perancangan Sistem

Perancangan sistem dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektronik pendukungnya, hal ini dimaksudkan agar sistem pengendalian posisi *stamping rod* dapat berjalan sesuai dengan deskripsi awal yang telah direncanakan. Perancangan sistem yang dilakukan menggunakan peralatan di Laboratorium Sistem Kontrol Universitas Brawijaya, meliputi:

Penentuan sistem *pneumatic* yang digunakan.

- Penentuan spesifikasi sistem yang akan dibuat, meliputi :
 - a. Penentuan deskripsi kerja sistem secara keseluruhan.
 - b. Rangkaian elektronik pendukung.
- Mencari parameter PID dengan metode *root locus*.
- Perancangan program pada Arduino Uno sebagai pusat pengontrol sistem.

3.2. Realisasi Pembuatan Sistem

Untuk merealisasikan alat atau sistem yang telah dirancang sebelumnya maka perlu diperhatikan beberapa tahapan dalam pembuatan alat tersebut, diantaranya yaitu :

- Penyempurnaan mekanik sistem *stamping rod* dan *pneumatic* secara keseluruhan yang meliputi penentuan batasan tekanan output *electric-pneumatic regulator*, penentuan batasan *pressure regulator* dan

pemasangan sensor serta konveyor untuk membawa barang yang akan di *stamp*.

- Pembuatan rangkaian elektronika meliputi rangkaian *V/I converter*, *Digital Analog Converter (DAC) PCF8591*, *Liquid Crystal Display (LCD)*, dan konveyor.
- Pembuatan program pengontrolan menggunakan *Arduino Uno*.

3.3. Pengujian dan Analisa Data

Setelah semua komponen pada alat sudah terhubung sesuai dengan diagram blok sistem yang telah dirancang dan perangkat lunak untuk mendukung sistem telah dibuat, maka diadakan pengujian dan analisa alat. Metode pengujian alat adalah sebagai berikut:

1. Pengujian pada sensor yang digunakan.

Pengujian sensor dilakukan dengan cara mencatat dan membandingkan pembacaan sensor dengan jarak yang sebenarnya secara manual atau menggunakan penggaris.

2. Pengujian rangkaian DAC berbasis PCF8591 yang digunakan.

Pengujian tersebut dilakukan dengan cara merangkai rangkaian DAC PCF8591 untuk mengetahui kelinieran tegangan yang dikeluarkan dari *Arduino Uno* dengan cara mengatur *Pulse Width Modulation (PWM)*.

3. Pengujian pada tiap-tiap blok rangkaian.

Pengujian tersebut meliputi pengujian *V/I converter* dan *electro-pneumatic regulator*.

4. Mengadakan pengujian sistem secara keseluruhan.

Pengujian ini dilakukan dengan cara menggabungkan semua bagian alat yang dibuat dan melihat kinerja alat. Pengujian ini bertujuan untuk mengetahui kinerja alat yang dibuat dan memberikan analisa terhadap kinerja alat. Mekanismenya, pengujian dilakukan dengan 4 macam *setpoint* yang berbeda-beda. Pengujian tersebut dilakukan dengan menggunakan parameter *Kp*, *Ki*, dan *Kd* yang sudah ditentukan sebelumnya untuk melihat apakah sistem bisa berjalan tanpa ada *error*

dengan penggunaan parameter PID yang sama dengan cara mengubah *setpoint* awal.

5. Mengevaluasi hasil pengujian sistem secara keseluruhan.

Evaluasi dilakukan untuk mengetahui kinerja alat secara keseluruhan untuk mengetahui apa yang menjadi kekurangan dalam sistem tersebut.

3.4. Pengambilan Kesimpulan dan Saran

Kesimpulan diambil berdasarkan data yang diperoleh dari pengujian sistem secara keseluruhan. Jika hasil yang didapatkan telah sesuai dengan yang direncanakan sebelumnya, maka sistem kendali tersebut telah berhasil memenuhi harapan dan tentunya memerlukan pengembangan lebih lanjut untuk penyempurnaan.



BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

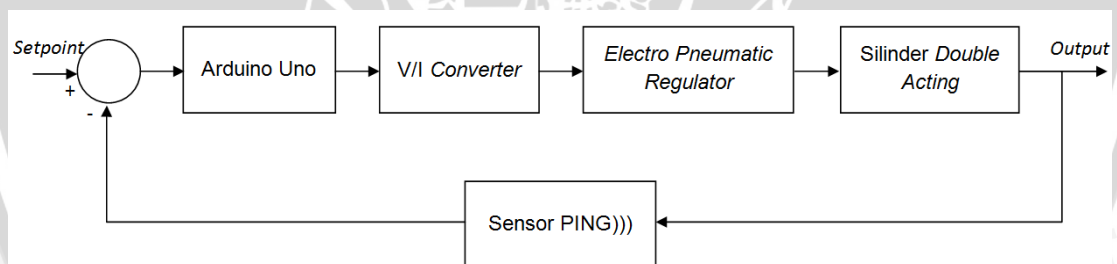
4.1. Perancangan Sistem

Perancangan alat ini dilakukan secara bertahap dalam bentuk blok sehingga akan memudahkan dalam analisis pada setiap bloknya maupun secara keseluruhan. Perancangan ini terdiri dari:

1. Cara kerja alat
2. Spesifikasi alat
3. Perancangan perangkat keras yang terdiri atas perancangan elektrik yang meliputi *V/I converter*, *Digital Analog Converter (DAC) PCF 8591*, *Liquid Crystal Display (LCD)*, sensor PING))), konveyor, dan Arduino Uno.
4. Perancangan perangkat lunak yaitu algoritma pencarian parameter PID menggunakan *root locus* kemudian menerapkan parameter tersebut pada kontroler Proporsional Integral Diferensial pada *software* Arduino Uno.

4.2. Blok Diagram Sistem

Diagram blok sistem yang dirancang ditunjukkan dalam Gambar 4.1.



Gambar 4.1 Diagram Blok Sistem
(Sumber: Perancangan, 2014)

Keterangan dari blok diagram sistem di atas adalah sebagai berikut :

- *Setpoint* sistem berasal dari program Arduino Uno. *Setpoint* digunakan untuk menentukan seberapa panjang *silinder rod* yang akan keluar.
- Pusat pengendalian sistem menggunakan Arduino Uno yang memberikan sinyal keluaran digital.
- Keluaran Arduino Uno berupa tegangan sehingga harus dikonversi terlebih dahulu ke arus dengan menggunakan *V/I converter* karena masukan dari *electro-pneumatic regulator* adalah arus sebesar 4-20mA.

- Silinder yang digunakan adalah silinder *double acting* yang berfungsi sebagai aktuator.
- Sebagai *feed back* digunakan sensor PING))) yang berfungsi sebagai sensor jarak.

4.3. Prinsip Kerja Sistem

Cara kerja sistem sebagai berikut :

- Sistem menggunakan *supply pressure* dari kompresor sebesar 7,2 bar.
- *Supply pressure* dari kompresor dihubungkan ke *air treatment unit* kemudian keluarannya dihubungkan ke *pressure regulator* dan *electro-pneumatic regulator* menggunakan T-junction.
- *Output pressure* dari *electro-pneumatic regulator* dihubungkan ke *port* silinder *double acting* yang digunakan untuk mengeluarkan batang silinder. *Output electro-pneumatic regulator* diatur agar mengeluarkan tekanan minimal 1,3 bar dan maksimal 3,0 bar.
- Pada *port* yang digunakan untuk memasukkan batang silinder dihubungkan dengan *pressure regulator* yang keluarannya diatur untuk mengeluarkan tekanan sebesar 2,75 bar.
- Terdapat konveyor berjalan membawa barang yang akan diberi stempel, ketika benda sudah berada di depan *stamping rod* dan mengenai *limit switch* maka konveyor akan berhenti dan *stamping rod* akan bergerak maju untuk melakukan pengecapan.
- *Limit switch* di samping konveyor untuk mendeteksi benda dan berfungsi logika *if else*. Jika tidak ada benda yang menekan *limit switch* konveyor akan terus berjalan dan *silinder rod* tidak akan melakukan proses *stamping* namun jika ada benda yang menekan *limit switch*, maka konveyor akan berhenti dan *silinder rod* akan keluar dan melakukan proses *stamping*.
- Sensor PING))) digunakan sebagai sensor posisi batang silinder. Sensor PING))) akan memberikan keluaran berupa level tegangan yang berubah-ubah sesuai dengan batang posisi batang silinder yang keluar.
- Setelah sensor PING))) memunculkan jarak yang dibaca pada batang silinder, maka hasil pembacaan jarak akan diolah Arduino Uno.

- Keluaran dari modul Arduino Uno akan yang berupa tegangan harus diolah dengan menggunakan rangkaian DAC PCF8591 yang berfungsi mengatur besar kecilnya tegangan yang keluar dengan cara mengatur PWM.
- Setelah tegangan yang dikeluarkan bisa diatur, kemudian tegangan tersebut harus dikonversi terlebih dahulu dengan *V/I converter* sebelum dihubungkan dengan *electro-pneumatic regulator* sehingga tekanan yang keluar dari *electro-pneumatic regulator* tergantung dari nilai PWM.
- Ketika sistem pada Arduino Uno diberi *setpoint* tertentu, maka hasil pembacaan dari sensor PING))) yang masuk di Arduino Uno akan terus diolah sampai posisi silinder yang dikehendaki terpenuhi.
- Setelah posisi silinder dikehendaki terpenuhi, Arduino Uno akan memberi sinyal PWM agar tekanan pada kedua port menyebabkan batang silinder diam.

4.4. Perancangan Perangkat Keras

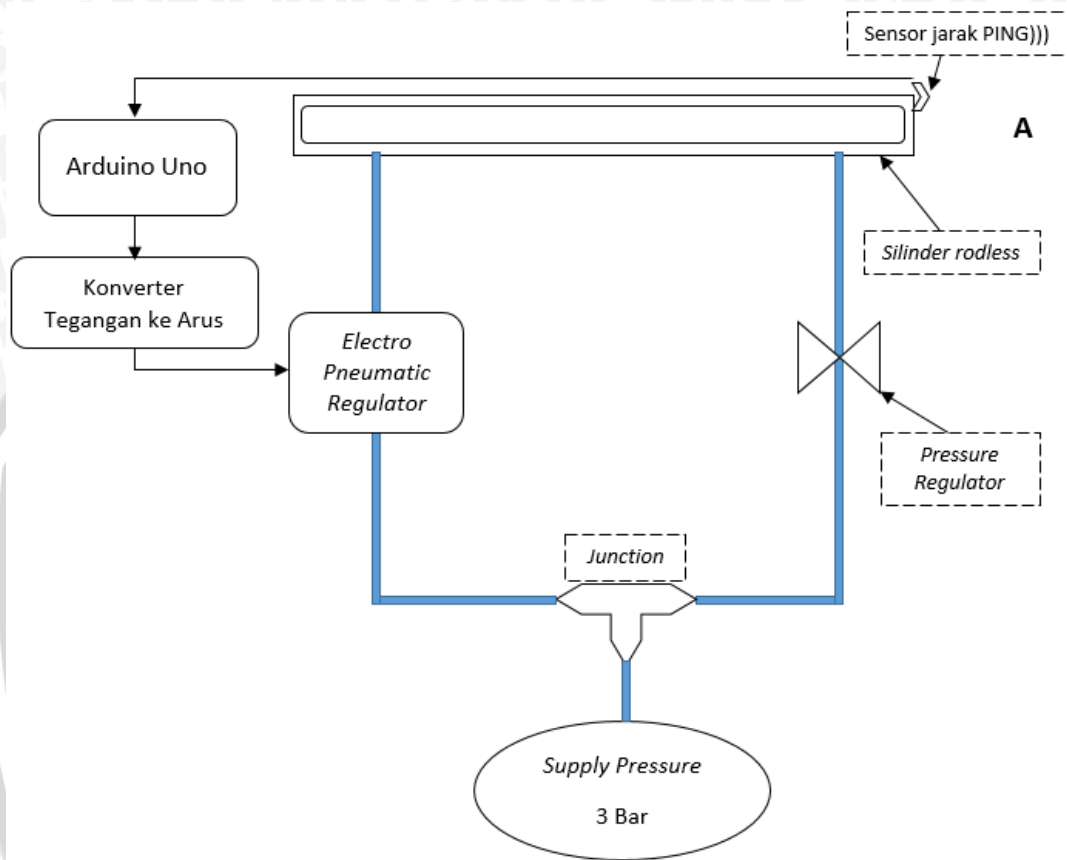
Berdasarkan diagram blok perancangan alat yang telah disusun, perancangan perangkat keras meliputi perancangan elektrik *V/I converter*, DAC PCF8591, LCD, sensor PING))), konveyor dan modul komunikasi Arduino Uno. Di bawah ini adalah penjelasan masing-masing rangkaian penyusun keseluruhan alat.

4.4.1 Spesifikasi Alat

Spesifikasi sistem pengendalian posisi *stamping rod* berbasis *pneumatic* menggunakan Arduino Uno adalah sebagai berikut :

- Sumber udara bertekanan sebesar 7,2 bar berasal dari kompresor tipe *rotary* keluaran hydrovane.
- Silinder *double acting* yang digunakan mempunyai panjang *piston rod* 30 cm dan diameter *bore* 25 mm.
- Pergerakan silinder *double acting* berasal dari tekanan udara dari *electro-pneumatic regulator* dan tekanan dengan arah yang berlawanan dari *pressure regulator*. Tekanan *output* dari *pressure regulator* diatur secara manual sebesar 2,5 bar.
- Sensor jarak menggunakan sensor PING))) 40 KHz produksi paralaks dengan jarak pembacaan 3cm-3m dan output berupa tegangan digital 0 atau 5V DC.

- Aktuator menggunakan bertekanan yang besarnya diatur *electro-pneumatic regulator* yang mendapat sinyal masukan dari *output* analog dari DAC. *Electro-pneumatic regulator* diatur agar tekanan udara di *port output* minimal 1,3 bar dan maksimal 3 bar.
- Skema perancangan sistem dapat dilihat dalam Gambar 4.2.



Gambar 4.2 Skema Keseluruhan Sistem
(Sumber: Perancangan, 2014)

4.4.2 Kompresor Hydrovane HV-01

Kompresor *hydrovane* HV-01 adalah seri kompresor terbuka (*unenclosed*) dari *hydrovane* dengan kecepatan motor tetap. HV-01 dapat menghasilkan tekanan udara sebesar 7,2 bar. Karena merupakan kompresor elektrik, HV-01 membutuhkan suplay tegangan sebesar 220V AC.

Berdasarkan prinsip kerjanya HV-01 adalah kompresor tipe *rotary vane*. Kompresor jenis ini terdiri dari sebuah rotor dengan beberapa baling-baling yang terpasang secara radial pada slot di rotor. Ketika rotor berputar terjadi penyempitan volume yang mengakibatkan tekanan udara semakin besar. Kompresor jenis ini menghasilkan panas karena rotor berputar secara terus

menerus, oleh karena itu ketika kompresor aktif oli terus disirkulasikan di bagian rotor untuk menurunkan suhu.

4.4.3 *Buffer Accumulator*

Alat ini digunakan untuk meminimalisir resiko kehilangan tekanan udara secara drastis ketika kompresor terjadi gangguan atau kegagalan digunakan *buffer accumulator*. Udara bertekanan dari kompresor terlebih dahulu ditampung di *buffer accumulator*. Dengan begitu apabila tekanan dari kompresor tiba-tiba turun, perangkat *pneumatic* tidak langsung kehilangan tekanan udara karena masih mendapat suplai dari *buffer accumulator*.

Buffer accumulator yang digunakan dalam penelitian ini buatan Martonair, seperti yang terlihat dalam Gambar 4.3 dengan tipe M16c200 yang dapat digunakan untuk tekanan udara maksimal 10 bar.



Gambar 4.3 *Buffer Accumulator*
(Sumber: Perancangan, 2014)

4.4.4 *Air Filter*

Udara bertekanan yang dihasilkan kompresor kualitasnya bergantung pada kompresor yang digunakan. Seringkali udara dari kompresor masih tercampur dengan partikel-partikel kecil seperti udara ataupun air. Oleh sebab itu digunakan *air filter* agar udara yang masuk ke komponen *pneumatic* lain bebas dari kotoran. Selain itu fungsi dari *air filter* adalah untuk memisahkan udara dari air. Sebab air dapat menyebabkan korosi pada komponen yang terbuat dari besi.

4.4.5 *Electro-Pneumatic Regulator ITV3051-013B*

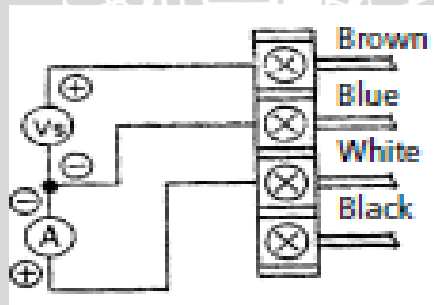
Alat ini digunakan untuk mengatur tekanan udara berdasarkan arus listrik yang masuk. Dibutuhkan *supply* tekanan udara yang lebih besar dibanding tekanan udara yang keluar. Pada penelitian ini *electro-pneumatic regulator* di-*setting* dengan batasan tekanan udara yang keluar 1,3-3 bar.



Gambar 4.4 *Electro-Pneumatic Regulator ITV3051-013B*

(Sumber: Perancangan, 2014)

Electro-pneumatic regulator mempunyai 4 kabel yang terletak pada bagian atas seperti ditunjukkan dalam Gambar 4.5. *Electro-pneumatic regulator* membutuhkan catu daya sebesar 24 VDC dan menggunakan sinyal *input* berupa arus 4-20mA DC. Juga terdapat kabel untuk memonitor tekanan dengan keluaran 1-5VDC. Terdapat satu layar yang berisikan informasi besar tekanan udara keluaran dalam satuan MPa seperti dalam Gambar 4.4.



Gambar 4.5 Rangkaian Wiring ITV3051-013B

(Sumber : ITV series catalog)

4.4.6 *Pressure Regulator*

Pressure regulator digunakan untuk mengatur tekanan udara yang masuk. Pada penelitian ini *air pressure regulator* berfungsi memberi udara bertekanan pada silinder *double acting* yang berlawanan arah dengan udara dari *electro-pneumatic regulator*. *Air pressure regulator* mempunyai 1 port *input*, 2

port *output*, dan 1 *exhaust valve* yang akan melepaskan udara agar tekanan udara yang dikeluarkan sesuai dengan keinginan. Terdapat sekrup atau *adjustment handle* seperti yang tampak dalam Gambar 4.6 yang digunakan untuk mengatur tekanan udara keluaran.



Gambar 4.6 Pressure Regulator
(Sumber: Perancangan, 2014)

Pressure regulator yang digunakan buatan Bestobell Mobrey tipe 40-50 mempunyai maksimal masukan 10 bar dan range keluaran 0.05 bar-3.5 bar.

4.4.7 Silinder *Double Acting*

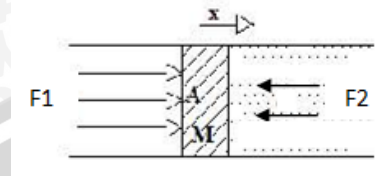
Silinder adalah aktuator *pneumatic* yang mempunyai gerakan linier. Silinder *double acting* adalah jenis silinder yang gerakan maju dan mundurnya tergantung oleh udara yang masuk pada kedua portnya. Silinder *double acting* ditunjukkan dalam Gambar 4.7



Gambar 4.7 Silinder *Double Acting* SMC CDM2b25-300
(Sumber: Perancangan, 2014)

Silinder ini mempunyai panjang *piston rod* 30cm dan diameter *bore* 25mm. Maksimal tekanan udara yang diperbolehkan adalah sebesar 10 bar. Untuk dihubungkan dengan *tubing* dibutuhkan *fitting* dengan ulir diameter 1/8".

Analogi silinder *double acting* yang digunakan dalam penelitian terlihat dalam Gambar 4.8.



Gambar 4.8 Analogi Silinder Double Acting
(Sumber : Palinggi, Wido S., 2006)

Fungsi alih silinder *double acting* dapat diperoleh melalui perhitungan dari spesifikasi yang dimiliki.

Tekanan (P) adalah suatu gaya yang bekerja merata pada permukaan bidang, dirumuskan dengan :

$$P = \frac{F}{A} \quad (4.1)$$

sehingga

$$F = P \cdot A \quad (4.2)$$

Analogi pada Gambar 4.8 sesuai dengan hukum kedua Newton, yaitu :

$$\sum F = m \cdot a \quad (4.3)$$

atau

$$\sum F = m \cdot \ddot{x} \quad (4.4)$$

Dari persamaan (4) didapat :

$$P \cdot A - F2 \cdot C\dot{x} = m \cdot \ddot{x} \quad (4.5)$$

Dengan transformasi Laplace diubah menjadi :

$$P(s)A - F2sX(s) = Ms^2X(s) \quad (4.6)$$

atau

$$P(s)A = X(s)(Ms^2 + F2s) \quad (4.7)$$

Sehingga didapatkan fungsi alih :

$$\frac{X(s)}{P(s)} = \frac{A}{Ms^2 + F2s} \quad (4.8)$$

atau

$$X(s) = \frac{A}{Ms^2 + F_2s} P(s) \quad (4.9)$$

dimana : A = luas penampang piston

M = massa stroke

F_2 = tekanan udara dari *port 2*

P = tekanan pada penampang piston

X = posisi piston

Diameter piston = 2,5 cm

$$A = \frac{1}{4}\pi D^2 \quad (4.10)$$

jadi

$$A = \frac{1}{4}\pi(2,5)^2 = 4,90625 \text{ cm}^2 \quad (4.11)$$

massa stroke = $M = 36 \text{ g}$

$F_2 = 2,5 \text{ bar} = 40 \text{ psi}$

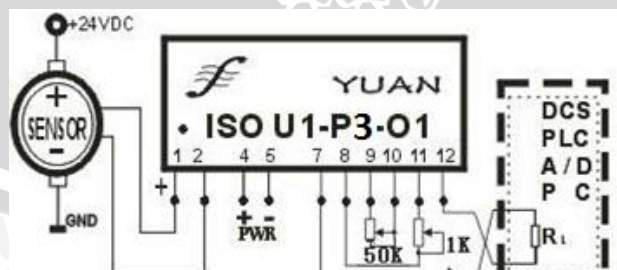
Nilai-nilai di atas dimasukkan dalam persamaan (4.9) sehingga didapat:

$$X(s) = \frac{4,90625}{36s^2 + 40s} P(s) \quad (4.12)$$

4.4.8 V/I Converter ISO U1-P3-O1

Pada penelitian ini digunakan V/I *converter* ISO U1-P3-O1 sebagai pengubah tegangan dari sensor menjadi arus 4-20mA. Konversi tegangan ke arus perlu dilakukan karena modul *input* analog AAI143 hanya dapat menerima sinyal masukan berupa arus 4-20mA.

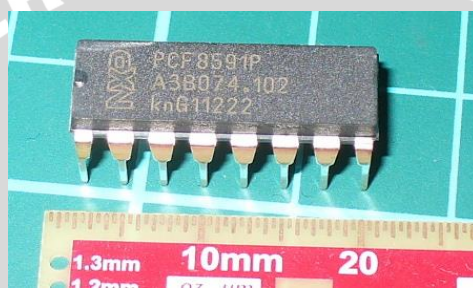
ISO U1-P3-O1 mempunyai 10 pin seperti yang terlihat dalam Gambar 4.9. Potensiometer pada pin 9-10 dan pin 8-11 digunakan untuk mengatur *gain* dan *zero* sehingga bisa dilakukan kalibrasi terhadap *output* IC.



Gambar 4.9 Rangkaian Wiring ISO U1-P3-O1
(Sumber : Datasheet ISO U1-P3-O1)

4.4.9 Digital Analog Converter (DAC) PCF8591

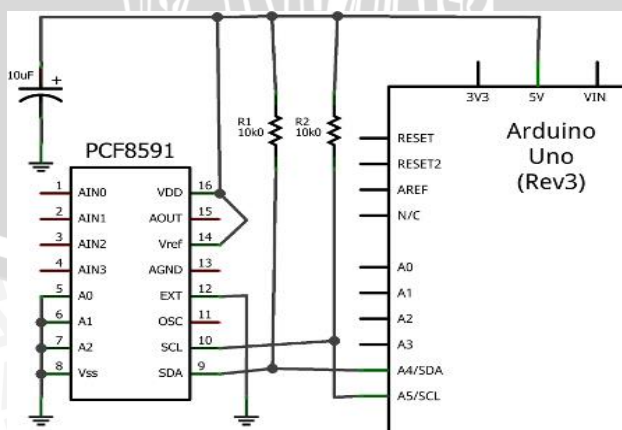
Arduino memiliki berbagai macam tipe, seperti contohnya Arduino Uno, Arduino Mega, dan masih banyak yang lainnya. Pada Arduino Uno dan Arduino Mega memiliki beberapa perbedaan, salah satunya banyaknya serial pin yang bisa digunakan. Sebagai perbandingan harga, harga Arduino Mega cukup mahal dibandingkan dengan Arduino Uno. Jadi apabila tidak berkenan menggunakan Arduino Mega tetapi dalam prakteknya butuh lebih banyak input serial pin analog, maka bisa digunakan Arduino Uno dengan ditambahkan DAC PCF8591. Selain itu, penggunaan rangkaian ini juga dapat membantu menghasilkan sinyal analog. Dalam Gambar 4.10 akan ditunjukkan IC PCF8591 yang digunakan untuk rangkaian tersebut.



Gambar 4.10 IC PCF8591

(Sumber: <http://tronixstuff.com/2013/06/17/tutorial-arduino-and-pcf8591-adc-dac-ic/>)

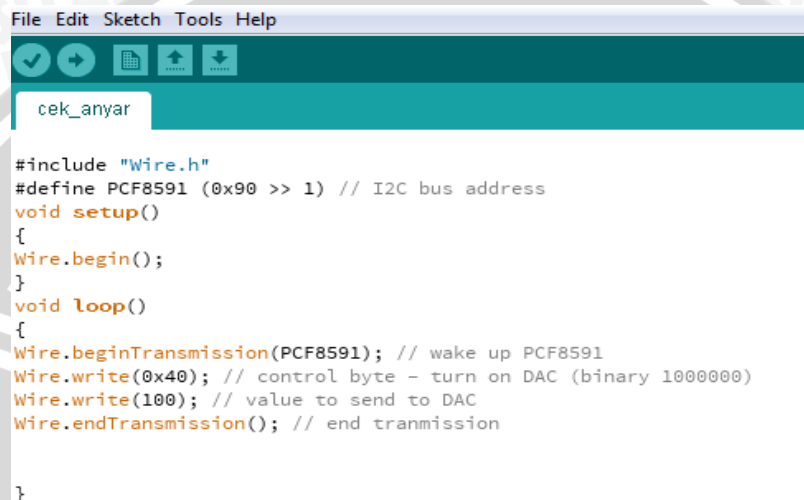
DAC pada PCF8591 memiliki resolusi 8 bit, sehingga dapat menghasilkan sinyal teoritis antara nol volt dan tegangan referensi (V_{ref}) dalam 255 langkah atau biasa yang lebih familiar dengan sebutan PWM. Dalam Gambar 4.11 akan ditunjukkan rangkaian demonstrasi DAC yang dihubungkan dengan Arduino Uno.



Gambar 4.11 Rangkaian PCF 8591

(Sumber: <http://tronixstuff.com/2013/06/17/tutorial-arduino-and-pcf8591-adc-dac-ic/>)

Menggunakan resistor 10k Ω resistor dan kapasitor 10 μ F di antara 5V dan GND. *Output analog* diambil dari pin 15 dan ada GND analog terpisah pada pin 13. Kemudian pin 13 dihubungkan ke GND dan sirkuit GND ke GND arduino. Untuk mengontrol DAC kita perlu mengirim dua *byte* data. Yang pertama adalah byte kontrol, yang hanya mengaktifkan DAC dan 1000000 (atau 0×40) dan *byte* berikutnya adalah nilai antara 0 dan 255 (tingkat *output*). Dalam Gambar 4.12 adalah program yang digunakan untuk menghubungkan komunikasi DAC PCF8591 dengan Arduino Uno.



```
File Edit Sketch Tools Help
cek_anyar
#include "Wire.h"
#define PCF8591 (0x90 >> 1) // I2C bus address
void setup()
{
  Wire.begin();
}
void loop()
{
  Wire.beginTransmission(PCF8591); // wake up PCF8591
  Wire.write(0x40); // control byte - turn on DAC (binary 1000000)
  Wire.write(100); // value to send to DAC
  Wire.endTransmission(); // end transmission
}
```

Gambar 4.12 Program DAC PCF8591 ke Arduino Uno
(Sumber: Perancangan, 2014)

4.4.10 Liquid Crystal Display (LCD)

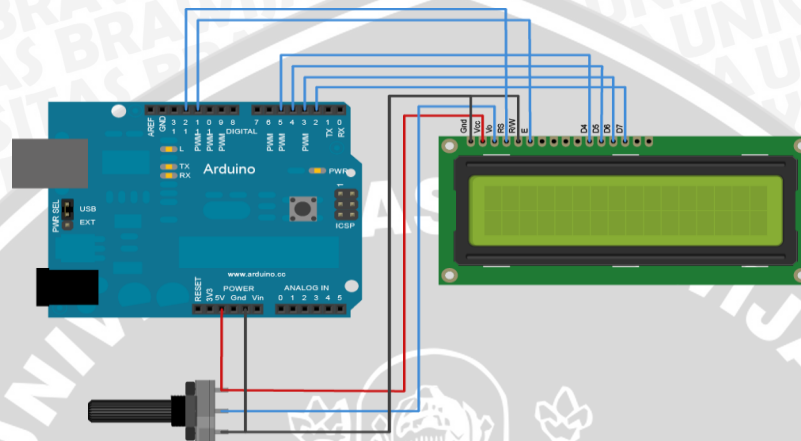
LCD yang digunakan dalam perancangan kali ini adalah Hitachi HD44780 dan di dalam sistem ini berfungsi sebagai penampil atau penunjuk waktu dan error yang ditunjukkan. Digunakannya LCD dalam perancangan ini diperlukan karena apabila hasil atau *output* waktu dan *error* ditunjukkan pada serial monitor 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*

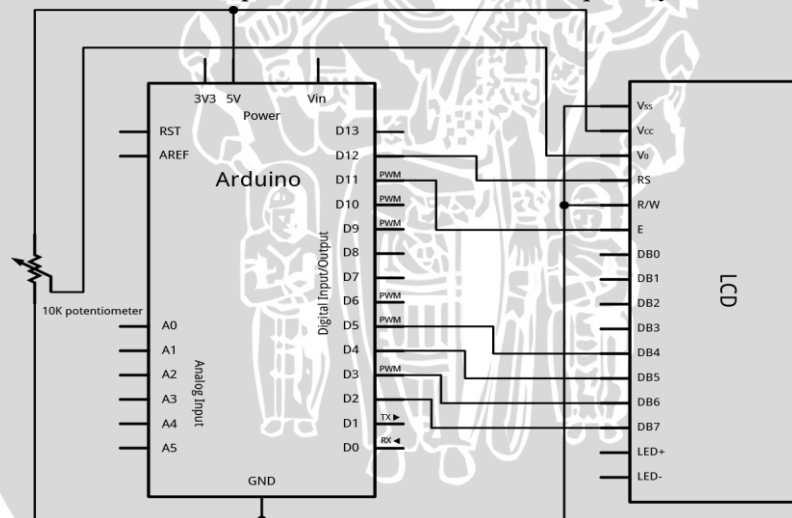
- 10k potensiometer
- *Project board*
- Kabel *male to female*

Dengan mensolder *pin header* terlebih dulu pada LCD, kemudian dengan menggunakan kabel *male to female* dihubungkan ke Arduino Uno seperti yang ditunjukkan dalam Gambar 4.13 dan 4.14 di bawah ini.



Gambar 4.13 LCD ke Arduino Uno

(Sumber : <http://arduino.cc/en/Tutorial/LiquidCrystal>)



Gambar 4.14 Rangkaian Wiring LCD ke Arduino Uno

(Sumber : <http://arduino.cc/en/Tutorial/LiquidCrystal>)

4.4.11 Sensor PING))

Sensor PING)) dalam perancangan ini digunakan untuk membaca jarak yang dikeluarkan oleh *piston rod* sekaligus untuk *feedback* dalam sistem. Apabila pembacaan jarak sensor PING)) belum sesuai dengan *setpoint* yang diharapkan, maka sistem akan terus berjalan sampai pembacaan jarak oleh

sensor PING))) sesuai dengan *setpoint* yang diharapkan. Sensor PING))) diletakkan di bawah silinder dengan posisi sejajar dengan *piston rod*.

Range pembacaan sensor PING))) sesuai *datasheet* adalah 3cm-3m. Sensor PING))) mempunyai keluaran sinyal analog 0-5V DC. Sensor PING))) memiliki 3 buah pin yaitu pin VCC, GND, dan SIG. *Gain* penguatan sensor-*transmitter* didefinisikan sebagai (Ikrom.H.,2008):

$$K = \frac{\text{span}_{\text{output}}}{\text{span}_{\text{input}}} \quad (4.13)$$

Dari data teknis didapatkan bahwa Δf (span) adalah 0-40 KHz memberi kesetaraan arus listrik 0-35 mA. Sehingga fungsi alih sensor PING)) ditunjukkan dalam persamaan 4.14.

$$K = \frac{35 - 0}{40 - 0} = 0.875 \text{ mA/KHz} \quad (4.14)$$

4.4.12 Konveyor

Konveyor adalah suatu sistem mekanik yang mempunyai fungsi memindahkan barang dari satu tempat ke tempat yang lain. Konveyor banyak dipakai di industri untuk transportasi barang yang jumlahnya sangat banyak dan berkelanjutan. Dalam kondisi tertentu, konveyor banyak dipakai karena mempunyai nilai ekonomis dibanding transportasi berat seperti truk dan mobil pengangkut. Jenis konveyor membuat penanganan alat berat tersebut / produk lebih mudah dan lebih efektif.

Konveyor yang digunakan di sini bertipe *belt conveyor* yakni konveyor yang dilingkari oleh sabuk biasanya berupa kain sepanjang konveyornya untuk memindahkan barang. Untuk menggerakkan konveyor ini digunakan motor 12 volt DC yang terletak di ujung. Karakteristik motornya, ketika tegangan yang masuk semakin besar maka putaran motornya semakin cepat. Penampilan fisik konveyor ditunjukkan dalam Gambar 4.15.



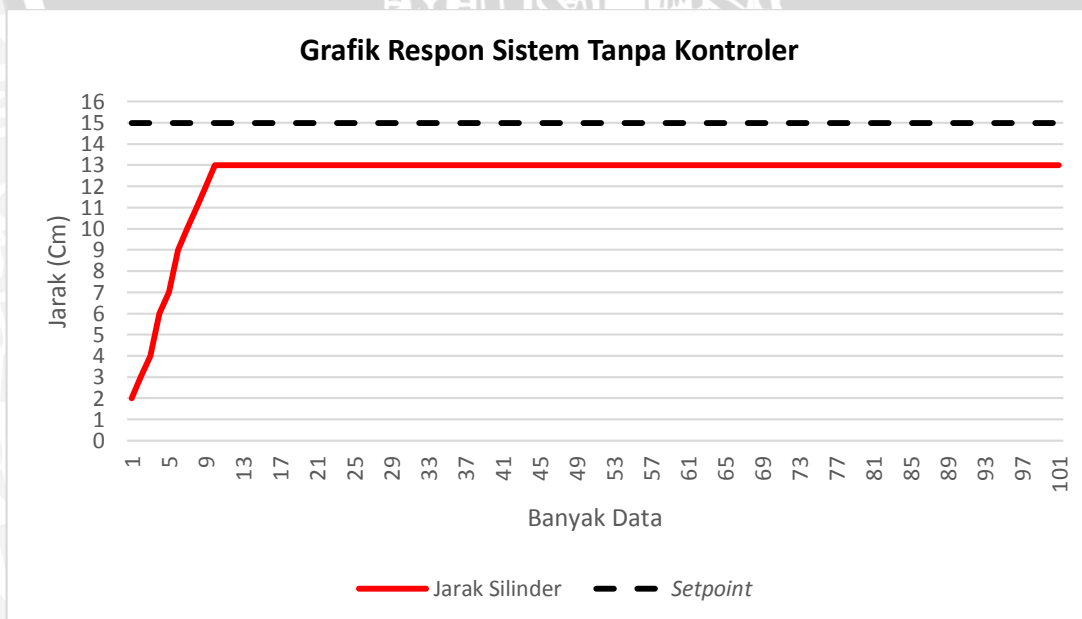
Gambar 4.15 Konveyor
(Sumber: Perancangan, 2014)

4.5. Perancangan Kontroler

Memenuhi tujuan performansi loop yang diinginkan, maka perlu ditambahkan kontroler pada sistem tersebut. Kontroler yang dipilih ialah kontroler Proporsional Integral Differensial (PID). Setelah didapatkan fungsi alih sistem yaitu

$$F(s) = \frac{4.90625}{36s^2 + 40s + 4.29296} \quad (4.15)$$

Selanjutnya adalah menentukan letak simpul *loop* tertutup. Respon sistem terhadap masukan fungsi unit *step* tanpa kontroler dapat dilihat pada Gambar 4.16.

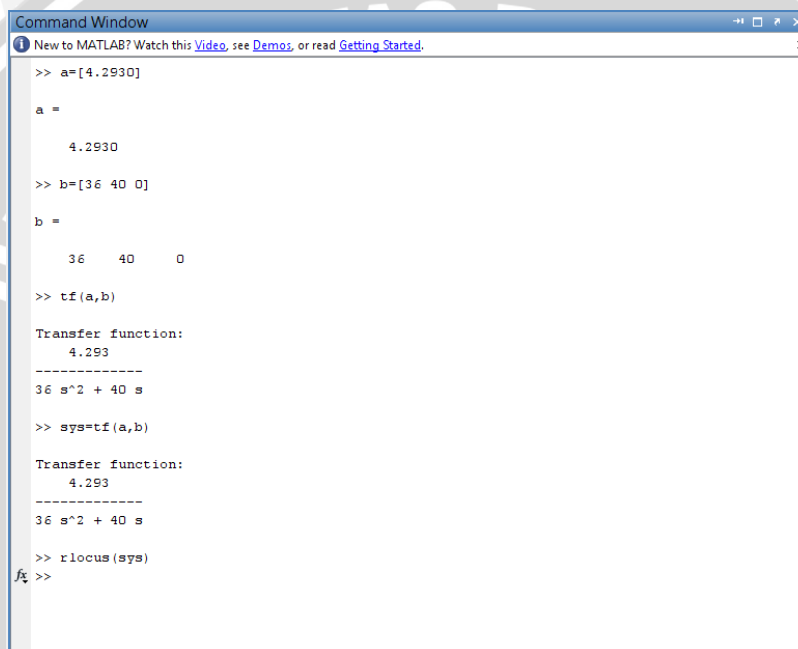


Gambar 4.16 Grafik Respon Tanpa Kontroler
(Sumber: Pengujian, 2014)

Dari respon yang didapat, sistem memiliki *error steady state* yang besar sehingga diperlukan kontroler untuk memperbaiki sistem. Memperbaiki sistem dapat diperoleh dengan memodifikasi *loop-loop* tertutupnya. Berdasarkan fungsi alih *loop* tertutup dapat diketahui sistem berorde dua. Nilai parameter PID ditentukan oleh pemilihan *pole* pada diagram *root locus*. Fungsi alih untuk *root locus* adalah

$$F(s) = \frac{4.2930}{36s^2 + 40s} \quad (4.16)$$

Pada penelitian ini digunakan $s_1 = -0.991$. Cara menunjukkan diagram *root locus* menggunakan program MATLAB 2010 terlihat dalam Gambar 4.17.

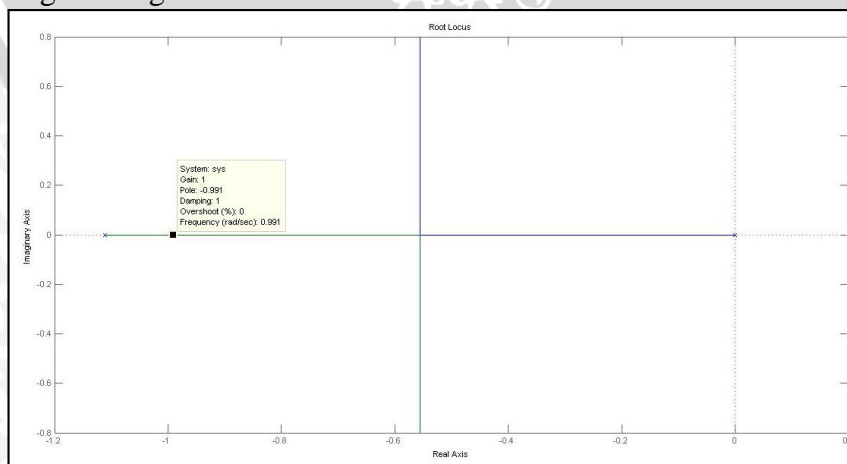


```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> a=[4.2930]
a =
    4.2930
>> b=[36 40 0]
b =
    36    40     0
>> tf(a,b)
Transfer function:
    4.293
-----
   36 s^2 + 40 s
>> sys=tf(a,b)
Transfer function:
    4.293
-----
   36 s^2 + 40 s
>> rlocus(sys)
f
>>
  
```

Gambar 4.17 Penulisan Program *Root Locus* pada MATLAB 2010
(Sumber: Perancangan, 2014)

Sedangkan diagram *root locus* terlihat dalam Gambar 4.18.



Gambar 4.18 Letak Pole pada Diagram *Root Locus*
(Sumber: Perancangan, 2014)

Titik *breakway root locus*, ditunjukkan dengan perhitungan di bawah ini

$$dG(s)H(s) = 0 \quad (4.17)$$

$$d36s^2 + 40s = 0 \quad (4.18)$$

$$72s + 40 = 0 \quad (4.19)$$

$$s = -\frac{40}{72} = -0,556 \quad (4.20)$$

Setelah ditentukan letak *pole* yang diinginkan kemudian dengan mensubstitusikan nilai s_1 dan nilai fungsi alih sistem dalam Persamaan 2.16 dan memvariasikan nilai K_i akan didapatkan parameter PID dalam Tabel 4.1. Pencarian parameter K_p , K_i , dan K_d dalam persamaan 2.16 dengan menggunakan MATLAB 2010 ditunjukkan pada *listing* program berikut :

```
%nilai pole yang ditentukan dari gambar root locus

s1=-0.991

KI=[0.1 0.5 1.5 2.5 5]

plant_num=[0 0 4.90625];
plant_den=[36 40 4.29296];

slmag = abs(s1)
beta = angle(s1)
plant_al = polyval(plant_num,s1)/polyval(plant_den,s1);
plantslmag = abs(plant_al)
psi = angle(plant_al)
t=0:1:20:300;

for k =1:5

KP = -sin(beta+psi) / (plantslmag*sin(beta)) -2*KI(k)*cos(beta) /slmag
nilai_KI = KI(k)
KD = sin(psi) / (slmag*plantslmag*sin(beta)) +KI(k) /slmag^2

Gcnum = [KD KP KI(k)];
Gcden = [0 1 0];

Tnum = conv(plant_num,Gcnum);
Tden = conv(plant_den,Gcden)+conv(plant_num,Gcnum);

r = roots(Tden)

step (Tnum,Tden,t)
hold on
end

hold off
figure, rlocus(Tnum,Tden)
```

Hasil pencarian parameter K_p , K_i dan K_d dari perhitungan pada program diatas ditunjukkan dalam Tabel 4.1.

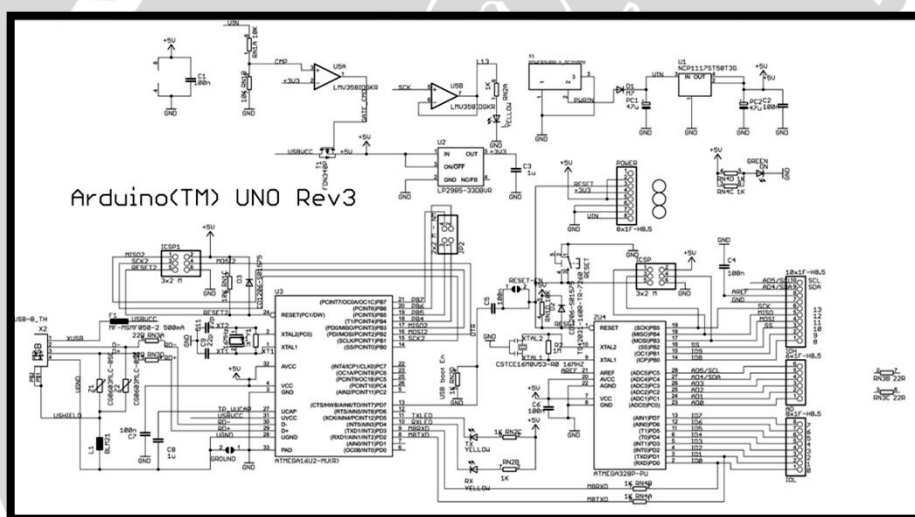
Tabel 4.1 Parameter PID dengan $s_1 = -0.991$
(Sumber : Perancangan, 2014)

No.	Kp	Ki	Kd	Pole 1	Pole 2	Pole 3
1	0,2002	0,1	0,1018	-0,991	-0,0670 + 0,0962i	-0,0670 - 0,0962i
2	2,3166	0,5	1,0182	-0,991	-0,1294 + 0,3475i	-0,1294 - 0,3475i
3	3,0256	1,5	1,5274	-0,991	-0,1641 + 0,4235i	-0,1641 - 0,4235i
4	5,0438	2,5	2,5456	-0,991	-0,2335 + 0,5378i	-0,2335 - 0,5378i
5	10,089	5	5,0912	-0,991	-0,4070 + 0,7225i	-0,4070 - 0,7225i

Setelah didapatkan nilai Kp, Ki dan Kd hasil perhitungan, kemudian dilakukan pengujian terhadap sistem, dan parameter yang sesuai dengan sistem.

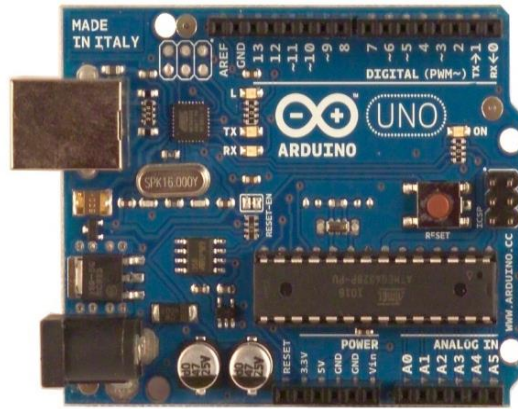
4.6. Modul Arduino Uno

Pada sistem pengendalian posisi *stamping rod* ini digunakan Arduino Uno sebagai pengolah data dalam proses pengaturan tekanan udara yang terdapat pada *electro-pneumatic regulator* untuk menggerakkan silinder *pneumatic*. Konfigurasi kaki I/O dari Arduino Uno ditunjukkan dalam Gambar 4.19.



Gambar 4.19 Desain Sistem Arduino Uno
(Sumber : Arduino Uno *datasheet*)

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 terhubung dengan adaptor DC atau baterai untuk menjalankannya. Modul Arduino Uno ditunjukkan dalam Gambar 4.20 beserta fungsi pin dalam Tabel 4.2.



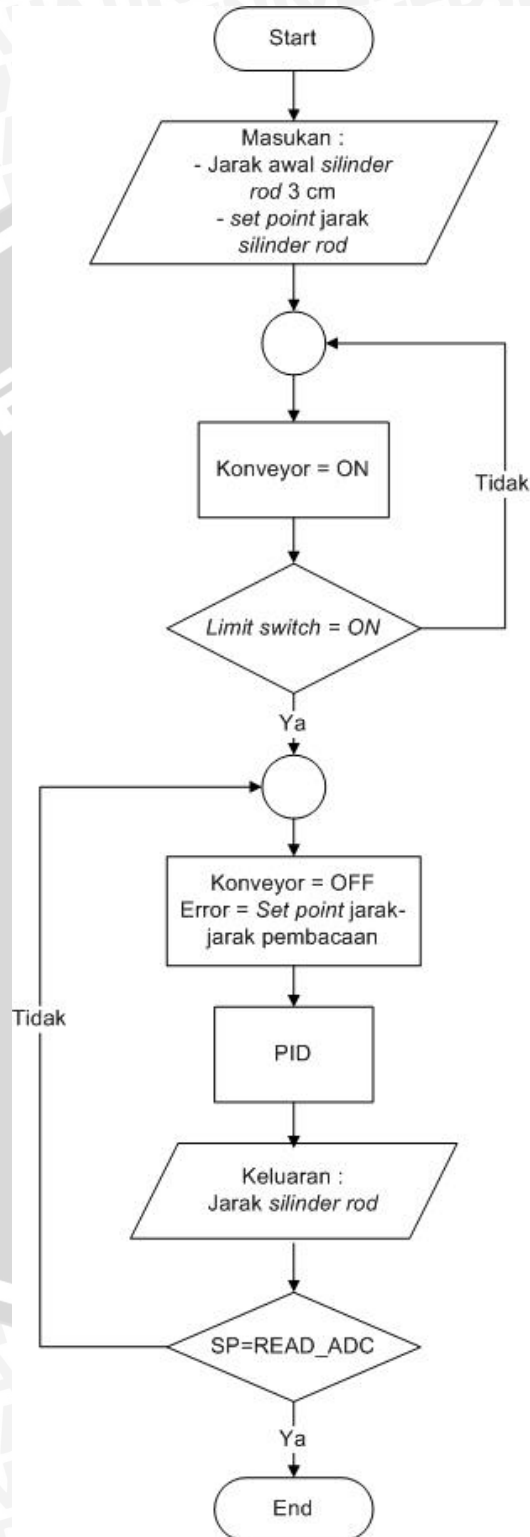
Gambar 4.20 Modul Arduino Uno
(Sumber: Perancangan 2014)

Tabel 4.2 Fungsi Pin Arduino Uno
(Sumber: Perancangan, 2014)

No	Pin	Fungsi
1	2	LCD D7 pin
2	3	LCD D6 pin
3	4	LCD D5 pin
4	5	LCD D4 pin
5	7	Jalur SIG sensor PING)))
6	8	Relay Arduino untuk motor konveyor
7	11	LCD enable pin
8	12	LCD RS pin
9	13	Masukan <i>limit switch</i>
10	A5	SCL PCF8591
11	A4	SDA PCF8591
11	GND	Jalur masukan GND seluruh sistem
12	5V	Jalur masukan 5V seluruh sistem

4.7 Perancangan Perangkat Lunak

Perancangan perangkat lunak ditunjukkan dalam *flowchart* dalam Gambar 4.21 di bawah ini.



Gambar 4.21 *Flowchart* program
(Sumber: Perancangan, 2014)

BAB V

PENGUJIAN DAN ANALISIS

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 analisis pada sistem apabila alat tidak bekerja sesuai dengan perancangan. Pengujian dibagi menjadi beberapa bagian, yaitu :

1. Pengujian *Digital Analog Converter* (DAC) PCF8591
2. Pengujian *V/I converter*
3. Pengujian *electro-pneumatic regulator*
4. Pengujian sensor (PING))
5. Pengujian sistem keseluruhan

5.1. Pengujian *Digital Analog Converter* (DAC) PCF8591

Pengujian ini dilakukan untuk mengetahui besar kecilnya *output* tegangan yang dikeluarkan oleh Arduino Uno dengan mengatur *Pulse Width Modulation* (PWM) dengan range 0-255.

5.1.1. Peralatan Pengujian

Peralatan pengujian terdiri dari:

- Laptop
- Arduino Uno
- Rangkaian DAC PCF8591
- Multimeter

5.1.2. Prosedur Pengujian

Pengujian dilakukan dengan menghubungkan rangkaian DAC PCF8591 ke Arduino Uno, kemudian menyambungkan port Arduino Uno dengan laptop. Dengan program PCF8591 kemudian diatur menggunakan PWM untuk mengetahui besar kecilnya tegangan.

5.1.3. Hasil Pengujian

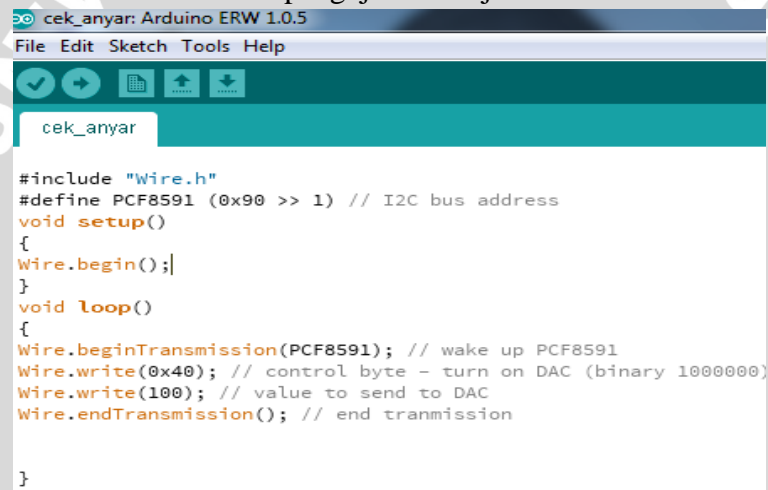
Setelah melakukan prosedur pengujian, maka didapatkan hasil untuk tegangan keluaran Arduino Uno yang ditunjukkan dalam Tabel 5.1

Tabel 5.1 Hasil Pengujian DAC PCF8591

(Sumber: Pengujian, 2014)

No	PWM	Tegangan (volt)
1	50	0.91
2	75	1.38
3	80	1.46
4	85	1.55
5	90	1.64
6	95	1.74
7	100	1.83
8	105	1.92
9	110	2.01
10	115	2.11
11	120	2.2
12	125	2.3
13	150	2.76

Program untuk melakukan pengujian ditunjukkan dalam Gambar 5.1



```

cek_anyar: Arduino ERW 1.0.5
File Edit Sketch Tools Help

cek_anyar

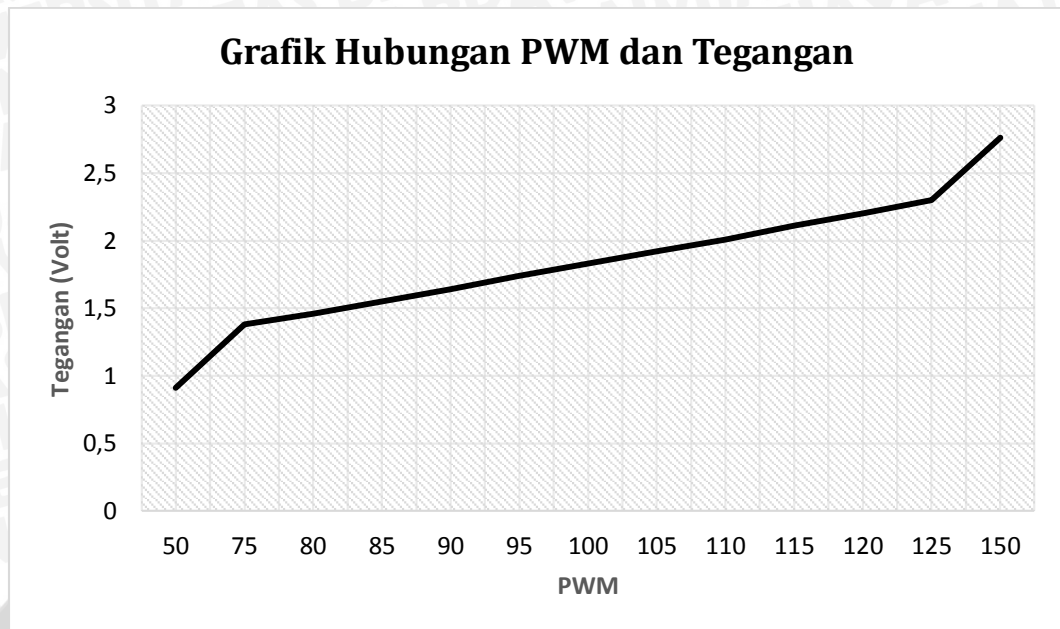
#include "Wire.h"
#define PCF8591 (0x90 >> 1) // I2C bus address
void setup()
{
  Wire.begin();
}
void loop()
{
  Wire.beginTransmission(PCF8591); // wake up PCF8591
  Wire.write(0x40); // control byte - turn on DAC (binary 1000000)
  Wire.write(100); // value to send to DAC
  Wire.endTransmission(); // end transmission
}

```

Gambar 5.1 Program PCF8591

(Sumber: Pengujian, 2014)

Grafik pengujian DAC PCF8591 ditunjukkan dalam Gambar 5.2.



Gambar 5.2 Grafik Hubungan PWM dan Tegangan

(Sumber: Pengujian, 2014)

Dari hasil pengujian yang dilakukan, semakin besar *setting* angka PWM yang dilakukan, maka *output* tegangan yang dikeluarkan semakin besar.

5.2. Pengujian V/I Converter

Pengujian ini dilakukan untuk mengetahui arus *output* V/I converter apabila diberi tegangan *input* yang berbeda-beda.

5.2.1 Peralatan Pengujian

Peralatan pengujian terdiri dari:

- Laptop
- Arduino Uno
- Rangkaian DAC PCF8591
- Rangkaian V/I converter
- Multimeter

5.2.2 Prosedur Pengujian

Seperti pada pengujian pada rangkaian DAC, pengujian dilakukan dengan menghubungkan rangkaian DAC PCF8591 ke Arduino Uno, kemudian menyambungkan port Arduino Uno dengan laptop. Pada rangkaian DAC PCF8591, kaki ke-15 pada IC yang merupakan AOUT dihubungkan dengan kaki ke-1 pada IC rangkaian V/I converter yang merupakan *input* positif tegangan. Sedangkan kaki ke-13 yang merupakan AGND pada IC rangkaian DAC

PCF8591, dihubungkan dengan kaki ke-2 pada IC rangkaian V/I *converter* yang merupakan *input* negatif tegangan. Dengan program PCF8591 kemudian diatur menggunakan PWM untuk mengetahui besar kecilnya tegangan untuk kemudian dikonversi menjadi arus.

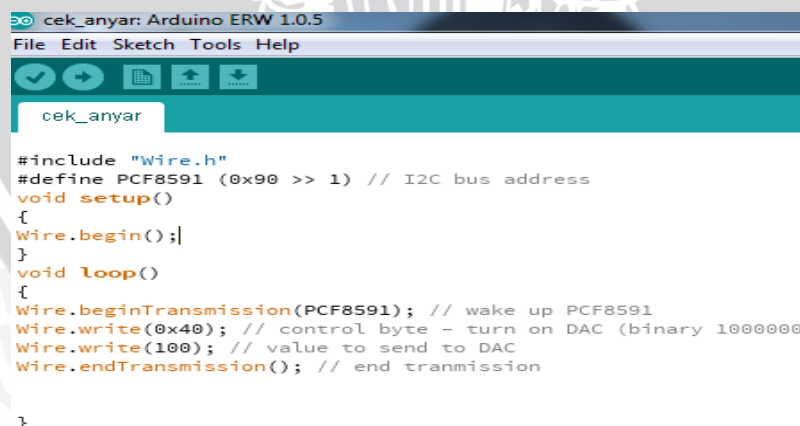
5.2.3 Hasil Pengujian

Setelah melakukan prosedur pengujian, maka didapatkan hasil untuk tegangan keluaran untuk dikonversi menjadi arus oleh V/I *converter* yang ditunjukkan dalam Tabel 5.2.

Tabel 5.2 Hasil Pengujian V/I Converter
(Sumber : Pengujian, 2014)

No	Masukan (Volt)	Keluaran (mA)
1	0.91	7
2	1.38	8.2
3	1.46	8.5
4	1.55	8.9
5	1.64	9.1
6	1.74	9.3
7	1.83	9.6
8	1.92	10
9	2.01	10.4
10	2.11	10.7
11	2.2	11
12	2.3	11.3
13	2.76	13

Program untuk melakukan pengujian V/I *converter* ditunjukkan dalam Gambar 5.3.



```

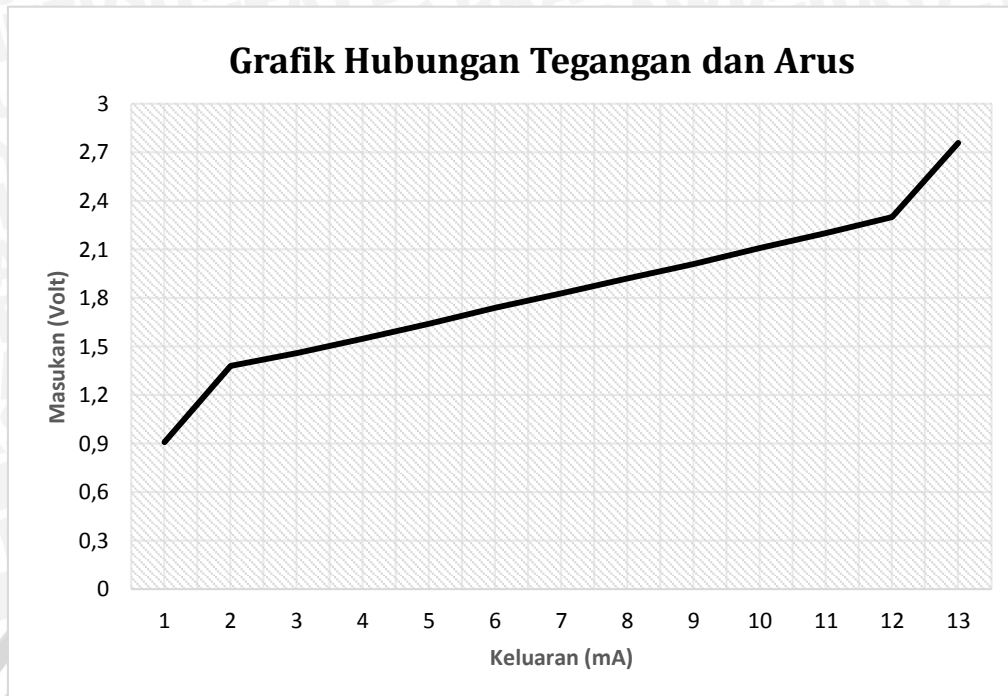
cek_anyar: Arduino ERW 1.0.5
File Edit Sketch Tools Help

cek_anyar

#include "Wire.h"
#define PCF8591 (0x90 >> 1) // I2C bus address
void setup()
{
  Wire.begin();
}
void loop()
{
  Wire.beginTransmission(PCF8591); // wake up PCF8591
  Wire.write(0x40); // control byte - turn on DAC (binary 1000000)
  Wire.write(100); // value to send to DAC
  Wire.endTransmission(); // end transmission
}
  
```

Gambar 5.3 Program PCF8591
(Sumber: Pengujian, 2014)

Grafik pengujian V/I *converter* ditunjukkan dalam Gambar 5.4.



Gambar 5.4 Grafik Hubungan Tegangan dan Arus
(Sumber: Pengujian, 2014)

Dari hasil pengujian yang dilakukan, semakin besar tegangan yang dikeluarkan, maka hasil konversi ke arus juga semakin besar.

5.3. Pengujian *Electro-Pneumatic Regulator*

Pengujian dilakukan untuk mengetahui besar tekanan apabila *electro-pneumatic regulator* diberi arus masukan yang berbeda-beda.

5.3.1 Peralatan Pengujian

Peralatan pengujian terdiri dari:

- Laptop
- Arduino Uno
- Rangkaian DAC PCF8591
- Rangkaian V/I *converter*
- *Electro-pneumatic regulator*
- *Switching AC/DC 12V*

5.3.2 Prosedur Pengujian

Melakukan prosedur pengujian yang sama dengan pengujian sebelumnya, yaitu merangkai sistem seperti pengujian V/I *converter*, kemudian kaki ke-7 pada IC V/I *converter* disambungkan pada kabel berwarna putih yang merupakan *input* dari *electro-pneumatic regulator*. Sedangkan kaki ke-12 pada

IC V/I *converter* disambungkan pada kabel warna biru yang merupakan GND dari *electro-pneumatic regulator*. *Switching AC/DC 12V* digunakan sebagai *supply electro-pneumatic regulator* dengan cara menyambungkan *output* positif DC ke kabel warna coklat dan *output* negatif DC ke kabel warna biru pada *electro-pneumatic regulator*.

5.3.3 Hasil Pengujian

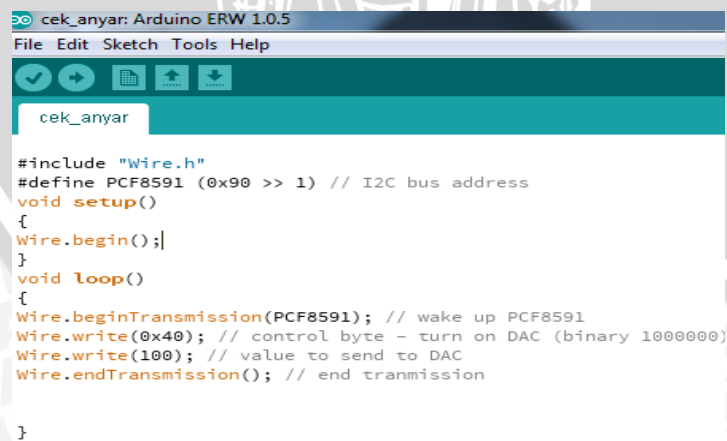
Setelah dilakukan prosedur pengujian, maka didapatkan hasil untuk tekanan yang dikeluarkan *electro-pneumatic regulator* dari *input* arus yang berbeda-beda yang ditunjukkan dalam Tabel 5.3.

Tabel 5.3 Hasil Pengujian *Electro-Pneumatic Regulator*

(Sumber : Pengujian, 2014)

No.	Masukan (mA)	Keluaran (bar)
1	7	1.62
2	8.2	1.79
3	8.5	1.83
4	8.9	1.86
5	9.1	1.9
6	9.3	1.94
7	9.6	1.97
8	10	2
9	10.4	2.03
10	10.7	2.07
11	11	2.1
12	11.3	2.13
13	13	2.3

Program untuk melakukan pengujian *electro-pneumatic regulator* ditunjukkan dalam Gambar 5.5.



```

cek_anyar: Arduino ERW 1.0.5
File Edit Sketch Tools Help

cek_anyar

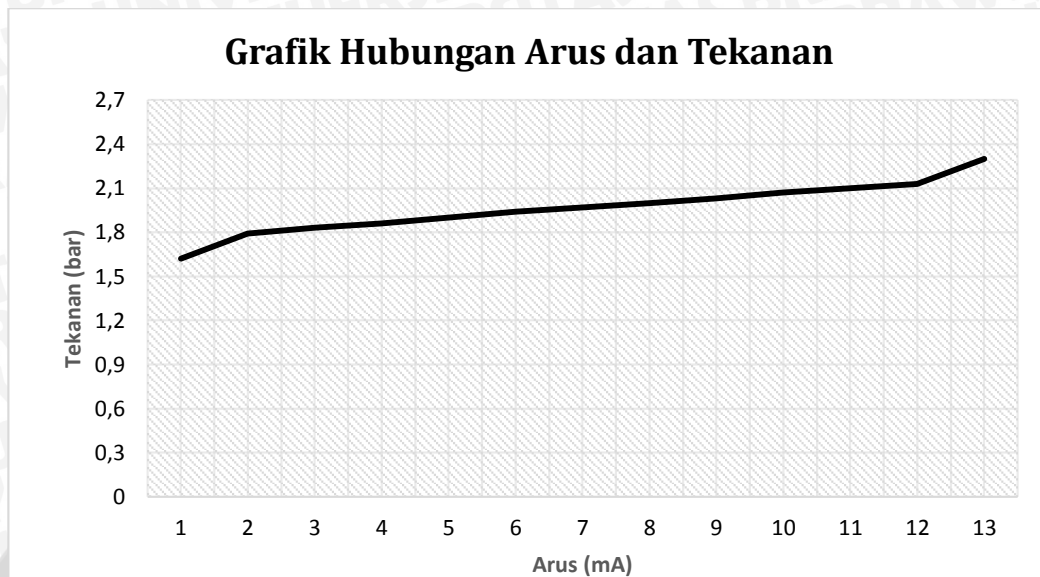
#include "Wire.h"
#define PCF8591 (0x90 >> 1) // I2C bus address
void setup()
{
  Wire.begin();
}
void loop()
{
  Wire.beginTransmission(PCF8591); // wake up PCF8591
  Wire.write(0x40); // control byte - turn on DAC (binary 1000000)
  Wire.write(100); // value to send to DAC
  Wire.endTransmission(); // end transmission
}

```

Gambar 5.5 Program PCF 8591

(Sumber: Pengujian, 2014)

Grafik pengujian *electro-pneumatic regulator* ditunjukkan dalam Gambar 5.6.



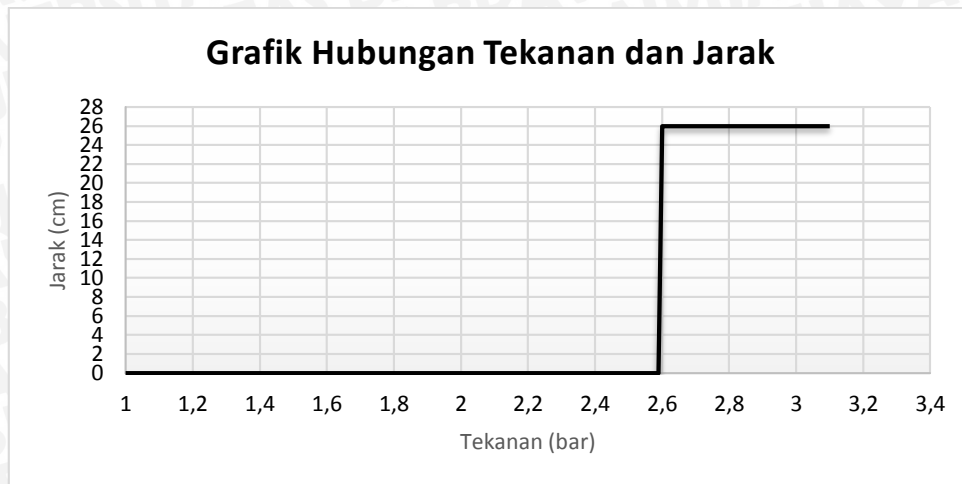
Gambar 5.6 Grafik Hubungan Tekanan dan Arus
(Sumber: Pengujian, 2014)

Dari hasil pengujian yang dilakukan, semakin besar arus yang dikeluarkan maka tekanan yang dihasilkan oleh *electro-pneumatic regulator* juga semakin besar. Hubungan antara jarak atau posisi silinder *pneumatic* dengan tekanan pada *electro-pneumatic regulator* dapat dilihat dalam Tabel 5.4.

Tabel 5.4 Hubungan antara Tekanan dengan Jarak Silinder *Pneumatic*
(Sumber : Pengujian, 2014)

No	P (bar)	X (cm)
1	1	0
2	1,25	0
3	1,5	0
4	1,75	0
5	2	0
6	2,25	0
7	2,5	0
8	2,59	0
9	2,6	26
10	2,75	26
11	3	26
12	3,1	26

Grafik hubungan antara tekanan dengan jarak silinder *pneumatic* ditampilkan dalam Gambar 5.7.



Gambar 5.7 Grafik Hubungan Tekanan dan Jarak
(Sumber: Pengujian, 2014)

5.4. Pengujian Sensor PING)))

Pengujian dilakukan untuk mengetahui seberapa besar *error* pembacaan jarak yang dilakukan oleh sensor PING))) dengan jarak sebenarnya dengan alat ukur, dalam hal ini menggunakan penggaris.

5.4.1 Peralatan Pengujian

Peralatan pengujian terdiri atas:

- Laptop
- Arduino Uno
- Sensor PING)))
- Penggaris
- *Acrilic* (mika)

5.4.2 Prosedur Pengujian

Pengujian dilakukan dengan cara menyambungkan sensor PING))) dengan Arduino Uno. 5V sensor PING))) disambungkan ke 5V Arduino Uno, GND sensor PING))) disambungkan ke GND Arduino Uno, dan SIG sensor PING))) disambungkan ke pin digital PWM Arduino Uno, yang mana pada pengujian ini diletakkan di pin ke-7. Setelah sensor PING))) sudah dapat membaca jarak *acrilic* yang sudah terpasang, jarak sensor PING))) yang ditampilkan oleh serial monitor Arduino Uno kemudian diukur dengan penggaris untuk mengetahui perbandingan jarak yang terbaca pada sensor PING))) dengan jarak yang sebenarnya.

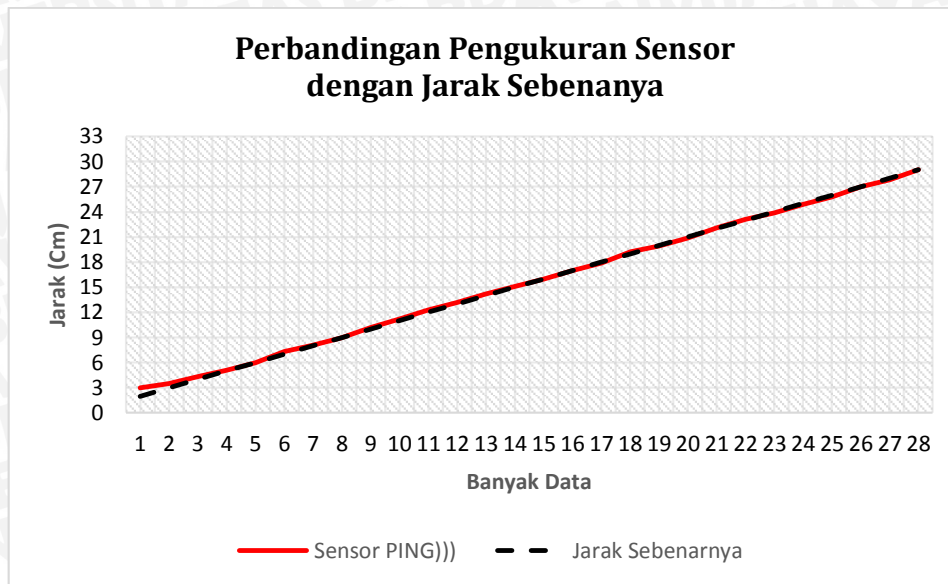
5.4.3 Hasil Pengujian

Setelah melakukan prosedur pengujian, maka didapatkan hasil pembacaan jarak sensor (PING))) dengan jarak sebenarnya yang diukur dengan penggaris dan ditunjukkan ke dalam Tabel 5.5.

Tabel 5.5 Hubungan antara Pembacaan Sensor dengan Jarak Sebenarnya
(Sumber : Pengujian, 2014)

No	Pembacaan Sensor (cm)	Jarak Sebenarnya (cm)
1	2	3
2	3	3,5
3	4	4,3
4	5	5,1
5	6	6
6	7	7,3
7	8	8,1
8	9	9
9	10	10,2
10	11	11,2
11	12	12,3
12	13	13,2
13	14	14,2
14	15	15,1
15	16	16
16	17	17
17	18	17,9
18	19	19,2
19	20	19,9
20	21	20,9
21	22	22,1
22	23	23,1
23	24	23,9
24	25	24,9
25	26	25,8
26	27	27
27	28	27,8
28	29	29

Grafik hubungan antara jarak pembacaan sensor dengan jarak sebenarnya ditunjukkan Gambar 5.8.



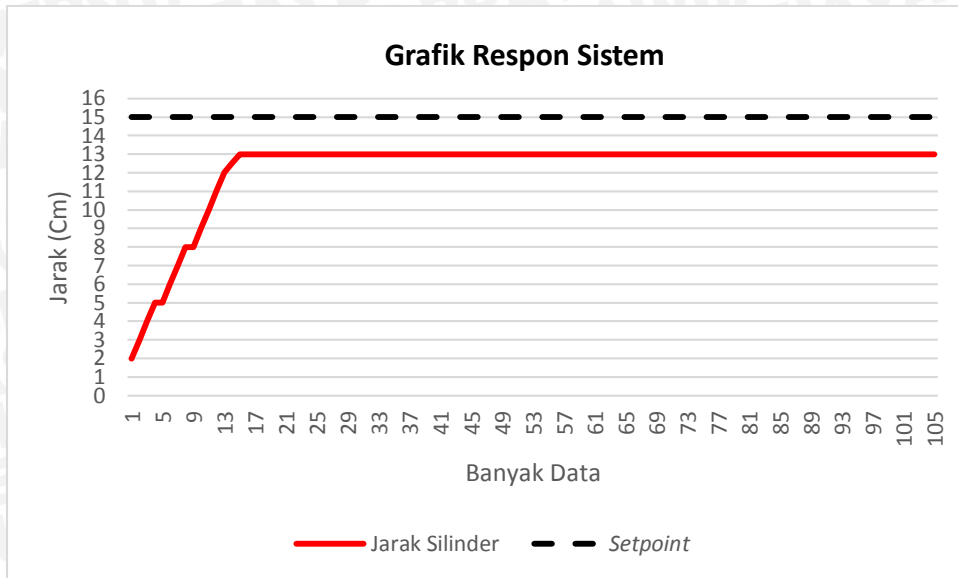
Gambar 5.8 Grafik Hubungan Pembacaan Sensor dan Jarak Sebenarnya
(Sumber: Pengujian, 2014)

Berdasarkan Gambar 5.8, dapat disimpulkan bahwa sensor PING)) dapat bekerja dengan baik.

5.5. Pengujian Sistem Secara Keseluruhan

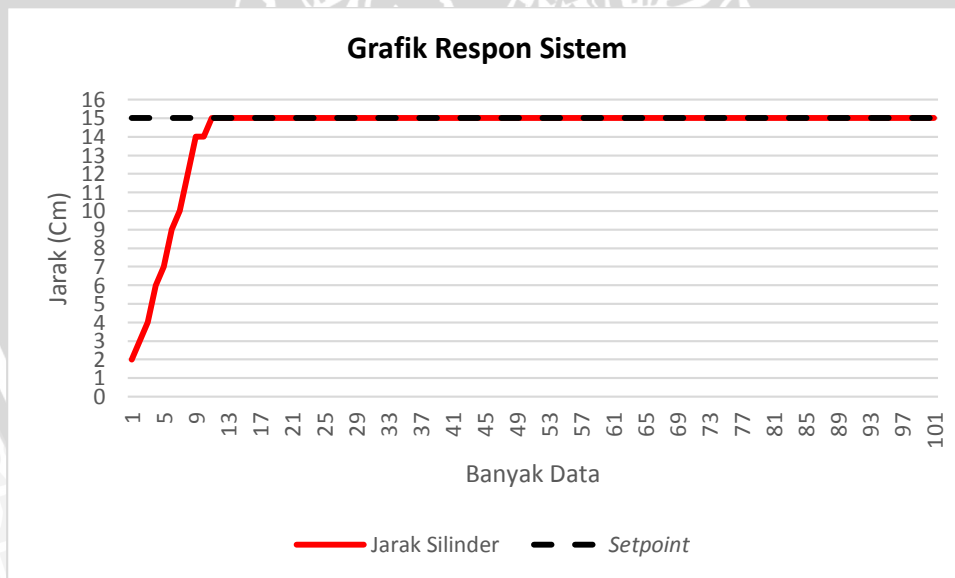
Pengujian keseluruhan sistem dilakukan untuk menguji apakah parameter yang sudah ditentukan dapat diaplikasikan pada alat dan sudah sesuai dengan *setpoint* yang diinginkan. Terdapat 5 parameter PID yang diperoleh dari perhitungan menggunakan metode *root locus*. Kelima parameter tersebut diuji satu-persatu untuk mengetahui respon sistemnya dan mencari parameter yang terbaik.

Grafik respon sistem menggunakan masukan unit *step* dengan nilai $K_p = 0,2002$, $K_i = 0,1$ dan $K_d = 0,1018$ dapat dilihat dalam Gambar 5.9.



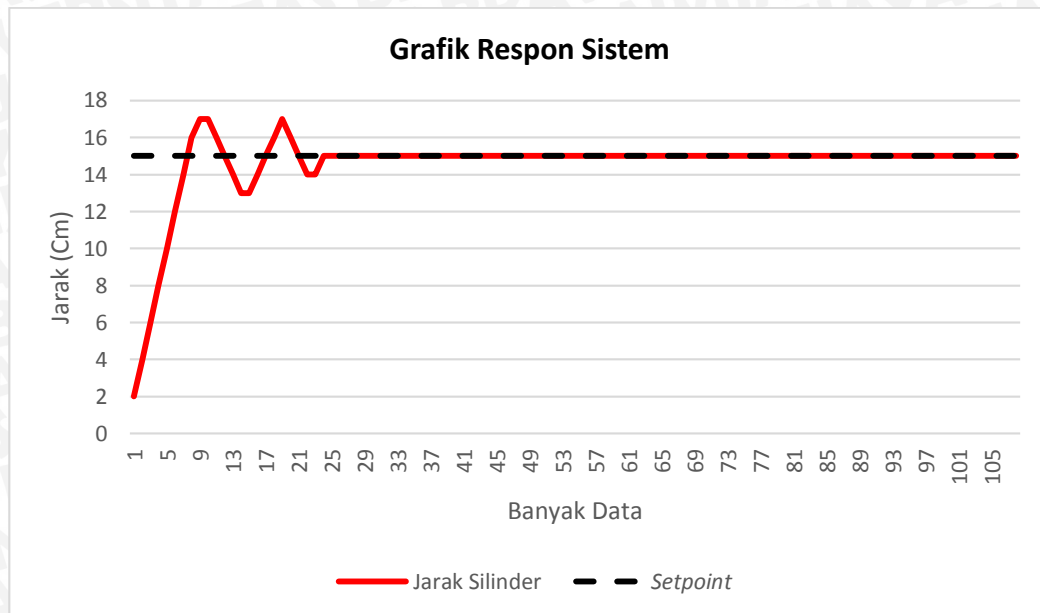
Gambar 5.9 Grafik Respon Sistem dengan Nilai $K_p= 0,2002$, $K_i= 0,1$ dan $K_d= 0,1018$
(Sumber : Perancangan, 2014)

Berdasarkan Gambar 5.9, dapat disimpulkan bahwa respon sistem terdapat *error steady state* (ess). Sehingga parameter tersebut tidak sesuai jika dimasukkan ke dalam sistem. Grafik respon sistem menggunakan masukan unit *step* dengan nilai $K_p= 2,3166$, $K_i= 0,5$ dan $K_d= 1,0182$ dapat dilihat dalam Gambar 5.10.



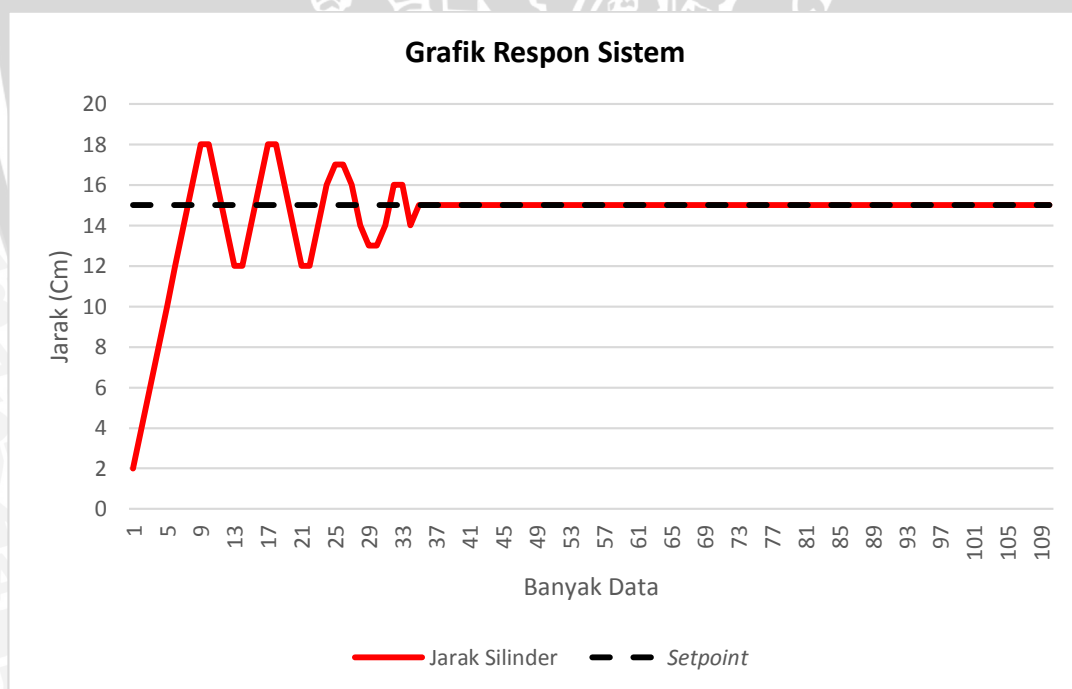
Gambar 5.10 Grafik Respon Sistem dengan Nilai $K_p= 2,3166$, $K_i= 0,5$ dan $K_d= 1,0182$
(Sumber : Perancangan, 2014)

Berdasarkan Gambar 5.10, dapat disimpulkan bahwa respon sistem cepat dan tidak terdapat ess. Sehingga parameter tersebut sesuai jika dimasukkan ke dalam sistem. Grafik respon sistem menggunakan masukan unit *step* dengan nilai $K_p= 3,0256$, $K_i= 1,5$ dan $K_d= 1,5274$ dapat dilihat dalam Gambar 5.11.



Gambar 5.11 Grafik Respon Sistem dengan Nilai $K_p= 3,0256$, $K_i= 1,5$ dan $K_d= 1,5274$ (Sumber : Perancangan, 2014)

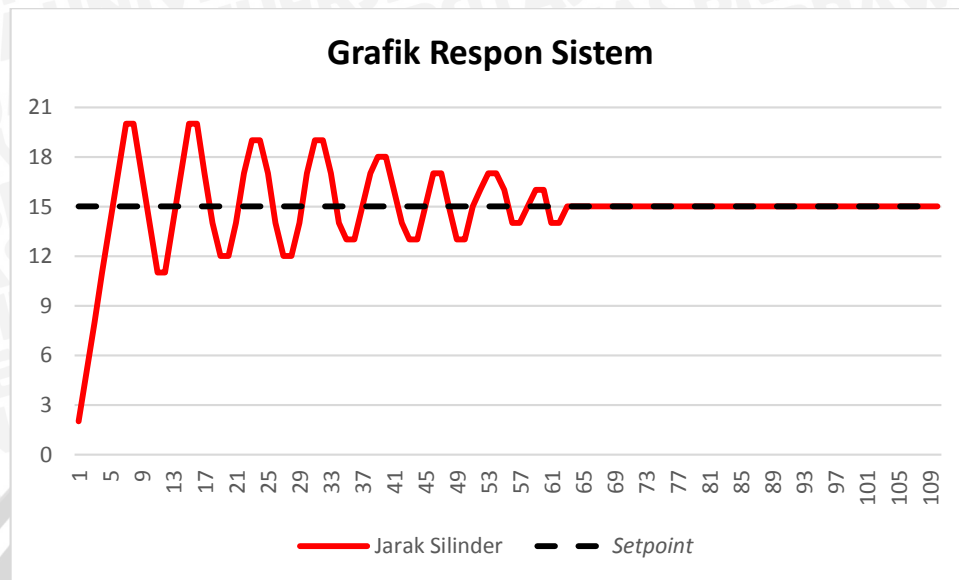
Berdasarkan Gambar 5.11, dapat disimpulkan bahwa respon sistem tidak terdapat *ess* namun terdapat *overshoot*. Grafik respon sistem menggunakan masukan unit *step* dengan nilai $K_p= 5,0438$, $K_i= 2,5$ dan $K_d= 2,5456$ dapat dilihat dalam Gambar 5.12.



Gambar 5.12 Grafik Respon Sistem dengan Nilai $K_p= 5,0438$, $K_i= 2,5$ dan $K_d= 2,5456$ (Sumber : Perancangan, 2014)

Berdasarkan Gambar 5.12, dapat disimpulkan bahwa respon sistem tidak terdapat *ess* namun terdapat *overshoot*. Grafik respon sistem menggunakan masukan

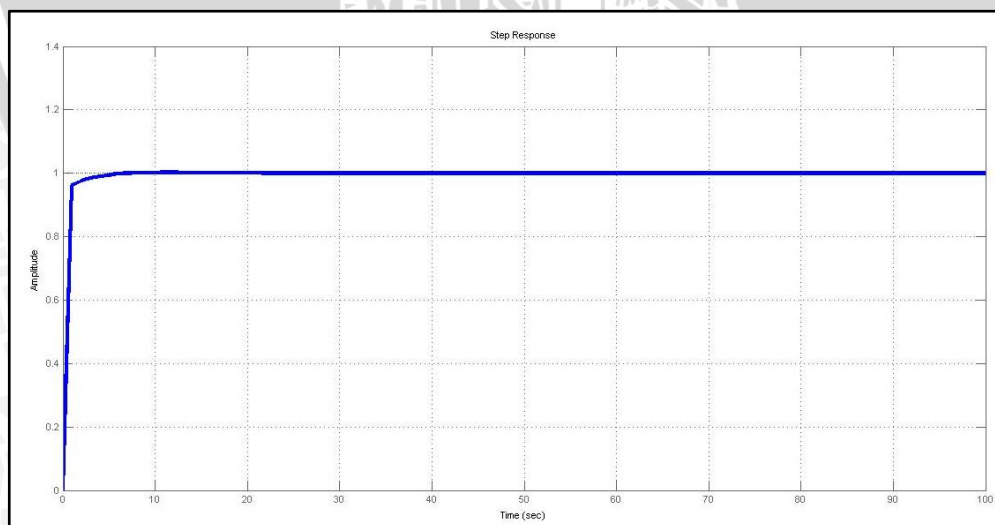
unit step dengan nilai $K_p= 10,089$, $K_i= 5$ dan $K_d= 5,0912$ dapat dilihat dalam Gambar 5.13.



Gambar 5.13 Grafik Respon Sistem dengan Nilai $K_p= 10,089$, $K_i= 5$ dan $K_d= 5,0912$
(Sumber : Perancangan, 2014)

Berdasarkan Gambar 5.13, dapat disimpulkan bahwa respon sistem tidak terdapat *ess* namun terdapat *overshoot*. Jadi, parameter yang paling sesuai untuk sistem ini adalah $K_p=2,3166$, $K_i= 0,5$ dan $K_d= 1,0182$.

Gambar 5.14 di bawah menunjukkan keluaran respon dengan kontroler menggunakan simulasi MATLAB 2010. Masukan berupa unit *step*. Dari hasil simulasi respon yang dihasilkan sesuai dengan *setpoint* yang diinginkan.



Gambar 5.14 Grafik Respon Sistem dengan $K_p= 2.3166$, $K_i= 0.5$ $K_d= 1.0182$ pada MATLAB 2010

(Sumber: Perancangan, 2014)

Hasil parameter PID hasil perhitungan kemudian diimplementasikan pada sistem.

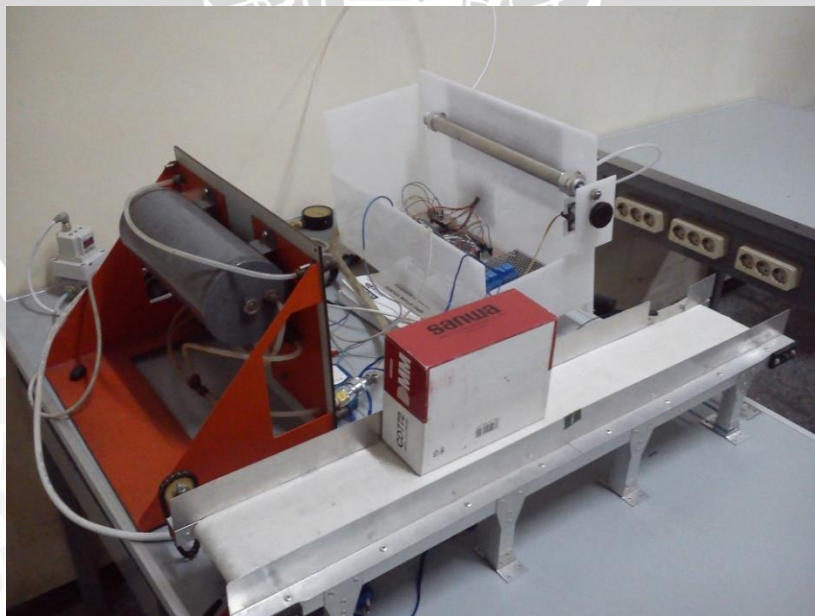
5.5.1 Peralatan Pengujian

Peralatan Pengujian terdiri atas :

- Laptop
- Arduino Uno
- Rangkaian DAC PCF8591
- Rangkaian V/I *converter*
- Sistem *pneumatic*
- *Electro-pneumatic regulator*
- *Switching AC/DC 12V*
- *Liquid Crystal Display (LCD)*
- Rangkaian sensor (PING))
- Konveyor
- *Box*

5.5.2 Prosedur Pengujian

Pengujian dilakukan dengan merangkai peralatan seperti dalam Gambar 5.15 dan kemudian mengunduh nilai $K_p=2.3166$, $K_i=0.5$ dan $K_d=1.0182$ pada *software* Arduino Uno. Mekanismenya, terdapat 4 macam pengujian dengan *setpoint* yang berbeda-beda yakni 10cm, 15cm, 20cm dan 25cm. Pengujian seperti ini dengan tujuan mengetahui bahwa sistem ini bisa digunakan ketika mempunyai *setpoint* yang berbeda-beda.



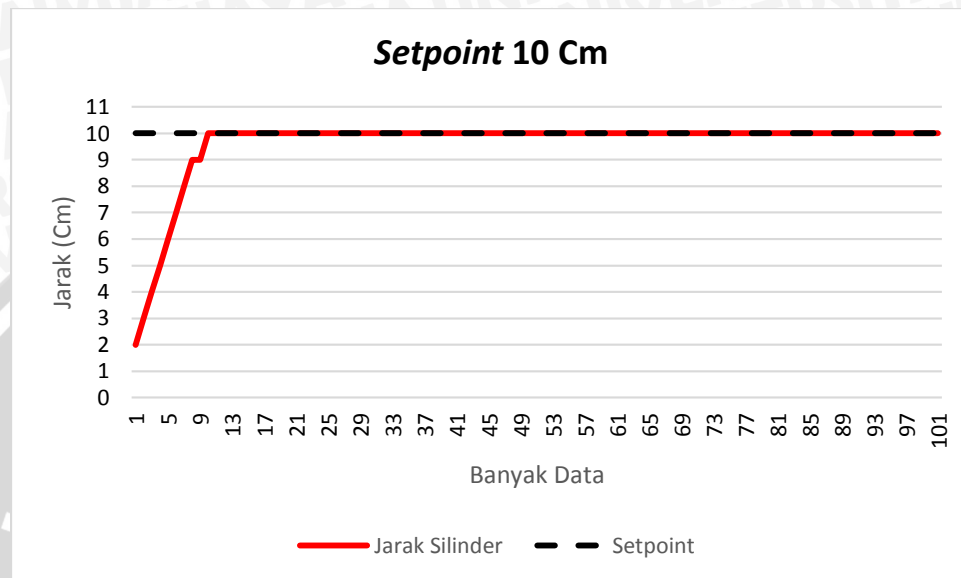
Gambar 5.15 Rangkaian Pengujian *Setpoint* Berbeda
(Sumber: Pengujian, 2014)

5.5.3 Hasil Pengujian

Setelah dilakukan pengujian, didapatkan hasil respon keluaran sistem dengan *setpoint* berbeda.

A. Pengujian dengan *Setpoint* 10cm

Grafik respon dengan *setpoint* 10cm ditunjukkan dalam Gambar 5.16



Gambar 5.16 Hasil Respon dengan *Setpoint* 10cm.

(Sumber: Pengujian, 2014)

Berdasarkan Gambar 5.16, respon keluaran sistem *steady state* tanpa terjadi osilasi. Namun, dengan pembacaan sensor PING))) yang tidak sesuai dengan jarak sebenarnya, maka *error* pada *setpoint* 10cm adalah :

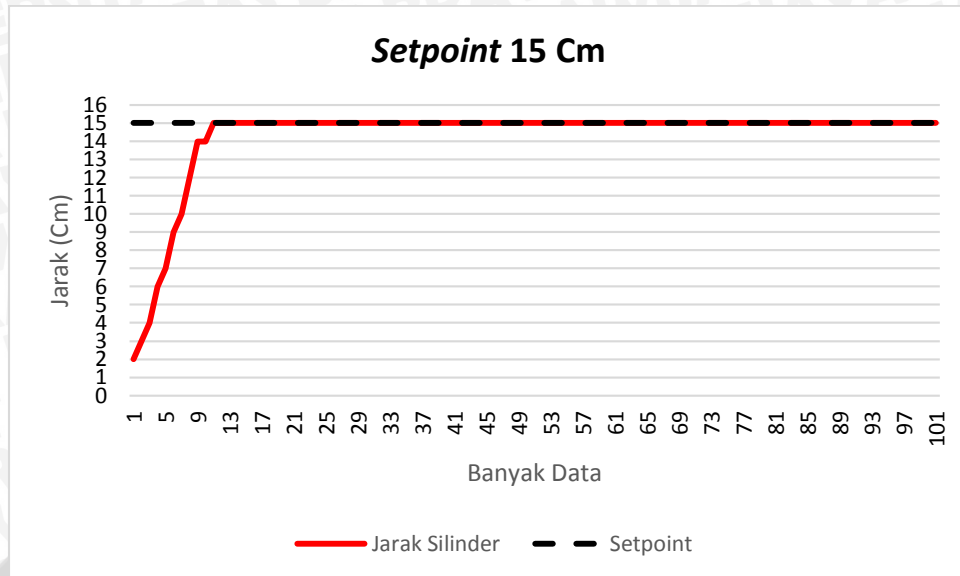
$$\begin{aligned}
 \% \text{ error} &= \frac{|R_d - S_d|}{S_d} \times 100\% \\
 &= \frac{|10.2 - 10|}{10} \times 100\% \\
 &= 0.02 \times 100\% \\
 &= 2\%
 \end{aligned}$$

Keterangan: R_d = Jarak Pembacaan Sebenarnya

S_d = Jarak Pembacaan Sensor

B. Pengujian dengan *Setpoint* 15cm

Grafik respon dengan *setpoint* 15cm ditunjukkan dalam Gambar 5.17



Gambar 5.17 Hasil Respon dengan Setpoint 15cm.

(Sumber: Pengujian, 2014)

Berdasarkan Gambar 5.17, respon keluaran sistem *steady state* tanpa terjadi osilasi. Namun, dengan pembacaan sensor PING))) yang tidak sesuai dengan jarak sebenarnya, maka *error* pada *setpoint* 15cm adalah :

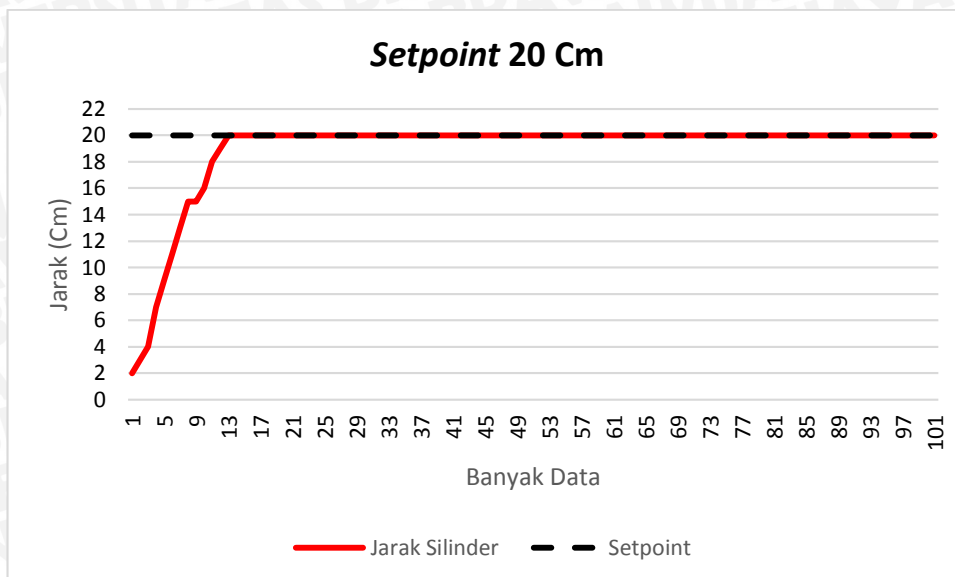
$$\begin{aligned}
 \% \text{ error} &= \frac{|R_d - S_d|}{S_d} \times 100\% \\
 &= \frac{|15,1 - 15|}{15} \times 100\% \\
 &= 0,005 \times 100\% \\
 &= 0,667\%
 \end{aligned}$$

Keterangan: R_d = Jarak Pembacaan Sebenarnya

S_d = Jarak Pembacaan Sensor

C. Pengujian dengan Setpoint 20cm

Grafik respon dengan *setpoint* 20cm ditunjukkan dalam Gambar 5.18.



Gambar 5.18 Hasil Respon dengan Setpoint 20cm.

(Sumber: Pengujian, 2014)

Berdasarkan Gambar 5.18, respon keluaran sistem *steady state* tanpa terjadi osilasi. Namun dengan pembacaan sensor PING))) yang tidak sesuai dengan jarak sebenarnya, maka *error* pada *setpoint* 20cm adalah :

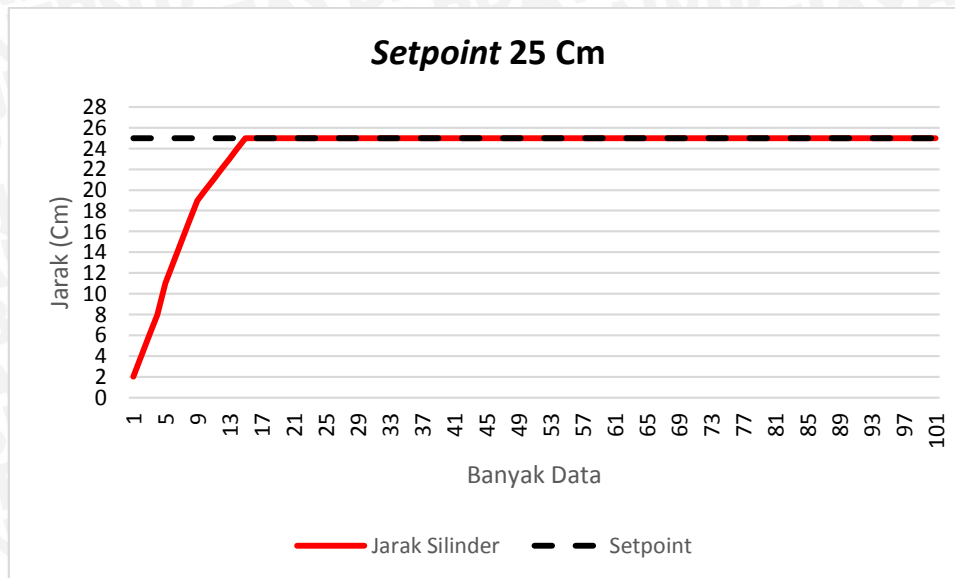
$$\begin{aligned}
 \% \text{ error} &= \frac{|R_d - S_d|}{S_d} \times 100\% \\
 &= \frac{|19.9 - 20|}{20} \times 100\% \\
 &= 0.018 \times 100\% \\
 &= 0,5\%
 \end{aligned}$$

Keterangan: R_d = Jarak Pembacaan Sebenarnya

S_d = Jarak Pembacaan Sensor

D. Pengujian dengan Setpoint 25cm

Grafik respon dengan *setpoint* 25cm ditunjukkan dalam Gambar 5.19.



Gambar 5.19 Hasil Respon dengan Setpoint 25cm.

(Sumber: Pengujian, 2014)

Berdasarkan Gambar 5.19, respon keluaran sistem *steady state* tanpa terjadi osilasi. Namun dengan pembacaan sensor PING))) yang tidak sesuai dengan jarak sebenarnya, maka *error* pada *setpoint* 25cm adalah :

$$\begin{aligned}
 \% \text{ error} &= \frac{|R_d - S_d|}{S_d} \times 100\% \\
 &= \frac{|24.9 - 25|}{25} \times 100\% \\
 &= 0.016 \times 100\% \\
 &= 0,4\%
 \end{aligned}$$

Keterangan: R_d = Jarak Pembacaan Sebenarnya
 S_d = Jarak Pembacaan Sensor

Dari keempat pengujian keseluruhan sistem di atas, dapat dibuat rincian hasilnya dalam Tabel 5.6 di bawah ini.

Tabel 5.6 Hasil 4 pengujian

No.	Setpoint	Settling time (ts)	Error
1.	10cm	1,6 detik	2%
2.	15cm	1,7 detik	0,667%
3.	20cm	1,7 detik	0,5%
4.	25cm	2 detik	0,4%

Berdasarkan Tabel 5.6, *settling time* (ts) rata-rata dari sistem adalah sebesar 1,75 detik dan *error* rata-rata sebesar 0,892%. Dikarenakan *error*-nya di bawah 5% maka termasuk dalam toleransi *error steady state* (ess).



BAB VI PENUTUP

6.1 Kesimpulan

Perancangan parameter PID menggunakan metode *root locus* dengan nilai *pole* $s = -0,991$ didapatkan nilai $K_p = 2.3166$, $K_i = 0.5$ dan $K_d = 1.0182$. Sistem ini menggunakan kontroler Arduino Uno beserta *software*-nya “Arduino ERW 1.05”. Berdasarkan hasil pengujian dengan 4 macam setpoint yang berbeda-beda (10cm, 15cm, 20cm, dan 25cm) diperoleh *settling time* (ts) rata-rata sebesar 1,75 detik dan terdapat *error* sebesar 0,892%. *Error* tersebut di bawah toleransi *error steady state* (ess) sebesar 5%.

6.2 Saran

Dalam perancangan dan pembuatan alat ini masih terdapat kelemahan. Untuk memperbaiki kinerja alat dan pengembangan lebih lanjut disarankan :

1. Disarankan menggunakan sensor jarak dengan tingkat kesalahan pembacaan atau *error* seminimal mungkin.
2. Disarankan untuk mencari parameter baru dengan cara melakukan peletakan posisi silinder dengan posisi vertikal, karena pada penelitian ini posisi silinder diletakkan secara horisontal.
3. Disarankan menggunakan pengontrolan dengan metode selain PID.

DAFTAR PUSTAKA

- Anonim, 2014, prooduk kompresor [online], (<http://www.nuvair.com/>, diakses tanggal 6 Januari 2014).
- Budi, Dimas. 2014. Aplikasi Kontroler Pid Dalam Pengendalian Posisi *Stamping Rod* Berbasis *Pneumatic* Menggunakan Arduino Uno. Universitas Brawijaya.
- Fajar, Andi, 2013, Prinsip Kerja Sensor PING))) [online], (<http://kuliah.andifajar.com/sensor-ultrasonic/>, diakses tanggal 7 Januari 2014).
- Kustanti, Ika. 2014. Pengendalian Kadar Keasaman (Ph) pada Sistem Hidroponik Stroberi Menggunakan Kontroler Pid Berbasis Arduino Uno. Universitas Brawijaya.
- Palinggi, Wido S. 2006. Kontrol Posisi Silinder Linear Pneumatik Dengan Katup On/Off Menggunakan Kontroler Proportional Integral Derivative (PID). Universitas Kristen Petra.
- Philip, C. L. & Harbor, R. D. 1996. *Feedback Control System*. Diterjemahkan oleh Widodo.R.J. Prentice Hall. New Jersey
- Rahmat, 2013, Sistem Pneumatik [online], (<http://monangcidakkal.blogspot.com/>, diakses tanggal 5 Januari 2014).
- SMC Corporation. 2012. *CM2 datasheet manual.pdf*
- SMC. *The Pneubook*. SMC Corporation.
- Wiradhana, Raditya. 2013. Sistem Pengendalian Suhu Pada Tungku Bakar Menggunakan Kontroler PID. Universitas Brawijaya
- Waller, D., Werner, H. 2002. *Pneumatics Workboos Basic Level*. Festo Didactic GmbH & Co.
- Wikipedia, 2014, Komponen *Pressure Regulator* beserta cara kerjanya [online], (<http://en.wikipedia.org/wiki/file:single-stage-regulator.svg>, diakses 10 Januari 2014).
- Zen, Ahmad D. 2013. Pengendalian Posisi *Stamping Rod* Berbasis *Pneumatic* Menggunakan DCS Centum VP. Universitas Brawijaya.

LAMPIRAN 1

Foto Alat





Gambar 1. Alat tampak depan



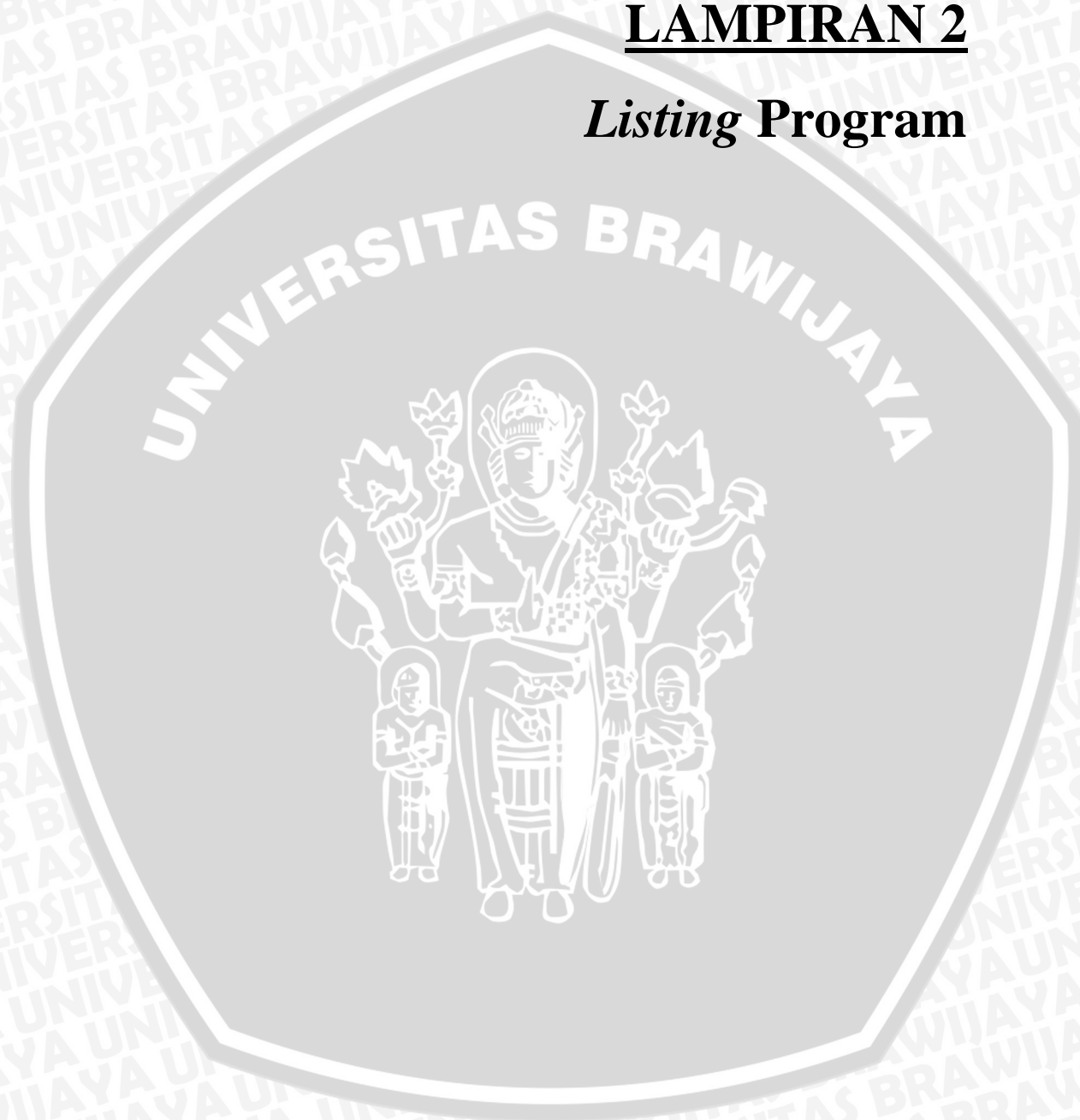
Gambar 2. Sistem *Pneumatic* tampak dari belakang



Gambar 3. Sistem *Pneumatic* tampak dari samping

LAMPIRAN 2

Listing Program



```
#include <LiquidCrystal.h>
#include "Wire.h"
#define PCF8591 (0x90 >> 1) // I2C bus address
int PinMotor = 13;
int LSwitch = 10;
float pingPin = 7;
int error,deleerror,sigmaerror,out,setpoin,lasterror,lasterror2;
unsigned char kp=2.3166,ki=0.5,kd=1.0182;
long duration, inches, cm;
long previousMillis = 0;
long currentMillis;
long interval = 1000;
int i=0;
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

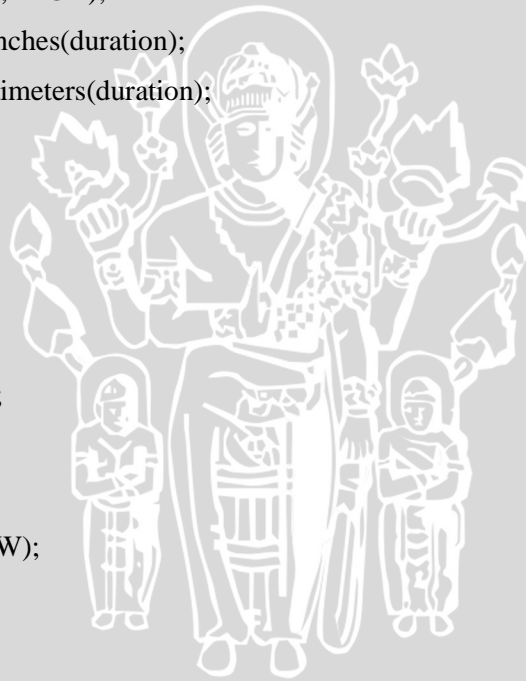
void setup()
{
  Wire.begin();
  Serial.begin(9600);
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("Ade A. Yana");
}

void loop()
{
  unsigned long currentMillis = millis();

  if(currentMillis - previousMillis >= interval) {
    // save the last time you blinked the LED
    i=i+1;
    previousMillis = currentMillis;
    //Serial.print(i);
    //Serial.println("  sekon");
  }
}
```



```
}  
  
setpoin = 10;  
//ambil data ping  
pinMode(pingPin, OUTPUT);  
digitalWrite(pingPin, LOW);  
delayMicroseconds(2);  
digitalWrite(pingPin, HIGH);  
delayMicroseconds(5);  
digitalWrite(pingPin, LOW);  
pinMode(pingPin, INPUT);  
duration = pulseIn(pingPin, HIGH);  
inches = microsecondsToInches(duration);  
cm = microsecondsToCentimeters(duration);  
  
//selesai  
Serial.print(cm);  
//Serial.print("cm");  
Serial.println();  
pinMode(LSwitch, INPUT);  
if(LSwitch=HIGH){  
error = setpoin-cm;  
digitalWrite (PinMotor, LOW);  
  
}  
else{  
error = 0;  
digitalWrite (PinMotor, HIGH);  
}  
delay(5);  
  
sigmaerror+=error;  
error = setpoin-cm;  
out = (105 + (kp*(error-lasterror) + ki*(sigmaerror/2) + kd*(error-(2*lasterror)+lasterror2)));
```



```
if(out>250)out=250;
else if(out<1)out=1;
//Serial.print(error);
//Serial.print("e");
//Serial.println();

delay(100);
Wire.beginTransmission(PCF8591); // wake up PCF8591
Wire.write(0x40); // control byte - turn on DAC (binary 1000000)
Wire.write(out); // value to send to DAC
Wire.endTransmission(); // end transmission
lasterror2=lasterror;
lasterror=error;
deleerror=error-lasterror;

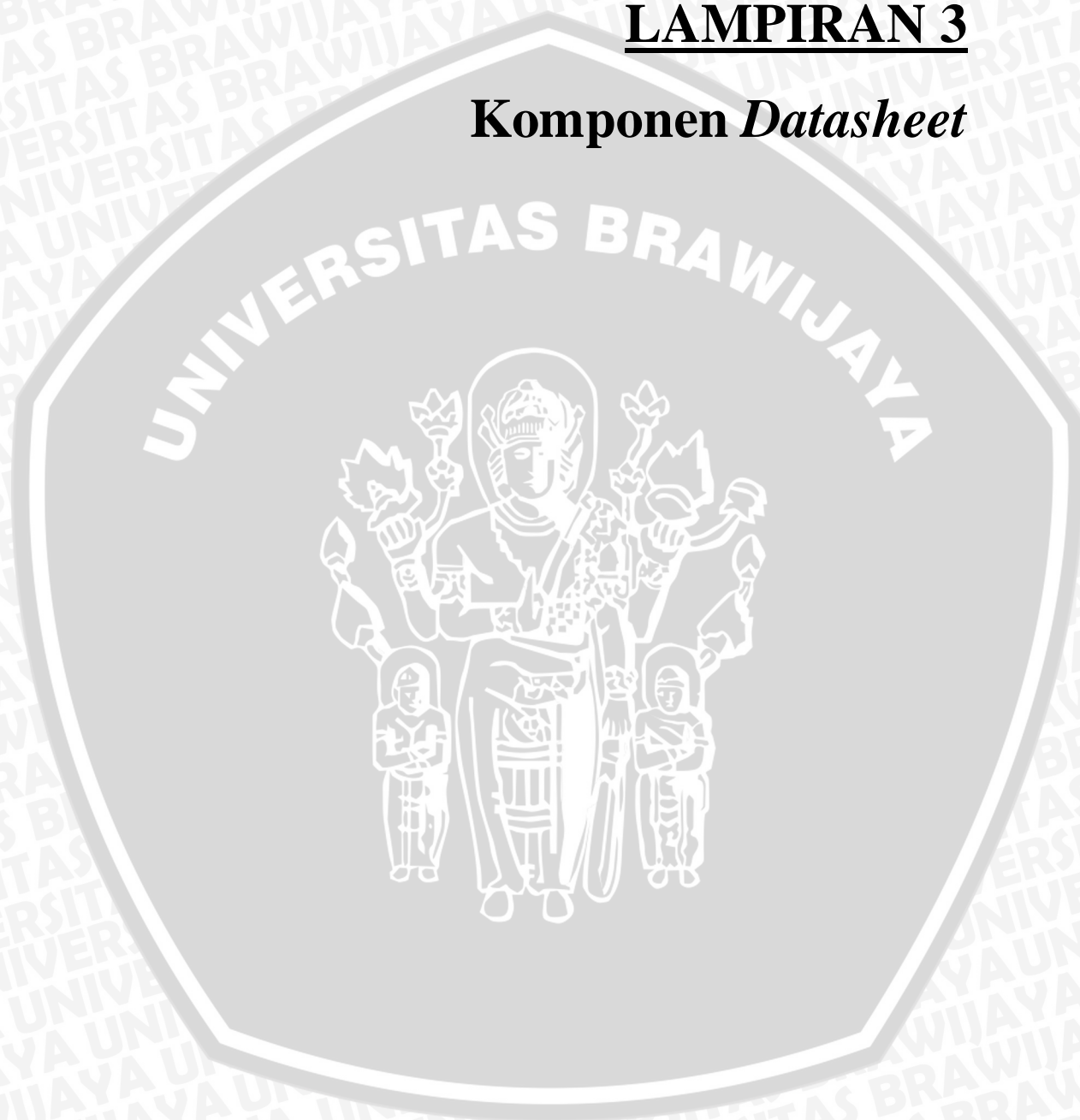
lcd.begin(16, 2);
lcd.setCursor(0,0);
lcd.print("setpoint= ");
lcd.setCursor(10,0);
lcd.print(setpoin);
lcd.setCursor(0,1);
lcd.print("Jarak Benda= ");
lcd.setCursor(14,1);
lcd.print(cm);
}
long microsecondsToInches(long microseconds)
{
    return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds)
{
    return microseconds / 29 / 2;
}
```

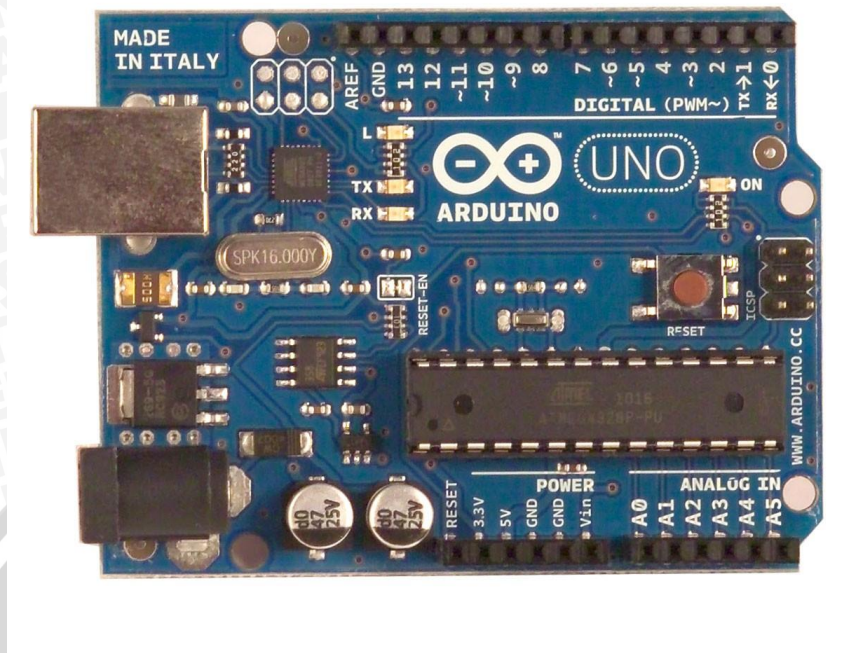


LAMPIRAN 3

Komponen *Datasheet*



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 1.0 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](#).

Index

Technical Specifications	Page 2
How to use Arduino Programming Enviroment, Basic Tutorials	Page 6
Terms & Conditions	Page 7
Enviromental Policies half sqm of green via Impatto Zero®	Page 7



radiospares

RADIONICS



Technical Specification

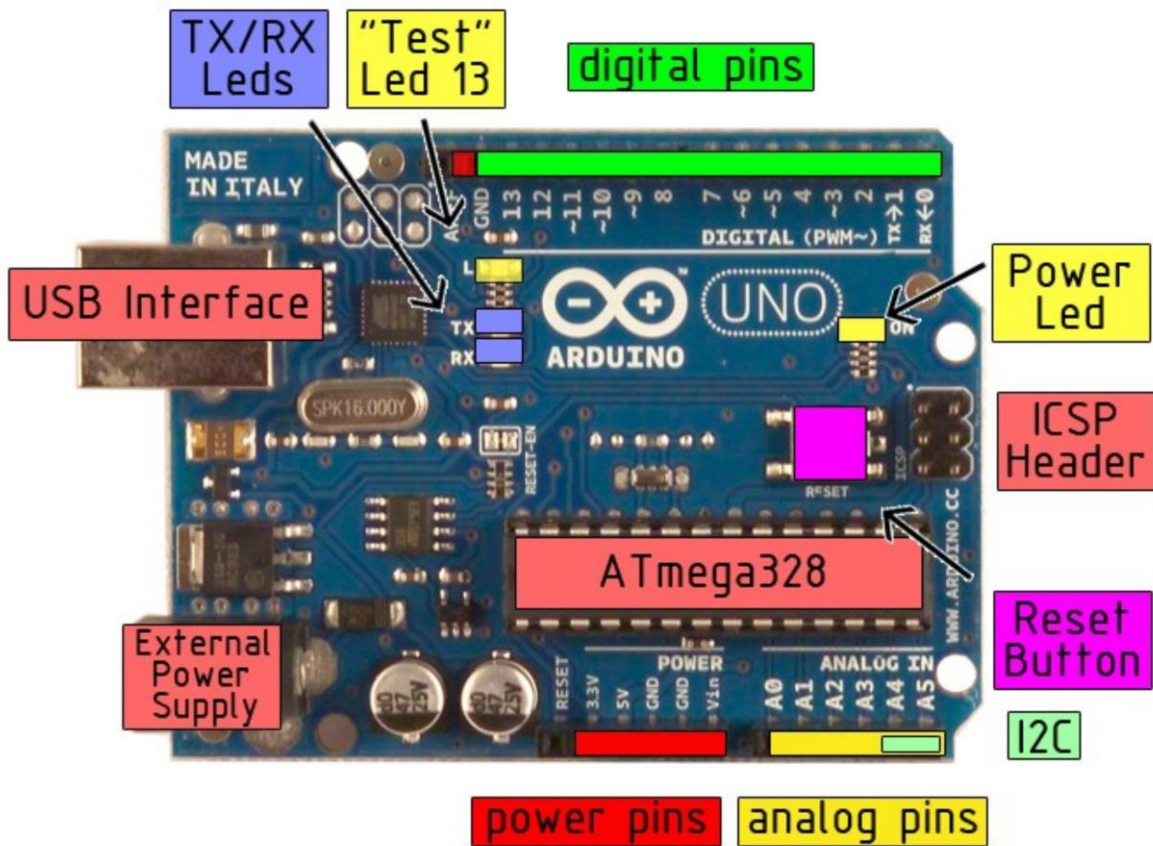


EAGLE files: [arduino-duemilanove-uno-design.zip](#) Schematic: [arduino-uno-schematic.pdf](#)

Summary

Microcontroller	ATmega328
Operating Voltage	
5V Input Voltage	
(recommended) 7-12V Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	
40 mA DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	
2 KB EEPROM	
1 KB Clock Speed	
16 MHz	

the board



radiospares

RADIONICS





599 Menlo Drive, Suite 100
 Rocklin, California 95765, USA
 Office: (916) 624-8333
 Fax: (916) 624-8003

General: info@parallax.com
 Technical: support@parallax.com
 Web Site: www.parallax.com
 Educational: www.stampsinclass.com

PING)))™ Ultrasonic Distance Sensor (#28015)

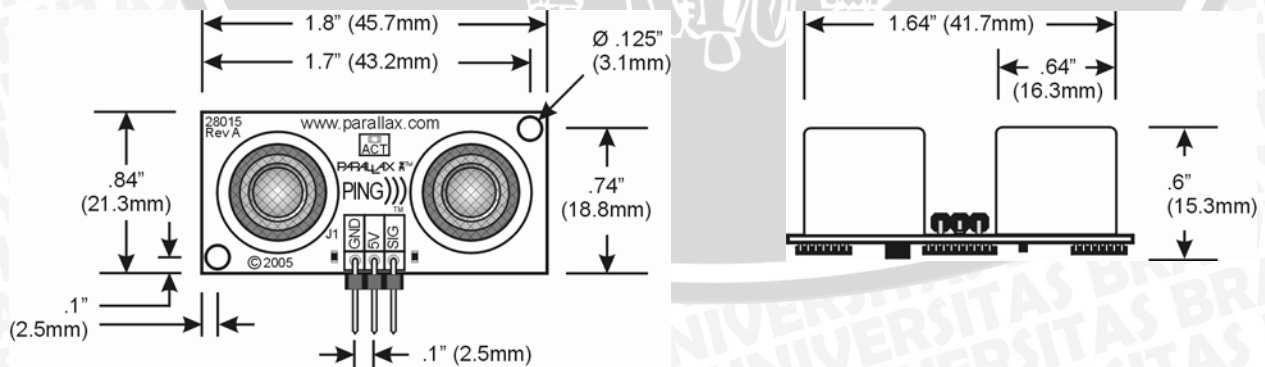
The Parallax PING))) ultrasonic distance sensor provides precise, non-contact distance measurements from about 2 cm (0.8 inches) to 3 meters (3.3 yards). It is very easy to connect to BASIC Stamp® or Javelin Stamp microcontrollers, requiring only one I/O pin.

The PING))) sensor works by transmitting an ultrasonic (well above human hearing range) burst and providing an output pulse that corresponds to the time required for the burst echo to return to the sensor. By measuring the echo pulse width the distance to target can easily be calculated.

Features

- Supply Voltage – 5 VDC
- Supply Current – 30 mA typ; 35 mA max
- Range – 2 cm to 3 m (0.8 in to 3.3 yds)
- Input Trigger – positive TTL pulse, 2 μ S min, 5 μ S typ.
- Echo Pulse – positive TTL pulse, 115 μ S to 18.5 ms
- Echo Hold-off – 750 μ S from fall of Trigger pulse
- Burst Frequency – 40 kHz for 200 μ S
- Burst Indicator LED shows sensor activity
- Delay before next measurement – 200 μ S
- Size – 22 mm H x 46 mm W x 16 mm D (0.84 in x 1.8 in x 0.6 in)

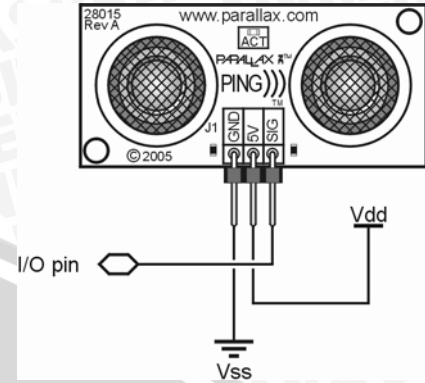
Dimensions



Pin Definitions

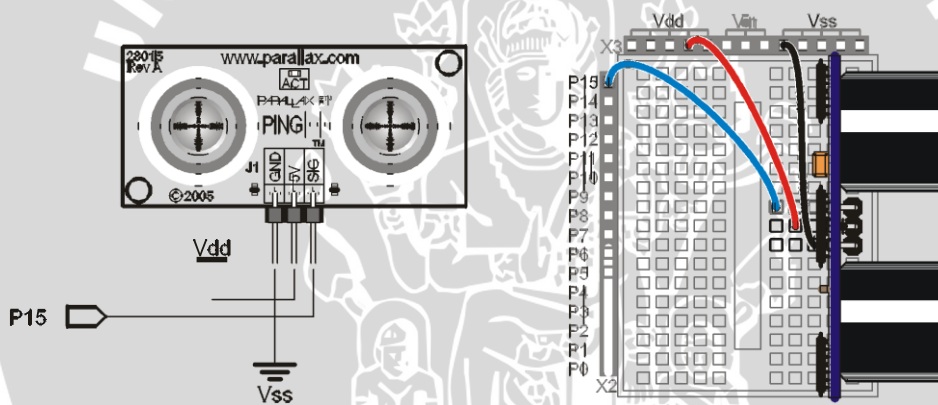
GND	Ground (Vss)
5 V	5 VDC (Vdd)
SIG	Signal (I/O pin)

The PING))) sensor has a male 3-pin header used to supply power (5 VDC), ground, and signal. The header allows the sensor to be plugged into a solderless breadboard, or to be located remotely through the use of a standard servo extender cable (Parallax part #805-00002). Standard connections are shown in the diagram to the right.



Quick-Start Circuit

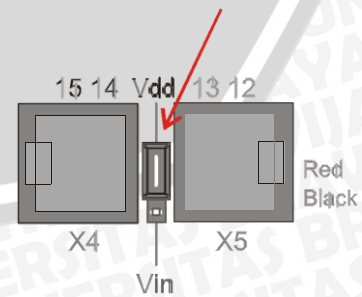
This circuit allows you to quickly connect your PING))) sensor to a BASIC Stamp® 2 via the Board of Education® breadboard area. The PING))) module's GND pin connects to Vss, the 5 V pin connects to Vdd, and the SIG pin connects to I/O pin P15. This circuit will work with the example program Ping_Demo.BS2 listed on page 7.



Servo Cable and Port Cautions

If you want to connect your PING))) sensor to a Board of Education using a servo extension cable, follow these steps:

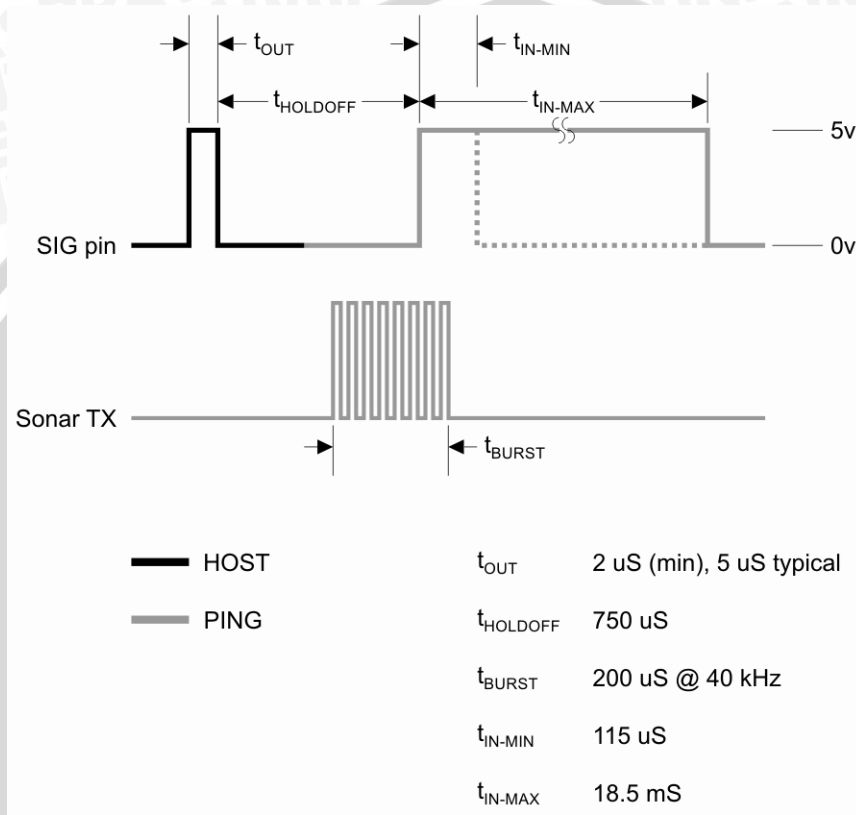
1. When plugging the cable onto the PING))) sensor, connect Black to GND, Red to 5 V, and White to SIG.
2. Check to see if your Board of Education servo ports have a jumper, as shown at right.
3. If your Board of Education servo ports have a jumper, set it to Vdd as shown.
4. If your Board of Education servo ports do not have a jumper, do not use them with the PING))) sensor. These ports only provide Vin, not Vdd, and this may damage your PING))) sensor. Go to the next step.
5. Connect the servo cable directly to the breadboard with a 3-pin header. Then, use jumper wires to connect Black to Vss, Red to Vdd, and White to I/O pin P15.



Board of Education Servo Port Jumper, Set to Vdd

Theory of Operation

The PING))) sensor detects objects by emitting a short ultrasonic burst and then "listening" for the echo. Under control of a host microcontroller (trigger pulse), the sensor emits a short 40 kHz (ultrasonic) burst. This burst travels through the air at about 1130 feet per second, hits an object and then bounces back to the sensor. The PING))) sensor provides an output pulse to the host that will terminate when the echo is detected, hence the width of this pulse corresponds to the distance to the target.



Test Data

The test data on the following pages is based on the PING))) sensor, tested in the Parallax lab, while connected to a BASIC Stamp microcontroller module. The test surface was a linoleum floor, so the sensor was elevated to minimize floor reflections in the data. All tests were conducted at room temperature, indoors, in a protected environment. The target was always centered at the same elevation as the PING))) sensor.

Test 1

Sensor Elevation: 40 in. (101.6 cm)

Target: 3.5 in. (8.9 cm) diameter cylinder, 4 ft. (121.9 cm) tall – vertical orientation

