

**PERANCANGAN DAN IMPLEMENTASI SISTEM KENDALI
UNTUK ROTARY PENDULUM DENGAN KONTROLLER PID
MENGUNAKAN SCILAB-ARDUINO**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Mesin**



Disusun Oleh :

Nama : Faza Hadiyan

No. Mahasiswa : 16525097

NIRM : 2016080690

**JURUSAN TEKNIK MESIN
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2021

Pernyataan Keaslian

“Dengan ini saya menyatakan bahwa karya tulis ilmiah yang saya buat merupakan karya sendiri dan bukan hasil plagiasi dari karya tulis orang lain. Semua referensi dan kutipan yang terdapat pada karya tulis ini telah saya cantumkan sitasi dan sumber pustakanya mengikuti tata cara pengutipan karya ilmiah yang benar. Apabila dikemudian hari saya dianggap melakukan pelanggaran hak kekayaan intelektual dan terbukti melanggar hak tersebut, maka saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.”



LEMBAR PENGESAHAN DOSEN PEMBIMBING

**PERANCANGAN DAN IMPLEMENTASI SISTEM KENDALI UNTUK
ROTARY PENDULUM DENGAN KONTROLLER PID
MENGUNAKAN SCILAB-ARDUINO**

TUGAS AKHIR

Disusun Oleh :

Nama : Faza Hadiyan
No. Mahasiswa : 16525097
NIRM : 2016080690

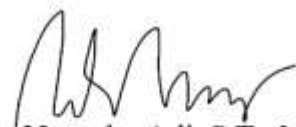
Yogyakarta, 21 Juni 2021

Pembimbing I,



Purtojo, S.T., M.Sc.

Pembimbing II,



Agung Nugroho Adi, S.T., M.T.

LEMBAR PENGESAHAN DOSEN PENGUJI

PERANCANGAN DAN IMPLEMENTASI SISTEM KENDALI UNTUK ROTARY PENDULUM DENGAN KONTROLLER PID MENGUNAKAN SCILAB-ARDUINO

TUGAS AKHIR

Disusun Oleh :

Nama : Faza Hadiyan


No. Mahasiswa : 16525097

NIRM : 2016080690

Tim Penguji

Purtojo, S.T., M.Sc.

Ketua


Tanggal : 7 Juli 2021


Donny Suryawan, S.T., M.Eng.

Anggota I


Tanggal : 1 Juli 2021

Finny Pratama Putera, S.T., M.Eng.


Anggota II


Tanggal : 1 Juli 2021

Mengetahui

Ketua Jurusan Teknik Mesin




Dr. Eng. Risdiyono, S.T., M.Eng.

HALAMAN PERSEMBAHAN

“Sesungguhnya bersama kesulitan ada kemudahan. Maka apabila engkau telah selesai (dari sesuatu urusan), tetaplah bekerja keras (untuk urusan yang lain).

Dan hanya kepada Tuhanmulah engkau berharap.”

(QS. Al-Insyirah: 6-8)

Tugas Akhir ini saya persembahkan kepada keluarga, saudara, sahabat dan teman-teman yang senantiasa percaya dan mendukung saya disetiap langkah yang saya ambil, terkhusus kepada kedua orang tua saya:

Ayah dan Ibu tercinta

Sonny Sandra Atmodjo dan Siti Nur Rohmah

Sebagai tanda bukti atas amanah yang diberikan kepada saya untuk melanjutkan pendidikan ke jenjang S1. Terimakasih telah berusaha mendidik dan mengantarkan saya mencapai gelar sarjana. Semoga gelar ini dapat menjadi jembatan saya untuk membanggakan kalian, dan segala proses perjuangan kalian dapat menjadi pahala yang bermanfaat di akhirat kelak.

Aamiin ya rabbal alamin.

HALAMAN MOTTO

Rasulullah saw bersabda, barang siapa menempuh jalan untuk mendapatkan ilmu, Allah SWT akan memudahkan baginya jalan menuju surga.

(Hadits Riwayat Muslim)

Education is the key to unlocking the world, a passport to freedom.

(Oprah Winfrey)

Work hard in silence, let success be your noise.

(Frank Ocean)

A smooth sea never made a skilled sailor.

(Franklin D Roosevelt)

المعهد الإسلامي
الاستاذ الأندلسي

KATA PENGANTAR ATAU UCAPAN TERIMA KASIH



Assalamu'alaikum warrahmatullahi wabarakatuh,

Alhamdulillahirabbil'alamin, puji syukur dipanjatkan atas kehadiran Allah SWT yang telah memberikan rahmat, hidayah, dan karunia-Nya sehingga tugas akhir yang berjudul "***Perancangan dan Impementasi Sistem Kendali untuk Rotary Pendulum dengan Kontroller PID Menggunakan Scilab-Arduino***" dapat diselesaikan dengan sebaik-baiknya. Tidak lupa shalawat serta salam dipanjatkan kepada junjungan kita Nabi Muhammad saw.

Tidak lupa pula ucapan terimakasih kepada pihak-pihak yang selalu mendukung dan memberi bantuan baik secara moril ataupun materil sehingga tugas akhir ini dapat diselesaikan dengan sebaik-baiknya. Terkhusus kepada:

1. Sonny Sandra Atmodjo dan Siti Nur Rohmah, selaku kedua orang tua penulis yang dengan usaha serta doa-nya lah penulis dapat sampai dititik ini. Terimakasih banyak atas kasih sayang dan nasihat yang diberikan kepada penulis sampai akhirnya penulis dapat menyelesaikan tugas akhir ini.
2. Bapak Purtojo, S.T., M.Sc., sebagai dosen pembimbing pertama yang telah berjasa meluangkan waktunya untuk membimbing, mengarahkan, memberikan ilmu serta mendukung penulis selama pembuatan dan penyusunan tugas akhir ini. Semoga ilmu yang diberikan dapat menjadi berkah yang berguna di kemudian hari. *Aamiin ya rabbal alamin.*
3. Bapak Agung Nugroho Adi, S.T., M.T., sebagai dosen pembimbing kedua yang juga berjasa karena meluangkan waktunya untuk membimbing, mengarahkan, memberikan ilmu serta mendukung penulis selama pembuatan dan penyusunan tugas akhir ini. Semoga ilmu yang diberikan juga dapat menjadi berkah yang berguna di kemudian hari. *Aamiin ya rabbal alamin.*

4. Hafizha Hudiyatmo dan Alya Putri Dzalika, selaku saudara dan saudari penulis. Terimakasih atas dukungan dan doa kalian selama ini.
5. Terimakasih untuk seluruh keluarga besar penulis baik dari keluarga ayah ataupun keluarga ibu, yang selalu siap sedia membantu dan memberi dukungan baik dari segi rohani ataupun jasmani.
6. Terimakasih untuk sahabat dan teman-teman yang selalu memberikan candaan dan solusi atas masalah-masalah yang dihadapi.
7. Terimakasih untuk semua orang yang memiliki jasa atas kehidupan penulis. Tentunya penulis tidak akan bisa sampai dititik ini tanpa jasa-jasa dari kalian.

Semoga amal dan perbuatan kalian dapat dibalas dengan kebaikan oleh Allah SWT. Penulis berharap pembaca laporan ini dapat memberikan kritik dan saran yang membangun untuk perkembangan tugas akhir ini. Semoga tugas akhir ini dapat memberikan manfaat bagi masyarakat umum.

Wassalamualaikum warrahmatullahi wabarakatuh.

Yogyakarta, 17 Juni 2021

Penulis,

FAZA HADIYAN

**PERANCANGAN DAN IMPLEMENTASI SISTEM KENDALI UNTUK
ROTARY PENDULUM DENGAN KONTROLLER PID
MENGUNAKAN SCILAB-ARDUINO**

Faza Hadiyan

ABSTRAK

Tugas akhir kali ini melakukan penelitian terkait kendali kontrol untuk mengurangi ayunan akibat pergerakan rotasi dari sebuah objek kendali. Untuk melakukan penelitian terkait kendali ayunan akibat gerakan rotasi, digunakan prototipe rotary pendulum. Komponen utama pada alat rotary pendulum adalah motor DC, pendulum ayun, dan dua sensor encoder. Sensor encoder digunakan untuk melacak perubahan posisi dari motor DC dan pendulum. Metode sistem kendali yang digunakan merupakan metode kendali PID (Proportional, Integral, Derivative). Penggunaan metode kendali PID bertujuan untuk meminimalisir ayunan yang terjadi pada pendulum akibat dari pergerakan rotasi yang disebabkan oleh motor DC.

Kata kunci : kendali ayunan, rotary pendulum, PID

**DESIGN AND IMPLEMENTATION OF CONTROL SYSTEM
FOR ROTARY PENDULUM WITH PID CONTROLLER USING
SCILAB-ARDUINO**

Faza Hadiyan

ABSTRACT

This final project is conducting research related to reduce the swing due to the rotational movement of a control object. To conduct research related to swing control due to rotational motion, a rotary pendulum prototype was used. The main components of a rotary pendulum device are a DC motor, swing pendulum, and two encoder sensors. The encoder sensor is used to track the change in position of the DC motor and pendulum. The software used is Scilab Xcos and Arduino IDE. Sci-lab X-cos has functions to simulate rotary pendulum control objects and designing a systematic model of the control system. The PID (Proportional, Integral, Derivative) control is used as a control system method. The use of the PID control method aims to minimize the swing that occurs in the pendulum as a result of the rotational movement caused by the DC motor.

Keyword: swing control, rotary pendulum, PID

DAFTAR ISI

Halaman Judul	i
Lembar Pengesahan Dosen Pembimbing	iii
Lembar Pengesahan Dosen Penguji	iv
Halaman Persembahan	v
Halaman Motto	vi
Kata Pengantar atau Ucapan Terima Kasih	vii
Abstrak	ix
Daftar Isi	xi
Daftar Tabel	xiii
Daftar Gambar	xiv
Daftar Notasi	xvi
Bab 1 Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian atau Perancangan	3
1.5 Manfaat Penelitian atau Perancangan	4
1.6 Sistematika Penulisan	4
Bab 2 Tinjauan Pustaka	5
2.1 Kajian Pustaka	5
2.2 Dasar Teori	6
2.2.1 <i>Rotary Pendulum</i>	6
2.2.2 Sistem kontrol	10
2.2.3 Scilab Xcos	13
2.2.4 Model Matematis	13
2.2.5 PID	15
Bab 3 Metode Penelitian	20
3.1 Alur Penelitian	20
3.2 Perancangan	21
3.2.1 Perancangan Perangkat Keras	21

3.2.2	Perancangan Perangkat Lunak.....	23
3.3	Metode pengujian	26
3.3.1	Pemeriksaan sensor encoder.....	26
3.3.2	Pemeriksaan respon motor DC.....	27
3.3.3	Pemeriksaan respons pendulum ayun.....	28
3.3.4	Simulasi model matematis.....	28
3.3.5	Pengujian kendali PID objek kendali <i>rotary pendulum</i>	29
Bab 4	Hasil dan Pembahasan	31
4.1	Hasil Pemeriksaan objek kendali <i>rotary pendulum</i>	31
4.1.1	Hasil pemeriksaan sensor encoder.....	31
4.1.2	Hasil pemeriksaan respon motor DC.....	32
4.1.3	Pemeriksaan respon pendulum ayun	33
4.2	Hasil perhitungan Model Matematis.....	34
4.2.1	Hasil Fungsi Transfer Lengan Pendulum (Motor DC).....	34
4.2.2	Hasil Fungsi Transfer Pendulum Ayun	35
4.2.3	Simulasi fungsi transfer alat <i>rotary pendulum</i>	37
4.3	Hasil Pengujian kendali PID objek kendali <i>rotary pendulum</i>	39
4.3.1	Pengujian tanpa Kendali PID	39
4.3.2	Pengujian kendali PID+PD metode ZNFD	41
4.3.3	Pengujian kendali PID+PD metode Cohen-Coon	43
4.3.4	Pengujian kendali PID+PD manual tuning.....	46
4.4	Pembahasan	48
Bab 5	Penutup.....	51
5.1	Kesimpulan	51
5.2	Saran atau Penelitian Selanjutnya.....	52
Daftar Pustaka	53
Lampiran	55

DAFTAR TABEL

Tabel 2-1 Spesifikasi Arduino Mega 2560.....	8
Tabel 2-2 pengaruh masing-masing konstanta PID (Toochinda, 2016).....	16
Tabel 2-3 Perhitungan konstanta PID metode ZNFD manual (Toochinda, 2016).	17
Tabel 2-4 Perhitungan konstanta PID menggunakan metode Cohen-Coon (Wicaksono, 2004).	18
Tabel 3-1 Blok-blok diagram yang digunakan.....	24
Tabel 4-1 <i>Identified Data</i> alat rotary pendulum.....	34



DAFTAR GAMBAR

Gambar 2-1 Mekanisme gerakan <i>rotary pendulum</i>	7
Gambar 2-2 Arduino Mega 2560 (Datasheet Arduino Mega 2560).....	8
Gambar 2-3 Kerangka motor DC (Datasheet Motor DC Maxon S2322).....	9
Gambar 2-4 Motor Driver L298N	10
Gambar 2-5 Optical Rotary Encoder (Datasheet Encoder Autonics E40s).....	10
Gambar 2-6 sistem kontrol sederhana.	11
Gambar 2-7 Sistem <i>open loop</i>	11
Gambar 2-8 Sistem <i>closed loop</i>	12
Gambar 2-9 grafik sistem <i>closed loop</i> (Toochinda, 2016)	12
Gambar 2-10 Contoh penggunaan aplikasi Xcos pada Scilab.....	13
Gambar 2-11 Ilustrasi pergerakan pendulum sederhana (Akbar, Rahmdhani, & Wibowo, 2016).	14
Gambar 2-12 Respon sistem menggunakan <i>damping</i>	15
Gambar 2-13 Respon sistem menggunakan kendali PID (Nise, 2015).	17
Gambar 2-14 respon <i>quarter amplitude decay</i> (Wicaksono, 2004).	18
Gambar 3-1 Alur penelitian.....	20
Gambar 3-2 Model rancangan perangkat keras <i>rotary pendulum</i>	21
Gambar 3-3 Koneksi elektrikal alat <i>rotary pendulum</i>	22
Gambar 3-4 Objek kendali <i>rotary pendulum</i>	23
Gambar 3-5 Xcos pengecekan pulsa sensor encoder.....	26
Gambar 3-6 Konfigurasi sensor encoder motor DC.....	26
Gambar 3-7 Konfigurasi sensor encoder pendulum ayun	27
Gambar 3-8 Konfigurasi Xcos untuk respon posisi motor DC	27
Gambar 3-9 Skema Xcos untuk respon osilasi pendulum ayun.	28
Gambar 3-10 skema Xcos untuk simulasi model matematis objek kendali <i>rotary pendulum</i>	29
Gambar 3-11 Skema Xcos uji coba kendali PID pada alat <i>rotary pendulum</i>	30
Gambar 4-1 Grafik pembacaan pulsa encoder motor DC.	31
Gambar 4-2 Grafik pembacaan pulsa encoder pendulum ayun.....	32
Gambar 4-3 Respon posisi motor DC.	32

Gambar 4-4 grafik pergerakan pendulum ayun.....	33
Gambar 4-5 Rancangan pendulum ayun (Sutejo, 2020).....	36
Gambar 4-6 Skema simulasi fungsi transfer pada Xcos.....	37
Gambar 4-7 Grafik motor DC fungsi transfer.	38
Gambar 4-8 Grafik pendulum fungsi transfer.	38
Gambar 4-9 Konfigurasi uji coba tanpa kendali PID	39
Gambar 4-10 Respon motor DC tanpa kendali PID	40
Gambar 4-11 Respon pendulum ayun tanpa kendali PID	41
Gambar 4-12 Respon objek kendali ketika diberikan nilai K_P 0,3.	42
Gambar 4-13 Respon objek kendali ketika diberikan nilai K_P 0,3; K_I 0,46; K_D 0,02.	42
Gambar 4-14 respons pendulum ayun ketika diberikan nilai K_P 0,05.....	44
Gambar 4-15 Xcos metode PID+PD Cohen-Coon alat <i>rotary pendulum</i>	45
Gambar 4-16 Grafik respons motor DC kendali PID+PD Cohen-Coon.	45
Gambar 4-17 Grafik respons pendulum ayun kendali PID+PD Cohen-Coon.	46
Gambar 4-18 Xcos kendali PID+PD <i>manual tuning</i> alat <i>rotary pendulum</i>	47
Gambar 4-19 Respon motor DC dengan kendali PID+PD.....	47
Gambar 4-20 Respon pendulum ayun dengan kendali PID+PD.....	48
Gambar 4-21 perbandingan respons motor DC.....	49
Gambar 4-22 perbandingan respons pendulum ayun.	49

DAFTAR NOTASI

PID	: <i>Proportional, integral, dan derivative</i>	
PD	: <i>Proportional, dan derivative</i>	
K_P	: Konstanta proporsional pada motor DC	
K_I	: Konstanta integral pada motor DC	
K_D	: Konstanta derivatif pada motor DC	
K_{Ps}	: Konstanta proporsional pada pendulum ayun	
K_{Ds}	: Konstanta derivatif pada pendulum ayun	
$P(s)$: Fungsi transfer motor DC	
J_m	: Momen inersia beban	$(kg \cdot m^2)$
B	: Koefisien gesek	$(Nm/[rad/s])$
k_{tm}	: Konstanta torsi motor	(Nm/A)
I_m	: Arus motor	(A)
ω	: Kecepatan motor	(rad/s)
ω_n	: Frekuensi natural	
ζ	: Konstanta <i>damping</i>	

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Secara umum, sistem kendali merupakan suatu usaha untuk mengolah masukan yang ditentukan untuk mendapatkan keluaran yang diinginkan. Penggunaan sistem kendali ini sudah banyak diterapkan di segala bidang baik di industri manufaktur ataupun dibidang lain. Pengembangan sistem kendali sejalan dengan kebutuhan teknologi saat ini yang mengutamakan otomatisasi dan kendali kontrol untuk mempermudah manusia.

Salah satu penggunaan sistem kendali dapat diterapkan pada alat *crane*. *Crane* merupakan sebuah derek jangkung yang dioperasikan oleh operator untuk mengangkut material konstruksi ataupun material lain dari satu tempat ke tempat yang lain. Penggunaan derek jangkung ini sangat bermanfaat karena meminimalisir tenaga manusia untuk mengangkut barang-barang yang berat.

Pengangkutan barang-barang yang berat memiliki resiko yang muncul ketika *crane* dioperasikan. Perpindahan barang yang terjadi beresiko menimbulkan ayunan yang dapat membahayakan operator ataupun orang-orang disekitar area operasi. Ayunan yang terjadi akan semakin berbahaya ketika ayunan tersebut mengakibatkan terjatuhnya material yang sedang diangkut.

Untuk meminimalisir hal tersebut, dilakukan pengendalian sistem *anti sway* pada alat *crane*. Sistem *anti sway* ini akan meminimalisir terjadinya ayunan ketika alat beroperasi. Sudah terdapat banyak penelitian yang membahas penerapan *anti sway* seperti penelitian oleh Solihin, Legowo, & Wahyudi (2010) yang menggunakan kendali PID dan Fuzzy untuk mengendalikan ayunan pada *gantry crane*. Penelitian oleh Hai, et al. (2017) juga menggunakan metode pengendalian Fuzzy dan PID untuk mengendalikan ayunan pada *overhead crane*. Sistem Kendali Anti Sway Overhead Crane (Akbar, Rahmdhani, & Wibowo, 2016) menggunakan metode sistem kendali PD untuk mengendalikan *anti sway* pada prototipe *overhead crane*. Penelitian-penelitian di atas merupakan penelitian yang membahas kendali sistem bagi objek kendali yang memiliki pergerakan

linier. Lalu bagaimana dengan pengendalian ayunan yang terjadi ketika objek kendali bergerak secara rotasi seperti yang terjadi pada *tower crane*?

Untuk melakukan penelitian terkait kendali ayunan akibat gerakan rotasi, digunakan prototipe *rotary pendulum*. *Rotary pendulum* ini sudah lebih dulu dirancang pada penelitian sebelumnya (Sutejo, 2020). Referensi yang digunakan pada penelitian sebelumnya mengacu pada perancangan model objek kendali *rotary inverted pendulum*, seperti pada perancangan kendali *rotary inverted pendulum* menggunakan kendali PID (Nusantoro, Muslim, Purwanto, & C, 2012). Hasil dari penelitian Sutejo (2020) merupakan perancangan *prototype* dari alat *rotary inverted pendulum*. Dimana *prototype* yang dirancang sudah dapat terintegrasi dengan perangkat lunak Scilab Xcos dan mampu menunjukkan grafik respon pergerakan objek kendali. Tetapi, Penelitian Sutejo (2020) ini belum memiliki pengendalian untuk mengendalikan respon pergerakan alatnya.

Penelitian kali ini juga menggunakan perangkat lunak Scilab Xcos dan Arduino IDE. Sci-lab X-cos berfungsi untuk melakukan simulasi objek kendali *rotary pendulum* dan merancang model sistematis penggunaan sistem kendali. Metode sistem kendali yang digunakan merupakan metode kendali PID (*Proportional, Integral, Derivative*). Penggunaan metode kendali PID bertujuan untuk meminimalisir ayunan yang terjadi pada pendulum akibat dari pergerakan rotasi yang disebabkan oleh motor DC. Sehingga, tujuan dari penelitian ini adalah menganalisis objek kendali dengan merancang model sistematis untuk sistem *anti sway* menggunakan PID dan menerapkan sistem *anti sway* tersebut pada objek kendali *rotary pendulum*.

1.2 Rumusan Masalah

Berikut adalah rumusan masalah penelitian terkait Perancangan dan Implementasi Sistem Kendali untuk Rotary Pendulum dengan Kontroller PID menggunakan Scilab Arduino:

1. Bagaimana cara mengidentifikasi karakteristik objek kendali dan menyusun model matematis objek kendali *rotary pendulum*?
2. Bagaimana cara mensimulasikan model matematis objek kendali *rotary pendulum*?

3. Bagaimana cara merancang model kendali sistem *anti sway* dengan metode PID dan mengimplementasikan sistem tersebut pada objek kendali *rotary pendulum*?
4. Bagaimana hasil dari penerapan sistem *anti sway* dengan metode sistem kendali PID pada objek kendali *rotary pendulum*?

1.3 Batasan Masalah

Berikut adalah batasan masalah untuk penelitian Perancangan dan Implementasi Sistem Kendali untuk Rotary Pendulum dengan Kontroller PID menggunakan Scilab Arduino:

1. Alat uji yang digunakan adalah alat *rotary pendulum* yang telah dirancang di penelitian sebelumnya (Sutejo, 2020).
2. Metode kendali yang digunakan adalah PID. Tidak membahas penggunaan kendali yang lain seperti Fuzzy ataupun NCTF.
3. Penelitian yang dilakukan sampai dengan implementasi sistem kendali PID pada alat *rotary pendulum*.
4. Perangkat lunak yang digunakan adalah Sci-lab versi 5.5.2 dan Arduino IDE.
5. Pemrograman Arduino menggunakan arduino toolbox v3. Program toolbox v3 ini menghubungkan *board* Arduino dengan Scilab versi 5.5.2 sehingga tidak perlu melakukan pemrograman ulang di Arduino.
6. Arduino Mega 2560 digunakan sebagai mikrokontroler.

1.4 Tujuan Penelitian atau Perancangan

Tujuan yang ingin dicapai dari penelitian ini adalah:

1. Mengidentifikasi karakteristik objek kendali dan menyusun model matematis objek kendali *rotary pendulum*.
2. Mensimulasikan model matematis objek kendali *rotary pendulum*.
3. Merancang model kendali sistem *anti sway* dengan kendali PID dan mampu diimplementasikan pada objek kendali *rotary pendulum*.
4. Mampu menganalisa penerapan sistem *anti sway* dengan kendali PID pada objek kendali *rotary pendulum*.

1.5 Manfaat Penelitian atau Perancangan

Manfaat dari penelitian ini adalah dapat membandingkan kinerja objek kendali yang menggunakan *anti sway* sistem kendali PID dengan kinerja objek kendali yang tidak menggunakan sistem *anti sway*. Peneliti juga mampu menerapkan pembelajaran yang telah dipelajari pada masa kuliah. Penelitian ini juga dapat digunakan sebagai referensi pembelajaran penggunaan perangkat lunak Scilab Xcos untuk simulasi sistem *anti sway* yang menggunakan PID sebagai sistem kendalinya. Bagi dunia industri, penelitian ini dapat menunjukkan efektifitas penggunaan kendali PID untuk sebuah sistem kendali, seperti pada *crane*, pengendalian *balancing system*, dan sistem-sistem kendali lainnya.

1.6 Sistematika Penulisan

Sistematika penulisan pada penelitian ini dibagi menjadi lima bab, diantaranya:

BAB I PENDAHULUAN

Bab ini terdiri dari latar belakang penelitian, rumusan masalah yang dihadapi, batasan masalah, tujuan serta manfaat dari pelaksanaan penelitian, dan sistematika penulisan laporan.

BAB II TINJAUAN PUSTAKA

Bab ini terdiri dari landasan teori untuk melakukan penelitian terkait seperti rumus-rumus yang akan digunakan dan juga referensi-referensi dari penelitian sebelumnya.

BAB III METODE PENELITIAN

Bab ini membahas tentang alur penelitian yang akan dikerjakan, perangkat keras yang digunakan, serta pengaturan perangkat lunak seperti pengaturan pada aplikasi arduino dan Scilab Xcos.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menunjukkan dan membahas hasil dari penelitian yang telah dilakukan.

BAB V PENUTUP

Bab ini berisi tentang kesimpulan dari penelitian serta kritik dan saran untuk penelitian selanjutnya.

BAB 2

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Penelitian tentang penerapan sistem *anti sway* sudah banyak dilakukan dan dikembangkan oleh para peneliti sebelumnya. Namun, kebanyakan referensi yang ada membahas penerapan *anti sway* pada sistem derek akibat pergerakan linear. Sedangkan, untuk penelitian sistem *anti sway* akibat pergerakan rotasi seperti pada alat *rotary pendulum* masih jarang dilakukan.

Sebelum memulai penelitian lebih lanjut, diperlukan kajian pustaka yang dapat menjadi referensi penelitian. Kajian pustaka pada penelitian ini terfokus pada beberapa topik bahasan, diantaranya adalah:

1. Menerapkan sistem peredam ayunan atau *anti sway* untuk mengendalikan ayunan pendulum.
2. Penggunaan suatu sistem kendali (seperti PID, Fuzzy, NCTF, dsb) untuk mendapatkan sistem peredam ayunan yang diinginkan.
3. Memiliki fungsi transfer yang menggunakan model matematis pendulum dan/atau motor DC.

Berdasarkan topik bahasan di atas, didapatkan beberapa referensi yang membahas penerapan sistem *anti sway*. Seperti pada jurnal yang membahas tentang penerapan kontrol *anti-swing* pada alat *gantry crane* otomatis (Solihin, Legowo, & Wahyudi, 2010). Penelitian ini menggunakan metode sistem kendali PID+PD yang digabungkan dengan kendali Fuzzy untuk mengendalikan *gantry crane* otomatis. Kendali PID digunakan untuk mengendalikan motor penggerak, sedangkan kendali PD digunakan untuk mengendalikan ayunan pendulum. Skema sistem kendali PID+PD dari jurnal tersebut menyerupai skema objek kendali pada penelitian kali ini. Sehingga skema sistem kendali dari jurnal tersebut dapat digunakan sebagai acuan skema sistem kendali pada penelitian kali ini.

Selain dari jurnal tersebut, terdapat penelitian yang membahas pengendalian ayunan pada *simulator crane* (Triyogi, 2017). Penelitian ini mengendalikan ayunan akibat motor DC yang menggerakkan pendulum secara

linier. Pengendalian ayunan menggunakan kendali PID menggunakan Matlab. Perbedaan penelitian ini dan penelitian yang akan dilakukan adalah pada perangkat lunak yang digunakan. Penelitian ini menggunakan Matlab yang memang lebih sering digunakan oleh pelaku industri, namun perangkat ini berbayar. Sedangkan penelitian yang akan dilakukan menggunakan perangkat lunak Scilab. Scilab memiliki fungsi yang kurang lebih sama dengan Matlab, namun Scilab merupakan perangkat lunak yang gratis. Perangkat lunak Scilab lebih cocok digunakan untuk penelitian-penelitian di tingkat akademis karena perangkat lunak ini bersifat *open source*.

Penelitian sebelumnya membahas perancangan objek kendali *rotary inverted pendulum* dan pengimplementasiannya secara langsung (Sutejo, 2020). Penelitian ini membahas bagaimana merancang objek kendali *rotary inverted pendulum* dengan komponen yang mudah didapat dan dengan karakteristik yang sudah ditentukan. Pembahasan penelitian ini hingga objek kendali *rotary inverted pendulum* selesai dibuat, dan dapat beroperasi dengan baik.

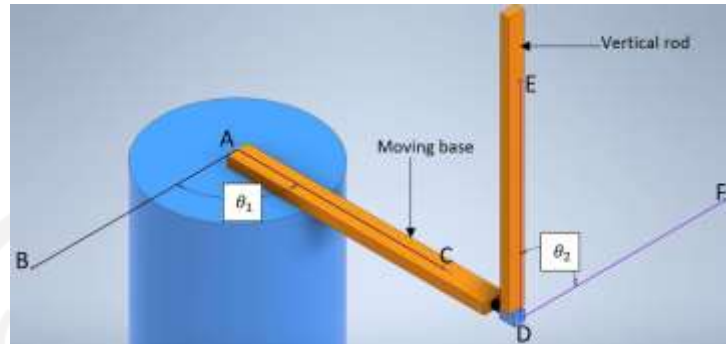
Dua dari tiga referensi di atas memiliki pokok bahasan yang sama yaitu membahas tentang penerapan kendali peredam ayunan meskipun dengan metode kendali yang berbeda-beda. Objek kendali yang digunakan pada kedua referensi di atas juga merupakan objek yang memiliki pergerakan linier. Perbedaan kedua referensi tersebut dengan penelitian yang akan dilakukan adalah ayunan pendulum yang terjadi berasal dari pergerakan rotasi akibat dari perputaran lengan pendulum oleh motor DC.

2.2 Dasar Teori

2.2.1 Rotary Pendulum

Rotary pendulum merupakan sebuah alat yang memiliki komponen lengan pendulum berputar yang terhubung dengan motor DC dan pendulum lain yang diletakkan pada ujung dari lengan pendulum (Muntari & Nurhadi, 2013). penggunaan *rotary pendulum* sebagai alat yang akan dikendalikan pada penelitian kali ini. Pengendalian diberikan pada mekanisme gerak. Pada *rotary pendulum*, terdapat dua komponen yang bergerak yaitu, batang lengan pendulum yang

terhubung dengan motor DC dan batang pendulum vertikal yang tersambung dengan batang lengan pendulum. Komponen dan mekanisme gerakan *rotary pendulum* ditunjukkan pada Gambar 2-1.



Gambar 2-1 Mekanisme gerakan *rotary pendulum*

Mekanisme pergerakan dari *rotary pendulum* adalah ketika batang lengan pendulum (*moving base*) digerakkan ke posisi tertentu oleh motor DC, maka batang pendulum vertikal (*vertical rod*) akan ikut bergerak. Ketika lengan pendulum sudah mencapai posisi yang diinginkan dan berhenti, pergerakan pendulum akan berayun dan perlahan-lahan berhenti. Untuk mengurangi ayunan tersebut, diberikan kontroller PID yang akan dibahas pada sub bab selanjutnya.

Alat *rotary pendulum* terdiri dari beberapa komponen penting yang masing-masing mempunyai kegunaan tertentu. Berikut adalah Komponen-komponen yang ada di alat *rotary pendulum*.

2.2.1.1 Arduino

Arduino merupakan sebuah papan mikrokontroler yang bersifat *open-source* yang sudah banyak digunakan oleh berbagai kalangan mulai dari pelajar hingga profesional. Terdapat banyak jenis perangkat keras arduino yang beredar, jenis-jenis ini dapat disesuaikan dengan kebutuhan penggunaannya. Selain dari hardware, arduino juga memiliki perangkat lunak sendiri bernama arduino IDE sebagai aplikasi pemrogramannya. Arduino menggunakan bahasa C++ sebagai bahasa pemrograman. Pemrograman yang dapat dilakukan menggunakan arduino sangat beragam, mulai dari program pembacaan serial data, sampai dengan pemrograman robot yang kompleks.

Untuk alat *rotary pendulum* pada penelitian kali ini arduino yang digunakan merupakan arduino mega 2560. Kelebihan dari arduino mega 2560

adalah pin digital *input/output* (I/O) yang mencapai 54 pin. Ketersediaan pin ini lebih banyak daripada perangkat keras arduino jenis yang lain. **Error! Reference source not found.** dan Gambar 2-2 masing-masing menunjukkan spesifikasi dari Arduino Mega 2560 dan gambar Arduino Mega 2560.

Tabel 2-1 Spesifikasi Arduino Mega 2560

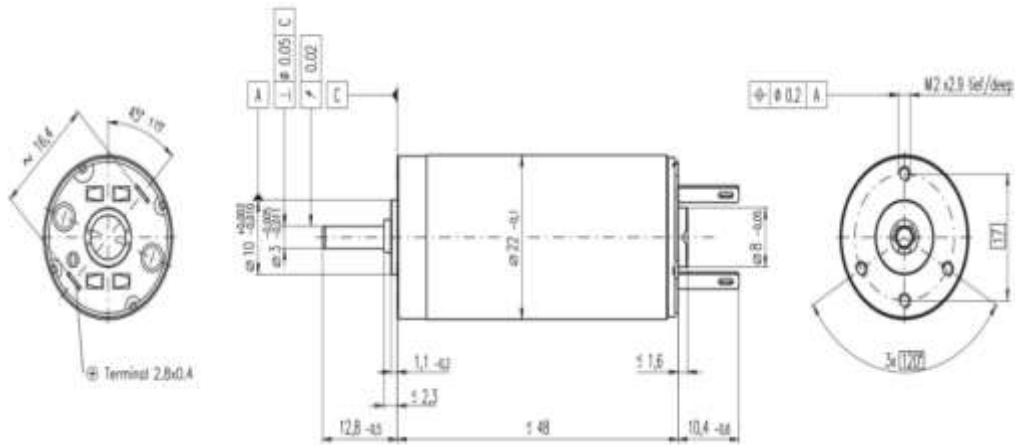
Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz



Gambar 2-2 Arduino Mega 2560 (Datasheet Arduino Mega 2560)

2.2.1.2 Motor DC

Motor DC merupakan sebuah motor listrik yang terdiri dari kerangka magnet (*yoke*), kutub motor (*field pole*), kumparan magnet (*field winding*), kumparan jangkar (*armature*), komutator, dan *brushes* (Ahmed & Alnaib, 2019). Ilustrasi kerangka motor DC ditunjukkan pada Gambar 2-3. Motor DC digunakan sebagai alat penggerak pada alat *rotary pendulum* dimana motor DC terhubung dengan batang lengan pendulum untuk menggerakkan batang tersebut ke sudut tertentu dengan kecepatan yang ditentukan.



Gambar 2-3 Kerangka motor DC (Datasheet Motor DC Maxon S2322).

Pada sistem kontrol, Motor DC memiliki nilai fungsi transfer tersendiri untuk memformulakan karakteristiknya. Berikut adalah fungsi transfer dari motor DC (Toochinda, 2016).

$$P(s) = \frac{1}{s(Jms+B)} \quad (2.1)$$

Dimana $P(s)$, Jm , dan B masing-masing merupakan fungsi transfer motor DC, momen inersia beban ($kg.m^2$), dan koefisien gesek ($Nm/[rad/s]$). Nilai momen inersia beban tergantung dengan beban yang diberikan kepada motor DC. Sedangkan untuk nilai koefisien gesek dapat dicari dengan formula sebagai berikut.

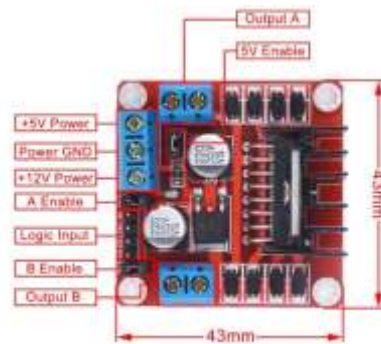
$$B = \frac{K_{tm} \cdot I_m}{\omega} \quad (2.2)$$

Nilai k_{tm} , I_m , dan ω masing-masing adalah konstanta torsi motor (Nm/A), arus motor (A) dan kecepatan motor (rad/s). Ketiga konstanta tersebut didapatkan dari *datasheet* motor DC yang digunakan untuk penelitian kali ini.

2.2.1.3 Driver Motor

Pada penelitian kali ini, digunakan *driver* motor L298N sebagai pengatur kerja motor DC. *Driver* L298N merupakan sebuah *driver* motor dengan *integrated circuit* (IC) L298 *dual* H-Bridge, penggunaan IC ini membuat *driver* motor L298N mampu menggerakkan dan mengarahkan 2 motor DC sekaligus dengan arus 2A. Pada *driver* motor ini tersedia pin-pin dengan kegunaannya masing-masing. Motor DC terhubung dengan pin *output* A atau B, pin +5V dan +12V merupakan pin

untuk *input* voltase, dan pin GND untuk koneksi ke *ground*. Gambar 2-4 menunjukkan skema motor driver L298N.



Gambar 2-4 Motor Driver L298N
(Datasheet L298N Dual H-Bridge Motor Driver).

2.2.1.4 Rotary Encoder

Rotary encoder merupakan sebuah sensor yang dapat mengukur posisi dan kecepatan sudut. Salah satu jenis encoder yang tersedia adalah optikal encoder. Optikal encoder memiliki tingkat akurasi pengukuran mencapai nanometer setelah dilakukan interpolasi (Zheng, Zhang, Wang, Hu, & Zhao, 2015). Optikal encoder pada alat rotary pendulum diperlukan untuk mengukur perubahan sudut batang lengan pendulum dan batang pendulum vertikal. Dari pengukuran kedua batang tersebut, akan didapat data berupa pulsa yang dapat dirubah menjadi data sudut posisi.



Gambar 2-5 Optical Rotary Encoder (Datasheet Encoder Autonics E40s)

2.2.2 Sistem kontrol

Sistem kontrol memiliki sebuah sub-sistem atau proses yang bertujuan untuk mendapatkan keluaran (*output*) yang diinginkan dengan masukan (*input*) yang tersedia atau ditentukan (Nise, 2015). Gambar 2-6 menunjukkan penjelasan sistem kontrol sederhana.

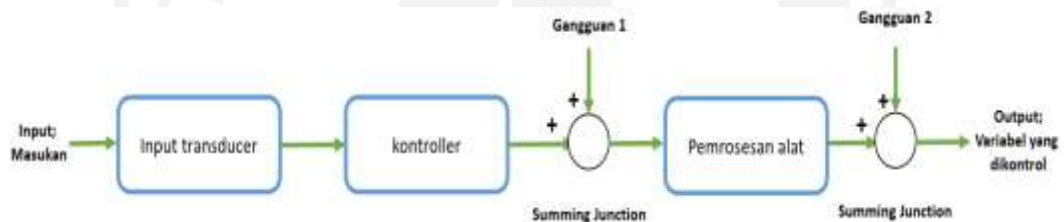


Gambar 2-6 sistem kontrol sederhana.

Diadaptasi dari: Nise (2015)

Beberapa alasan utama penggunaan sistem kontrol yaitu, amplifikasi daya, kendali jarak jauh, kemudahan perubahan bentuk masukan (*input*), dan kompensasi terhadap gangguan-gangguan yang terjadi. Terdapat 2 bentuk utama sistem kontrol, yaitu sistem *open loop* dan sistem *closed loop*.

2.2.2.1 Sistem Open Loop

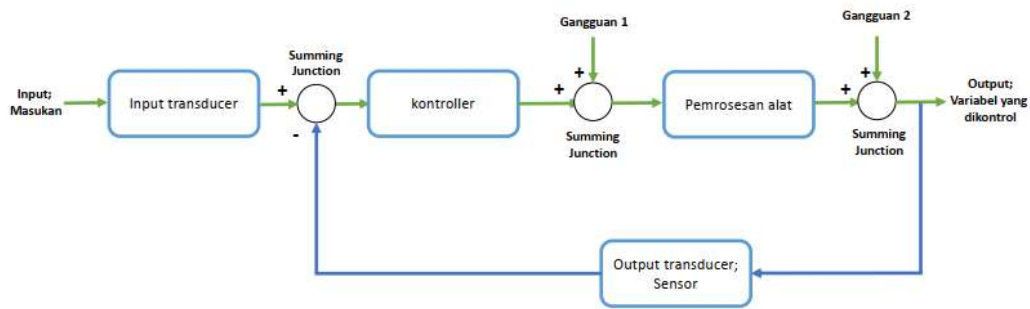


Gambar 2-7 Sistem *open loop*.

Diadaptasi dari: Nise (2015)

Dari Gambar 2-7 dapat dipahami alur kerja dari sistem *open loop*. Ketika *input transducer* menerima sebuah masukan (*input*), masukan tersebut akan diteruskan ke sebuah kontroller. Masukan yang diterima kontroller akan diproses untuk menghasilkan sebuah keluaran (*output*) yang diinginkan. Tentu dalam proses sistem yang bekerja, terdapat gangguan-gangguan yang terjadi. Gangguan-gangguan ini bisa mempengaruhi hasil dari proses sebuah sistem menjadi kurang efektif. Kekurangan dari sistem *open loop* adalah sistem ini tidak dapat mendeteksi gangguan-gangguan yang terjadi ketika proses kerja sistem sedang berlangsung (Nise, 2015).

2.2.2.2 Sistem closed loop

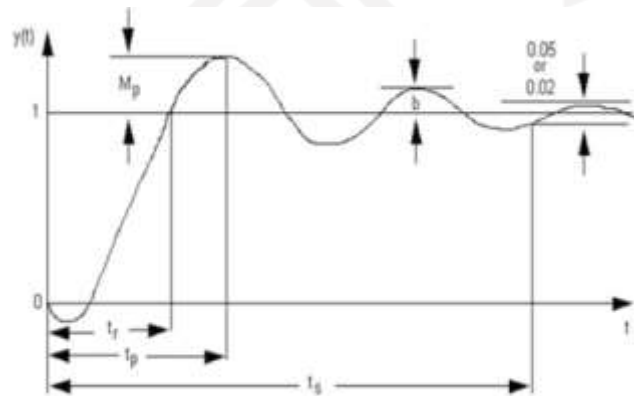


Gambar 2-8 Sistem *closed loop*.

Diadaptasi dari: Nise (2015)

Berbeda dengan sistem *open loop*, sistem *closed loop* mampu mendeteksi gangguan yang terjadi ketika sistem beroperasi. Dari Gambar 2-8 terlihat perbedaan dengan skema *open loop*. Sistem *closed loop* memiliki *sensor* yang mengembalikan data hasil keluaran (*output*). *Sensor* ini mengembalikan data keluaran ke bagian *swimming junction* untuk diteruskan ke *kontroller*. *Kontroller* lalu menyesuaikan *error* yang terdeteksi dari data pembacaan *sensor* untuk menghasilkan keluaran yang lebih efektif (Nise, 2015).

Pengembalian data hasil keluaran ke bagian *swimming junction* pada sistem *closed loop* ini juga dikenal dengan istilah *feedback*. Penerapan *feedback* pada sistem *closed loop* ini berfungsi untuk melacak performa sistem dan juga mendeteksi *error* sistem (Toochinda, 2016). Gambar 2-9 menunjukkan contoh dari grafik sistem *closed loop*.



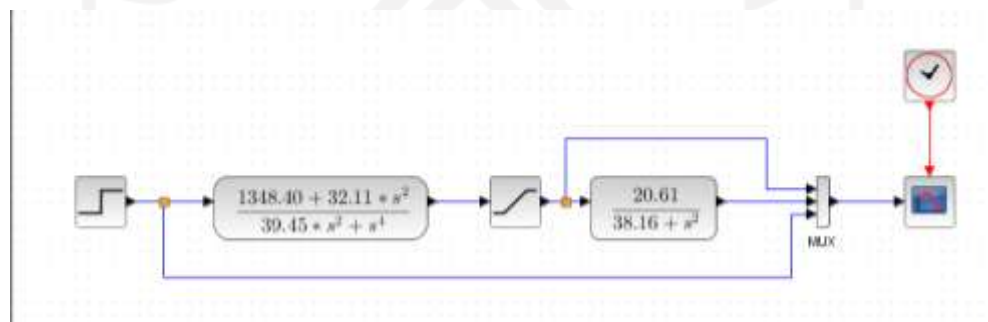
Gambar 2-9 grafik sistem *closed loop* (Toochinda, 2016)

Berdasarkan gambar di atas, dapat terlihat karakteristik dari grafik kendali *closed loop*. karakteristik yang dapat diidentifikasi diantaranya adalah waktu sistem

mengalami kenaikan respon yang disebut dengan *rise time* (t_r), waktu sistem berada di puncak atau *time peak* (t_p), waktu sistem sudah berada di kondisi *steady state* atau *settling time* (t_s), dan persen *overshoot* sistem kendali (M_p). Dari beberapa karakteristik tersebut, dapat ditentukan respon sistem yang diinginkan. Untuk mencapai respon sistem yang diinginkan dapat dilakukan dengan mengatur kendali menggunakan controller yang akan dibahas pada sub-bab selanjutnya.

2.2.3 Scilab Xcos

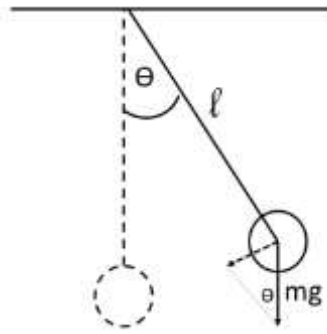
Scilab merupakan sebuah perangkat lunak yang dapat digunakan untuk melakukan pemodelan matematis seperti kalkulasi integral, diferensial, aljabar, statistik dan lain sebagainya (Handayani, Pramudya, Suparwoto, & Muchlas, 2018). Scilab juga merupakan perangkat lunak yang bersifat *open source* sehingga dapat digunakan oleh berbagai kalangan. Terdapat berbagai macam fitur yang tersedia di scilab, salah satu fitur tersebut adalah Xcos yang digunakan untuk melakukan simulasi pada penelitian kali ini. Xcos merupakan bagian dari scilab GUI (*Graphic User Interface*) yang digunakan sebagai pemodelan simulasi suatu sistem (Kurniawan, Simbolon, & MN, 2020). Sistem kontrol dapat disimulasikan dengan Xcos, hal ini karena Xcos memiliki model-model blok matematis yang memiliki fungsinya masing-masing. Blok matematis ini juga dapat saling dihubungkan antara satu dengan yang lainnya untuk membuat diagram sistem kendali. Gambar 2-10 menunjukkan aplikasi Xcos pada perangkat lunak Scilab.



Gambar 2-10 Contoh penggunaan aplikasi Xcos pada Scilab.

2.2.4 Model Matematis

Model matematis digunakan untuk memodelkan pergerakan pendulum sederhana. Gambar 2-11 menunjukkan pergerakan pendulum sederhana.



Gambar 2-11 Ilustrasi pergerakan pendulum sederhana (Akbar, Rahmdhani, & Wibowo, 2016).

Pemodelan matematis osilasi pendulum sederhana dapat diformulakan menggunakan hukum Newton II yang berbentuk:

$$\sum F = m a \quad (2.3)$$

Dimana F merupakan resultan gaya (N), m merupakan massa benda (kg), dan a merupakan percepatan benda (m/s^2). Pendulum memiliki pergerakan rotasi, sehingga hukum Newton II berubah menjadi:

$$\tau = J \alpha \quad (2.4)$$

Dimana τ adalah torsi yang bekerja pada pendulum, J merupakan momen inersia pendulum ($kg \cdot m^2$), dan $\alpha = \ddot{\theta}$ adalah percepatan sudut pendulum (m/s^2). Dari formula di atas, maka didapatkan:

$$-(m g \sin\theta)l = J \ddot{\theta} = m l \ddot{\theta} \quad (2.5)$$

$$-g \sin\theta = l \ddot{\theta} \quad (2.6)$$

$$\ddot{\theta} + \frac{g}{l} \sin\theta = 0 \quad (2.7)$$

Dimana θ merupakan posisi sudut, g merupakan gravitasi (m/s^2), dan l merupakan panjang dari pendulum (m). Persamaan (2.7) memiliki kesamaan dengan persamaan osilasi harmonik sederhana yang berbentuk:

$$\ddot{x} + \frac{k}{m} x = 0 \quad (2.8)$$

Dengan asumsi nilai θ kecil, maka nilai $\sin \theta \approx \theta$. Sehingga persamaan (2.7) menjadi:

$$-\theta \frac{g}{l} = \ddot{\theta} \quad (2.9)$$

Berdasarkan persamaan (2.9) maka nilai frekuensi natural ω_n dapat diformulakan menjadi:

$$\omega_n = \frac{1}{2\pi} \sqrt{\frac{g}{l}} \quad (2.10)$$

Nilai frekuensi natural digunakan untuk mendapatkan fungsi transfer bagi pendulum ayun pada penelitian kali ini. Fungsi transfer yang digunakan merupakan model matematis untuk sistem kontrol orde 2 yang berbentuk:

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.11)$$

Nilai ζ merupakan konstanta *damping*. Apabila nilai ζ kurang dari 0 maka fungsi transfer sistem ordo 2 akan mengalami osilasi secara terus menerus (*undamped*). Sedangkan apabila nilai ζ ada diantara 0 dan 1, maka keluaran sistem orde 2 akan berosilasi dengan teredam hingga mencapai keadaan *steady state* (*under damped*), dan apabila nilai ζ sama dengan 1, maka keluaran sistem orde 2 akan langsung menuju ke keadaan *steady state* tanpa mengalami osilasi (*critically damped*).



Gambar 2-12 Respon sistem menggunakan *damping*.

Nilai ζ dapat dicari dengan menggunakan teori *viscous damping*, dimana nilai ζ dapat dicari menggunakan rumus sebagai berikut:

$$\zeta = \frac{\delta}{\sqrt{(2\pi)^2 + \delta^2}} \quad (2.12)$$

Dimana;

$$n\delta = \ln \frac{X_i}{X_{i+n}} \quad (2.13)$$

2.2.5 PID

PID (*Proportional, Integral, Derivative*) merupakan salah satu controller yang digunakan untuk mengatur respon sistem objek kendali. Penggunaan PID sudah banyak diaplikasikan di berbagai alat yang memerlukan kontrol sistem. Persamaan dari kontrol PID dituliskan sebagai berikut.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (2.14)$$

Variabel u dan e masing-masing merupakan variabel keluaran yang diinginkan, dan masukan yang diberikan ke sistem objek kendali. Selain dari kedua variabel tersebut, terdapat 3 variabel kontrol pada persamaan PID. Ketiga variabel kontrol tersebut adalah nilai K_p sebagai nilai konstanta *error* proporsional, variabel K_i sebagai nilai konstanta *error* integral, dan variabel K_d sebagai nilai konstanta *error* derivatif (Toochinda, 2016). Konstanta K_p , K_i , K_d masing-masing didefinisikan sebagai berikut.

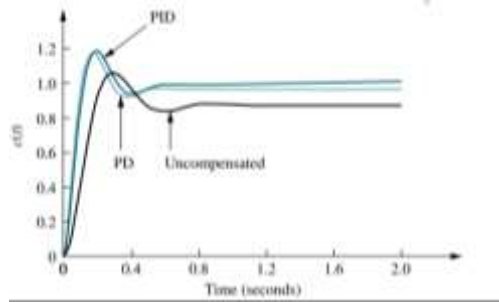
$$K_p = K ; K_i = \frac{K}{T_i} ; K_d = KT_d \quad (2.15)$$

Masing-masing konstanta PID memiliki pengaruh untuk mengendalikan respon sistem. kendali PID dapat meningkatkan atau mengurangi karakteristik *overshoot*, *settling time*, *rise time*, dan *steady state error*. Berikut tabel penggunaan masing-masing konstanta PID.

Tabel 2-2 pengaruh masing-masing konstanta PID (Toochinda, 2016).

Respon closed loop	Rise time	Overshoot	Settling time	SS error
K_p	Berkurang	Bertambah	Berubah sedikit	Berkurang
K_i	Berkurang	Bertambah	Bertambah	Lenyap
K_d	Berubah sedikit	Berkurang	Berkurang	Berubah sedikit

Tabel di atas menunjukkan pengaruh umum penggunaan masing-masing konstanta terhadap sistem *closed loop*. Berdasarkan karakteristik respon sistem masing-masing konstanta tersebut, nilai konstanta PID dapat dicari untuk mendapatkan respon sistem yang diinginkan. Gambar 2-13 menunjukkan contoh respon sistem menggunakan kendali PID.



Gambar 2-13 Respon sistem menggunakan kendali PID (Nise, 2015).

2.2.5.1 Metode ZNFD

Terdapat banyak metode untuk melakukan penyetelan nilai konstanta PID, salah satu metode umum yang digunakan adalah metode *Ziegler-Nichols Frequency Domain (ZNFD)*. Penyetelan menggunakan metode ZNFD memiliki 2 cara yaitu, penyetelan manual dan penyetelan otomatis. Cara menentukan konstanta PID menggunakan metode manual dapat dilakukan dengan cara sebagai berikut (Toochinda, 2016).

1. Konstanta integral dan derivatif terlebih dahulu dianggap bernilai 0 ($K_i = K_d = 0$).
2. Nilai konstanta K_p di atur hingga sistem kontrol close-loop mulai berosilasi. Osilasi sistem kontrol yang terjadi haruslah bersifat *continue*. Catat nilai konstanta sebagai nilai K_u .
3. Ukur periode osilasi sebagai nilai T_u .
4. Gunakan Tabel 2-3 sebagai perhitungan konstanta PID.

Tabel 2-3 Perhitungan konstanta PID metode ZNFD manual (Toochinda, 2016).

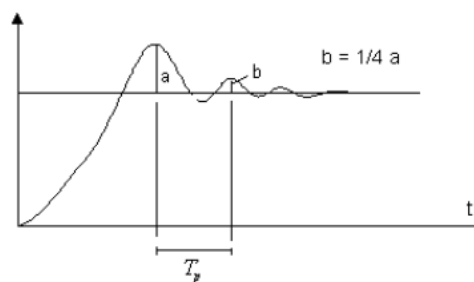
Bentuk kontroller	K_p	K_i	K_d
P	$0,5 K_u$	-	-
PI	$0,4 K_u$	$0,5 \frac{K_u}{T_u}$	-
PID	$0,6 K_u$	$1,2 \frac{K_u}{T_u}$	$0,075 K_u T_u$

Dari tabel di atas, akan didapat nilai konstanta masing-masing PID. Nilai yang didapat menggunakan metode ZNFD manual bukan menjadi nilai mutlak. Perhitungan menggunakan metode ZNFD ini dapat digunakan sebagai dasar nilai

konstanta, selanjutnya nilai konstanta PID dapat disesuaikan untuk mendapatkan hasil respon sistem yang sesuai dengan karakteristik yang diinginkan.

2.2.5.2 Metode Cohen-Coon

Metode Cohen-Coon merupakan perkembangan dari metode tuning PID ZNFD. Metode Cohen-Coon mengatasi model yang tidak memiliki keadaan osilasi secara tetap. Metode ini dapat mengatasi respon dengan kondisi teredam atau memiliki respon *quarter amplitude decay*. Respon tersebut ditunjukkan pada gambar



Gambar 2-14 respon *quarter amplitude decay* (Wicaksono, 2004).

Penyetelan kendali PID menggunakan metode ini dapat dicari dengan langkah sebagai berikut:

1. Kontroller K_P diatur hingga model memiliki respon teredam.
2. Dari respon tersebut, dapat diidentifikasi nilai T_P (waktu periode 1 amplitudo) dan L (waktu awal respon mengalami peningkatan).
3. Dari data yang sudah didapat, dapat dihitung nilai konstanta PID menggunakan tabel berikut.

Tabel 2-4 Perhitungan konstanta PID menggunakan metode Cohen-Coon (Wicaksono, 2004).

Type Kontroller	K_P	K_I	K_D
P	$\frac{1}{K} \left(\frac{T}{L} \right) \left[1 + \frac{1}{3} \left(\frac{L}{T} \right) \right]$	-	-
PI	$\frac{1}{K} \left(\frac{T}{L} \right) \left[0,9 + \frac{1}{12} \left(\frac{L}{T} \right) \right]$	$L \left[\frac{30 + 3 \left(\frac{L}{T} \right)}{9 + 20 \left(\frac{L}{T} \right)} \right]$	-
PD	$\frac{1}{K} \left(\frac{T}{L} \right) \left[\frac{5}{4} + \frac{1}{6} \left(\frac{L}{T} \right) \right]$	-	$L \left[\frac{6 - 2 \left(\frac{L}{T} \right)}{22 + 3 \left(\frac{L}{T} \right)} \right]$

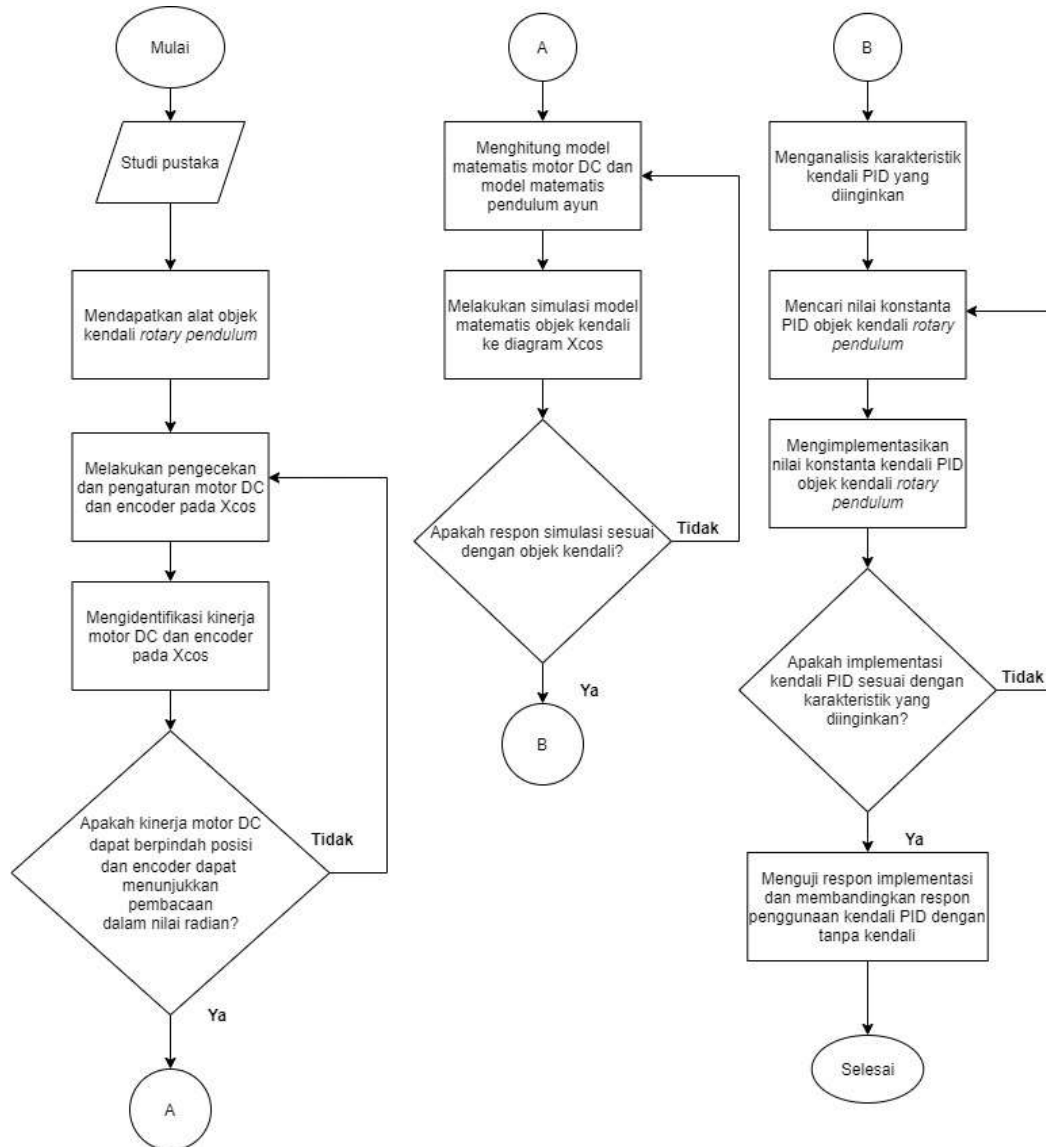
PID	$\frac{1}{K} \left(\frac{T}{L} \right) \left[\frac{5}{4} + \frac{1}{6} \left(\frac{L}{T} \right) \right]$	$L \left[\frac{30 + 3 \left(\frac{L}{T} \right)}{9 + 20 \left(\frac{L}{T} \right)} \right]$	$L \left[\frac{4}{11 + 2 \left(\frac{L}{T} \right)} \right]$
-----	--	--	--

Berdasarkan tabel di atas, akan didapat nilai masing-masing konstanta yang ingin digunakan. Metode ini juga dapat digunakan sebagai dasar tuning dari model objek yang ingin dikendalikan.



BAB 3 METODE PENELITIAN

3.1 Alur Penelitian



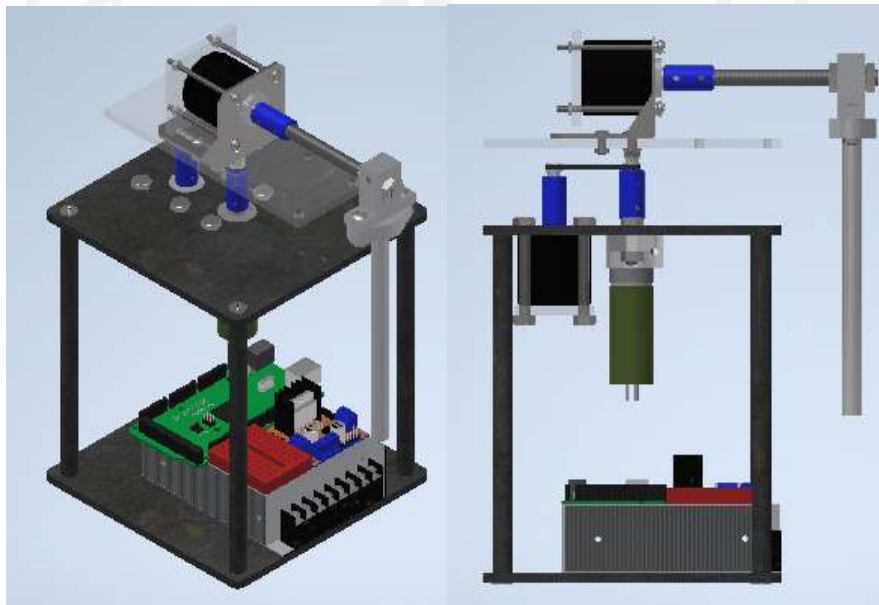
Gambar 3-1 Alur penelitian

3.2 Perancangan

Perancangan pada penelitian kali ini terbagi menjadi dua yaitu perancangan perangkat keras dan perancangan perangkat lunak. Objek kendali *rotary pendulum* yang digunakan pada penelitian kali ini sudah terlebih dahulu dirancang pada penelitian sebelumnya.

3.2.1 Perancangan Perangkat Keras

Perancangan perangkat keras *rotary pendulum* menggunakan referensi Quanser sebagai acuan bentuk objek kendali. Susunan perangkat keras disusun seperti Gambar 3-2.

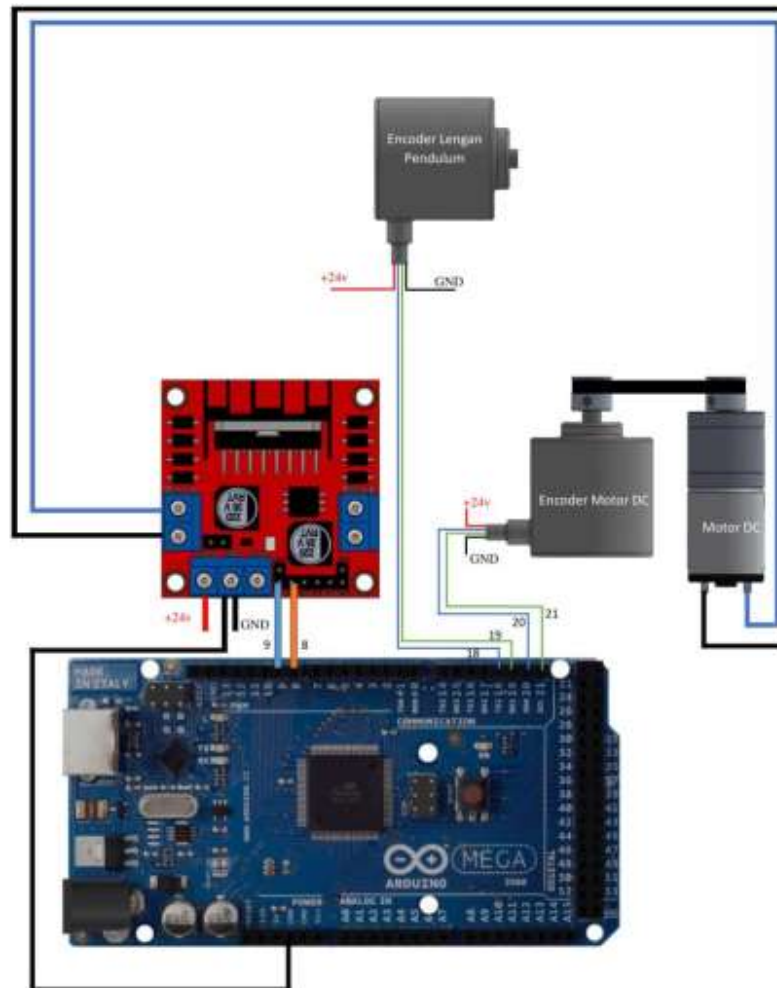


Gambar 3-2 Model rancangan perangkat keras *rotary pendulum*.

Berdasarkan gambar di atas, dapat dilihat perancangan model objek kendali *rotary pendulum* diatur dengan 2 alas yang disusun bertingkat. Alas pertama merupakan tempat perangkat keras elektrik yang terdiri dari *power supply*, mikrokontroler Arduino Mega 2560, *motor driver* L298, dan *breadboard*. Alas kedua tersambung dengan komponen-komponen mekanikal yang terdiri dari motor DC, sensor encoder, lengan pendulum, *coupling*, *bracket*, poros, dan juga pendulum ayun.

Setelah merancang dan menyusun komponen perangkat keras di atas, langkah selanjutnya adalah melakukan koneksi elektrikal untuk alat *rotary*

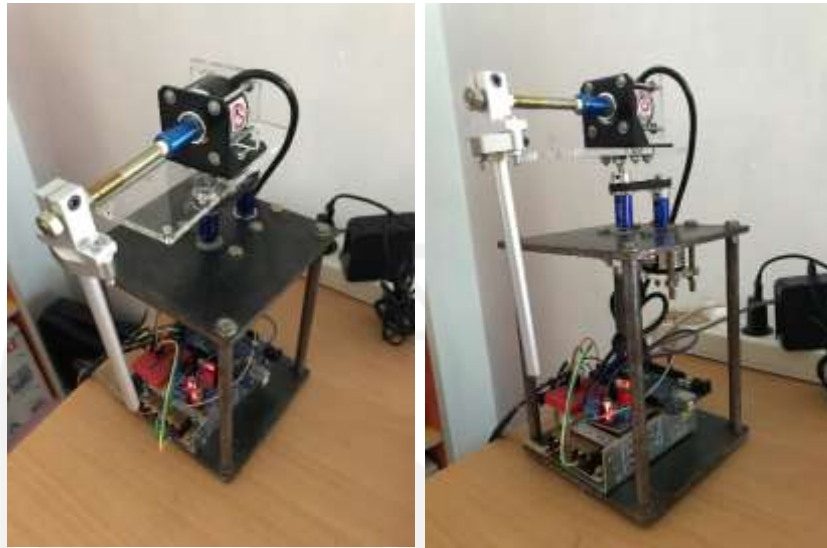
pendulum. Gambar 3-3 menunjukkan skema koneksi elektrik alat *rotary pendulum*.



Gambar 3-3 Koneksi elektrik alat *rotary pendulum*.

Berdasarkan gambar di atas, terlihat bahwa motor DC tidak langsung terhubung ke Arduino Mega 2560, melainkan dihubungkan terlebih dahulu ke motor driver L298n. Hal tersebut bertujuan agar motor driver L298n dapat mengatur arah dan kecepatan motor DC. Motor driver L298n dihubungkan dengan pin digital 8 dan 9 (PWM) pada Arduino Mega dan juga dikoneksikan ke *power supply* dan ground (GND). Selanjutnya mengkoneksikan dua sensor encoder ke Arduino Mega. Salah satu encoder dihubungkan ke motor DC dengan menggunakan *timing belt* dan sensor encoder lain terhubung ke pendulum ayun. Masing-masing encoder terkoneksi ke pin digital 18, 19 dan 20, 21. Koneksi pin sensor encoder sesuai dengan pengaturan pada Xcos yang akan ditunjukkan pada

sub-bab selanjutnya. Gambar 3-4 menunjukkan objek kendali *rotary pendulum* setelah dirancang.



Gambar 3-4 Objek kendali *rotary pendulum*.

3.2.2 Perancangan Perangkat Lunak


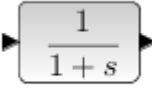
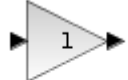


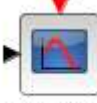
Perangkat lunak yang digunakan pada penelitian ini diantaranya adalah:

1. Arduino IDE yang digunakan untuk mengupload program toolbox V3. Program toolbox V3 merupakan program untuk menghubungkan mikrokontroler Arduino ke perangkat lunak Scilab.
2. Scilab Xcos yang berfungsi untuk memodelkan sistem kontrol dari objek kendali *rotary pendulum*. Scilab memiliki beberapa fitur yang dapat digunakan untuk berbagai macam keperluan, salah satu fiturnya adalah Xcos. Perancangan model sistem kontrol dapat dibuat menggunakan Xcos. Oleh karenanya, pada Xcos terdapat blok-blok diagram yang merepresentasikan fungsi-fungsi yang masing-masing fungsi dapat diatur sesuai dengan model sistem kontrol yang telah dirancang.

Langkah pertama melakukan koneksi objek kendali *rotary pendulum* ke perangkat lunak scilab adalah dengan mengunggah program toolbox arduino V3. Selanjutnya, mengunggah loader toolbox V3 ke dalam perangkat lunak Scilab versi 5.5.2, program loader dan loader telah sukses terunggah ke dalam perangkat lunak Scilab 5.5.2 akan ditunjukkan pada bagian lampiran.

Setelah *loader* telah terunggah, selanjutnya dapat membuka Xcos untuk merancang blok diagram sistem kendali. Berikut adalah blok-blok diagram yang digunakan pada penelitian kali ini.

Tabel 3-1 Blok-blok diagram yang digunakan

Blok diagram	Nama	Fungsi
 STEP_FUNCTION	STEP_FUNCTION	berfungsi untuk memberikan input berupa step. Bisa dalam nilai voltase dan juga radian.
 CLR	CLR	Berfungsi untuk menyatakan nilai fungsi transfer pada Xcos.
 GAINBLK	GAINBLK	Sebagai blok pengkali nilai pada Xcos.
 SUMMATION	SUMMATION	Berfungsi untuk melakukan perbandingan antara nilai input dan output pada sistem kendali.
 MUX	MUX	Berfungsi untuk menyatukan beberapa input menjadi satu output.
 CSCOPE	CSCOPE	Berfungsi untuk menggambarkan output sebagai grafik nilai terhadap waktu.

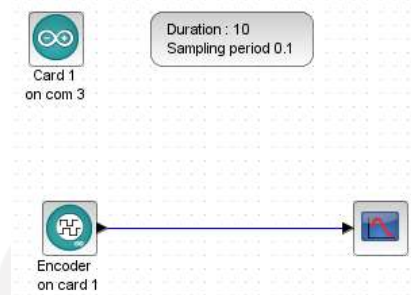
 CLOCK_c	CLOCK_c	Befungsi untuk mengatur periode waktu sistem kendali.
 PID	PID	Berguna untuk memasukkan nilai kendali PID pada sistem.
	TIME_SAMPLE	Befungsi untuk mengatur periode pengambilan sampel pada arduino.
	ARDUINO_SETUP	Melakukan pengaturan pembacaan port arduino yang akan dibaca oleh sistem Xcos.
	ARDUINO_SCOPE	Menunjukkan grafik keluaran dari sistem yang menggunakan arduino
	DCMOTOR_SB	Befungsi untuk mengatur dan menjalankan motor DC yang terhubung dengan arduino.
	ENCODER_SB	Befungsi mengatur dan membaca data dari sensor encoder yang terhubung dengan arduino.

Dari blok-blok diagram di atas, objek sistem kendali *rotary pendulum* dapat dilakukan uji coba respon untuk mengetahui performa dari motor DC dan pembacaan kedua sensor encoder.

3.3 Metode pengujian

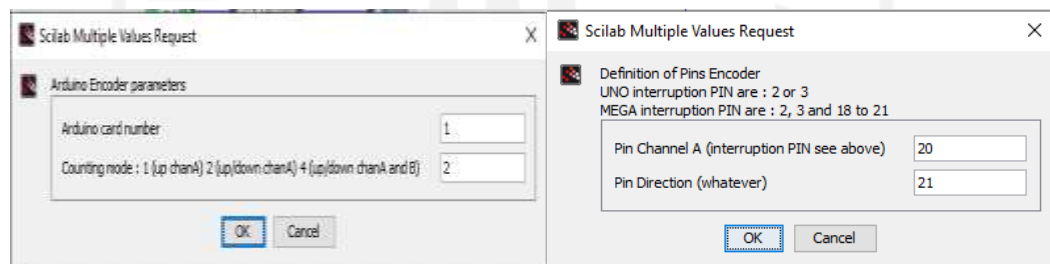
3.3.1 Pemeriksaan sensor encoder

Hal pertama yang dilakukan adalah memeriksa sensor encoder yang terhubung dengan motor DC dan pendulum. Pengecekan sensor encoder dilakukan dengan mencocokkan pulsa bacaan sensor pada spesifikasi dan pulsa bacaan sensor secara langsung. Berikut adalah konfigurasi Xcos untuk pengecekan sensor encoder.



Gambar 3-5 Xcos pengecekan pulsa sensor encoder

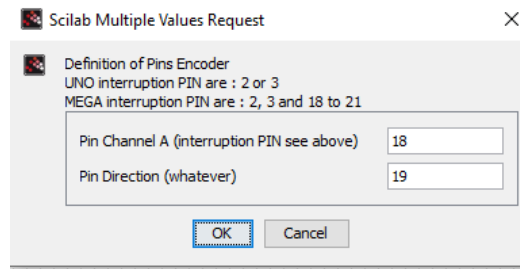
Skema Xcos gambar di atas hanya menghubungkan blok diagram encoder ke blok diagram *arduino scope* untuk menunjukkan grafik. Pada blok diagram encoder, diatur konfigurasi sensor yang terkoneksi pada Arduino. Pada sensor encoder motor DC, sensor terhubung ke pin 20 dan 21. Berikut adalah konfigurasi sensor encoder pada motor DC.



Gambar 3-6 Konfigurasi sensor encoder motor DC

Berdasarkan spesifikasinya, encoder yang terhubung dengan motor DC memiliki spesifikasi pembacaan 400 pulsa setiap 1 kali putaran. Pengecekan ini bermaksud untuk membuktikan pulsa yang terbaca oleh sensor encoder ketika diputar 1 kali.

Pada sensor encoder pendulum ayun, sensor terhubung ke pin 18 dan 19. Berikut adalah konfigurasi sensor encoder pada pendulum ayun.

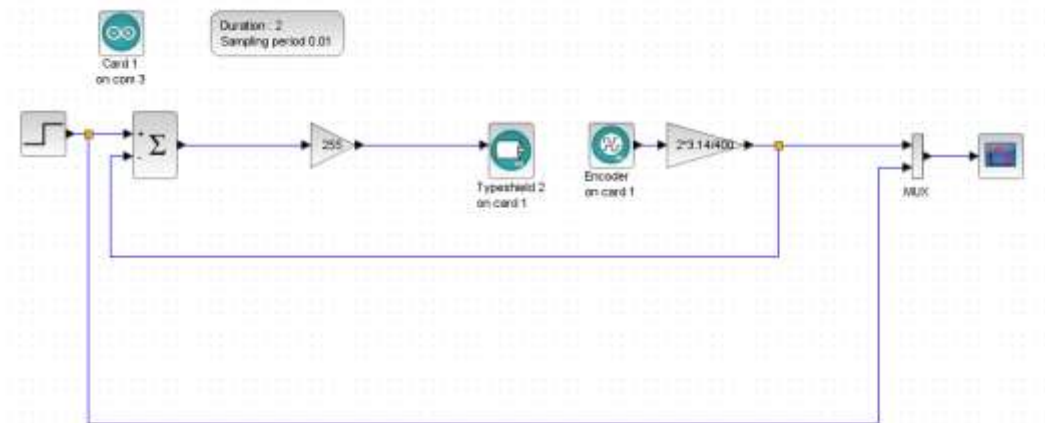


Gambar 3-7 Konfigurasi sensor encoder pendulum ayun

Sensor encoder yang terhubung dengan pendulum ayun memiliki spesifikasi pembacaan 100 pulsa per 1 kali putaran.

3.3.2 Pemeriksaan respon motor DC

Setelah nilai pulsa sensor encoder pada motor DC telah diketahui, selanjutnya dapat melakukan pemeriksaan respon karakteristik motor DC dengan skema Xcos sebagai berikut.

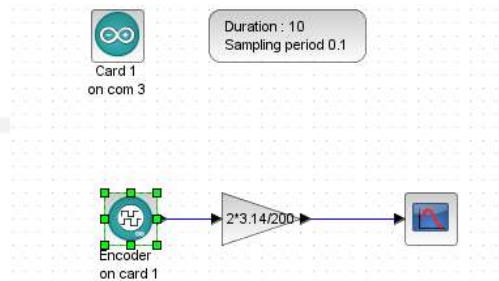


Gambar 3-8 Konfigurasi Xcos untuk respon posisi motor DC

Pada percobaan respon posisi, posisi yang ingin dicapai adalah 90° atau $1,57$ rad. Sehingga pada blok *step input* diberikan nilai $1,57$ rad. Nilai *gain* PWM 255 berfungsi untuk mengatur motor DC agar memiliki akselerasi yang maksimal. Konfigurasi motor DC dan sensor encoder motor DC sama dengan konfigurasi sebelumnya. Hasil pembacaan sensor encoder motor DC selanjutnya dikalikan dengan nilai $2\pi/400$, perhitungan ini bertujuan agar hasil pembacaan sensor encoder motor DC membaca nilai dalam satuan radian.

3.3.3 Pemeriksaan respons pendulum ayun

Setelah didapat nilai pulsa sensor encoder pendulum ayun. Langkah selanjutnya adalah memeriksa respon karakteristik osilasi dari pendulum ayun. Skema Xcos untuk respon osilasi ditunjukkan oleh Gambar 3-9.



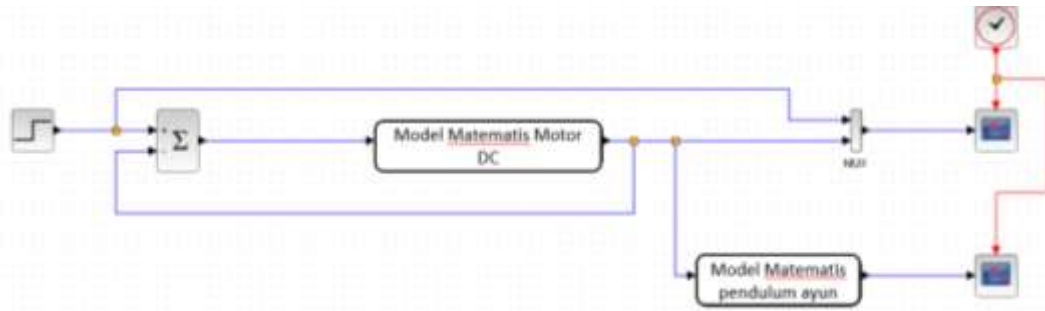
Gambar 3-9 Skema Xcos untuk respon osilasi pendulum ayun.

Konfigurasi blok diagram encoder sama dengan konfigurasi saat melakukan pemeriksaan pulsa encoder pendulum ayun. Saat uji coba, pendulum diayunkan secara manual sehingga pendulum berayun secara bebas. Untuk menunjukkan hasil pembacaan dalam satuan radian, hasil pembacaan sensor encoder pendulum dikalikan dengan nilai $2\pi/200$.

3.3.4 Simulasi model matematis

Ketika karakteristik objek kendali sudah diketahui, selanjutnya adalah melakukan simulasi model matematis. Terdapat 2 model matematis yang perlu dihitung yaitu model matematis untuk motor DC dan model matematis untuk pendulum ayun. Perhitungan model matematis untuk motor DC dapat dilakukan dengan mengumpulkan data dari *datasheet*. Perhitungan model matematis untuk pendulum ayun dapat dilakukan dengan data natural frekuensi pendulum dan data yang ditunjukkan pada pemeriksaan respon osilasi pendulum ayun.

Simulasi model matematis juga dilakukan menggunakan Xcos. Skema simulasi dari model matematis adalah sebagai berikut.



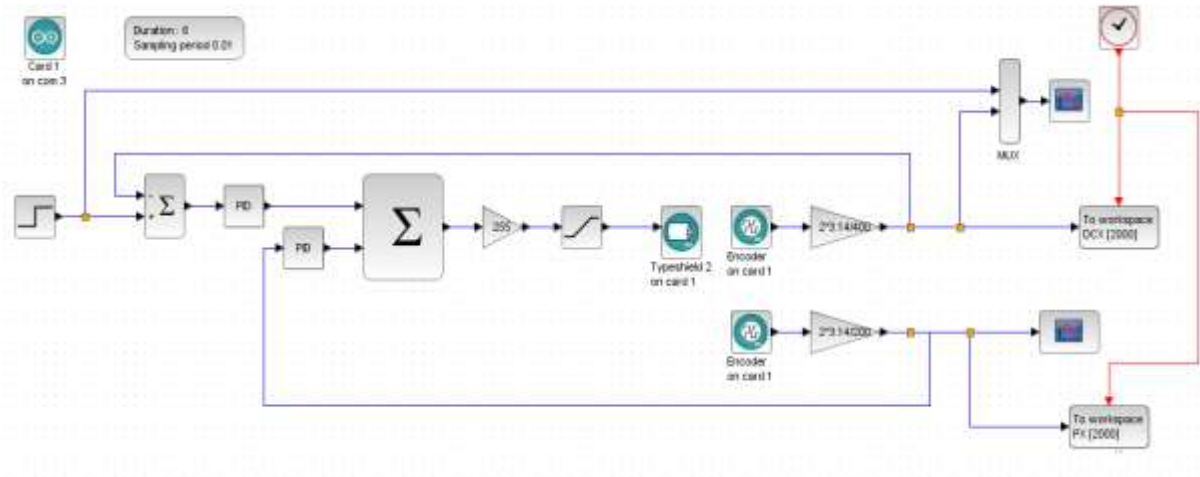
Gambar 3-10 skema Xcos untuk simulasi model matematis objek kendali *rotary pendulum*.

Input step yang diberikan merupakan *input* posisi dalam radian. Posisi yang diinginkan adalah motor DC berpindah 90° yang apabila dirubah ke nilai radian adalah 1,57 rad. *Input step* selanjutnya diteruskan ke blok *error junction* yang berguna untuk membandingkan *error* dari masukan dan keluaran sistem motor DC. Dari *error junction* diteruskan ke model matematis motor DC. Keluaran model matematis motor DC selanjutnya diteruskan ke model matematis pendulum ayun. Perpindahan posisi motor DC secara tiba-tiba akan mengayunkan pendulum. Selanjutnya barulah hasil keluaran model matematis motor DC, model matematis pendulum, dan input ditampilkan pada grafik.

3.3.5 Pengujian kendali PID objek kendali *rotary pendulum*

Terdapat tiga tahapan dalam melakukan uji coba kendali PID pada objek kendali *rotary pendulum*. Yang pertama adalah uji coba alat sebelum diberikan kendali PID, yang kedua adalah menguji alat dengan kendali PID+PD dimana pencarian nilai PD menggunakan metode yang ada, dan yang ketiga adalah uji coba alat dengan kendali PID+PD dimana nilai PD dicari menggunakan metode *manual tuning*.

Uji coba dilakukan menggunakan skema Xcos yang ditunjukkan pada Gambar 3-11. Skema yang digunakan mengikuti alur referensi kendali PID+PD (Solihin, Legowo, & Wahyudi, 2010).



Gambar 3-11 Skema Xcos uji coba kendali PID pada alat *rotary pendulum*.

Skema di atas memiliki nilai step input 1,57 rad. Nilai masukan ke motor DC diberikan PWM sebesar 255 dan dibatasi menggunakan blok saturasi. Penggunaan blok saturasi ini bertujuan agar nilai input PWM Motor DC tidak lebih dari 255. Motor DC dikendalikan menggunakan konfigurasi PID, sedangkan pendulum ayun dikendalikan menggunakan konfigurasi PD.

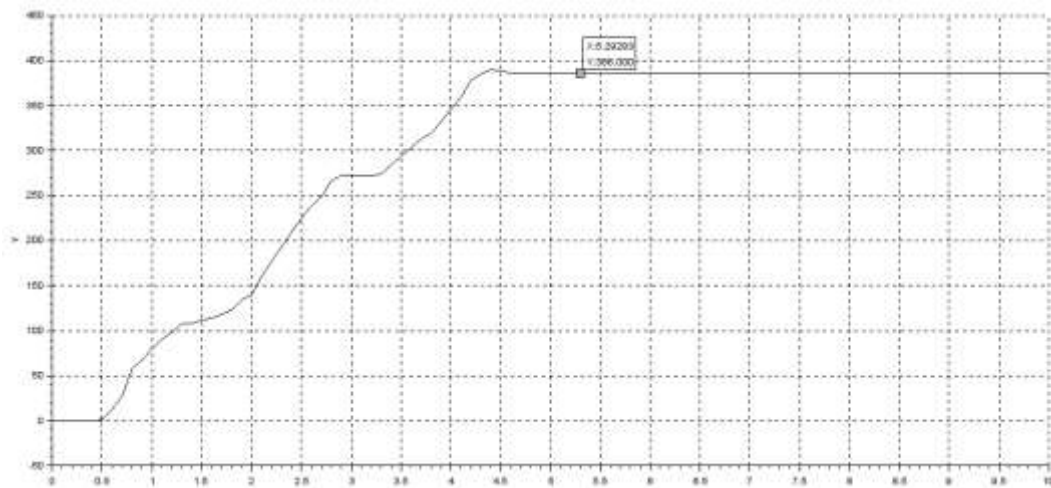
BAB 4

HASIL DAN PEMBAHASAN

4.1 Hasil Pemeriksaan objek kendali *rotary pendulum*

4.1.1 Hasil pemeriksaan sensor encoder

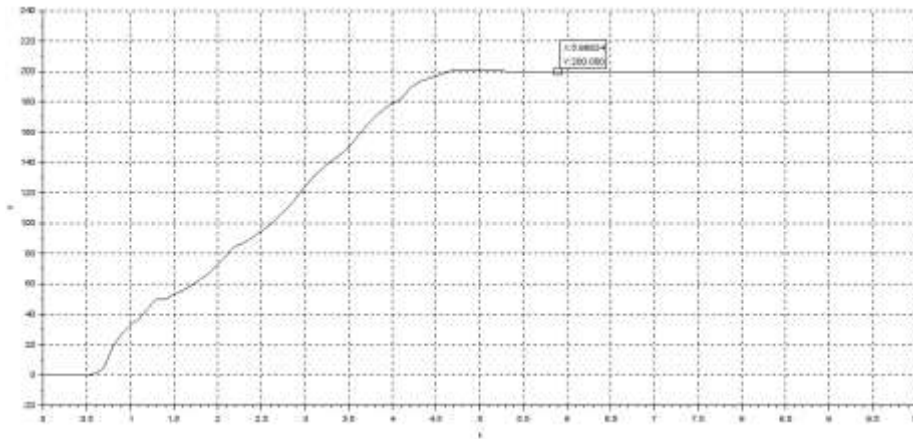
Sensor encoder yang diperiksa pertama kali adalah sensor encoder pada motor DC. Gambar 4-1 menunjukkan grafik pembacaan pulsa encoder motor DC.



Gambar 4-1 Grafik pembacaan pulsa encoder motor DC.

Dari grafik tersebut, terlihat bahwa untuk 1 kali putaran pulsa yang terbaca adalah 386 pulsa. Berdasarkan pembacaan tersebut, dianggap bahwa pulsa bacaan sensor secara langsung sudah sesuai dengan pulsa bacaan sensor spesifikasi. Hal ini karena nilai pembacaan pulsa secara langsung sudah mendekati nilai 400 pulsa per putaran.

Selanjutnya adalah Hasil pemeriksaan sensor encoder pendulum ayun yang ditunjukkan pada Gambar 4-2.

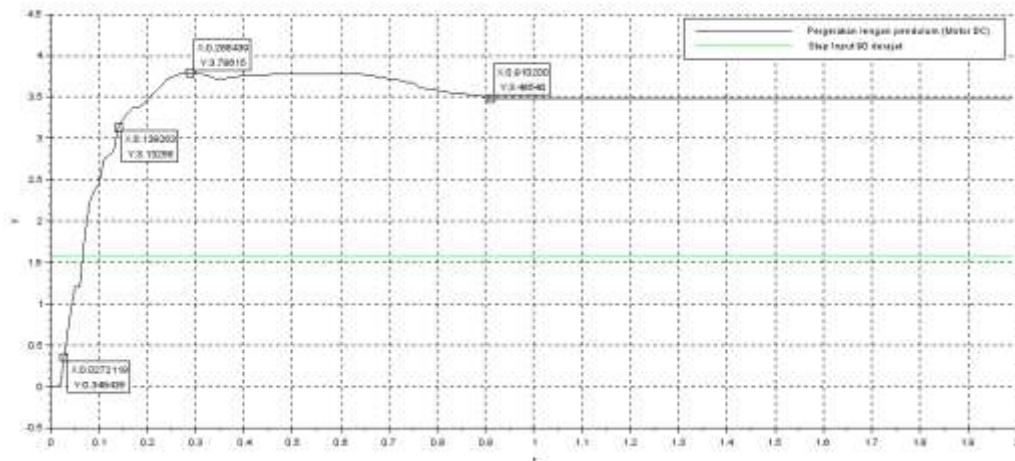


Gambar 4-2 Grafik pembacaan pulsa encoder pendulum ayun

Pembacaan pulsa encoder pendulum ayun eksperimen menunjukkan bahwa dalam 1 kali berputar, sensor encoder membaca 200 pulsa. Hasil dari eksperimen ini berbeda dengan spesifikasinya. Berdasarkan hal tersebut, diputuskan untuk menggunakan pembacaan eksperimen sebagai nilai pulsa sensor encoder pendulum pada penelitian kali ini.

4.1.2 Hasil pemeriksaan respon motor DC

Setelah melakukan pemeriksaan pulsa sensor encoder. Selanjutnya dapat memeriksa respon motor DC. Gambar 4-3 menunjukkan hasil pembacaan respon posisi motor DC.



Gambar 4-3 Respon posisi motor DC.

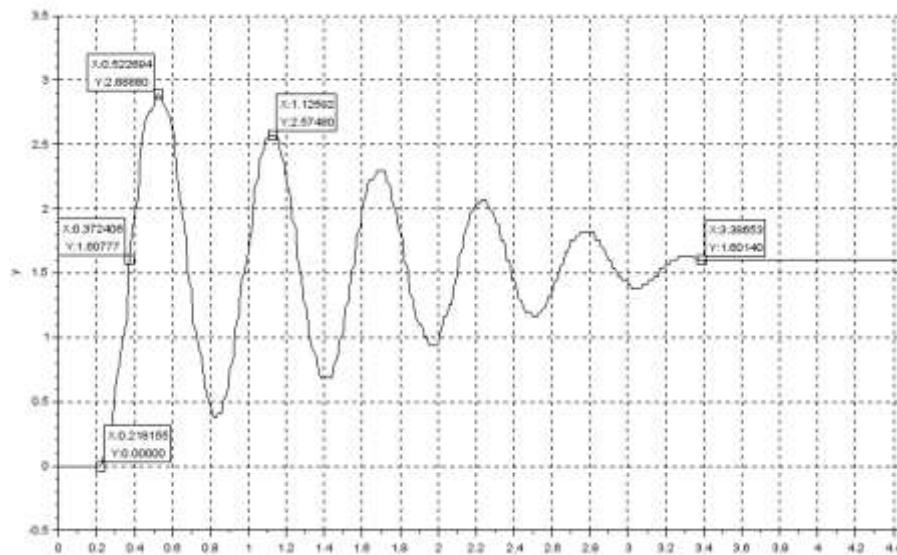
Dari respon posisi motor DC di atas, diketahui untuk mengalami perpindahan posisi motor DC membutuhkan waktu 0,13 detik. Posisi puncak berada di nilai 3,78 radian atau pada posisi $216,5^\circ$ di waktu 0,28 detik. Sedangkan respon motor DC

yang diinginkan berada di nilai 1,57 radian atau pada posisi 90° . Berdasarkan hal tersebut diketahui error steady state bernilai 1,91 radian dan overshoot yang muncul sebesar 140,76% dari hasil *steady state* yang diinginkan. Respon motor DC membutuhkan waktu 0,9 detik untuk sampai ditahap *steady state*.

Berdasarkan informasi yang didapat, diketahui bahwa respon posisi motor DC masih belum sesuai dengan respons yang diinginkan. Sehingga perlu dilakukan kendali sistem agar respon yang diinginkan dapat tercapai.

4.1.3 Pemeriksaan respon pendulum ayun

Dari hasil pemeriksaan respon pendulum ayun, didapat grafik yang ditunjukkan oleh Gambar 4-4.



Gambar 4-4 grafik pergerakan pendulum ayun.

Informasi dari grafik respon pergerakan pendulum ayun menunjukkan pendulum yang berayun bebas memerlukan waktu 3,38 detik untuk berhenti berayun. Nilai redaman di setiap 1 fase amplitudo adalah 1,12 (*decay ratio*). Pada fase kelima, amplitudo mengalami redaman sebesar 80% dari amplitudo awal. Ayunan pendulum ini perlu dikendalikan agar waktu yang diperlukan untuk berhenti dan besarnya ayunan yang terjadi bisa dikurangi.

4.2 Hasil perhitungan Model Matematis

Berdasarkan perumusan model matematis pada bab 2, diketahui terdapat 2 fungsi transfer pada objek kendali *rotary pendulum*. Fungsi transfer yang pertama menunjukkan respon keluaran dari motor DC atau lengan pendulum, dan fungsi transfer yang kedua menunjukkan respon keluaran dari pendulum yang berayun. Data yang telah teridentifikasi didapat dari datasheet motor DC (terlampir) serta pengukuran langsung pada objek kendali *rotary pendulum*. Data yang teridentifikasi selanjutnya digunakan untuk menghitung fungsi transfer objek kendali *rotary pendulum*.

Tabel 4-1 *Identified Data* alat *rotary pendulum*.

No	<i>Identified Data</i>	Nilai	Sumber
1	$J_{\text{motor DC}}$	$5,68 \text{ gcm}^2$ $= 5,68 \times 10^{-7} \text{ kgcm}^2$	Datasheet Motor DC (dengan voltase 24V)
2	K_{tm}	$30,7 \text{ mNm/A}$ $= 0,0307 \text{ Nm/A}$	Datasheet Motor DC (dengan voltase 24V)
3	I_{m}	0,42 A	Datasheet Motor DC (dengan voltase 24V)
4	ω	$\omega = \frac{\text{rpm datasheet}}{\text{rpm reduksi}}$ $= \frac{7190 \text{ rpm}}{504,84}$ $= 14,24 \text{ rpm}$ $= 1,49 \text{ rad/s}$	Datasheet Motor DC (dengan voltase 24V)
5	$l_{\text{pendulum ayun}}$	0,18 m	Pengukuran langsung
6	$M_{\text{pendulum ayun}}$	0,06 kg	Pengukuran langsung
7	$l_{\text{lengan pendulum}}$	0,15 m	Pengukuran langsung

4.2.1 Hasil Fungsi Transfer Lengan Pendulum (Motor DC)

Berdasarkan *problem data* di atas dapat dihitung fungsi transfer dari motor DC, Diketahui fungsi transfer motor DC ditunjukkan pada persamaan (2.1).

$$P(s) = \frac{1}{s(J_m s + B)} \quad (2.1)$$

Dengan asumsi bahwa beban yang bertumpu pada motor DC diabaikan, sehingga nilai momen inersia menggunakan nilai dari *datasheet* motor. Nilai koefisien gesek (B) terlebih dahulu dicari dengan rumus (2.2).

$$B = \frac{K_{tm} \cdot I_m}{\omega} \quad (2.2)$$

Nilai konstanta torsi motor (K_{tm}), arus motor (I_m), dan kecepatan motor (ω) didapat dari *datasheet* motor DC. Nilai yang diambil dengan anggapan voltase yang diberikan ke motor DC sebesar 24 V, sehingga nilai koefisien gesek dapat dihitung seperti pada persamaan (4.1) dan (4.2).

$$B = \frac{0,03 \frac{Nm}{A} \cdot 0,42 A}{1,49 \frac{rad}{s}} \quad (4.1)$$

$$B = 0.0084 \text{ Nm/(rad/s)} \quad (4.2)$$

Ketika nilai koefisien gesek sudah didapat, selanjutnya menentukan nilai fungsi transfer motor DC. Nilai fungsi transfer motor DC pada alat *rotary pendulum* ditunjukkan pada persamaan (4.3).

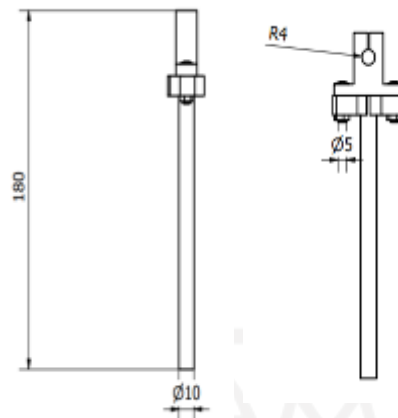
$$P(s) = \frac{1}{0.0000000568 s^2 + 0.00864 s} \quad (4.3)$$

Nilai fungsi transfer motor DC dapat disederhanakan menjadi seperti pada persamaan (4.4).

$$P(s) = \frac{1760563,38}{s^2 + 15211,27 s} \quad (4.4)$$

4.2.2 Hasil Fungsi Transfer Pendulum Ayun

Berdasarkan model matematis yang digambarkan pada Gambar 2-11, diketahui bahwa model ilustrasi pendulum merupakan pendulum tali dengan beban diujung talinya. Hal ini berbeda dengan model pendulum dari alat *rotary pendulum* yang ditunjukkan pada Gambar 4-5.



Gambar 4-5 Rancangan pendulum ayun (Sutejo, 2020).

Dari gambar tersebut, diasumsikan bahwa model rancangan pendulum memiliki sistem yang sama dengan model pendulum menggunakan tali. Asumsi ini dikarenakan frekuensi natural dari pendulum tidak bergantung dengan massa beban pendulum seperti yang ditunjukkan pada persamaan (2.10). Dari persamaan tersebut, dapat dihitung frekuensi natural untuk pendulum ayun.

$$\omega_n = \frac{1}{2\pi} \sqrt{\frac{g}{l}} \quad (2.10)$$

$$\omega_n = \frac{1}{2\pi} \sqrt{\frac{9,81 \text{ m/s}^2}{0,18 \text{ m}}} \quad (4.5)$$

$$\omega_n = 1,18 \quad (4.6)$$

Sehingga, nilai frekuensi natural dari alat *rotary pendulum* adalah 1,18. Langkah selanjutnya adalah mencari nilai fungsi transfer untuk model pendulum ayun. Pertama-tama dicari nilai untuk konstanta damponya, dimana nilai konstanta damponya dapat dicari dengan menghitung rumus (2.12) dan (2.13).

$$\zeta = \frac{\delta}{\sqrt{(2\pi)^2 + \delta^2}} \quad (2.12)$$

Dimana;

$$n\delta = \ln \frac{X_i}{X_{i+n}} \quad (2.13)$$

Berdasarkan data percobaan ayunan pendulum secara bebas yang ditunjukkan pada Gambar 4-4. Diketahui amplitudo pada fase kelima mengalami penurunan nilai sebesar 80%, data ini dapat digunakan untuk menghitung nilai δ . Berikut ditunjukkan perhitungan nilai δ .

$$5\delta = \ln \frac{X_i}{(100-80)\%X_i} \quad (4.7)$$

$$5\delta = \ln \frac{x_1}{0,02x_i} \quad (4.8)$$

$$5\delta = \ln 5 \quad (4.9)$$

$$\delta = \frac{1,61}{5} \quad (4.10)$$

$$\delta = 0,322 \quad (4.11)$$

Sehingga nilai konstanta damping dapat dihitung sebagai berikut.

$$\zeta = \frac{0,322}{\sqrt{(2\pi)^2 + 0,322^2}} \quad (4.12)$$

$$\zeta = 0,05 \quad (4.13)$$

Berdasarkan persamaan (2.11) nilai fungsi transfer pendulum ayun didefinisikan sebagai berikut.

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.11)$$

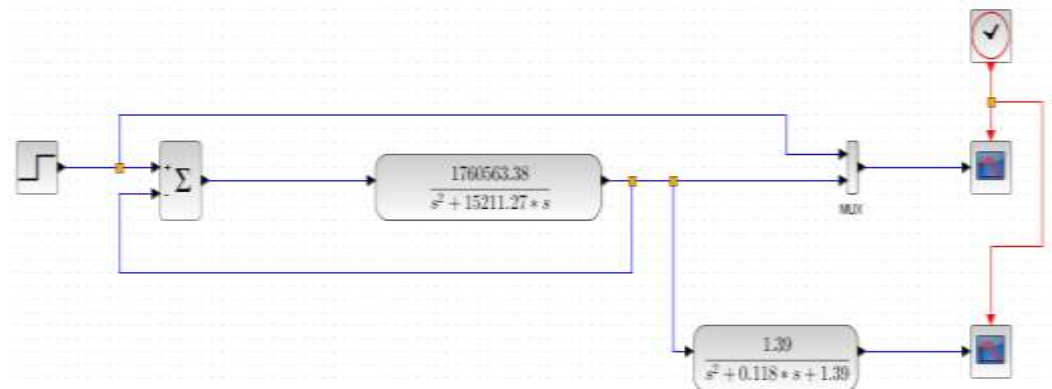
Diketahui dari perhitungan pada persamaan (4.6) frekuensi natural (ω_n) bernilai 1,18. Maka fungsi transfer pendulum ayun dapat dihitung menjadi.

$$H(s) = \frac{1,18^2}{s^2 + 2(0,05)(1,18)s + 1,18^2} \quad (4.14)$$

$$H(s) = \frac{1,39}{s^2 + 0,118s + 1,39} \quad (4.15)$$

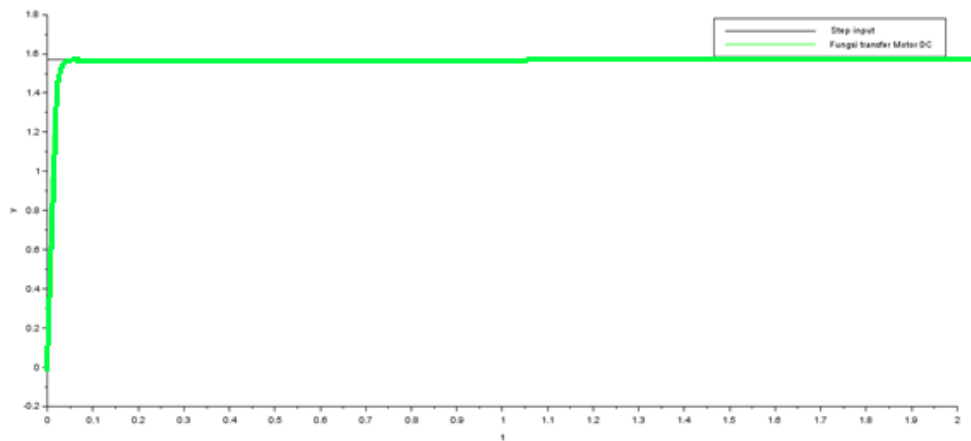
4.2.3 Simulasi fungsi transfer alat *rotary pendulum*

Setelah didapatkan fungsi transfer transfer dari alat rotary pendulum, selanjutnya dilakukan simulasi pada aplikasi Scilab Xcos. Skema simulasi digambarkan pada Gambar 4-6.



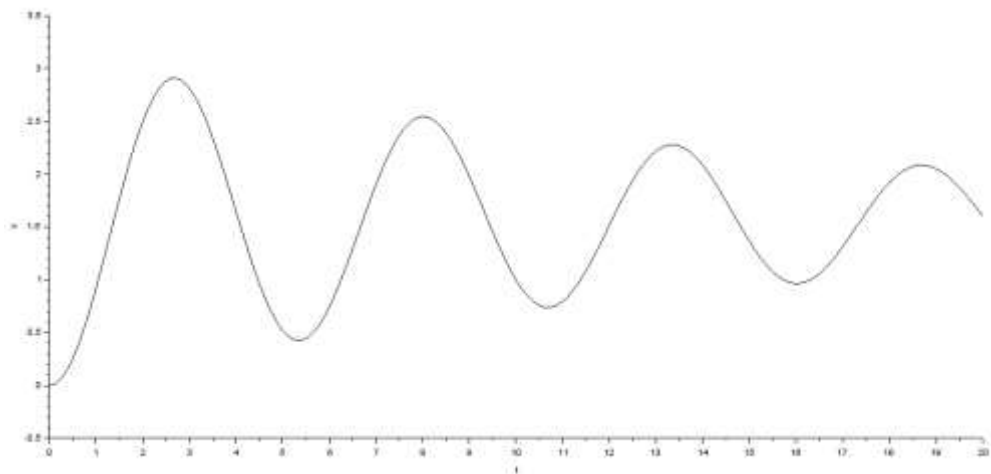
Gambar 4-6 Skema simulasi fungsi transfer pada Xcos.

Gambar 4-7 dan Gambar 4-8 masing-masing menunjukkan grafik motor DC dan grafik pendulum ayun simulasi fungsi transfer alat *rotary pendulum*.



Gambar 4-7 Grafik motor DC fungsi transfer.

Dari gambar di atas terlihat terdapat dua garis grafik. Garis berwarna hitam menunjukkan nilai *input* masukan dan grafik berwarna hijau menunjukkan respons fungsi transfer untuk motor DC. Dari grafik terlihat bahwa fungsi transfer motor DC mengalami kenaikan menuju ke nilai *step input* dan ketika sudah berada di nilai *step input*, respons motor DC mengalami keadaan *steady state*. Respons fungsi transfer motor DC ini sudah sesuai dengan respons implementasi motor DC secara langsung di alat *rotary pendulum*.



Gambar 4-8 Grafik pendulum fungsi transfer.

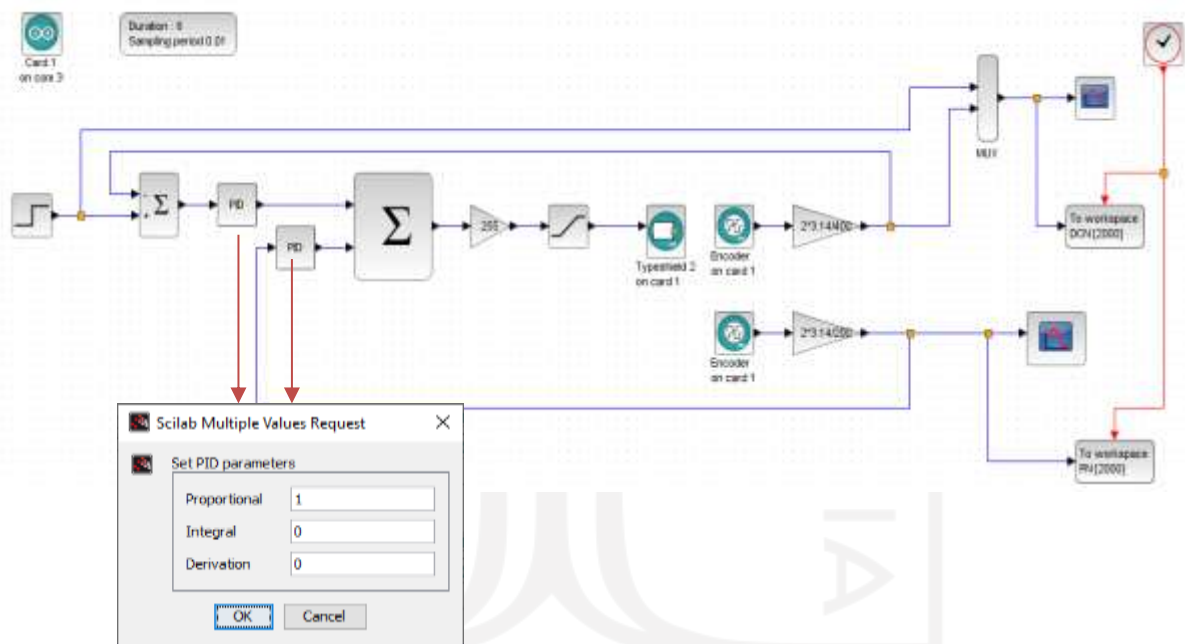
Gambar 4-8 menunjukkan hasil respons fungsi transfer pendulum ayun. Dari gambar di atas, terlihat bahwa fungsi transfer pendulum ayun menunjukkan hasil osilasi yang teredam atau yang biasa disebut dengan respons *underdamped*. Secara teori, respons *underdamped* akan terus memiliki respons osilasi meskipun respons tersebut mengalami redaman. Sedangkan pada respons implementasi pendulum

ayun secara langsung, pendulum berosilasi, lalu teredam dan, pada waktu tertentu akan berhenti berosilasi.

4.3 Hasil Pengujian kendali PID objek kendali *rotary pendulum*

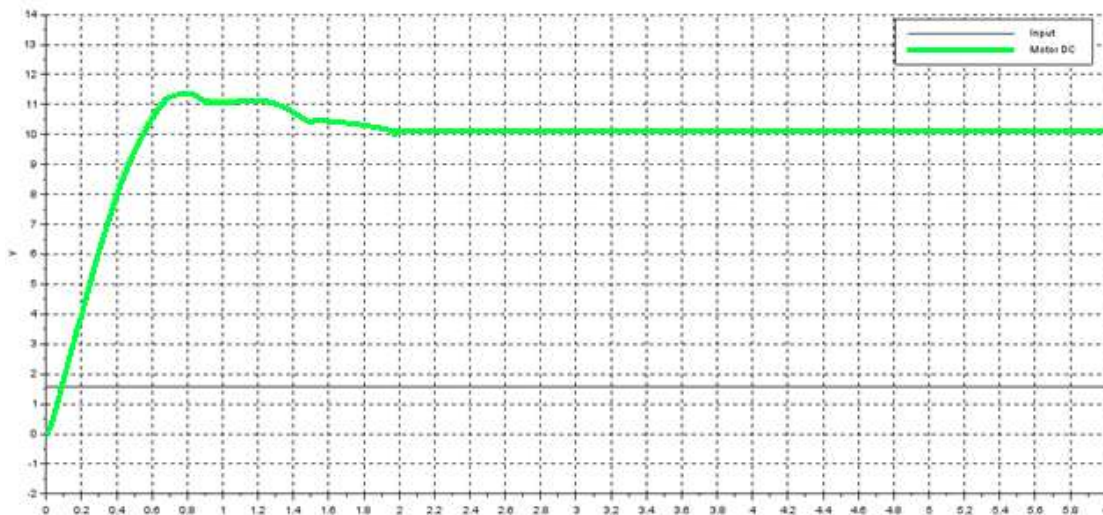
4.3.1 Pengujian tanpa Kendali PID

Pengujian alat *rotary pendulum* dapat dilakukan dengan skema yang ditunjukkan oleh Gambar 3-11. Caranya adalah dengan tidak memasukkan nilai konstanta integral dan derivatif pada kendali PID dan konstanta proporsional diberikan nilai 1. Konfigurasi ini ditunjukkan pada Gambar 4-9.



Gambar 4-9 Konfigurasi uji coba tanpa kendali PID

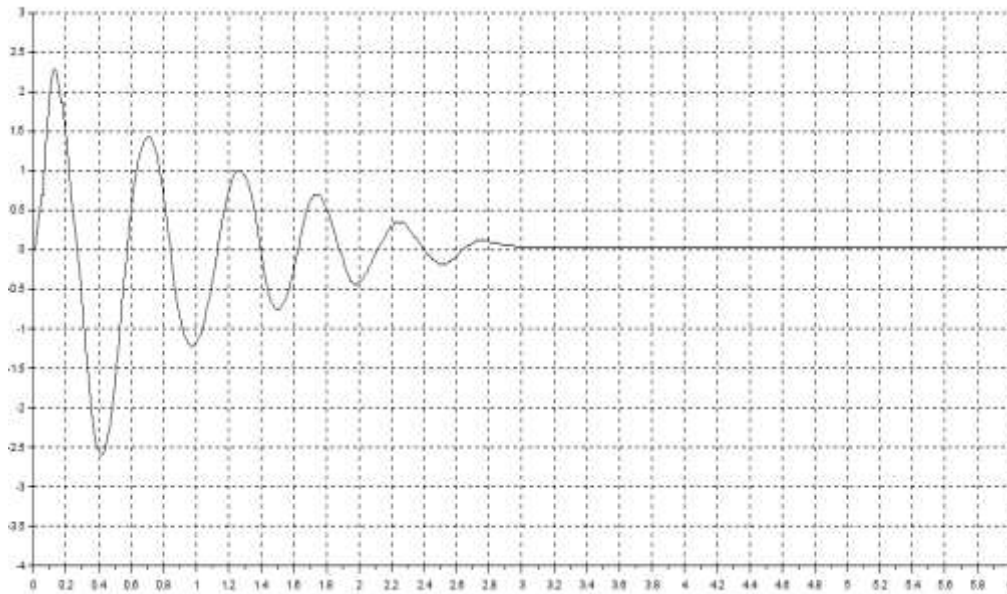
Dari skema Xcos di atas, Hasil uji coba menunjukkan respon perpindahan posisi motor DC dan pendulum ayun. Masing-masing hasil ditunjukkan pada Gambar 4-10 dan Gambar 4-11.



Gambar 4-10 Respon motor DC tanpa kendali PID

Respon di atas menunjukkan pergerakan motor DC tanpa kendali PID. Input yang diberikan adalah 1,57 rad. Garis hitam pada gambar di atas menunjukkan nilai input masukan pada alat *rotary pendulum*, sedangkan garis hijau menunjukkan respon aktual motor DC.

Dari grafik di atas diketahui motor DC memiliki waktu perubahan posisi di 0,6 detik dengan posisi *steady* di nilai 10,15 radian atau berhenti di posisi $581,55^\circ$ di waktu 2 detik. *Overshoot* yang terjadi adalah sebesar 546,9% jika dibandingkan dengan kondisi *steady state* yang diinginkan. Berdasarkan hal tersebut diketahui bahwa nilai posisi *steady* aktual motor DC sangat jauh berbeda dari respon yang diinginkan. Sedangkan untuk respon pendulum ayun ditunjukkan pada Gambar 4-11 .

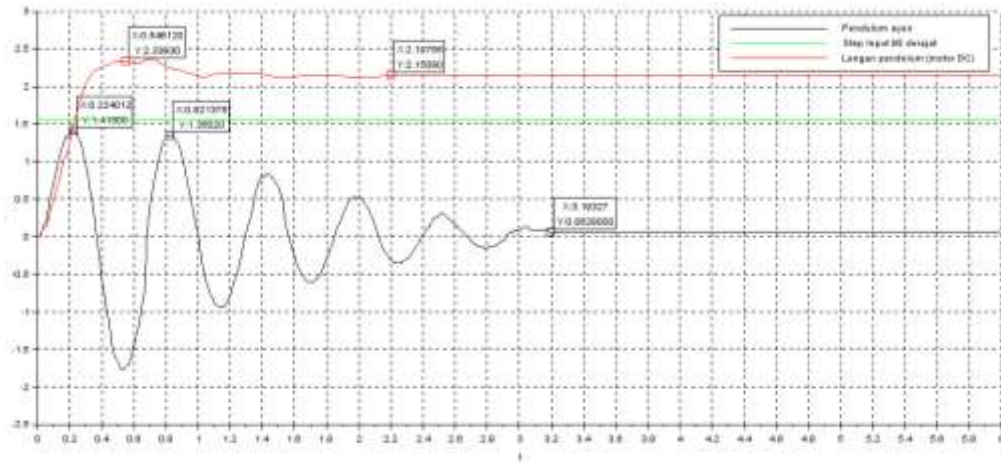


Gambar 4-11 Respon pendulum ayun tanpa kendali PID

Dari grafik di atas didapatkan informasi bahwa pendulum memerlukan waktu 3 detik untuk berhenti berayun. Nilai redaman disetiap 1 fase amplitudo sebesar 1,62. Pada grafik dapat dilihat juga pendulum masih mengalami ayunan yang besar. Pada waktu 0,13 detik pendulum memiliki ayunan yang paling besar. Metode pengendalian diperlukan untuk mengecilkan ayunan yang terjadi pada pendulum.

4.3.2 Pengujian kendali PID+PD metode ZNFD

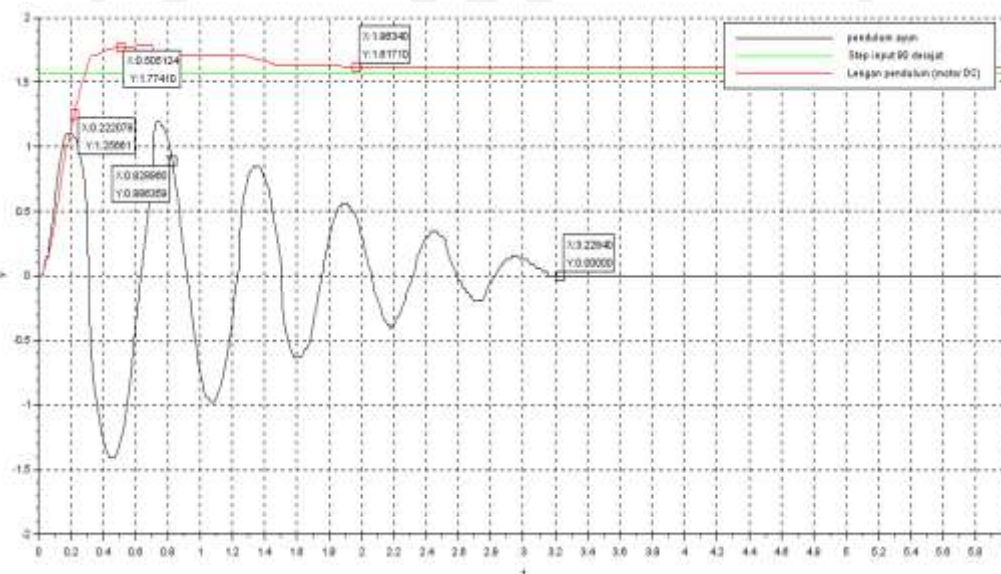
Pengujian selanjutnya adalah mengimplementasikan konfigurasi kendali PID+PD pada alat *rotary pendulum*. Kendali PID digunakan untuk mengendalikan motor DC, dan kendali PD digunakan untuk mengendalikan pendulum ayun. Hal pertama yang dilakukan adalah mencari nilai PID pada motor DC. Nilai konstanta kendali PID didapatkan dengan melakukan metode *manual tuning*. Langkah pertama, dilakukan pencarian nilai konstanta *proportional* (K_P) untuk kendali PID pada motor DC. Dari hasil iterasi didapatkan nilai konstanta *proportional* 0,3. Gambar 4-12 menunjukkan hasil respon objek kendali *rotary pendulum* ketika diberikan nilai K_P 0,3.



Gambar 4-12 Respon objek kendali ketika diberikan nilai K_P 0,3.

Berdasarkan gambar di atas, terlihat bahwa respon motor DC (garis merah) sudah mendekati nilai inputnya (garis hijau). Motor DC sudah *settle* di waktu 2.19 detik dan berada di posisi 2.15 rad. Nilai kondisi *steady state* dari respon motor DC dianggap sudah lebih mendekati ke nilai posisi yang diinginkan.

Setelah nilai K_P sudah didapat, selanjutnya adalah menentukan nilai konstanta integral (K_I) dan derivatif (K_D) dari kendali PID pada motor DC. Menggunakan metode iterasi, didapat nilai K_I 0,46 dan K_D 0,02 yang menunjukkan hasil respon seperti pada Gambar 4-13.



Gambar 4-13 Respon objek kendali ketika diberikan nilai K_P 0,3; K_I 0,46; K_D 0,02.

Karena pengendalian objek kendali baru diterapkan pada motor DC yang mengendalikan posisi perputaran lengan pendulum, maka terlihat pada Gambar

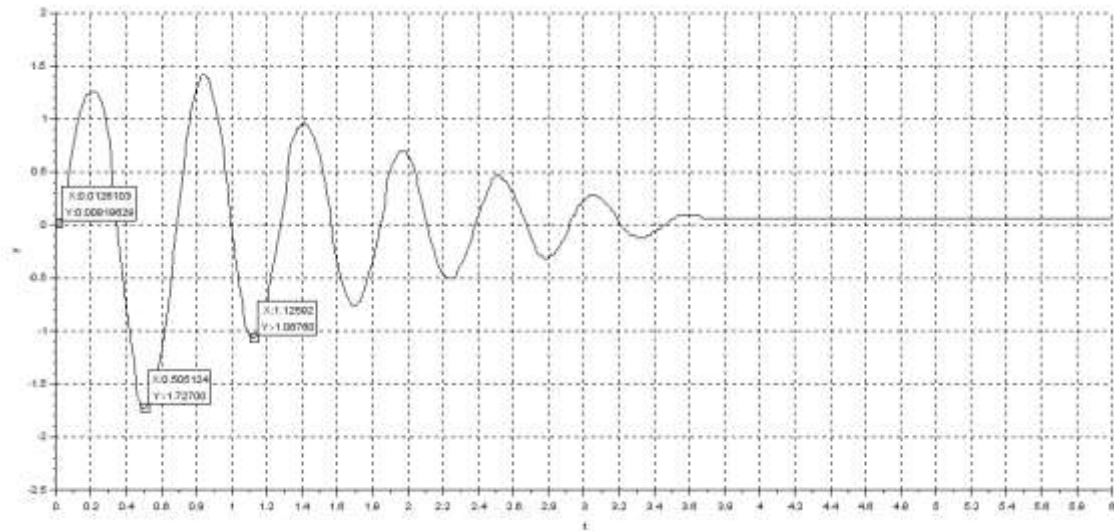
4-13 pergerakan pendulum ayun (garis hitam) masih belum terkendali. Hal ini karena pendulum masih membutuhkan waktu 3,2 detik untuk berhenti berayun. Sedangkan, posisi perputaran motor DC sudah mulai mendekati ke nilai *input* yang diberikan. Berdasarkan grafik di atas, respon motor DC sudah mengalami peningkatan karena *error steady state* mengecil yaitu hanya sebesar 0,04 radian. Respon mulai stabil di waktu 1,96 detik dan respon memiliki overshoot sebesar 12,7% jika dibandingkan dengan respon yang diinginkan.

Ketika nilai PID untuk kendali motor DC sudah didapatkan, langkah selanjutnya adalah mencari nilai konstanta PD bagi pendulum ayun. Pencarian konstanta PD tidak dapat menggunakan metode ZNFD. Hal ini karena metode ZNFD memiliki syarat respon model kendali yang berosilasi secara kontinyu. Sedangkan model pendulum ayun memiliki respon redaman. Sehingga nilai PD pendulum ayun tidak dapat dianalisis menggunakan metode ZNFD

4.3.3 Pengujian kendali PID+PD metode Cohen-Coon

Selanjutnya mencoba mencari nilai konstanta PD menggunakan metode Cohen-Coon. Karena respons pendulum ayun menunjukkan fenomena teredam (*underdamped*), sehingga memenuhi syarat penggunaan metode Cohen-Coon untuk mencari nilai konstanta PD.

Untuk mencari nilai konstanta PD, langkah pertama adalah mengatur nilai konstanta P pada kendali pendulum (K_{Ps}) sampai hasil grafik menunjukkan respon redaman. Dari hasil tersebut, akan didapat data waktu 1 fase amplitudo (T_P) dan waktu awal respon mulai mengalami kenaikan (L). dari hasil beberapa uji coba pengaturan nilai K_{Ps} , diketahui nilai K_{Ps} yang sesuai adalah 0,05. Gambar 4-13 menunjukkan respon pendulum ayun ketika diberikan nilai K_{Ps} 0,05.



Gambar 4-14 respons pendulum ayun ketika diberikan nilai K_{Ps} 0,05.

Dari Gambar 4-14, didapat data sebagai berikut. Dengan konstanta proporsional pendulum (K_{Ps}) 0,05, diketahui 1 fase amplitudo (T_{Ps}) membutuhkan waktu 0,01 detik dan respon mulai mengalami kenaikan (L) diwaktu 0,62 detik. Informasi tersebut dapat digunakan untuk menghitung konstanta derivatif pendulum ayun (K_{Ds}). Berdasarkan Tabel 2-4, nilai K_{Ds} untuk pengendalian PD dapat dihitung menjadi sebagai berikut

$$K_{Ds} = L \left[\frac{6-2\left(\frac{L}{T_p}\right)}{21+3\left(\frac{L}{T_p}\right)} \right] \quad (4.15)$$

$$K_{Ds} = 0,01 \left[\frac{6-2\left(\frac{0,01}{0,62}\right)}{21+3\left(\frac{0,01}{0,62}\right)} \right] \quad (4.16)$$

$$K_{Ds} = 0,01 \left[\frac{6-2\left(\frac{0,01}{0,62}\right)}{21+3\left(\frac{0,01}{0,62}\right)} \right] \quad (4.17)$$

$$K_{Ds} = 0,01 \left[\frac{6-0,032}{21+0,048} \right] \quad (4.18)$$

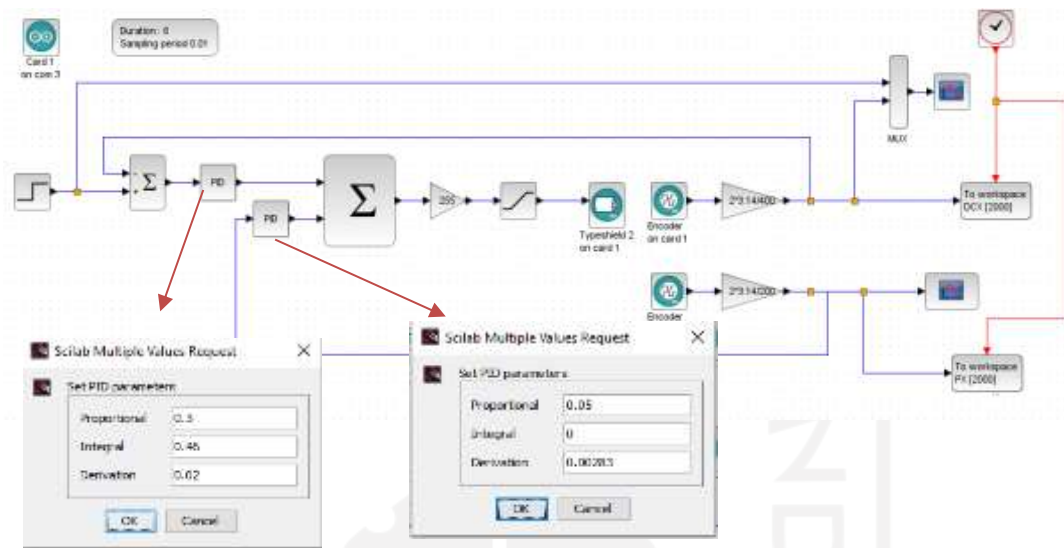
$$K_{Ds} = 0,01 \left[\frac{5,968}{21,048} \right] \quad (4.19)$$

$$K_{Ds} = 0,01(0,283) \quad (4.20)$$

$$K_{Ds} = 0,00283 \quad (4.21)$$

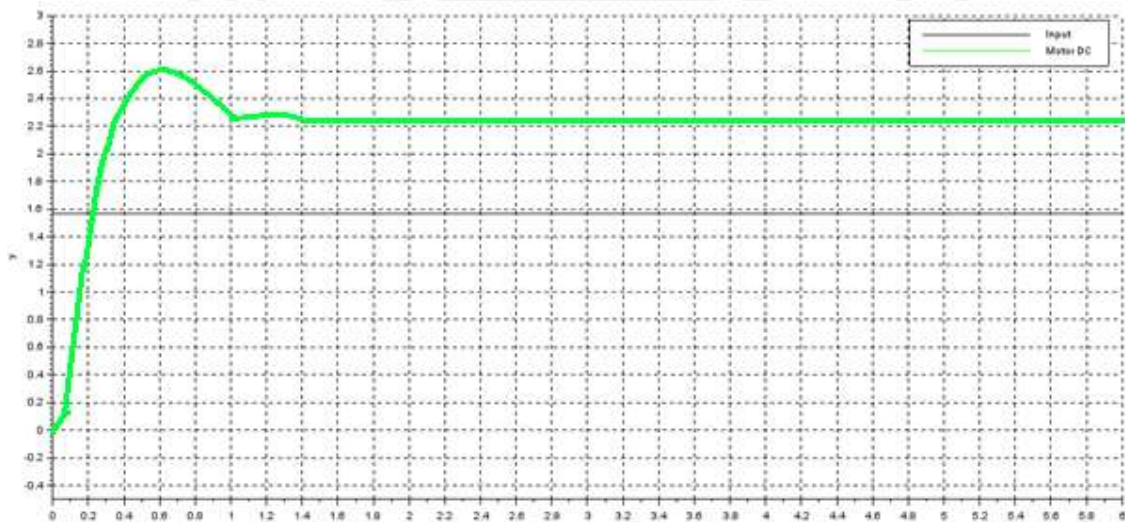
Hasil perhitungan menunjukkan K_{Ds} bernilai 0,00283. Sehingga nilai konstanta PD yang menggunakan metode Cohenn-Coon adalah K_{Ps} 0,05 dan K_{Ds} 0,00283. Xcos

untuk konfigurasi kendali PID+PD menggunakan metode Cohen Coon ditunjukkan pada Gambar 4-15.



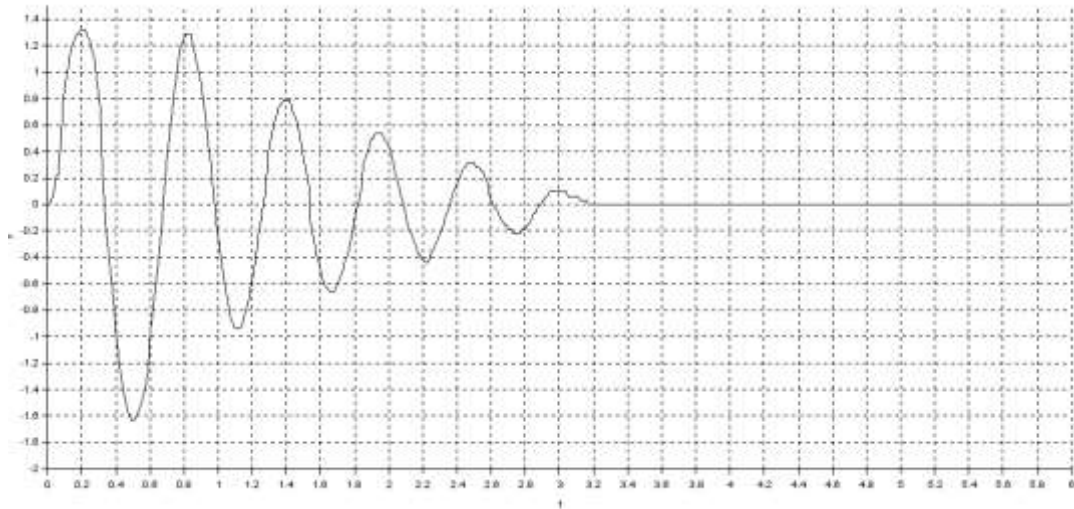
Gambar 4-15 Xcos metode PID+PD Cohen-Coon alat *rotary pendulum*.

Berdasarkan konfigurasi di atas, didapatkan hasil sebagai berikut.



Gambar 4-16 Grafik respons motor DC kendali PID+PD Cohen-Coon.

Berdasarkan grafik respons di atas diketahui bahwa motor DC membutuhkan waktu 1,41 detik untuk berada di kondisi *steady state*. *Error steady state* sebesar 0,67 radian yang dinilai sudah cukup bagus. Namun, *overshoot* yang ditunjukkan oleh respons ini cukup tinggi yaitu di 42,6% dari respons yang diinginkan. Sedangkan untuk grafik pendulum ayun menggunakan kendali PID+PD metode Cohen-Coon ditunjukkan pada Gambar 4-17.

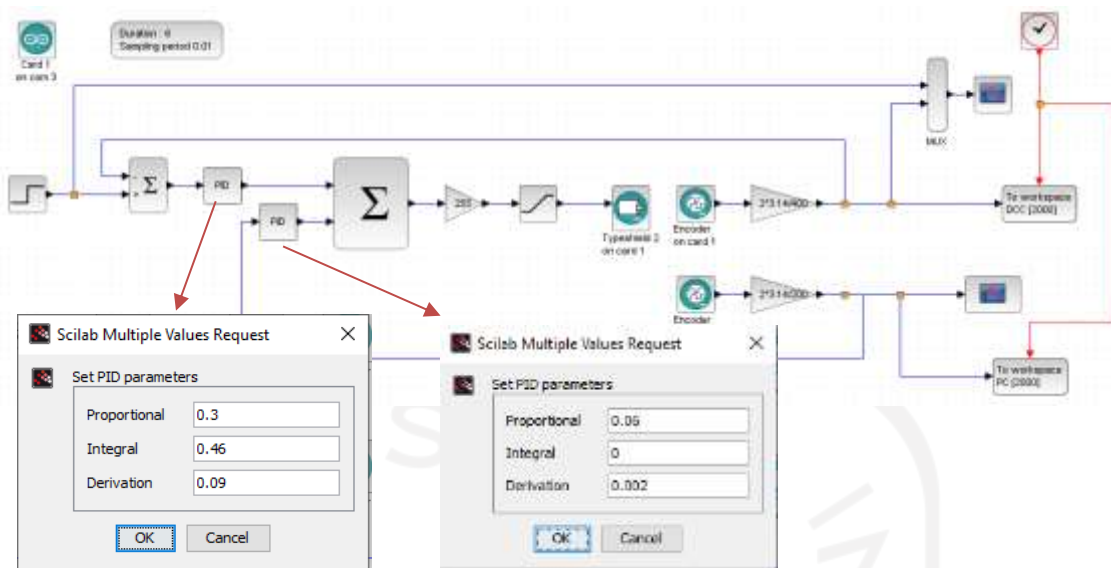


Gambar 4-17 Grafik respons pendulum ayun kendali PID+PD Cohen-Coon.

Berdasarkan respon pendulum di atas diketahui bahwa pendulum membutuhkan waktu 3,17 detik untuk berhenti berayun dengan decay ratio sebesar 1,02. Terlihat juga ayunan pendulum lebih kecil apabila dibandingkan dengan respon pendulum tanpa menggunakan kendali PID+PD. Konstanta dari PID+PD ini dapat digunakan sebagai acuan untuk melakukan *tuning* secara *manual*.

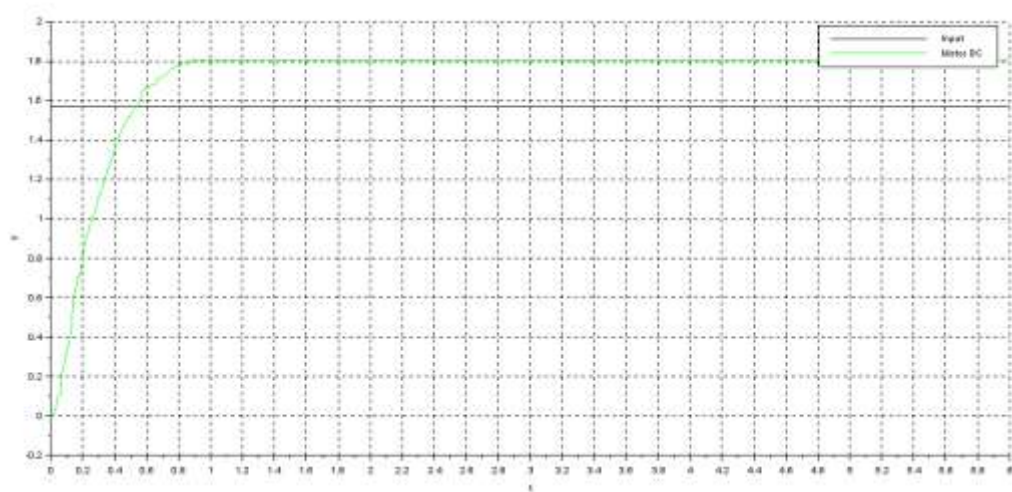
4.3.4 Pengujian kendali PID+PD manual tuning

langkah selanjutnya adalah mencari nilai konstanta kendali PD pendulum ayun menggunakan metode *manual tuning*. Dari metode tersebut didapat konstanta $K_{Ps} = 0,06$ dan $K_{Ds} = 0,002$. Pada nilai konstanta PID motor DC terdapat penyesuaian untuk konstanta K_D menjadi 0,09. konstanta K_P dan K_I yang digunakan sama dengan percobaan sebelumnya. Kofigurasi Xcos pengujian ini ditunjukkan pada Gambar 4-18.



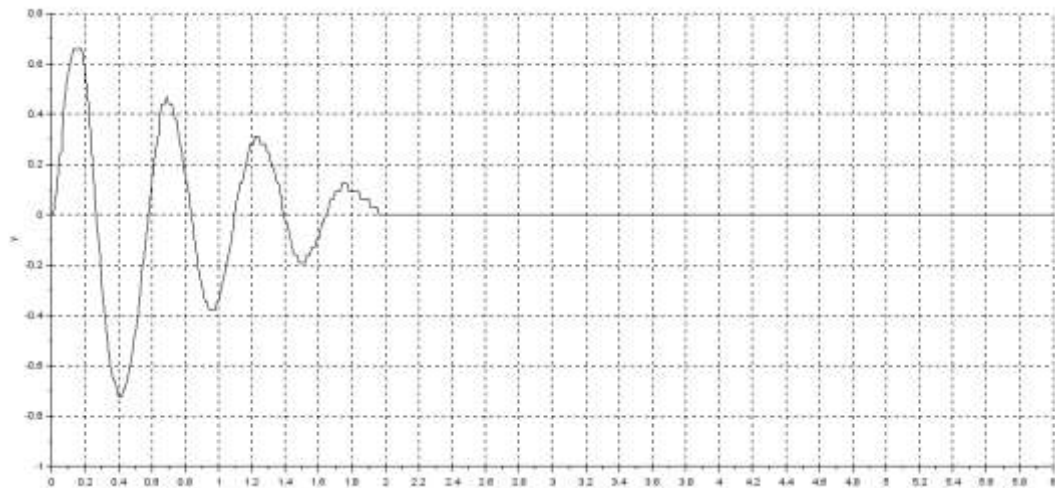
Gambar 4-18 Xcos kendali PID+PD *manual tuning* alat rotary pendulum.

Berdasarkan hasil kendali dengan konfigurasi PID+PD didapat grafik yang ditunjukkan pada Gambar 4-19 untuk respon motor DC dan Gambar 4-20 untuk respon pendulum ayun.



Gambar 4-19 Respon motor DC dengan kendali PID+PD.

Dari grafik di atas didapat informasi respon pergerakan motor DC dengan kendali PID+PD menggunakan metode *manual tuning*. Respons motor DC membutuhkan waktu 0,89 detik untuk berada pada kondisi *steady state*. *Error steady state* sebesar 0,23 radian dan persen overshoot di nilai 14,96%.

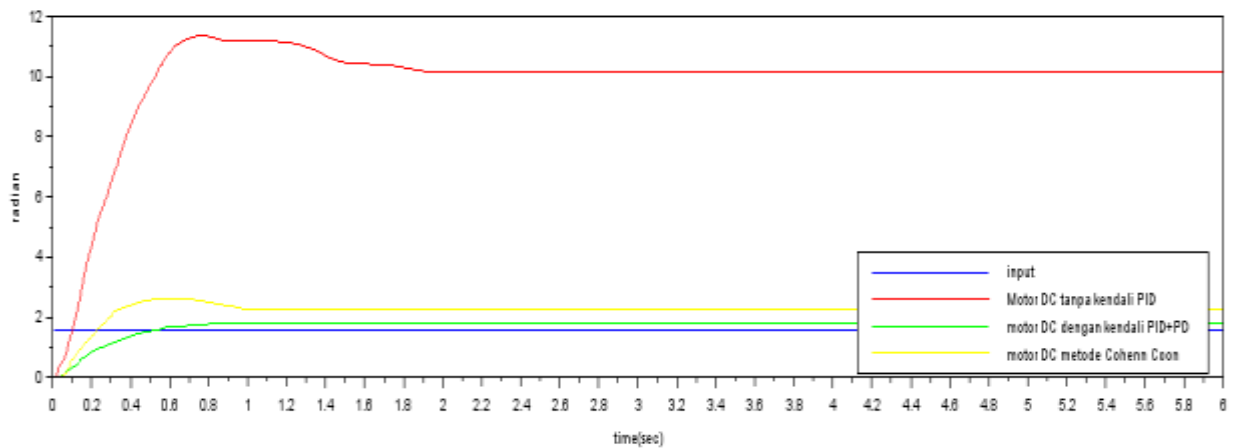


Gambar 4-20 Respon pendulum ayun dengan kendali PID+PD.

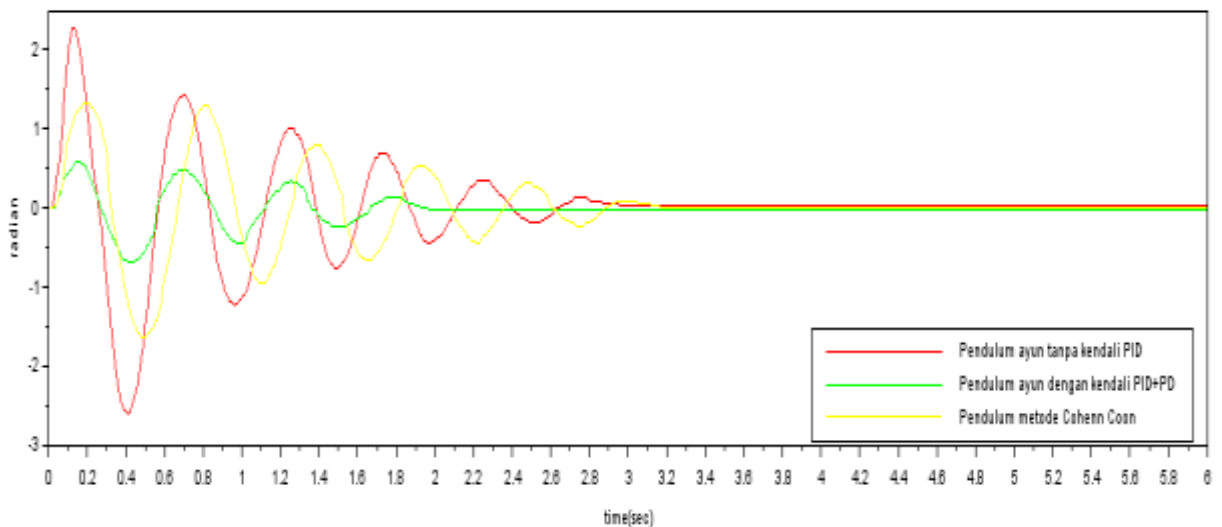
Berdasarkan respon pendulum ayun menggunakan kendali PID+PD metode manual tuning diketahui bahwa pendulum membutuhkan waktu 1,96 detik untuk berhenti berayun. Redaman pada setiap 1 fase amplitudo memiliki nilai 1,41. Menggunakan metode ini, pendulum memiliki ayunan yang lebih kecil. Hal ini akan jelas terlihat apabila dibandingkan dengan percobaan lain.

4.4 Pembahasan

Berdasarkan pengujian, dilakukan tiga uji coba untuk mengendalikan ayunan pada alat *rotary pendulum*. Ketiga uji coba tersebut adalah, pengujian tanpa kendali PID, pengujian dengan kendali PID+PD dengan metode Cohen-Coon, dan pengujian dengan kendali PID+PD dengan metode *manual tuning*. Selanjutnya, ketiga hasil tersebut dibandingkan sehingga diketahui mana yang lebih baik performanya. Gambar 4-21 dan Gambar 4-22 menunjukkan perbandingan dari respons motor DC dan respons pendulum ayun.



Gambar 4-21 perbandingan respons motor DC.



Gambar 4-22 perbandingan respons pendulum ayun.

Gambar 4-21 menunjukkan perbandingan respons motor DC antara tanpa diberikan kendali PID (merah), diberikan kendali PID+PD metode Cohen-Coon (kuning), dan diberikan kendali PID+PD metode *manual tuning* (hijau). Berdasarkan gambar tersebut diketahui dengan nilai input step 1,57 radian (biru), respons alat *rotary pendulum* menunjukkan hasil yang sangat jauh berbeda dengan nilai yang diinginkan. Hasil itu menunjukkan bahwa alat *rotary pendulum* memang memerlukan kendali agar hasil keluaran sesuai dengan nilai masukan yang diberikan.

Ketika diberikan kendali PID+PD menggunakan metode Cohen-Coon, grafik respons alat uji menunjukkan perbaikan. Hal ini dikarenakan respons dari motor DC mulai mendekati ke nilai keluaran yang diinginkan. Tetapi, masih

terdapat *overshoot* yang terjadi ketika diberikan konfigurasi kendali PID+PD menggunakan metode Cohen-Coon ini. Sedangkan, ketika objek kendali *rotary pendulum* dikendalikan menggunakan PID+PD *manual tuning*. Terlihat respons grafik lebih mendekati ke nilai yang diinginkan dan tidak ada *overshoot* yang terjadi pada respons metode ini.

Gambar 4-22 menggabungkan respon pendulum ayun antara ketiga metode pengujian. Hal yang perlu diperhatikan adalah seberapa lama waktu pendulum untuk berhenti berayun dan seberapa besar ayunan yang terjadi pada pendulum. Dari gambar tersebut ketika alat tidak diberikan suatu kendali, pendulum berayun sangat besar dan membutuhkan waktu 3 detik sebelum pendulum benar-benar berhenti. Menggunakan metode PID+PD Cohen-Coon terlihat bahwa ayunan pendulum tidak sebesar ayunan yang terjadi ketika tidak diberikan kendali, tetapi metode ini membutuhkan waktu yang sedikit lebih lama agar pendulum berhenti berayun. Pada aplikasi metode kendali PID+PD *manual tuning*, ayunan pendulum terlihat semakin kecil dan waktu untuk pendulum berhenti lebih singkat dibandingkan kedua metode sebelumnya.

Hasil ini masih bisa dikembangkan agar mendapatkan hasil yang lebih baik. Seperti mengembangkan pengendalian dengan menggabungkan kendali PID+PD dengan metode kendali lain seperti NCTF ataupun Fuzzy. Atau dengan menggunakan perangkat lunak Matlab untuk mencari nilai konstanta PID. Pada perangkat lunak Matlab, terdapat fungsi *auto tuning* untuk mendapatkan konstanta PID sesuai dengan respons yang diinginkan. Sehingga lebih mempermudah pencarian nilai konstanta PID.

BAB 5

PENUTUP

5.1 Kesimpulan

Kesimpulan dari penelitian ini adalah:

1. Objek kendali *rotary pendulum* memiliki dua karakteristik objek kendali. Motor DC memiliki karakteristik *critical damped* dan pendulum memiliki karakteristik *underdamped*. Model matematis objek kendali *rotary pendulum* telah mampu disusun sesuai dengan karakteristik tersebut.
2. Simulasi objek kendali telah dilakukan dengan merancang blok-blok fungsi pada Xcos, dengan melakukan simulasi dapat diketahui respons dari objek kendali *rotary pendulum*.
3. Sistem kendali PID menggunakan blok-blok fungsi pada Xcos untuk objek kendali *rotary pendulum* telah dirancang. Skema sistem kendali dapat menunjukkan hasil respons dengan tiga mode yang berbeda yaitu tanpa kendali PID, dengan kendali PID+PD metode Cohen-Coon, dan kendali PID+PD metode *manual tuning*.
4. Berdasarkan hasil implementasi sistem kendali pada alat, disimpulkan bahwa alat memang membutuhkan pengendalian untuk mencapai respons yang diinginkan. Penggunaan metode kendali Cohen-Coon sudah dapat memberikan hasil yang mendekati nilai keluaran yang diinginkan. Tetapi hasil yang didapat masih kurang maksimal, untuk mendapatkan hasil yang lebih maksimal dapat digunakan metode *manual tuning*. Penggunaan metode *manual tuning* memiliki hasil yang lebih baik dibandingkan dengan metode Cohen-Coon. Hal ini dikarenakan setiap model kendali memiliki spesifikasi dan kerumitan yang berbeda. Metode-metode perumusan konstanta PID yang ada dapat digunakan sebagai landasan untuk melakukan *tuning* konstanta PID secara manual.

5.2 Saran atau Penelitian Selanjutnya

Saran untuk penelitian selanjutnya:

1. Penelitian selanjutnya dapat menggabungkan dua metode kendali yang berbeda, seperti penggunaan metode kendali PID+PD ditambah dengan Fuzzy atau metode kendali lainnya.
2. Penelitian selanjutnya dapat menggunakan perangkat lunak Matlab jika memungkinkan, hal ini karena perangkat lunak Matlab lebih sering digunakan dibidang industri secara umum. Pengendalian nilai PID menggunakan Matlab juga akan lebih mudah karena pada Matlab terdapat fitur *auto tuning* untuk kendali PID. Penggunaan fitur ini akan lebih mempermudah pencarian konstanta PID.



DAFTAR PUSTAKA

- Ahmed, & Alnaib, I. (2019). DC Motors.
- Akbar, M. J., Rahmdhani, M., & Wibowo, A. S. (2016). Sistem Kendali Anti Sway Overhead Crane. *Jurnal Penelitian dan Pengembangan Telekomunikasi, Kendali, Elektrik, dan Elektronika (TEKTRIKA)*, 160-166.
- Anyakrawati, A. (2015). Implementasi Model Reference Adaptive Systems (MRAS) untuk Kestabilan pada Rotary Inverted Pendulum. *Jurnal Seminar Hasil Aretasiwi Anyakrawati*.
- Hai, L. X., Nguyen, T. H., Khanh, T. G., Thanh, N. T., Duong, B. T., & Minh, P. X. (2017). Anti Sway Tracking Control of Overhead Crane System Based on PID and Fuzzy Sliding Mode Control. *Journal of Science and Technology*, 116-127.
- Handayani, D. N., Pramudya, Y., Suparwoto, & Muchlas. (2018). The Application of Scilab Software in Frequency Mode Simulation on the Circular Membrane. *Journal of Physics: Theories and Applications*, 83-94.
- Kurniawan, E., Simbolon, R. S., & MN, N. (2020). Analysis and Simulation of PI and PID Control System Using Xcos Scilab. *Journal of Technomaterials*, 108-116.
- Muntari, & Nurhadi, H. (2013). Desain Sistem Kendali Rotary Pendulum dengan Sliding-PID. *JURNAL TEKNIK POMITS*, F243-F249.
- Nise, N. S. (2015). *Control System Engineering*. John Willey & Sons, Inc.
- Nusantoro, G. D., Muslim, M. A., Purwanto, P., & C, R. I. (2012). Rancang Bangun Rotary Inverted Pendulum Menggunakan Kontrol PID.
- Solihin, M. I., Legowo, A., & Wahyudi, W. (2010). Fuzzy-tuned PID Anti-swing Control of Automatic Gantry Crane. *Journal of Vibration and Control*.
- Sutejo, H. R. (2020). *Pemodelan dan Pembuatan Prototipe Sistem Kendali Posisi Rotary Inverted Pendulum*. Yogyakarta: Jurusan Teknik Mesin, Fakultas Teknologi Industri, Universitas Islam Indonesia.
- Technology, H. (n.d.). *L298N Dual H-Bridge Motor Driver*.
- Toochinda, V. (2016). *Feedback Control with Scilab and Arduino*. Bangkok: Varodom Toochinda, Ph.D.

Venturin, M., & Poles, S. (n.d.). *Controller for Inverted Pendulum*. Retrieved from Openeering: <http://www.openeering.com>

Wicaksono, H. (2004). Analisa Performansi dan Robustness Beberapa Metode Tuning Kontroller PID pada Motor DC. *Jurnal Teknik Elektro vol. 4*, 70-78.

Zheng, D., Zhang, S., Wang, S., Hu, C., & Zhao, X. (2015). *A Capacitive Rotary Encoder Based on Quadrature Modulation and Demodulation*.



LAMPIRAN

Lampiran 1

Datasheet Motor DC Maxon S2322

maxon special program

S 2322 Ø22 mm, Graphite Brushes, 6 Watt

M 1:1

		Part Numbers							
		2322 ... -11, 225-260 (insert winding number)							
Winding number		000	001	000	002	003	005	007	
Motor Data									
Values at nominal voltage									
1	Nominal voltage	V	12	15	18	18	24	30	48
2	No load speed	rpm	7010	7000	7330	8690	7190	7340	6070
3	No load current	mA	42.2	33.3	30.6	25.8	21.3	14.3	6.88
4	Nominal speed	rpm	4300	4310	4840	3670	4490	4500	3590
5	Nominal torque (max. continuous torque)	mNm	11.8	12.2	12.2	12.3	12.4	12.3	12.4
6	Nominal current (max. continuous current)	A	0.807	0.664	0.583	0.530	0.420	0.279	0.164
7	Stall torque	mNm	32.8	33.5	35.8	31.4	34.1	33.4	33.4
8	Starting current	A	2.14	1.72	1.64	1.28	1.11	0.724	0.411
9	Max. efficiency	%	70	71	72	71	73	73	73
Characteristics									
10	Terminal resistance	Ω	5.61	6.71	10.9	14.1	21.6	49.7	157
11	Terminal inductance	mH	0.492	0.700	0.935	1.29	1.97	4.43	10.2
12	Torque constant	mNm/A	15.4	18.4	21.8	24.8	30.7	46.1	89.9
13	Speed constant	rpm/V	622	491	437	309	311	207	137
14	Speed/torque gradient	rpm/mNm	227	220	219	223	219	224	258
15	Mechanical time constant	ms	13.9	13.5	13.3	13.3	13.0	12.9	12.5
16	Motor inertia	gcm ²	5.94	5.40	5.79	5.70	5.48	5.50	5.37

Specifications	Operating Range	Comments
<p>Thermal data</p> <p>17 Thermal resistance housing-ambient 14.1 K/W</p> <p>18 Thermal resistance winding-housing 3.9 K/W</p> <p>19 Thermal time constant winding 3.24 s</p> <p>20 Thermal time constant motor 0.49 s</p> <p>21 Ambient temperature -25...+100°C</p> <p>22 Max. permissible winding temperature +125°C</p> <p>Mechanical data (ball bearings)</p> <p>23 Max. permissible speed 9000 rpm</p> <p>24 Axial play 0.05 - 0.15 mm</p> <p>25 Radial play 0.025 mm</p> <p>26 Max. axial load (dynamic) 3.8 N</p> <p>27 Max. force for press fit (static) 64 N</p> <p>28 Max. radial loading, 5 mm from flange 14 N</p> <p>Other specifications</p> <p>29 Number of pole pairs 7</p> <p>30 Number of commutator segments 90</p> <p>31 Weight of motor</p> <p>Values listed in the table are nominal. Explanation of the figures on page 71.</p> <p>Option</p> <p>Sleeve bearings in place of ball bearings</p>		<p>■ Continuous operation In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient. = Thermal limit.</p> <p>□ Short term operation The motor may be briefly overloaded (recurring).</p> <p>— Assigned power rating</p>

Maxon Modular System		Overview on page 20 - 25
<p>Planetary Gearhead</p> <p>Ø22 mm</p> <p>0.5 - 3.0 Nm</p> <p>Page 250</p> <p>Planetary Gearhead</p> <p>Ø29 mm</p> <p>0.5 - 2.0 Nm</p> <p>Page 251</p> <p>Planetary Gearhead</p> <p>Ø26 mm</p> <p>0.5 - 2.0 Nm</p> <p>Page 302</p>		

364 maxon special program
May 2015 edition / subject to change

Lampiran 2

Autonics Rotary Encoder E40-S

E40 Series

Shaft Type/Hollow Shaft Type/Blind Hollow Shaft Type Ø40mm Incremental Rotary Encoder

■ Features

- Easy installation at narrow space
- Low moment of inertia
- Power supply: 5VDC, 12-24VDC ±5%
- Various output types

⚠ Please read "Safety Considerations" in the instruction manual before using.



■ Ordering Information

E40	H	8	5000	3	N	24	
Series	Shaft diameter	Pulses/revolution	Output phase	Control output	Power supply	Cable	
Ø40mm S: shaft type	External 6: Ø6mm 8: Ø8mm	Refer to resolution	2: A, B 3: A, B, Z 4: A, \bar{A} , B, \bar{B} 6: A, \bar{A} , B, \bar{B} , Z, Z	T: Totem pole output N: NPN open collector output V: Voltage output L: Line driver output	5: 5VDC ±5% 24: 12-24VDC ±5%	No mark : Radial cable type C: Radial cable connector type	
Ø40mm H: hollow shaft type, HB: blind hollow shaft type	Inner 6: Ø6mm 8: Ø8mm 10: Ø10mm 12: Ø12mm						

■ Specifications

Item	Shaft Type/Hollow Shaft Type/Blind Hollow Shaft Type Ø40mm Incremental Rotary Encoder		
Resolution (PPR)*1	*1, *2, *5, 10, *12, 15, 20, 23, 25, 30, 35, 40, 45, 50, 60, 75, 100, 120, 150, 192, 200, 240, 250, 256, 300, 360, 400, 500, 512, 600, 800, 1000, 1024, 1200, 1500, 1800, 2000, 2048, 2500, 3000, 3600, 5000		
Electrical specification	Output phase	A, B, Z phase (line driver A, \bar{A} , B, \bar{B} , Z, Z phase)	
	Phase difference of output	Phase difference between A and B: $\frac{T}{4} \pm \frac{T}{8}$ (T=1 cycle of A phase)	
	Control output	Totem pole output	*[Low] - Load current: max. 30mA, residual voltage: max. 0.4VDC= *[High] - Load current: max. 10mA, output voltage (power voltage 5VDC=): min. (power voltage-2.0)VDC= Output voltage (power voltage 12-24VDC=): min. (power voltage-3.0)VDC=
		NPN open collector output	Load current: max. 30mA, residual voltage: max. 0.4VDC=
		Voltage output	Load current: max. 10mA, residual voltage: max. 0.4VDC=
	Response time (rise/fall)	Line driver output	*[Low] - Load current: max. 20mA, residual voltage: max. 0.5VDC= *[High] - Load current: max. -20mA, output voltage (power voltage 5VDC=): min. 2.5VDC= Output voltage (power voltage 12-24VDC=): min. (power voltage-3.0)VDC=
		Totem pole output	
		NPN open collector output	Max. 1µs (cable length: 2m, I sink = 20mA)
		Voltage output	
		Line driver output	Max. 0.5µs (cable length: 2m, I sink = 20mA)
Max. response frequency	300kHz		
Power supply	* 5VDC= ±5% (ripple P-P: max. 5%) * 12-24VDC= ±5% (ripple P-P: max. 5%)		
Current consumption	Max. 80mA (disconnection of the load), line driver output max. 50mA (disconnection of the load)		
Insulation resistance	Over 100MΩ (at 500VDC megger between all terminals and case)		
Dielectric strength	750VAC 50/60Hz for 1 minute (between all terminals and case)		
Connection	Radial cable type, Radial cable connector type		
Mechanical specification	Starting torque	* S type: max. 40gf·cm (0.004N·m) * H/HB type: max. 50gf·cm (0.005N·m)	
	Moment of inertia	Max. 40g·cm ² (4×10 ⁻⁹ kg·m ²)	
	Shaft loading	Radial: max. 2kgf, Thrust: max. 1kgf	
Max. allowable revolution*2	5,000rpm		
Vibration	1.5mm amplitude at frequency of 10 to 55Hz (for 1 min) in each X, Y, Z direction for 2 hours		
Shock	Approx. max. 50G		
Environment	Ambient temperature	-10 to 70°C, storage: -25 to 85°C	
	Ambient humidity	35 to 85%RH, storage: 35 to 90%RH	
Protection structure	IP50 (IEC standard)		
Cable	Ø5mm, 5-wire (line driver output: 8-wire), 2m, Shield cable (AWG24, core diameter: 0.08, number of cores: 40, insulator out diameter: Ø1mm)		
Accessory	* S type: Ø6mm coupling, Ø8mm coupling * H/HB type: bracket		
Approval	CE (except line driver output)		
Unit weight	Approx. 120g		

*1: ** pulse is only for A, B phase (line driver output is for A, \bar{A} , B, \bar{B} phase). Not indicated resolutions are customizable.

*2: Make sure that max. response revolution should be lower than or equal to max. allowable revolution when selecting the resolution.

$$[\text{Max. response revolution (rpm)}] = \frac{\text{Max. response frequency}}{\text{Resolution}} \times 60 \text{ sec}$$

※Environment resistance is rated at no freezing or condensation.

H-24

Autonics

Lampiran 3

Loader Scilab

```
1 // - This file is released under the 3-clause BSD license. See COPYING-BSD.
2 // - Generated by builder.sce: Please, do not edit this file
3
4 try
5   .getversion("scilab");
6 catch
7   .error("Scilab-5.0-or-more-is-required.");
8 end;
9
10 exec(get_absolute_file_path("loader.sce")+"etc\"+"arduino.start");
11
```

Lampiran 4

Sketsa alat *Rotary pendulum*

