

**PEMODELAN DAN PERANCANGAN SISTEM KENDALI  
*ROTARY PENDULUM* DENGAN WAKTU DISKRIT  
MENGUNAKAN SCILAB/XCOS**

**TUGAS AKHIR**

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



**Disusun Oleh :**

**Nama : ALIY NURFIRYAL**  
**No. Mahasiswa : 16525087**  
**NIRM : 2016080680**

**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.”

Yogyakarta, 01 April 2022



Aliy Nurfiryal

**LEMBAR PENGESAHAN DOSEN PEMBIMBING**

**PEMODELAN DAN PERANCANGAN SISTEM KENDALI  
*ROTARY PENDULUM* DENGAN WAKTU DISKRIT  
MENGUNAKAN SCILAB/XCOS**

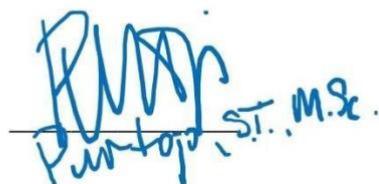
**TUGAS AKHIR**

**Disusun Oleh :**

**Nama** : ALIY NURFIRYAL  
**No. Mahasiswa** : 16525087  
**NIRM** : 2016080680

Yogyakarta, 10-03-2022

Pembimbing,

  
Pembimbing, ST., M.Sc.

**LEMBAR PENGESAHAN DOSEN PENGUJI**

**PEMODELAN DAN PERANCANGAN SISTEM KENDALI**  
**ROTARY PENDULUM DENGAN WAKTU DISKRIT**  
**MENGGUNAKAN SCILAB/XCOS**

**TUGAS AKHIR**

**Disusun Oleh :**

**Nama** : ALIY NURFIRYAL  
**No. Mahasiswa** : 16525087  
**NIRM** : 2016080680

Tim Penguji

Purtojo, S.T., M.Sc.

Ketua

  
Tanggal : 01-04-2022

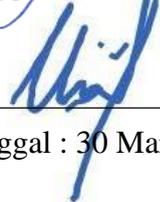
Santo Ajie Dhewanto, S.T., M.M.

Anggota I

  
Tanggal : 31 Maret 2022

Dr. Muhammad Khafidh, S.T., M.T.

Anggota II

  
Tanggal : 30 Maret 2022

Mengetahui

Ketua Jurusan Teknik Mesin



  
Dr. Eng. Risdiono, S.T., M.Eng.

## HALAMAN PERSEMBAHAN

Saya ucapkan terimakasih dan penghargaan setinggi-tingginya kepada :

Orang tua tercinta

**Achyar dan Kostini Wati Arnawijaya**

Sebagai tanda bukti untuk saya dalam menjalankan amanah di jenjang pendidikan. Terimakasih karena telah berusaha semaksimal mungkin untuk menghantarkan saya mencapai gelar sarjana. Semoga dengan gelar yang saya peroleh bisa menjadikan bekal orang tua di akhirat kelak.

*Aamiin ya Rabbal'alamin*

Teruntuk adik dan kakak

**Muhammad Sulthan Hassanain dan Norray Lammalif**

Terima kasih atas dukungan dan semangat yang diberikan, semoga kita bisa terus berjuang bersama agar bisa membanggakan dan membahagiakan orang tua kita.

Teruntuk Pasangan Hidup Kelak (*InshaAllah*)

**Devanie Aulia Sherliantie**

Terima kasih telah hadir untuk memberikan semangat dan dorongan, Menemani proses diri ini agar menjadi individu yang lebih baik lagi. Semoga dengan terselesaikannya tugas akhir ini dapat mempermudah langkah kita untuk kedepannya, semoga Allah SWT akan selalu memberikan kemudahan atas segala urusan kita.

*Aamiin ya Rabbal'alamin*

Teruntuk Baba dan Ibunda

**Mohammad Danan Andoyono dan Elly Evasari Piay**

Terima kasih telah memberikan semangat dan dorongan, menjadi teman tempat berkeluh kesah. Semoga Allah SWT selalu memberikan kesehatan, kebahagiaan serta rejeki yang tiada henti,

*Aamiin ya Rabbal'alamin.*

## **HALAMAN MOTTO**

*“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)**

“Orang yang hebat adalah orang yang memiliki kemampuan menyembunyikan kesusahan, sehingga orang lain mengira ia selalu senang.”

**Imam Syafi’i**

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya.”

**QS Al Baqarah 286**

“Jangan menilai saya dari kesuksesan, tetapi nilai saya dari seberapa sering saya jatuh dan berhasil bangkit kembali.”

**Nelson Mandela**

“Sukses berjalan dari satu kegagalan ke kegagalan yang lain, tanpa kita kehilangan semangat.”

**Abraham Lincoln**

## KATA PENGANTAR ATAU UCAPAN TERIMA KASIH

*Assalamu'alaikum Warrahmatullahi Wabarakatuh*

Alhamdulillahirabbil'alamiin, dengan segala bentuk kebaikan dan izin Allah SWT sehingga saya dapat menyelesaikan skripsi yang berjudul "**Pemodelan dan perancangan sistem kendali rotary pendulum dengan waktu diskrit menggunakan Scilab/Xcos**". Skripsi ini disusun sebagai salah satu syarat bagi mahasiswa untuk memperoleh gelar Sarjana Teknik Mesin (S.T) pada Program Studi Teknik Mesin Fakultas Teknologi Industri di Universitas Islam Indonesia. Saya haturkan terimakasih banyak kepada pihak yang telah bersedia memberi bantuan dan saran sehingga saya dapat menyelesaikan skripsi ini dengan baik.

1. Purtojo, S.T., M.Sc sebagai dosen pembimbing pertama yang telah meluangkan waktu dan berjasa dalam memberikan ilmu, membimbing, mengarahkan, mendukung, dan memberikan kemudahan kepada saya selama proses penyusunan skripsi ini. Semoga Allah memberikan ridhonya, *Aamiin ya Rabbal'alamin*.
2. Santo Ajie Dhewanto, S.T., M.M. yang telah bersedia memberikan waktunya untuk menguji dan memberikan arahan pada saya dalam penyusunan skripsi ini.
3. Dr. Muhammad Khafidh, S.T., M.T. yang telah bersedia memberikan waktunya untuk menguji dan memberikan arahan pada saya dalam penyusunan skripsi ini.

4. Teman seperjuangan angkatan Teknik Mesin 2016, semoga kita dipertemukan kembali dalam kondisi yang jauh lebih baik dan tetap membanggakan almamater UII.
5. Faza Hadiyan, terima kasih untuk waktu yang telah diluangkan dalam proses penyusunan skripsi ini. Semoga semua kebaikan yang telah dilakukan mendapat balasan yang setimpal.
6. Achmad Ramdhani Gamestya Muchlis, terima kasih telah menjadi teman yang baik, selalu ada baik suka maupun duka. Menjadi tempat berkeluh kesah, semoga kebaikanmu dibalas dengan kebaikan yang setimpal.
7. Semua pihak yang tidak dapat dituliskan satu persatu, terimakasih telah membentuk saya.
8. *Last but not least, I wanna thank me. I wanna thank me for believing in me. I wanna thank me or doing all this hard work. I wanna thank me for having no days off. I wanna thank me for, for never quitting. I wanna thank me for always being a giver. And tryna give more than I receive. I wanna thank me for tryna do more right than wrong. I wanna thank me for just being me at all times.*

Saya berharap kepada semua pihak yang membaca naskah skripsi ini untuk bisa memberikan kritik dan saran sebagai perbaikan dikemudian hari. Semoga Allah SWT memberikan ridho-Nya dan semoga skripsi ini dapat bermanfaat bagi para pembaca.

***Wassalamu'alaikum Warrahmatullahi Wabarakatuh***

Yogyakarta, 29 Juli 2020

Penulis,



Aliy Nurfiryal

**PEMODELAN DAN PERANCANGAN SISTEM KENDALI ROTARY  
PENDULUM DENGAN WAKTU DISKRIT MENGGUNAKAN**

**SCILAB/XCOS**

**Aliy Nurfiryal**

**ABSTRAK**

*Berkembangnya industri 4.0 menjadikan semua teknologi sensor, interkoneksi dan analisis data saling terintegrasi terutama dalam mengendalikan objek kendali. Salah satu objek kendali yang memiliki beberapa permasalahan yaitu rotary pendulum. Pada 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 sistem waktu diskrit. Penggunaan metode kendali sistem waktu diskrit bertujuan untuk meminimalisir pergerakan rotasi yang disebabkan oleh motor DC sehingga nilai output bisa hampir mencapai nilai input yang diinginkan. Kesimpulan dari penelitian ini adalah mendapatkan bahwa sistem kendali waktu diskrit ini sudah bisa pergerakan rotasi yang disebabkan oleh motor DC sehingga nilai output yang dihasilkan hampir mencapai nilai input yang diinginkan.*

*Kata kunci: kendali ayunan, rotary pendulum, diskrit*

# **MODELING AND DESIGN OF DISCRETE TIME ROTARY PENDULUM CONTROL SYSTEM USING SCILAB/XCOS**

Aliy Nurfiryal

*The development of industry 4.0 makes all sensor technologies, interconnections and data analysis integrated with each other, especially in controlling control objects. One of the control objects that has several problems is the rotary pendulum. In this final project, conducting research related to control control to reduce swing due to 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, a swing pendulum, and two encoder sensors. The encoder sensor is used to track changes in the position of the DC motor and pendulum. The control system method used is a discrete time system control method. The use of the discrete time system control method aims to minimize the rotational movement caused by the DC motor so that the output value can almost reach the desired input value. The conclusion of this study is to find that this discrete time control system is capable of rotational movement caused by a DC motor so that the resulting output value almost reaches the desired input value.*

*Keywords: swing control, rotary pendulum, discrete*

## DAFTAR ISI

Halaman Judul .....	i
Lembar Pengesahan Dosen Pembimbing .....	ii
Lembar Pengesahan Dosen Penguji .....	iii
Halaman Persembahan .....	iv
Halaman Motto .....	v
Kata Pengantar atau Ucapan Terima Kasih .....	vi
Abstrak .....	viii
Daftar Isi .....	x
Daftar Tabel .....	xii
Daftar Gambar .....	xiii
Daftar Notasi .....	xiv
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 Sistem Kendali Diskrit .....	6
2.3 <i>Rotary Pendulum</i> .....	6
2.3.1 <i>Pendulum</i> .....	8
2.3.2 <i>Rotary Arm</i> .....	8
2.3.3 Motor DC .....	9
2.3.4 <i>Rotary Encoder</i> .....	10
2.3.5 Driver Motor .....	10
2.3.6 Arduino .....	11
2.4 Model Matematika .....	12

Bab 3 Metode Penelitian .....	16
3.1 Alur Penelitian.....	16
3.2 Peralatan dan Bahan .....	16
3.3 Perancangan.....	23
3.3.1 Perancangan Perangkat Keras .....	23
3.3.2 Perancangan Perangkat Lunak.....	28
Bab 4 Hasil dan Pembahasan .....	36
4.1 Hasil perhitungan Model Matematis .....	36
4.1.1 Hasil Fungsi Transfer Lengan <i>Pendulum</i> (Motor DC).....	37
4.1.2 Hasil Fungsi Transfer <i>Pendulum</i> Ayun .....	38
4.1.3 Simulasi fungsi transfer alat <i>rotary pendulum</i> .....	39
4.2 Hasil Pengujian.....	41
4.2.1 Pengujian tanpa Kendali Diskrit.....	42
4.2.2 Pengujian kendali diskrit manual tuning .....	46
4.3 Pembahasan .....	52
Bab 5 Penutup.....	57
5.1 Kesimpulan.....	57
5.2 Saran atau Penelitian Selanjutnya .....	58
Daftar Pustaka .....	59

## DAFTAR TABEL

Tabel 3-1 Tabel perbandingan prototipe <i>rotary inverted pendulum</i> .....	28
Tabel 4-1 <i>Problem data</i> alat <i>rotary pendulum</i> .....	36
Tabel 4-2 Hasil <i>steady state</i> dari alat uji dengan <i>input</i> 1,57 radian.....	53
Tabel 4-3 Hasil <i>settling time</i> dari alat uji dengan <i>input</i> 1,57 radian .....	53
Tabel 4-4 Hasil <i>steady state</i> dari alat uji dengan <i>input</i> 3,14 radian.....	54
Tabel 4-5 Hasil <i>settling time</i> dari alat uji dengan <i>input</i> 3,14 radian .....	54

## DAFTAR GAMBAR

Gambar 2-1 Model Skematis <i>Rotary Pendulum</i> .....	7
Gambar 2-2 (a) FBD pada lengan (b) FBD pada <i>pendulum</i> .....	7
Gambar 3-1 Alur Penelitian.....	16
Gambar 3-2 Maxon motor DC 2322 .....	17
Gambar 3-3 Sensor <i>encoder</i> .....	17
Gambar 3-4 <i>Pendulum</i> .....	18
Gambar 3-5 Lengan <i>pendulum</i> .....	18
Gambar 3-6 Poros penghubung. ....	19
Gambar 3-7 <i>Coupling</i> .....	19
Gambar 3-8 <i>Pulley</i> .....	19
Gambar 3-9 <i>Timing belt</i> .....	20
Gambar 3-10 <i>Bracket</i> motor. ....	20
Gambar 3-11 <i>Base rotary pendulum</i> .....	20
Gambar 3-12 Arduino Mega 2560. ....	21
Gambar 3-13 <i>Driver</i> motor IC L298. ....	21
Gambar 3-14 <i>Power supply</i> 24 VDC.....	21
Gambar 3-15 Kabel <i>jumper</i> .....	22
Gambar 3-16 <i>Breadboard</i> .....	22
Gambar 4-1 Rancangan <i>pendulum</i> ayun (Sutejo, 2020).....	38
Gambar 4-2 Skema simulasi fungsi transfer pada Xcos.....	40
Gambar 4-3 Grafik motor DC fungsi transfer. ....	40
Gambar 4-4 Grafik <i>pendulum</i> fungsi transfer. ....	41

## DAFTAR NOTASI

PID	: <i>Proportional, integral dan derivative</i>	
PD	: <i>Proportional dan derivative</i>	
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)
$\omega$	: Kecepatan motor	(rad/s)
$\omega_n$	: Frekuensi natural	
$\zeta$	: Konstanta <i>damping</i>	

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan pada era industri 4.0 membuat teknologi interkoneksi, sensor dan analisis data menjadi terintegrasi di dalam berbagai bidang industri. Peran yang dilakukan manusia pada industri 4.0 sangat sedikit terutama dalam pengendalian dan pengoperasian mesin. Sistem kendali pada saat berkembangnya industri 4.0 berperan besar dan akan digunakan dalam mengendalikan mesin otomatis. Tuntutan serta kebutuhan yang banyak dalam pengendalian objek kendali mendorong banyak penelitian mengenai perancangan sistem kendali.

Penelitian yang bertema sistem kendali dalam beberapa tahun terakhir ini banyak dilakukan, salah satu contohnya adalah sistem kendali *pendulum*. Hal ini terjadi karena *pendulum* memiliki permasalahan yang cukup kompleks. Jenis *pendulum* yang digunakan adalah *rotary pendulum*, untuk mekanisme dari *pendulum* adalah *rotary arm* yang bergerak sesuai dengan lintasannya yang melingkar untuk mempertahankan posisi tegak *pendulum*.

Pada proses pembuatan *pendulum* terdapat berbagai tahapan agar respon objek kendali bisa sesuai. Pertama dengan membuat model matematis dari *pendulum*. Kemudian, memproses model matematis dan menyambungkan dengan objek kendali dengan perangkat lunak. Grafik respon objek kendali dapat menampilkan permasalahan yang harus dikendalikan melalui kontroler. Ketiga adalah perancangan kontroler yang diperlukan sesuai dengan permasalahan berdasarkan grafik respon dari objek kendali. Terakhir mengendalikan *pendulum* dengan kontroler yang sudah dirancang menggunakan perangkat lunak. Perangkat lunak yang digunakan adalah Scilab dan Xcos. Perangkat lunak ini dipilih karena sci-lab x-cos ini merupakan *open-source application*. Sehingga perangkat lunak ini dapat diakses oleh semua termasuk mahasiswa.

Dalam hal pengambilan data dari objek kendali yang bisa diambil dari *output* yang ada. Pengambilan data pada penelitian ini menggunakan sistem waktu diskrit, karena pada sistem kontrol diskrit hanya memiliki satu nilai pada *range*

tertentu. Jenis variabel diskrit yang umum adalah biner yang berarti memiliki 2 kemungkinan yaitu *on* atau *off*, terbuka atau tertutup. Pengambilan data menggunakan sistem waktu diskrit ini berguna untuk pengambilan data yang lebih akurat karena sistem diskrit merupakan sistem yang perubahannya dalam waktu terpisah seperti dalam sistem manufaktur dimana bagian datang dan keluarnya di waktu yang spesifik, seperti ketika mesin hidup dan mati lagi di waktu spesifik. Pada pemodelan simulasi sistem diskrit biasanya disebut juga dengan model simulasi sistem kejadian-diskrit (*Discrete-Event Simulation Model, DES Model*). Variabel waktu dan keadaan merupakan dua variabel penting yang digunakan untuk mendeskripsikan karakteristik model dari objek kendali. Penelitian yang menerapkan diskrit ini sudah banyak seperti Pemodelan dan Simulasi Sistem Industri Manufaktur Menggunakan Metode Simulasi *Hybrid* (Pasirulloh & Suryani, 2017).

Tujuan dari penelitian adalah menganalisis objek kendali dengan merancang model sistematis dengan menggunakan sistem kendali diskrit dengan parameter pendukung yang ada dan menerapkan sistem kendali diskrit tersebut pada prototipe *rotary pendulum*.

## **1.2 Rumusan Masalah**

Berikut adalah rumusan masalah penelitian terkait pemodelan dan perancangan sistem kendali *rotary pendulum* dengan waktu diskrit menggunakan Scilab/Xcos:

1. Bagaimana cara memilih simulasi objek kendali *rotary pendulum*?
2. Bagaimana cara merancang model sistematis objek kendali dengan penerapan sistem kendali waktu diskrit pada mesin *rotary pendulum*?
3. Bagaimana cara mengimplementasikan sistem kendali waktu diskrit pada objek kendali *rotary pendulum*?
4. Bagaimana mengevaluasi kinerja kontroler pada sistem kendali waktu diskrit?

### **1.3 Batasan Masalah**

Berikut adalah batasan masalah untuk penelitian pemodelan dan perancangan sistem kendali *rotary pendulum* dengan waktu diskrit menggunakan Scilab/Xcos:

1. Alat uji yang digunakan adalah alat *rotary pendulum* yang telah dirancang oleh peneliti dengan referensi penelitian tentang “Pemodelan dan Pembuatan Prototipe Sistem Kendali Posisi *Rotary Inverted Pendulum*” (Sutejo, 2020).
2. Metode kendali yang digunakan adalah sistem waktu diskrit.
3. Penelitian yang dilakukan sampai dengan implementasi sistem kendali waktu diskrit pada alat *rotary pendulum*.
4. Perangkat lunak yang digunakan adalah Scilab versi 5.5.2 dan Arduino IDE.
5. Pemrograman Arduino menggunakan *arduino toolbox v3*.
6. Arduino mega 2560 digunakan sebagai mikrokontroler.
7. Terdapat dua *encoder* yang digunakan, masing-masing memiliki 100 pulsa per revolusi dan 400 pulsa per revolusi.
8. Motor DC yang digunakan memiliki kecepatan 570 rpm.

### **1.4 Tujuan Penelitian atau Perancangan**

Tujuan dari penelitian ini adalah:

1. Mampu mengimplementasikan sistem kendali waktu diskrit pada objek kendali *rotary pendulum*.
2. Mengetahui kinerja objek kendali *rotary pendulum* menggunakan simulasi pada scilab Xcos.

## **1.5 Manfaat Penelitian atau Perancangan**

Manfaat dari penelitian ini adalah untuk membandingkan kinerja objek kendali ketika menggunakan sistem kendali waktu diskrit dengan kinerja objek kendali yang tidak menggunakan sistem kendali waktu diskrit. Mampu untuk menerapkan pembelajaran yang telah didapat pada masa kuliah. Penelitian ini juga bisa dipergunakan untuk referensi pembelajaran perangkat lunak Scilab Xcos untuk simulasi sistem kendali waktu diskrit sebagai sistem kendalinya.

## **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 x-cos.

### **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 mengenai pemodelan sistem kendali diskrit pada *rotary pendulum* yang spesifik dalam pembuatan prototipe dan model matematis belum banyak dilakukan. Tetapi, untuk penelitian mengenai pengendalian *rotary pendulum* sudah banyak dilakukan. Penelitian yang sudah dilakukan tersebut rata-rata memiliki *output* perancangan sistem kendali *rotary pendulum* dengan berbagai metode pengendalian.

Referensi dalam proses pemodelan sistem kendali diskrit pada *rotary pendulum* berdasarkan penelitian yang memiliki alur pemodelan sama. Alur pemodelan dari *rotary pendulum* itu sendiri sebagai berikut:

1. Pembuatan skema untuk pergerakan dan diagram bebas *rotary pendulum*.
2. Pembuatan persamaan berdasarkan diagram benda bebas pada *rotary pendulum* menggunakan persamaan *Euler-Lagrange*.
3. Mendapatkan persamaan non linier untuk *rotary pendulum* dan melinierisasikan dengan asumsi sistem berada pada keadaan *equilibrium*.
4. Mendapatkan persamaan fungsi transfer berdasarkan persamaan yang telah dilinierisasikan.

Pada penelitian yang dilakukan oleh (Muntari & Nurhadi, 2013) membahas mengenai desain sistem kendali *rotary inverted pendulum* dengan *sliding-PID* (kontrol *Proportional, Integral, Derivative*). Alur untuk pemodelan *rotary pendulum* pada penelitian tersebut mempunyai kesamaan dengan alur pemodelan di atas, *output* untuk pemodelan adalah fungsi transfer. Perancangan mengenai *rotary inverted pendulum* menggunakan *control PID* dilakukan oleh (Nusantoro et al., 2012) memiliki alur pemodelan *rotary pendulum* yang sama juga. Pemodelan dari *rotary pendulum* pada penelitian tersebut menggunakan *output state space*. Sedangkan penelitian mengenai implementasi *model references adaptive systems* (MRAS) untuk kestabilan pada *rotary inverted pendulum* dilakukan oleh

(Anyakrawati, 2015). Alur penelitian tersebut mempunyai kesamaan dengan alur pemodelan di atas, *output* dari pemodelan yang digunakan adalah *state space*.

Berdasarkan tiga penelitian tersebut, secara garis besarnya mempunyai kesamaan pada alur pemodelan untuk *rotary pendulum*, tetapi pada *output* pemodelannya berbeda-beda. Pembahasan pada tiga penelitian tersebut cenderung pada pengendalian dengan metode kontroler yang berbeda-beda, pada prototipe sendiri tidak dibahas secara detail. Sehingga, mengacu pada tiga penelitian tersebut maka penelitian mengenai *rotary pendulum* yang dilakukan lebih spesifik pada pembuatan dan pemodelan sistem kendali waktu diskrit pada *rotary pendulum* dengan referensi alur diatas. Keluaran dari pemodelan sistem kendali waktu diskrit pada *rotary pendulum* adalah fungsi transfer dan *state space*. Tahap verifikasi dari prototipe dilakukan dengan simulasi dan pemeriksaan kinerja prototipe.

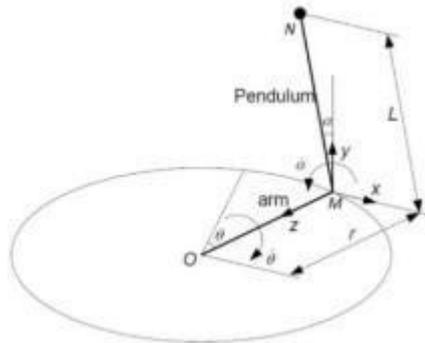
## **2.2 Sistem Kendali Diskrit**

Sistem diskrit merupakan sistem di mana perubahannya sebagian besar terputus-putus. Pemodelan sistem diskrit, variabel berstatus hanya berubah pada satu *set* titik waktu yang terpisah. Model simulasi dianalisis dengan numerik daripada dengan metode analitik. Metode analitik menggunakan penalaran deduktif matematika untuk menyelesaikan model. Misalnya: Kalkulus diferensial dapat digunakan untuk menentukan kebijakan biaya minimum untuk beberapa model persediaan. Metode numerik menggunakan prosedur komputasi dan 'berjalan', yang dihasilkan berdasarkan asumsi model dan pengamatan dikumpulkan untuk dianalisis dan untuk memperkirakan ukuran kinerja sistem yang sebenarnya. Simulasi dunia nyata sangat luas, yang dijalankan dengan bantuan komputer. Banyak *insight* yang dapat diperoleh dengan simulasi secara manual yang dapat diterapkan untuk sistem kecil (Banks et al., 1984).

## **2.3 Rotary Pendulum**

Sistem *rotary pendulum* merupakan *rotating arm* yang digerakkan oleh motor dan sebuah batang *pendulum*, dipasang pada tepi *rotating arm*. Dalam

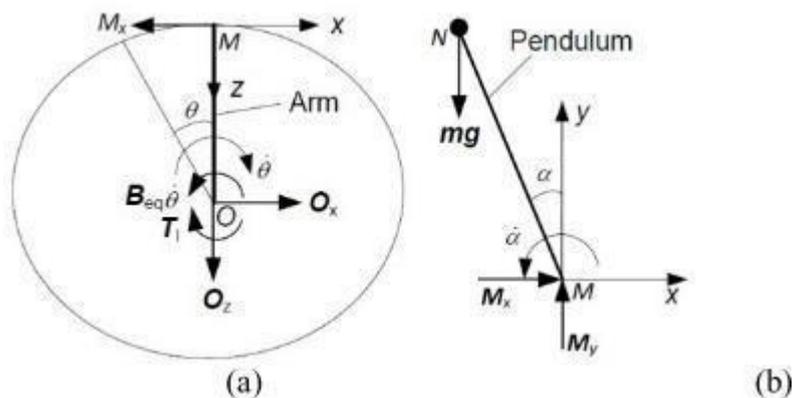
kehidupan sehari-hari prinsip ini sering digunakan, contohnya sistem pada alat pelempar bola *baseball* seperti ditunjukkan pada gambar 2-1 dibawah ini:



Gambar 2-1 Model Skematis Rotary Pendulum

Secara umum gambar model skematis dari system *rotary pendulum* dalam penelitian ini ialah sebagai berikut:

Sistem *rotary pendulum* terdiri dari *vertical pendulum* dan *horizontal arm*. Pusat *horizontal arm* terhubung dengan motor, sedangkan ujung dari *horizontal arm* terhubung dengan vertikal *pendulum*.  $\alpha$  dan  $\theta$  merupakan koordinat umum untuk menggambarkan sistem *rotary pendulum*. *Pendulum* dilokasikan dengan memberikan  $\alpha$  dengan sudut rotasi lengan  $\theta$  (Muntari & Nurhadi, 2013). *Free body diagram* dapat dilihat dalam gambar 2-2 dibawah ini:



Gambar 2-2 (a) FBD pada lengan (b) FBD pada pendulum

### 2.3.1 Pendulum

*Pendulum* adalah suatu benda yang dikaitkan dengan seutas tali dan beban yang ditempatkan diujung tali. Pada umumnya ada dua jenis *pendulum* yaitu *pendulum* biasa (*direct pendulum*) dan *pendulum* terbalik (*Inverted pendulum*). *Inverted pendulum* memiliki dua jenis pergerakan dalam pengendaliannya, yaitu *rotary inverted pendulum* dan *linear inverted pendulum*. *Rotary inverted pendulum* memiliki pergerakan rotasi atau melingkar. Sedangkan, *Linear rotary inverted pendulum* bergerak secara *linear* atau lurus, seperti ditunjukkan pada gambar 2-3 di bawah ini.



Gambar 2-3 Perbedaan linear *pendulum* dan *rotary pendulum*

Perbedaan ini mengakibatkan pemilihan *encoder*, *rotary inverted pendulum* menggunakan *rotary encoder* dan *linear inverted pendulum* menggunakan *linear encoder*. Masing-masing jenis *inverted pendulum* tersebut memiliki kelebihan dan kelemahan tersendiri (Sutejo, 2020).

### 2.3.2 Rotary Arm

*Rotary Arm* merupakan lengan penggerak dari *rotary inverted pendulum*. *Pendulum* pada dasarnya tidak stabil secara alamiah, agar *inverted pendulum* bisa tegak dan stabil maka harus secara aktif diseimbangkan. Metode untuk menyeimbangkan *inverted pendulum* adalah dengan cara memberikan gaya pada lengan *pendulum* secara rotasional dengan menggunakan sistem kendali umpan balik, seperti ditunjukkan pada gambar 2-4 di bawah ini (Sutejo, 2020).



Gambar 2-4 *Rotary arm*

### 2.3.3 Motor DC

Motor DC (*Direct Current*) merupakan perangkat elektromekanik dasar yang bertujuan untuk mengkonversi tenaga listrik menjadi tenaga mekanik. Motor DC adalah jenis motor yang memanfaatkan tegangan searah sebagai sumber tenaganya. Dengan memberikan beda tegangan pada kedua terminal tersebut, motor akan berputar pada satu arah, dan bila polaritas dari tegangan tersebut dibalik maka arah putaran motor akan terbalik pula. Polaritas dari tegangan yang diberikan pada dua terminal menentukan arah putaran motor sedangkan besar dari beda tegangan pada kedua terminal menentukan kecepatan motor, seperti ditunjukkan pada gambar 2-5 di bawah ini (Septerina, 2019).



Gambar 2-5 Motor DC

### 2.3.4 Rotary Encoder

*Encoder* atau penyandi merupakan rangkaian digital yang mempunyai fungsi yang berlawanan dengan rangkaian dekoder. Rangkaian *encoder* mempunyai sejumlah masukan yang pada suatu saat hanya ada satu masukan yang boleh aktif. Keluaran enkoder ini adalah *bit* jamak terkode yang akan dibangkitkan tergantung pada masukan yang diaktifkan. Enkoder merupakan rangkaian logika yang berfungsi mengubah data yang ada pada inputnya menjadi kode-kode biner pada *outputnya*, seperti ditunjukkan pada gambar 2-6 di bawah ini (Raihana, 2019).



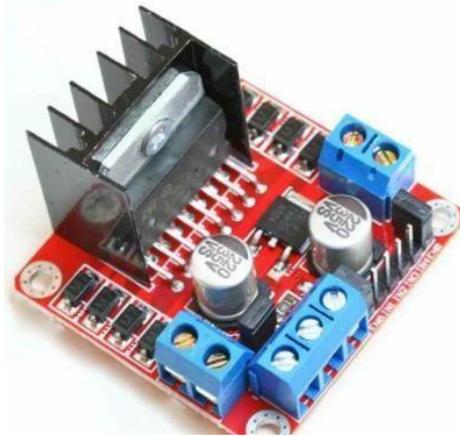
Gambar 2-6 Rotary Encoder

*Rotary encoder* adalah perangkat elektromekanik yang dapat memonitor gerakan dan posisi. Secara umum *rotary encoder* menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah. Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh *rotary encoder* untuk diteruskan oleh rangkaian kendali. *Rotary encoder* umumnya digunakan pada pengendalian robot, motor *driver*, dan sebagainya (Wahyudi, 2015).

### 2.3.5 Driver Motor

*Driver* IC L298 digunakan untuk mengontrol arah putaran motor DC. Satu buah L298 dapat digunakan untuk mengontrol arah putaran dua buah motor DC. IC *driver* L298 mempunyai kemampuan menggerakkan motor DC sampai arus 2A dan tegangan maksimum 40 *volt* DC untuk satu kanalnya. Mengendalikan jalan atau kecepatan motor diperuntungkan untuk *pin enable* A dan B. *Pin input* 1

sampai 4 digunakan untuk mengendalikan arah putaran, sedangkan *pin* output 13 dihubungkan ke motor DC yang sebelumnya melalui diode yang disusun secara *H-bridge*, seperti ditunjukkan pada gambar 2-7 di bawah ini (Sutejo, 2020b).

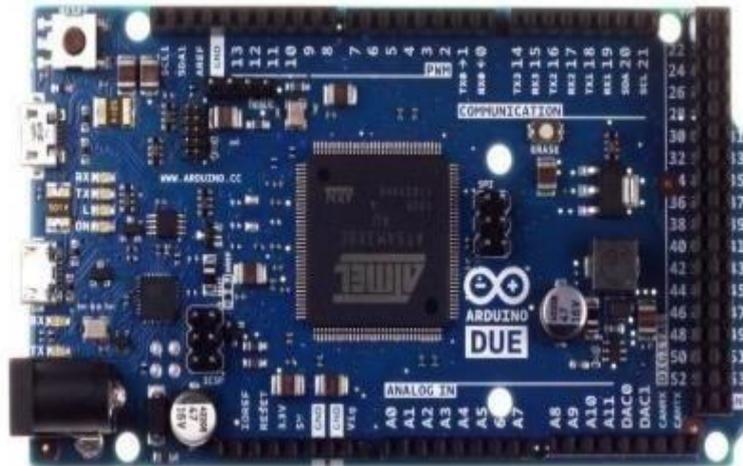


Gambar 2-7 Motor *Driver* L298N

### 2.3.6 Arduino

Mikrokontroler merupakan sebuah *chip* yang berfungsi sebagai pengontrol rangkaian elektronik yang terdiri dari CPU (*Central Processing Unit*), memori, I/O (*input/output*), bahkan sudah dilengkapi dengan ADC (*Analog-to-Digital Converter*) yang sudah terintegrasi di dalamnya. Kelebihan utama dari mikrokontroler adalah tersedianya RAM (*Random Access Memory*) dan peralatan I/O pendukung sehingga ukuran *board* mikrokontroler menjadi sangat ringkas.

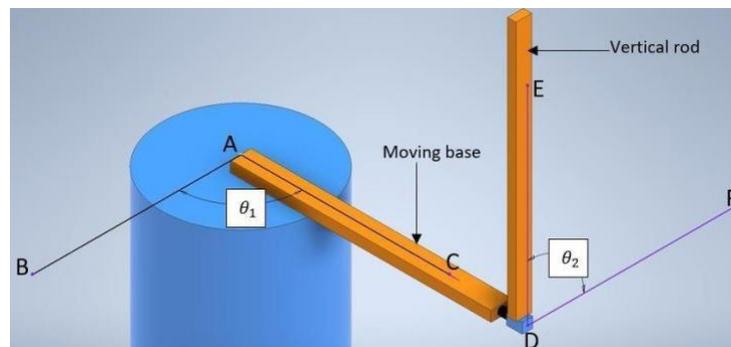
Arduino adalah sebuah *platform prototyping open-source hardware* yang dapat digunakan untuk membuat proyek berbasis pemrograman. *Hardware Arduino* memiliki prosesor mikrokontroler ATmega yang dirilis oleh Atmel AVR, tetapi *software* yang digunakan memiliki bahasa pemrograman tersendiri. Arduino dirancang untuk memudahkan pengguna elektronik atau siapapun yang ingin mengembangkan peralatan elektronik interaktif berdasarkan *hardware* dan *software* yang fleksibel dan mudah untuk digunakan, seperti ditunjukkan pada gambar 2-8 di bawah ini (Junaidi & Prabowo, 2018).



Gambar 2-8 Arduino IDE

## 2.4 Model Matematika

Perumusan model matematis dari konsep *rotary pendulum* didapatkan dengan melakukan ilustrasi mekanisme pergerakan objek kendali. Ilustrasi tersebut dapat dilihat pada gambar 2-9 di bawah ini.



Gambar 2-9 Batas Sudut dan Mekanisme Gerak *Rotary Inverted Pendulum*

Pendekatan model matematis untuk konsep *rotary pendulum* dapat menggunakan pendekatan dengan rumus Newton (Muntari & Nurhadi, 2013) dan rumus *Euler-Lagrange* (Venturin & Poles). Untuk kasus *rotary pendulum* pada penelitian kali ini, pendekatan yang digunakan merupakan pendekatan *Euler-Lagrange*. Pendekatan *Euler-Lagrange* menggunakan konsep energi potensial dan energi kinetik. Dari penggunaan energi potensial dan energi kinetik akan didapatkan rumus non-linier untuk posisi dan kecepatan sudut dari kedua batang

*pendulum* (Sutejo, 2020). Berikut dijabarkan persamaan energi kinetik dan energi potensial kedua batang pendulum:

1. Batang lengan pendulum

a. Energi kinetik:

$$T = \frac{1}{2} J \dot{\theta}^2 \quad (2.1)$$

b. Energi potensial:

$$U_1 = 0 \quad (2.2)$$

2. Batang pendulum vertikal

a. Energi kinetik

$$T = \frac{1}{2} J_1 \dot{\theta}_1^2 + \frac{1}{2} M_2 (l_1 \dot{\theta}_1)^2 + \frac{1}{2} M_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos \theta_2 \quad (2.3)$$

b. Energi potensial

$$U = M_2 g \frac{l_2}{2} \cos \theta_2 \quad (2.4)$$

Persamaan (2.1), (2.2), (2.3), dan (2.4) dimasukkan ke dalam perumusan sistem lagrangian yang dituliskan sebagai berikut:

$$L = T - U = (T_1 + T_2) - (U_1 + U_2) \quad (2.5)$$

Sehingga didapat:

$$L = \frac{1}{2} J_1 \dot{\theta}_1^2 + \frac{1}{2} J_2 \dot{\theta}_2^2 + \frac{1}{2} M_2 (l_2 \dot{\theta}_1)^2 + \frac{1}{2} M_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos \theta_2 - M_2 g \frac{l_2}{2} \cos \theta_2 \quad (2.6)$$

Persamaan gerak objek kendali *rotary pendulum* didapatkan menggunakan turunan dari persamaan *Euler-Lagrange*, persamaan itu dituliskan menjadi:

$$\frac{d}{dt} \left( \frac{dL}{dq} \right) - \frac{dL}{dq} = Q \quad (2.7)$$

Keterangan :

- $\theta_1$  = Posisi sudut batang lengan *pendulum* (rad)
- $\dot{\theta}_1$  = Kecepatan sudut batang lengan *pendulum* (rad/s)
- $\theta_2$  = Posisi sudut batang *pendulum* vertikal (rad)
- $\dot{\theta}_2$  = Kecepatan sudut batang *pendulum* vertikal (rad/s)

Dimana  $\frac{dL}{dq}$  merupakan sistem energi kinetik dan  $-\frac{dL}{dq}$  merupakan sistem energi potensial. Dari kalkulasi persamaan turunan (2.6) didapat sistem non-linier untuk persamaan gerak objek kendali *pendulum* sebagai berikut:

$$\begin{cases} (J_1 + M_2 l_1^2) \ddot{\theta}_1 + \frac{1}{2} M_2 l_1 l_2 \ddot{\theta}_2 \cos \theta_2 - \frac{1}{2} M_2 l_1 l_2 \dot{\theta}_2^2 \sin \theta_2 = r \\ J_2 \ddot{\theta}_2 + \frac{1}{2} M_2 l_1 l_2 \ddot{\theta}_1 \cos \theta_2 - \frac{1}{2} M_2 g l_2 \sin \theta_2 = 0 \end{cases} \quad (2.8)$$

Nilai torsi ( $r$ ) dihasilkan dari motor DC yang memutar batang lengan *pendulum*. Keluaran dari motor DC dikontrol oleh masukan voltase (V) sehingga persamaan dari nilai torsi dapat ditulis sebagai berikut:

$$r = NK_\phi I = NK_\phi \left( \frac{V}{R} - K_\phi \theta_1 \right) = NK_\phi \frac{V(t)}{R} - \frac{N^2 K_\phi^2}{R} \theta_1 \quad (2.9)$$

Untuk menyederhanakan penulisan substitusi persamaan (2.8) dan (2.9), digunakan notasi sebagai berikut:

$$\begin{aligned} a &= J_1 + M_2 \times l_1^2 \\ b &= \frac{1}{2} M_2 \cdot l_1 \cdot l_2 \\ c &= J_2 \\ d &= \frac{1}{2} M_2 \cdot g \cdot l_2 \\ e &= \frac{N \cdot K_\phi}{R} \\ f &= \frac{N^2 \cdot K_\phi^2}{R} \\ \delta &= ac - b^2 \end{aligned}$$

Keterangan :

- $J_1$  = Momen inersia batang lengan *pendulum* ( $\text{kgm}^2$ )
- $l_1$  = Panjang batang lengan *pendulum* (m)
- $M_2$  = Massa batang *pendulum* vertikal (kg)
- $l_2$  = Panjang batang *pendulum* vertikal (m)
- $J_2$  = Momen inersia batang *pendulum* vertikal ( $\text{kgm}^2$ )
- $K_\phi$  = Torsi motor
- $N$  = Rasio *gear*
- $R$  = Hambatan resistor (Ohm)

Sehingga sistem non-linier dari persamaan (2.8) dan (2.9) dapat dituliskan menjadi:

$$\begin{cases} a \cdot \ddot{\theta}_1 + b \cdot \ddot{\theta}_2 \cos \theta_2 - b \cdot \dot{\theta}_2^2 \sin \theta_2 = e \cdot V(t) - f \cdot \theta_1 \\ c \cdot \ddot{\theta}_2 + b \cdot \ddot{\theta}_1 \cos \theta_2 - d \cdot \sin \theta_2 = 0 \end{cases} \quad (2.10)$$

Persamaan non-linier objek kendali *rotary pendulum* (2.10) dapat dilinearisasikan dengan asumsi:

$$\theta_1 = \theta_2 = \dot{\theta}_1 = \dot{\theta}_2 = 0$$

$$\sin \theta \approx \theta \text{ dan } \cos \theta \approx 1$$

Asumsi diatas memberikan batasan dimana posisi dan kecepatan sudut kedua batang *pendulum* sama dengan 0. Hal itu memberikan nilai  $\cos \theta$  mendekati 1 dan nilai  $\sin \theta$  mendekati  $\theta$ . Hal Persamaan matriksnya dituliskan sebagai berikut:

$$\begin{bmatrix} J_1 & \frac{1}{2} M_2 l_1 l_2 \\ \frac{1}{2} M_2 l_1 l_2 & J_2 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix} \quad (2.11)$$

Dari persamaan (2.11) akan didapat matriks *state space* yang ditunjukkan pada persamaan (2.12).

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{c}{\delta} & -\frac{d}{\delta} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{\delta} \\ 0 \end{bmatrix} V(t) \quad (2.12)$$

Persamaan fungsi transfer didapatkan dengan mengaplikasikan transformasi laplace pada sistem non-linier objek kendali *rotary pendulum* dengan torsi sebagai variabel kontrol. Berikut adalah fungsi transfer pada persamaan non-linier objek kendali *rotary pendulum* (Venturin & Poles).

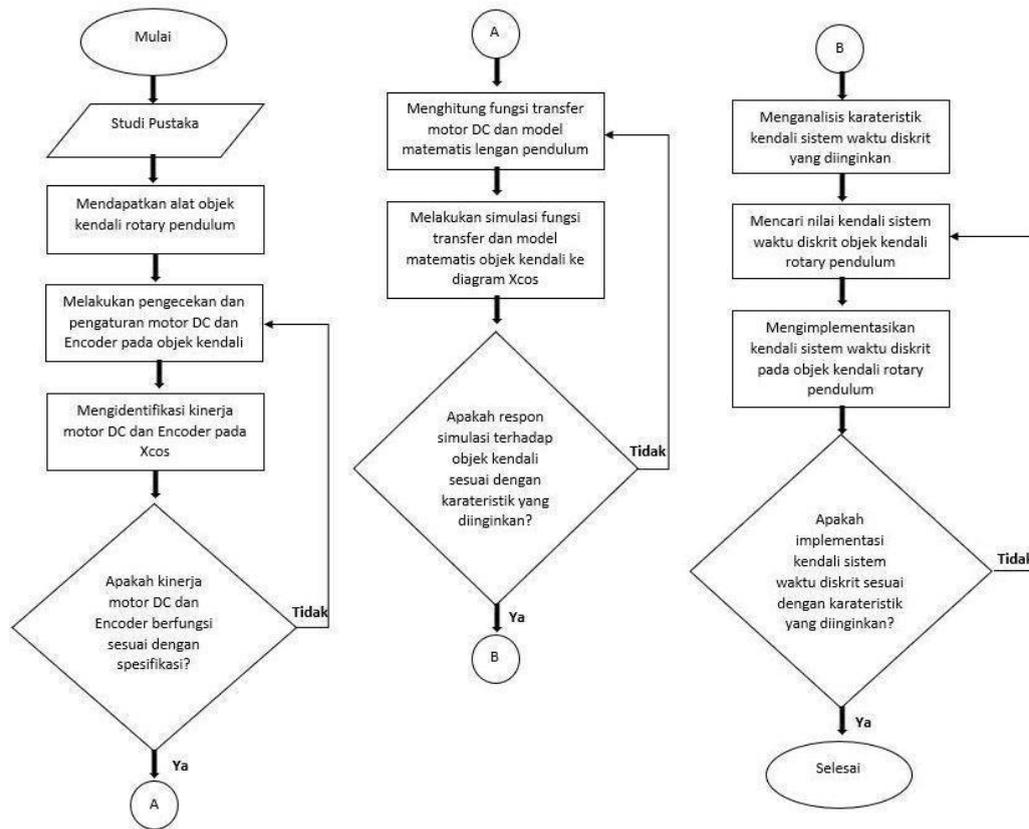
$$\omega_1(s) = \frac{\theta_1(s)}{c(s)} = \frac{cs^2 - d}{as^2 \left[ \left( \frac{b}{a} \right) s^2 - d \right]} = \frac{-d + c \frac{\delta}{\delta}}{\frac{ad}{\delta} \frac{2}{\delta} s^2 - \frac{d}{\delta} s + s^4} \quad (2.13)$$

$$\omega_2(s) = \frac{\theta_2(s)}{c(s)} = \frac{-b}{\left( \frac{b}{a} \right) s^2 - d} = \frac{-b}{\frac{da\delta}{\delta} + s^2} \quad (2.14)$$

# BAB 3

## METODE PENELITIAN

### 3.1 Alur Penelitian



Gambar 3-1 Alur Penelitian

### 3.2 Peralatan dan Bahan

Peralatan yang digunakan dalam melakukan penelitian ini ada dua macam, yaitu peralatan perangkat keras dan perangkat lunak.

### 3.2.1. Perangkat Keras

Perangkat keras yang digunakan untuk merancang objek kendali *rotary pendulum* ini diantaranya adalah:

1. Motor DC Maxon tipe 2322 diameter 22 mm sebagai *actuator* dari *rotary inverted pendulum*, seperti pada gambar 3-2 di bawah ini.



Gambar 3-2 Maxon motor DC 2322

2. Dua *rotary encoder* dengan merk Autonics dengan 100 pulsa per revolusi dan Orange dengan 600 pulsa per revolusi, seperti pada gambar 3-3 di bawah ini.



Gambar 3-3 Sensor *encoder* Autonics dan Orange.

3. *Pendulum* dengan material aluminium sebagai objek yang akan diamati, seperti pada gambar 3-4 di bawah ini.



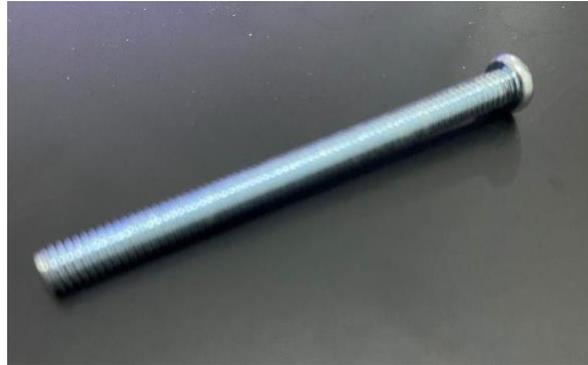
Gambar 3-4 *Pendulum*.

4. Lengan *pendulum* sebagai media untuk penggerak antar motor DC ke *pendulum*, seperti pada gambar 3-5 di bawah ini.



Gambar 3-5 Lengan *pendulum*.

5. Poros motor ke lengan *pendulum* dan *pendulum*, seperti pada gambar 3-6 di bawah ini.



Gambar 3-6 Poros penghubung.

6. *Coupling* sebagai sambungan antara poros satu dengan lainnya, seperti pada gambar 3-7 di bawah ini.



Gambar 3-7 *Coupling*.

7. *Pulley* dengan diameter 5 mm sebagai penghubung antar motor DC dengan *rotary encoder*, seperti pada gambar 3-8 dibawah ini.



Gambar 3-8 *Pulley*.

8. *Timing belt* GT2 dengan lebar 6 mm dan panjang 158 mm sebagai penghubung antara motor DC dengan *rotary encoder*, seperti pada gambar 3-9 di bawah ini.



Gambar 3-9 *Timing belt*.

9. *Bracket* motor DC diameter *shaft* 200 mm, seperti pada gambar 3-10 di bawah ini.



Gambar 3-10 *Bracket* motor.

10. Base objek kendali *rotary pendulum* dengan material akrilik dengan tebal 10 mm, seperti pada gambar 3-11 di bawah ini.



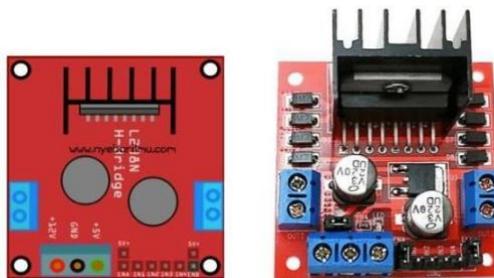
Gambar 3-11 *Base rotary pendulum*.

11. Arduino Mega 2560 sebagai mikrokontroler, seperti gambar 3-12 di bawah ini.



Gambar 3-12 Arduino Mega 2560.

12. *Driver* motor IC L298 untuk mengatur arah dan kecepatan putaran motor DC, seperti gambar 3-13 di bawah ini.



Gambar 3-13 *Driver* motor IC L298.

13. *Power supply* (input 220 VAC, output 24 VDC) sebagai penyuplai tegangan motor dan *driver* motor, seperti gambar 3-14 di bawah ini.



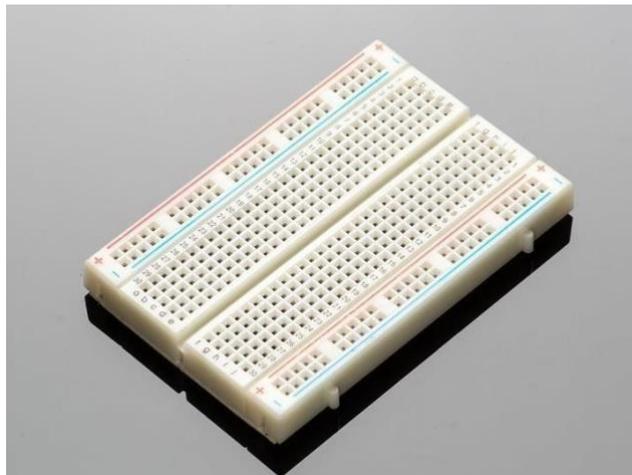
Gambar 3-14 Power supply 24 VDC.

14. Kabel *jumper* sebagai penghubung, seperti pada gambar 3-15 di bawah ini.



Gambar 3-15 Kabel *jumper*.

15. *Breadboard* sebagai penghubung dan pencabang antar rangkaian, seperti pada gambar 3-16 di bawah ini.



Gambar 3-16 *Breadboard*.

### 3.2.2. Perangkat Lunak

1. Arduino IDE digunakan untuk mengunggah program *toolbox V3*. Program *Toolbox V3* ini merupakan program yang digunakan untuk menghubungkan mikrokontroler Arduino ke perangkat lunak yang dimana pada penelitian ini menggunakan Scilab 5.5.2 Xcos.
2. Scilab 5.5.2 Xcos berfungsi untuk memodelkan sistem kontrol dari objek kendali *rotary pendulum*. Fitur yang dimiliki oleh Scilab ini beberapa

digunakan untuk berbagai keperluan, salah satunya adalah Xcos. Perancangan model sistem kontrol bisa dibuat menggunakan fitur Xcos.

### 3.3 Perancangan

Perancangan yang dilakukan pada penelitian ini dibagi menjadi dua, yaitu perancangan perangkat keras dan perancangan perangkat lunak. Pada perancangan perangkat lunak berkaitan dengan *setup* dikonfigurasi perangkat keras dan koneksi dengan perangkat lunak. Sedangkan perancangan perangkat lunak akan berkaitan dengan model Scilab Xcos. Sebelum melakukan penelitian lebih lanjut, terlebih dahulu melakukan pendeteksian apakah masih terdapat kendala yang mempengaruhi performa alat objek kendali *rotary pendulum*.

#### 3.3.1 Perancangan Perangkat Keras

*Rotary inverted pendulum* memiliki perangkat keras yang dapat dibagi menjadi dua, yaitu komponen elektrik dan komponen mekanik. Konfigurasi *base* dari *rotary inverted pendulum* berdasarkan desain yang sesuai dengan referensi modul quanser seperti pada gambar 3-17 di bawah ini.



Gambar 3-17 *Rotary inverted pendulum* versi Quanser.

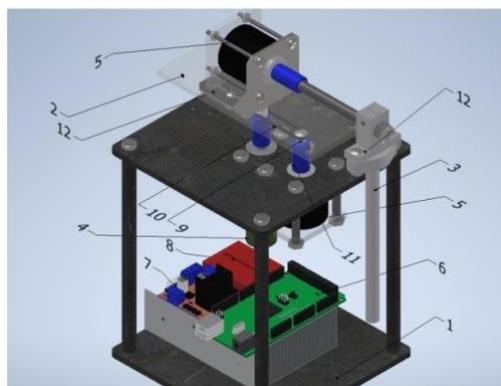
mengambil referensi untuk bentuk *base* dari gambar 3-16 di atas yaitu dimensi panjang dan lebar *base* prototipe sama seperti referensi quanser (Apkarian et al., 2012).

Dari pengamatan, komponen elektrik dari *rotary inverted pendulum* tidak boleh terpisah dari *base*, sehingga tinggi *base* ditinggikan melebihi referensi. Hal tersebut bertujuan agar prototipe lebih mudah dijalankan, cukup dengan menghubungkan *port* dari Arduino ke laptop atau komputer. Berdasarkan referensi

bentuk *base rotary inverted pendulum*, kondisi dari komponen yang ada tidak mendukung sepenuhnya untuk konfigurasi *base rotary inverted pendulum* yang sama seperti pada gambar 3-17. Hal ini menimbulkan beberapa masalah dan kendala yang terjadi, permasalahan dan kendala tersebut adalah:

1. Kebutuhan dari *encoder* dan motor DC dengan spesifikasi yang sesuai cukup susah untuk didapatkan. Komponen dengan spesifikasi yang sesuai bisa didapatkan, tetapi dengan konfigurasi motor DC dan *encoder* secara terpisah. Hal ini menyebabkan permasalahan untuk merancang posisi konfigurasi yang sesuai dengan referensi Quanser.
2. Pemilihan *bracket* standar pada komponen motor DC, *encoder* motor DC, *encoder* lengan *pendulum* dan *pendulum* hanya beberapa saja. *Bracket* standar yang tersedia untuk motor DC, *encoder* lengan *pendulum* dan *pendulum*. Untuk *bracket encoder* motor DC perlu dilakukan modifikasi atau pemanfaatan plat akrilik yang ada.
3. Poros *encoder* dan motor DC sedikit kurang panjang ketika perlu disambungkan dengan lengan *pendulum* dan *pendulum*.

Kendala dan permasalahan di atas diselesaikan dengan merancang ulang dan sedikit memodifikasi konfigurasi posisi dan membuat beberapa *part* yang dibutuhkan. Hal ini dilakukan agar bisa mendapatkan spesifikasi datasheet dari komponen utama yang diperlukan tanpa perlu mengganti. Perancangan konfigurasi *base rotary inverted pendulum* seperti desain pada gambar 3-18 di bawah ini.



Gambar 3-18 Desain *rotary inverted pendulum*.

Desain pada gambar 3-18 menunjukkan beberapa komponen pada *rotary inverted pendulum*. Komponen prototipe *rotary inverted pendulum* pada gambar 3-18 diberi keterangan sebagai berikut:

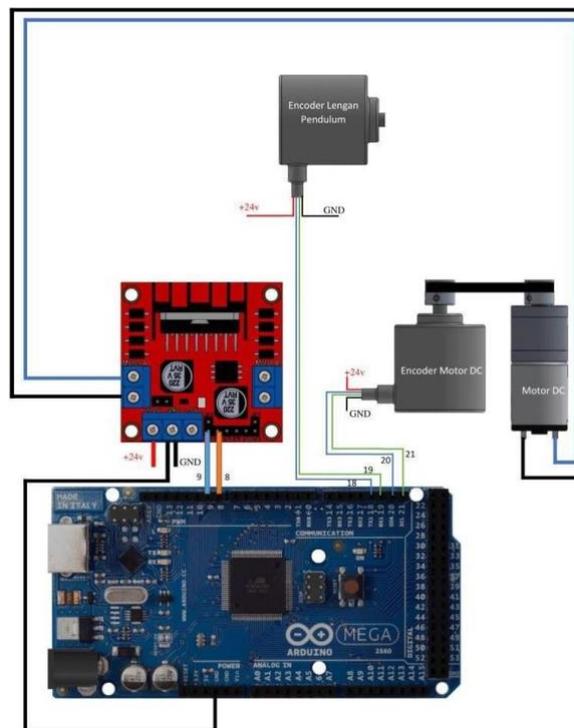
1. *Base rotary inverted pendulum.*
2. *Lengan pendulum.*
3. *Pendulum.*
4. Motor DC.
5. *Rotary encoder.*
6. Arduino mega 2560.
7. *Driver motor L298.*
8. *Bread board.*
9. *Pulley.*
10. *Belt.*
11. *Coupling.*
12. *Bracket.*

Penjelasan mengenai dimensi desain *rotary inverted pendulum* terdapat pada gambar teknik yang terlampir. Perancangan desain di gambar 3-18 dilakukan untuk menjawab kendala yang ditemukan, solusi tersebut dituangkan dalam desain yang memiliki keterangan sebagai berikut:

1. Sesuai dengan referensi bahwa motor DC sudah terhubung dalam satu poros dengan *rotary encoder*, namun komponen tersebut susah didapatkan. Komponen yang tersedia motor DC dan *rotary encoder* terpisah, tidak terhubung dalam poros yang sama. Maka motor DC dan *rotary encoder* dihubungkan menggunakan *belt* yang kemudian dikaitkan antara kedua poros menggunakan *pulley*. Menggunakan *belt* yang dikaitkan antar poros mempertimbangkan ketika pembacaan posisi oleh *rotary encoder* motor DC hasilnya lebih akurat, dikarenakan dimensi *pulley* dan *thread belt* sudah disesuaikan. Dibanding menggunakan *gear* yang mengharuskan menghitung rasio *gear*.
2. *Bracket* untuk *encoder* motor DC perlu dibuat penahan dengan memanfaatkan akrilik dengan tebal 5 mm dan baut dengan panjang 60 mm M5. *Encoder* lengan *pendulum* juga perlu dibuat penahan akrilik tebal 5 mm dan baut panjang 50 mm M3. Plat akrilik dengan ketebalan 5 mm sudah cukup untuk menahan *encoder* dengan baut yang menahannya.

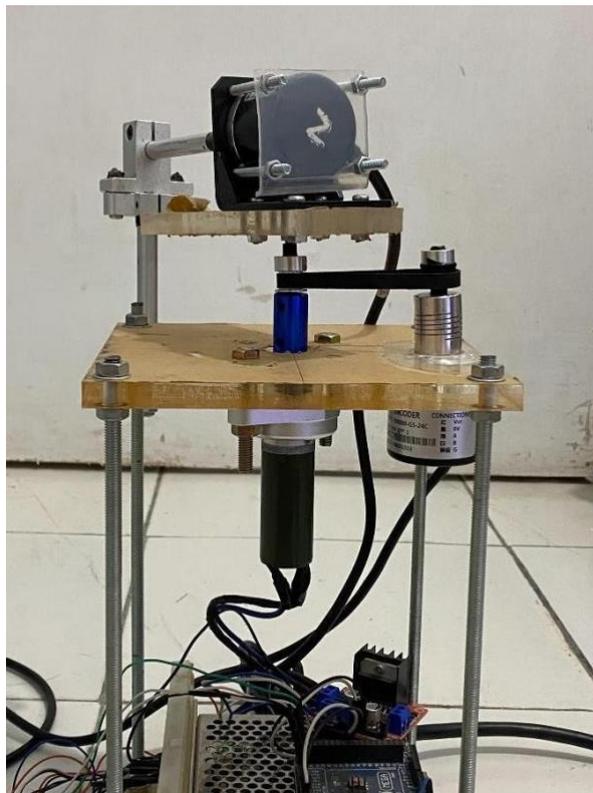
3. Poros motor DC dan *rotary encoder* kurang panjang sehingga tidak memungkinkan langsung disambung pulley, terutama komponen motor DC yang diperlukan untuk disambungkan dengan lengan pendulum. Penghubung antar poros menggunakan *coupling* sesuai dengan diameter yang dibutuhkan.
4. Konfigurasi dari lengan pendulum tidak dibuat dengan menggantung melainkan terpusat pada poros motor DC, hal ini dikarenakan *rotary encoder* pada bagian lengan *pendulum* memiliki massa yang cukup besar dan ditambah dengan lengan *pendulum*.

Tujuan dari desain selain untuk membantu melakukan perancangan ulang prototipe dari permasalahan yang ada, desain juga membantu untuk mendapatkan momen inersia lengan pendulum dan pendulum. Momen inersia digunakan dalam pemodelan *rotary inverted pendulum* untuk mendapatkan fungsi transfer. Kemudian setelah merancang komponen mekanik tersebut dilakukan perancangan komponen elektrik. Konfigurasi komponen elektrik sesuai dengan skema driver motor IC L298. Skema dari driver motor IC L298 disesuaikan dengan penggunaan komponen yang digunakan seperti pada gambar 3-19 di bawah ini.



Gambar 3-19 Rangkaian elektrik *rotary inverted pendulum*.

Rangkaian elektrik *rotary inverted pendulum* menggunakan skema dari *driver* motor IC L298 yang diterapkan pada motor DC dan dua *encoder*. Pada gambar 3-19 tanda 24V dan GND dalam rangkaian disambungkan pada *power supply input* 220 VAC, *output* 24 VDC. *Encoder* disambungkan ke Arduino pada *pin* 18-21, *encoder* pada motor DC pada *pin* 20-21 dan *encoder* lengan pendulum pada *pin* 18-19. Perancangan dari *hardware* yang terdiri dari *base* dan rangkaian elektrik digabungkan menjadi satu seperti pada gambar 3-20 dibawah ini.



Gambar 3-20 Prototipe *rotary inverted pendulum*

Berdasarkan hasil perancangan *hardware* yang dilakukan, membandingkan fungsi prototipe dengan referensi yang ada. Perbandingan tersebut dijelaskan pada tabel 3-1 di bawah ini.

Tabel 3-1 Tabel perbandingan prototipe *rotary inverted pendulum* dengan referensi

<i>Rotary inverted pendulum</i>	<b>Kelebihan</b>	<b>Kelemahan</b>
Prototipe	Komponen lebih mudah didapatkan.	Tidak dapat ditambah alat ukur lain seperti <i>tachometer</i> .
	Penggunaan Arduino membuat simulasi bisa menggunakan dua aplikasi yaitu Arduino IDE dan Scilab Xcos. Aplikasi yang mudah didapatkan ( <i>open source</i> ).	Kabel yang kurang tertata rapi, sehingga menimbulkan hambatan ketika dilakukan eksperimen berulang.
Referensi (Quanser)	Penggunaan <i>gear</i> menguntungkan untuk penambahan alat ukur seperti <i>tachometer</i> .	Komponen susah didapatkan.
	Penataan kabel rapih dan bagus, sehingga apabila dilakukan eksperimen berulang-ulang tidak terjadi hambatan.	Simulasi hanya bisa dilakukan pada satu aplikasi yaitu MATLAB, yang dimana aplikasi ini tidak gratis atau berbayar.

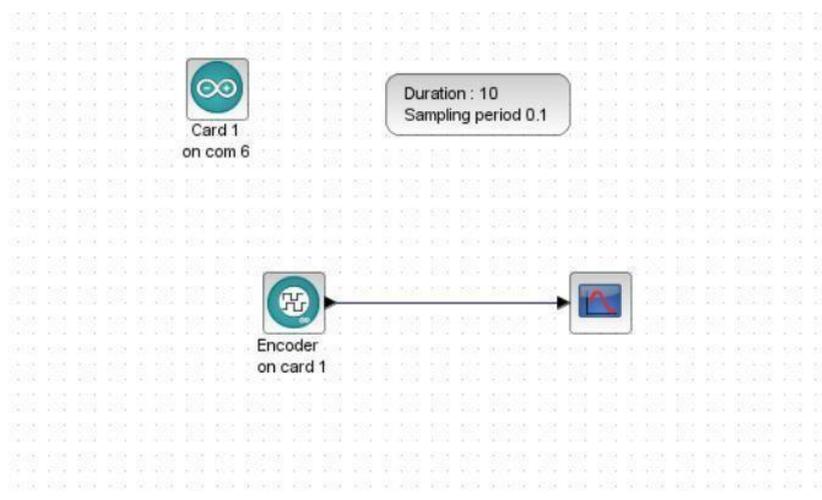
### 3.3.2 Perancangan Perangkat Lunak

Perangkat lunak yang digunakan adalah Scilab 5.5.2 dan *toolbox* Arduino 1.4 sebagai blok diagram arduino di aplikasi Scilab Xcos. Mengenai perancangan perangkat lunak Scilab terdiri dari pemeriksaan kinerja *rotary encoder* dan motor DC. Hal ini dilakukan dengan tujuan untuk memastikan bahwa komponen utama

yaitu *rotary inverted pendulum* telah terhubung dengan perangkat lunak Scilab dan bisa bekerja sesuai dengan spesifikasinya.

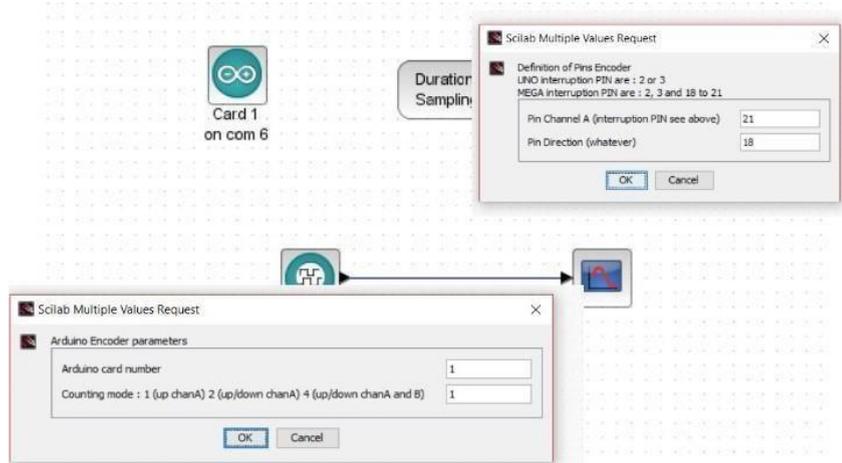
### 3.3.2.1. Pemeriksaan Kinerja *Rotary Encoder*

Pemeriksaan kinerja dari *rotary encoder* ini bertujuan untuk memastikan bahwa kinerja dari *rotary encoder* ini sudah sesuai dengan spesifikasi atau belum. Penulis menggunakan dua *rotary encoder*, yang pertama ada pada *pendulum* dan terhubung dengan motor DC. *Encoder* pertama dengan merk Autonics mempunyai kapasitas 100 pulsa per putaran, sedangkan *encoder* kedua dengan merk Orange memiliki kapasitas 600 pulsa per putaran. *Rotary encoder* pada *pendulum* berfungsi untuk mengetahui posisi radian atau sudut pada *pendulum* yang disambungkan pada *input pin* Arduino 21 dan 18. Pemeriksaan dari kinerja blok diagram *rotary encoder pendulum* dapat dilihat pada gambar 3-21 dibawah ini.



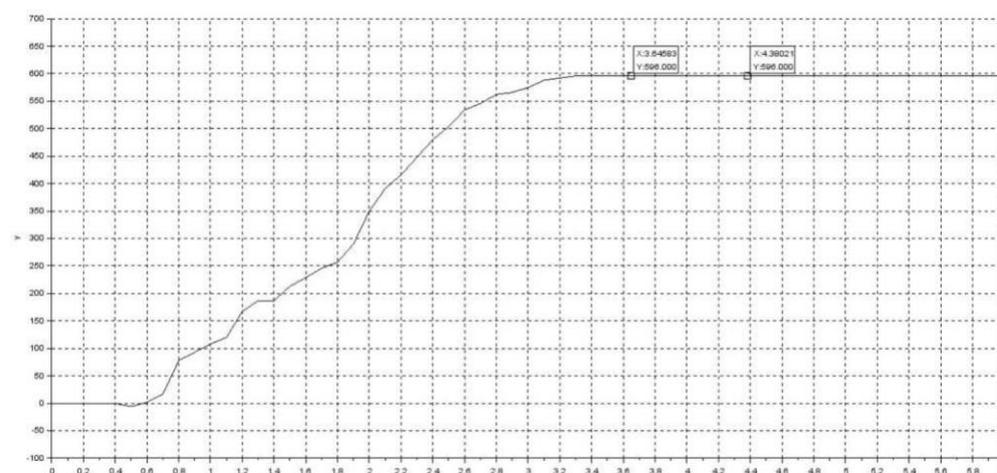
Gambar 3-21 Blok diagram pemeriksaan kinerja *rotary inverted pendulum*.

Blok diagram seperti gambar 3-20 diatas didapatkan dari *toolbox* Arduino 1.4. Setelah membuat blok diagram untuk pemeriksaan kinerja *rotary encoder* kemudian dimasukkan *pin* yang sesuai pada arduino seperti pada gambar 3-22 dibawah ini.



Gambar 3-22 Mengatur blok diagram *encoder pendulum*.

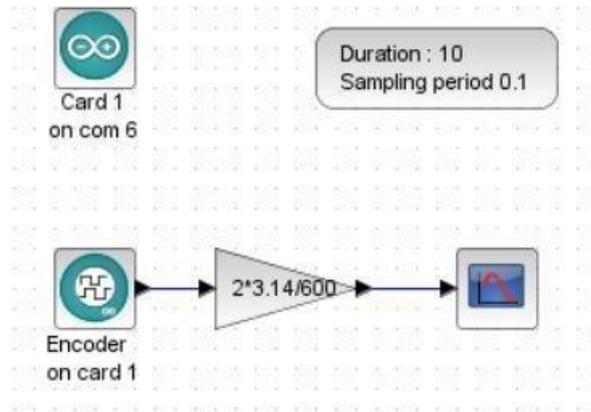
Setelah melakukan pengaturan pada blok diagram *encoder* seperti pada gambar 3-21 kemudian dilakukan eksperimen. Eksperimen dilakukan dengan menyambungkan langsung *rotary encoder pendulum* ke perangkat lunak Scilab. Dari eksperimen tersebut didapatkan hasil yang bisa dilihat dari grafik yang menunjukkan kapasitas pulsa yang terbaca seperti gambar 3-22 dibawah ini.



Gambar 3-22 Respon pemeriksaan pulsa *rotary encoder pendulum*.

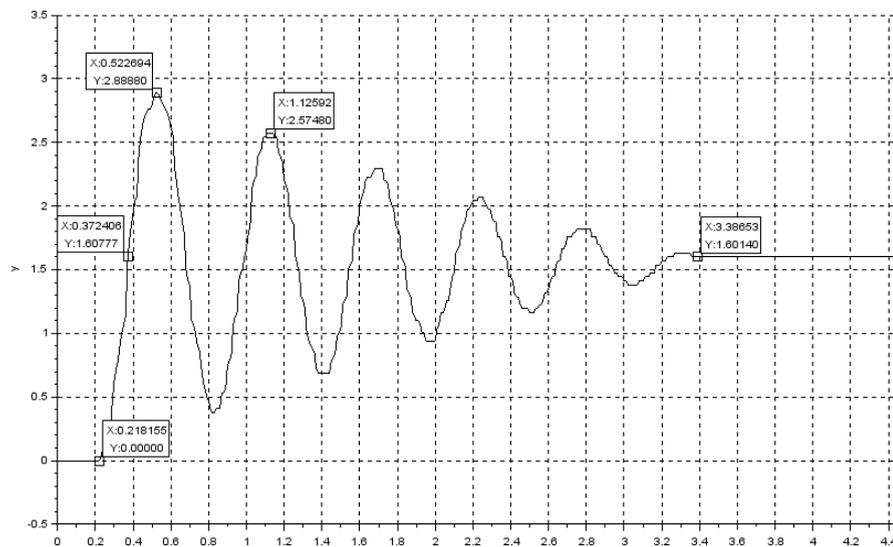
Sumbu Y menunjukkan pulsa dan sumbu X menunjukkan waktu. Respon dari kinerja *rotary encoder pendulum* ini menunjukkan pulsa yang didapat sebesar 596 per putaran. Hasilnya tetap menggunakan sesuai dengan spesifikasi yaitu 600 pulsa per putaran.

Setelah didapat nilai pulsa, selanjutnya dilakukan pemeriksaan respon osilasi dari pendulum ayun. Konfigurasi Xcos yang digunakan ditunjukkan pada Gambar 3-23.



Gambar 3-23 Konfigurasi respon osilasi pendulum ayun.

Pada saat pengujian, pendulum diayunkan secara manual sehingga pendulum berayun dengan bebas. Untuk bisa menunjukkan hasil pembacaan dalam satuan radian, hasil pembacaan sensor encoder pendulum dikalikan dengan nilai nilai  $2\pi/600$ . Kemudian hasil yang didapat dari konfigurasi, didapat grafik yang ditunjukkan oleh Gambar 3-24.



Gambar 3-24 grafik pergerakan pendulum ayun.

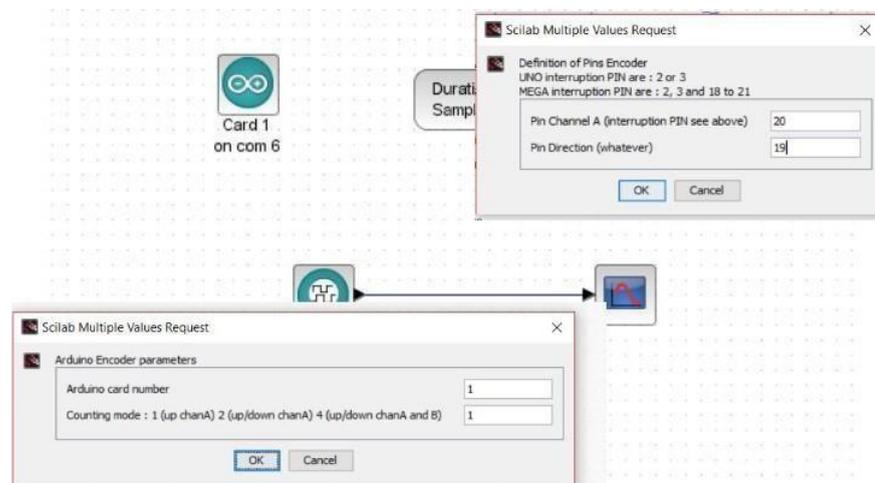
Dari hasil pergerakan pendulum ayun, didapat hasil:

- *Rise time* : 0,154 detik

- *Settling time* : 3,38 detik
- *Error SS* : -
- *Percent overshoot* : 80 % (dibandingkan oleh fase amplitudo ke-5)
- *Peak time* : 0,52 detik
- *Decay ratio* : 1,12

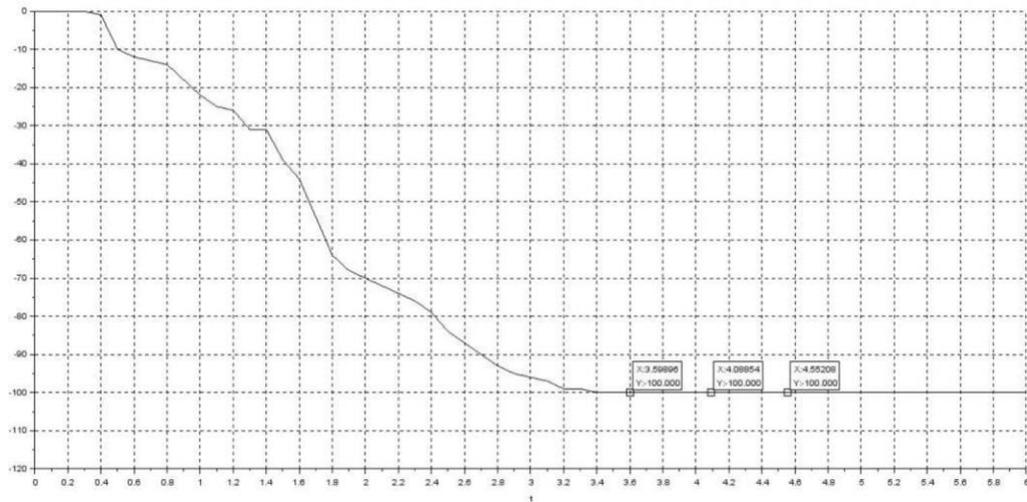
Dari respon pergerakan pendulum didapatkan informasi bahwa pendulum yang berayun bebas memerlukan waktu 3,38 detik untuk berhenti berayun. Ayunan pendulum ini akan dikendalikan agar waktu yang diperlukan agar bisa berhenti bisa dikurangi lagi.

Kemudian adalah pemeriksaan kinerja *rotary encoder* untuk motor DC, *rotary encoder* pada motor DC ini berfungsi untuk mengetahui posisi dari putaran motor DC. Pemeriksaan dari *rotary encoder* motor DC menggunakan blok diagram sama dengan gambar 3-20, tetapi yang berbeda adalah posisi *pin* pada blok diagram *encoder*. *Input pin rotary encoder* motor DC 20 dan 19 yang selanjutnya dimasukkan pada blok diagram seperti pada gambar 3-24 di bawah ini.



Gambar 3-24 Mengatur blok diagram *encoder* motor DC.

Setelah pengaturan pada blok diagram *encoder* motor DC seperti pada gambar 3-24 tersebut selanjutnya dilakukan eksperimen. Eksperimen yang dilakukan dengan menyambungkan langsung *rotary encoder* motor DC ke perangkat lunak Scilab. Respon dari eksperimen ini dapat dilihat dari grafik yang menunjukkan kapasitas pulsa yang terbaca seperti pada gambar 3-25 di bawah ini.

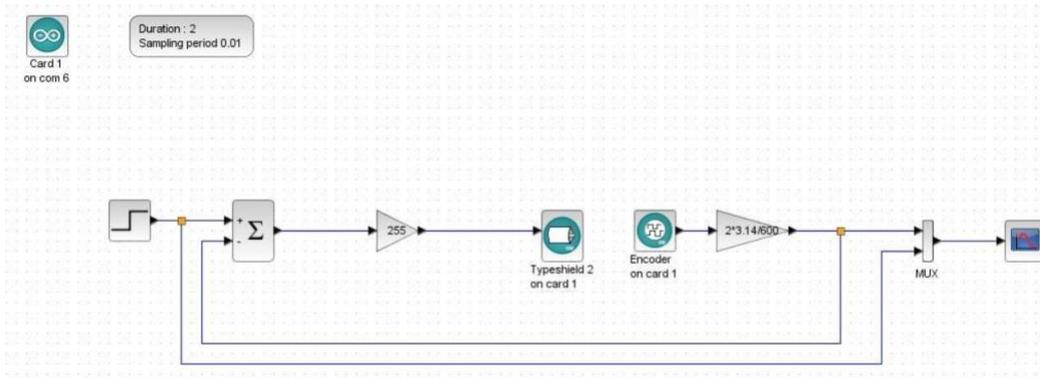


Gambar 3-25 Respon pemeriksaan pulsa *rotary encoder* motor DC.

Respon dari kinerja *rotary encoder* pada motor DC ini menunjukkan pulsa 100 per putaran. Sehingga hasil yang digunakan sesuai dengan spesifikasinya yaitu 100 pulsa per putaran, hasil dari *encoder* dengan merk Autonics ini tetap menghasilkan angka yang konstan yaitu 100 pulsa per putaran dibanding dengan merk Orange yang hasilnya mengalami perbedaan yang cukup signifikan di setiap eksperimennya.

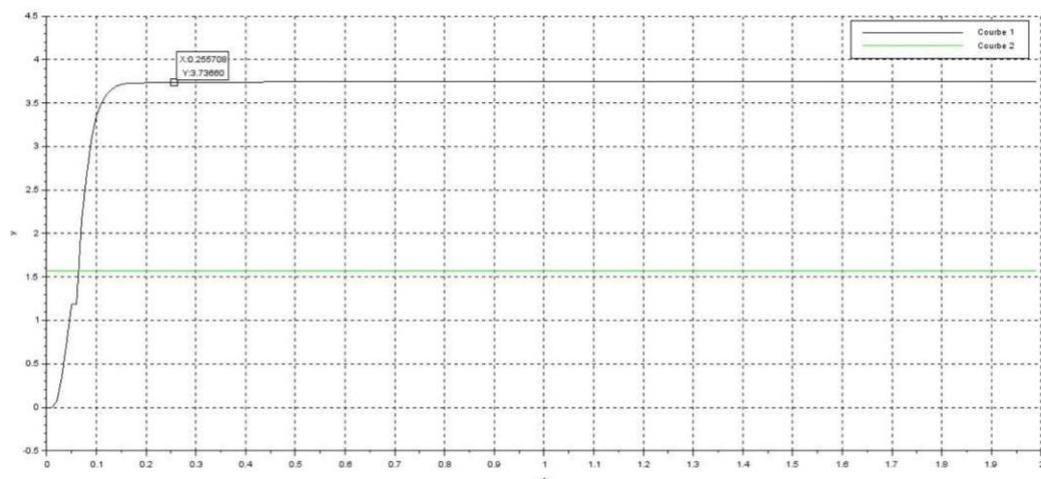
### 3.3.2.2 Pemeriksaan Kinerja Motor DC

Perancangan perangkat lunak yang selanjutnya menggunakan Scilab Xcos adalah pemeriksaan kinerja motor DC. Pemeriksaan kinerja motor DC adalah pemeriksaan respon posisi, karena parameter dari sistem kendali *rotary inverted pendulum* adalah posisi. Desain blok diagram Scilab Xcos untuk pemeriksaan kinerja motor DC dapat dilihat pada gambar 3-26 di bawah ini.



Gambar 3-26 Blok diagram pemeriksaan kinerja motor DC.

Desain blok diagram pemeriksaan kinerja motor DC menggunakan input posisi 1,57 radian atau  $90^0$  dan 3,14 radian atau  $180^0$  pada blok *step function*. Nilai *gain* PWM yaitu 255 bertujuan agar mendapatkan akselerasi yang maksimal. Pengaturan pada blok *typeshield 2 on card 1* dengan memilih tipe *driver* motor yang digunakan yaitu L298. Kemudian memasukkan *direction pin* 8 dan *enable (speed) pin* 9, hal ini harus disesuaikan dengan rangkaian elektrik pada gambar 3-18. Blok *encoder* sama seperti pada pemeriksaan kinerja *rotary encoder* motor DC pada gambar 3-23. Nilai *gain* pulsa *encoder* didapatkan dari hasil perhitungan  $2 \times 3,14/600$ , perhitungan tersebut bertujuan untuk merubah jumlah pulsa dalam satuan radian. Hasil dari respon pemeriksaan kinerja motor DC dapat dilihat pada gambar 3-27 di bawah ini.



Gambar 3-27 Respon pemeriksaan kinerja motor DC dengan *input* posisi.

Pada gambar 3-27 diatas bisa dilihat bahwa sumbu y menunjukkan posisi dalam satuan radian dan sumbu x menunjukkan waktu dalam satuan *second*. Sedangkan garis hijau menunjukkan *input* yang diinginkan dalam satuan radian. Respon kinerja motor DC yang ditunjukkan pada gambar 3-27 di atas menunjukkan posisi 3,73 radian. Hasil ini belum sesuai dengan input yang diberikan sebesar 1,57 radian, kemudian hal ini dapat diatasi dengan melakukan pengendalian menggunakan PID pada motor DC yang digunakan. Respon pada objek kendali yang diteliti memiliki parameter spesifikasi respon sistem sebagai berikut :

1. *Error steady state* = 1,98 rad/s
2. *Settling time* = 0,25 s
3. *Rise time* = 0,20 s
4. *Percent overshoot* = -
5. *Peak time* = -

## BAB 4

### HASIL DAN PEMBAHASAN

#### 4.1 Hasil Perhitungan Model Matematis

Bersumber pada formulasi model matematis pada bab 2, dikenal ada 2 guna transfer pada objek kendali *rotary pendulum*. Guna fungsi transfer yang pertama menampilkan reaksi keluaran dari motor DC ataupun lengan *pendulum*, serta guna transfer yang kedua menampilkan reaksi keluaran dari *pendulum* yang berayun. *Problem* informasi yang digunakan buat menghitung guna transfer didapat dari *datasheet* motor DC yang terdapat pada lampiran 1 dan pengukuran langsung pada objek kendali *rotary pendulum*.

Tabel 4-1 *Problem* data alat *rotary pendulum*.

No	<i>Problem Data</i>	Nilai	Sumber
1	$J_{\text{motor DC}}$	$5,68 \text{ gcm}^2$ $= 5,68 \times 10^{-7} \text{ kgcm}^2$	<i>Datasheet</i> Motor DC (dengan voltase 24V)
2	$K_{\text{tm}}$	$30,7 \text{ mNm/A}$ $= 0,0307 \text{ Nm/A}$	<i>Datasheet</i> Motor DC (dengan voltase 24V)
3	$I_m$	0,42 A	<i>Datasheet</i> Motor DC (dengan voltase 24V)
4	$\omega$	$\omega = \frac{\text{rpm datasheet}}{\text{rpm reduksi}}$ $= \frac{7190 \text{ rpm}}{504,84}$ $= 14,24 \text{ rpm}$ $= 1,49 \text{ rad/s}$	<i>Datasheet</i> 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

### 4.1.1 Hasil Fungsi Transfer Lengan *Pendulum* (Motor DC)

Bersumber pada problem informasi diatas bisa dihitung guna transfer dari motor DC, Dikenal guna transfer motor DC ditunjukkan pada persamaan (2.1).

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

Dengan anggapan kalau beban yang bertumpu pada motor DC diabaikan, sehingga nilai momen inersia memakai nilai dari *datasheet* motor. Nilai koefisien gesek (B) terlebih dulu dicari dengan rumus (2. 2).

$$B = \frac{K_{tm} \cdot I_m}{\omega}$$

Nilai konstanta torsi motor ( $K_{tm}$ ), arus motor ( $I_m$ ), serta kecepatan motor ( $\omega$ ) didapat dari *datasheet* motor DC. Nilai yang diambil dengan asumsi voltase yang diberikan ke motor DC sebesar 24 V, sehingga nilai koefisien gesekan bisa dihitung semacam pada persamaan ( 4.1).

$$B = \frac{\frac{0,03 \frac{Nm}{A} \cdot 0,42 A}{1,49 \frac{rad}{s}}}{1,49 \frac{rad}{s}} \quad (4.1)$$
$$B = 0,0084 \text{ Nm}/(\text{rad/s})$$

Kala nilai koefisien gesek telah didapat, berikutnya memastikan nilai guna transfer motor DC. Nilai guna transfer motor DC pada perlengkapan *rotary pendulum* ditunjukkan pada persamaan (4.2).

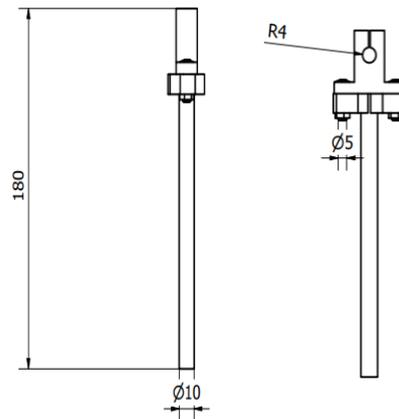
$$P(s) = \frac{1}{0,0000000568 s^2 + 0,00864 s} \quad (4.2)$$

Nilai fungsi transfer motor DC dapat disederhanakan menjadi seperti pada persamaan (4.3).

$$P(s) = \frac{1760563,38}{s^2 + 15211,27 s} \quad (4.3)$$

### 4.1.2 Hasil Fungsi Transfer *Pendulum* Ayun

Berdasarkan model matematis yang ditunjukkan pada Gambar 2-11, diketahui bahwa model ilustrasi bandul tersebut adalah tali bandul dengan beban di ujung tali. Ini berbeda dengan model *pendulum* yang ditunjukkan pada Gambar 4-1.



Gambar 4-1 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.

$$\omega_n = \frac{1}{2\pi} \sqrt{\frac{g}{l}}$$

$$\omega_n = \frac{1}{2\pi} \sqrt{\frac{9,81 \text{ m/s}^2}{0,18 \text{ m}}} \quad (4.4)$$

$$\omega_n = 1,18 \quad (4.5)$$

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 dampingnya, dimana nilai konstanta damping dapat dicari dengan menghitung rumus (2.12) dan (2.13).

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

Dimana;

$$n\delta = \ln \frac{X_i}{X_{i+n}}$$

Berdasarkan data percobaan ayunan *pendulum* secara bebas yang ditunjukkan pada Gambar 3-24. Diketahui amplitudo pada fase kelima mengalami penurunan nilai sebesar 80%, data ini dapat digunakan untuk menghitung nilai  $\delta$ . Berikut ditunjukkan perhitungan nilai  $\delta$ .

$$5\delta = \ln \frac{X_i}{(100-80)\%X_i} \quad (4.6)$$

$$5\delta = \ln \frac{X_1}{0,02X_i}$$

$$5\delta = \ln 5$$

$$\delta = \frac{1,61}{5}$$

$$\delta = 0,322$$

Sehingga nilai konstanta *damping* dapat dihitung sebagai berikut.

$$\zeta = \frac{0,322}{\sqrt{(2\pi)^2 + 0,322^2}} \quad (4.7)$$

$$\zeta = 0,05$$

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

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

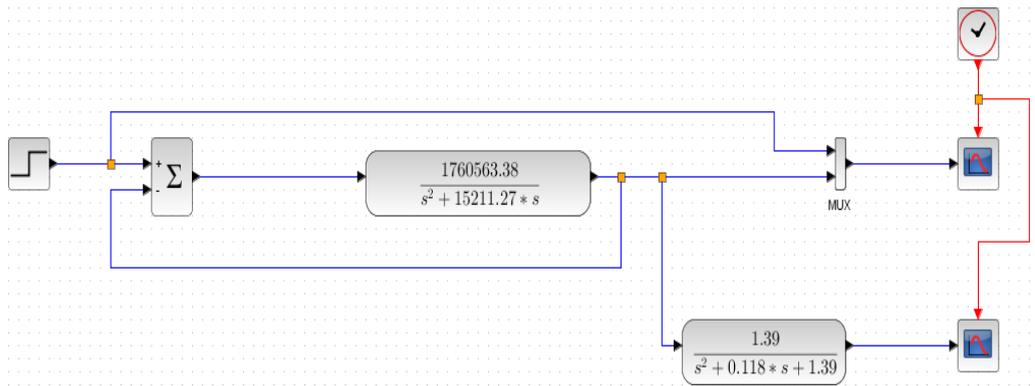
Diketahui dari perhitungan pada persamaan (4.5) frekuensi *natural* ( $\omega_n$ ) bernilai 1,18. Maka fungsi transfer *pendulum* ayun dapat dihitung menjadi.

$$H(s) = \frac{1,18^2}{s^2 + 2(0,05)(1,18)s + 1,18^2} \quad (4.8)$$

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

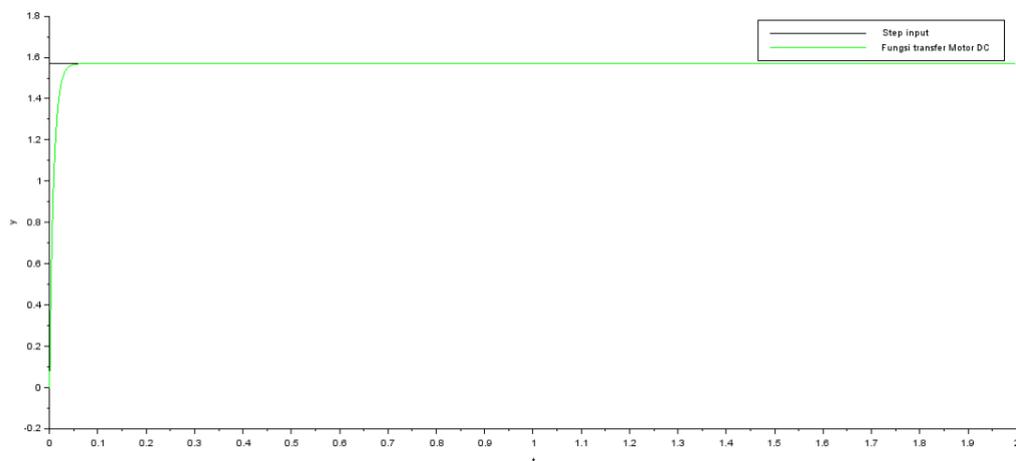
### 4.1.3 Simulasi Fungsi Transfer Alat Rotary Pendulum

Setelah didapatkan fungsi transfer dari alat *rotary pendulum*, selanjutnya dilakukan simulasi pada aplikasi Scilab Xcos. Skema simulasi digambarkan pada Gambar 4-2 di bawah ini.



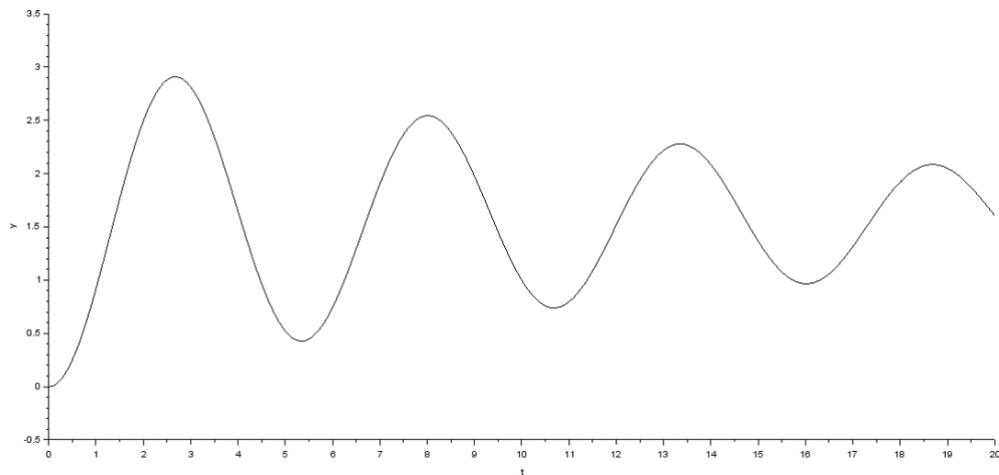
Gambar 4-2 Skema simulasi fungsi transfer pada Xcos.

*Input step* yang diberikan merupakan *input* posisi dalam radian. Posisi yang diinginkan adalah motor DC berpindah  $90^\circ$  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 fungsi transfer motor DC. Keluaran fungsi transfer motor DC selanjutnya diteruskan ke fungsi transfer *pendulum* ayun. Perpindahan posisi motor DC secara tiba-tiba akan mengayunkan *pendulum*. Selanjutnya barulah hasil keluaran fungsi transfer motor DC, fungsi transfer *pendulum*, dan input ditampilkan pada grafik. Grafik motor DC dan grafik *pendulum* ayun simulasi fungsi transfer alat *rotary pendulum* ditunjukkan pada gambar 4-3 dan gambar 4-4 di bawah ini.



Gambar 4-3 Grafik motor DC fungsi transfer.

Dari gambar diatas terlihat terdapat dua garis grafik. Garis berwarna hitam menunjukkan nilai *input* masukan dan grafik berwarna hijau menunjukkan respon 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*, respon motor DC mengalami keadaan *steady state*. Respon fungsi transfer motor DC ini sudah sesuai dengan respon implementasi motor DC secara langsung di alat *rotary pendulum*. Hasil respon fungsi transfer *pendulum* ayun ditunjukkan pada gambar 4-4 di bawah ini.



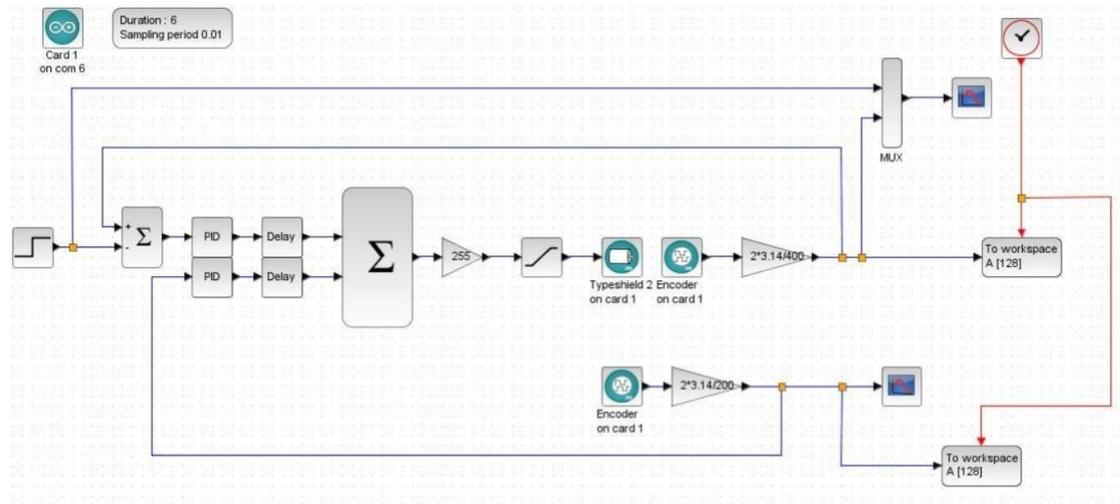
Gambar 4-4 Grafik *pendulum* fungsi transfer.

Dari gambar diatas, terlihat bahwa fungsi transfer *pendulum* ayun menunjukkan hasil osilasi yang teredam atau yang biasa disebut dengan respon *underdamped*. Secara teori, respon *underdamped* akan terus memiliki respon osilasi meskipun respon tersebut mengalami redaman. Sedangkan pada respon implementasi pendulum ayun secara langsung, pendulum berosilasi, lalu teredam dan, pada waktu tertentu akan berhenti berosilasi.

## 4.2 Hasil Pengujian

Terdapat dua tahapan dalam melakukan uji coba kendali sistem waktu diskrit pada alat *rotary pendulum*. Pada pengujian pertama, alat percobaan diuji sebelum diberikan kendali sistem waktu diskrit, yang kedua adalah melakukan pengujian menggunakan kendali sistem waktu diskrit dimana pencarian nilai sistem waktu diskrit menggunakan metode *manual tuning*.

Uji coba dilakukan menggunakan diagram atau skema Xcos. Skema yang digunakan disusun berdasarkan referensi kendali PID+PD ditunjukkan di gambar 4-5 di bawah ini (Solihin et al., 2010).

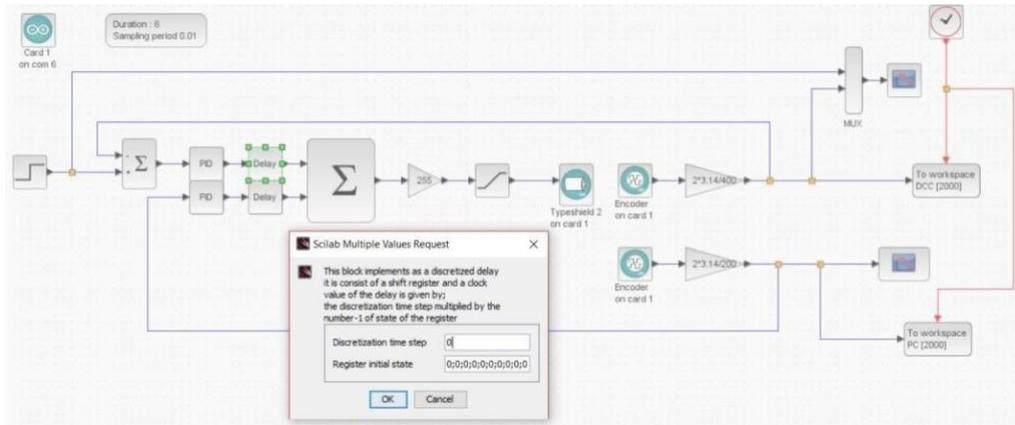


Gambar 4-5 Skema Xcos uji coba kendali diskrit pada alat *rotary pendulum*.

Skema di atas memiliki *step input* dengan nilai 1,57 rad dan 3,14 rad. Nilai untuk masukan motor DC dari PWM sebesar 255 dan dibatasi menggunakan blok saturasi. Penggunaan blok saturasi pada skema Xcos ini bertujuan agar nilai *input* PWM Motor DC tidak lebih dari 255. Motor DC pada skema ini menggunakan konfigurasi PID, sedangkan *pendulum* ayun dikendalikan menggunakan konfigurasi PD.

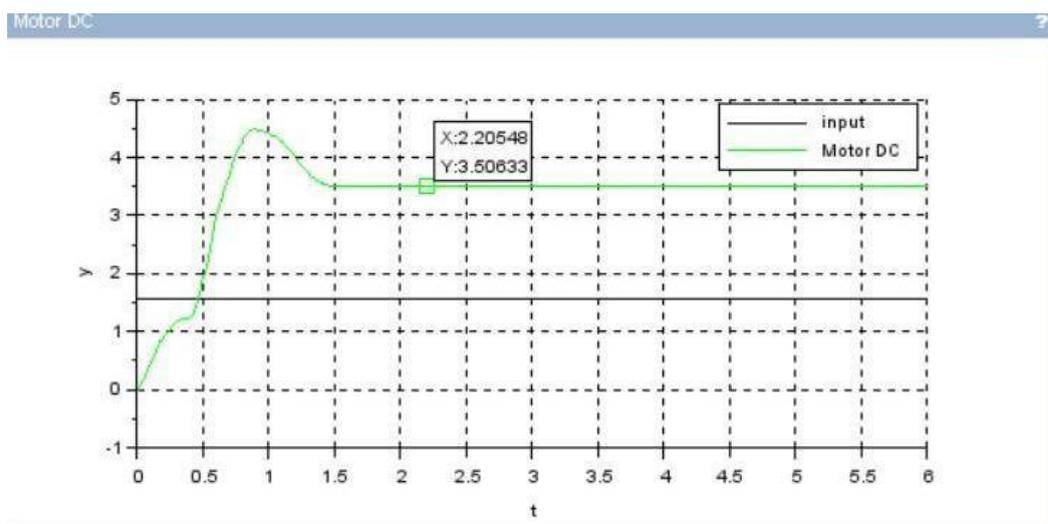
#### 4.2.1 Pengujian Tanpa Kendali Diskrit

Pengujian alat uji *rotary pendulum* dapat dilakukan dengan skema yang telah ditunjukkan pada Gambar 4-5. Caranya adalah dengan tidak memasukkan nilai konstanta *discretization time step* sedangkan *register initial state* tetap sama nilainya. Selanjutnya konstanta yang digunakan adalah konstanta “0”, konstanta “1” dan konstanta “2”. Pada percobaan pertama ini dilakukan dengan konstanta “0”. Konfigurasi ini ditunjukkan pada Gambar 4-6 di bawah ini.



Gambar 4-6 Konfigurasi Xcos tanpa kendali diskrit atau konstanta “0”

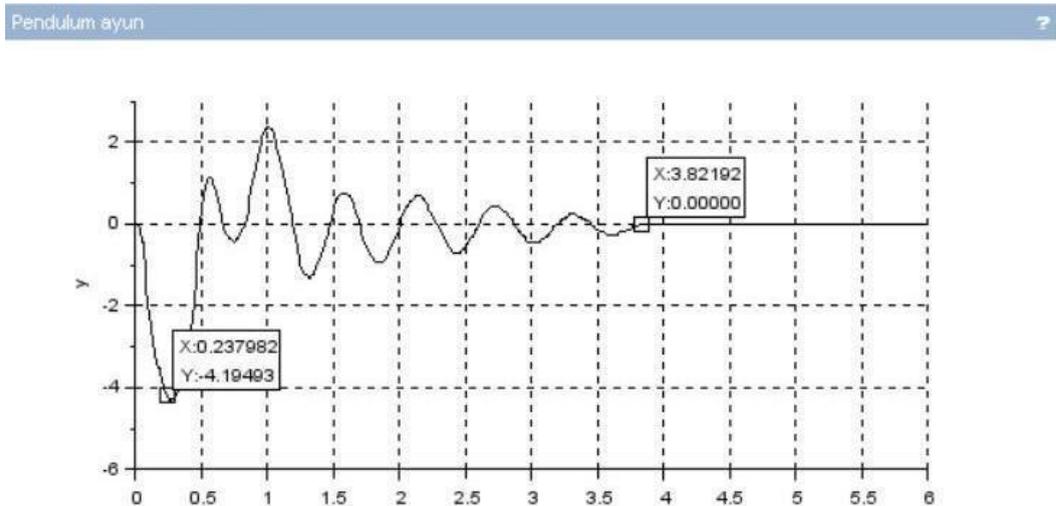
Dari skema Xcos diatas, hasil uji coba menunjukkan respon dari perpindahan posisi motor DC dan *pendulum* ayun. Masing-masing hasil ditunjukkan pada Gambar 4-7 dan Gambar 4-8 untuk *input* 1,57 radian, sedangkan untuk *input* 3,14 radian ditunjukkan pada Gambar 4-9 dan Gambar 4-10.



Gambar 4-7 Grafik respon Motor DC tanpa kendali diskrit atau konstanta “0”

Respon di atas menunjukkan pergerakan motor DC tanpa kendali diskrit. *Input* yang diberikan pada alat uji adalah 1,57 rad. Pada gambar di atas garis hitam menunjukkan nilai *input* pada alat *rotary pendulum*, sedangkan garis hijau menunjukkan respon yang aktual dari motor DC tersebut. Dari grafik di atas didapat data:

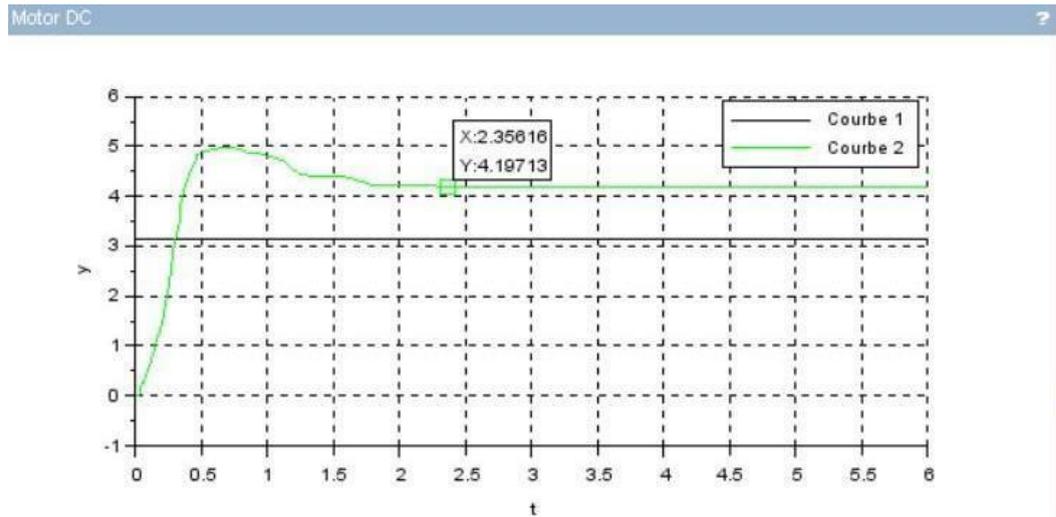
- *Rise time* : 0,6 detik
- *Settling time* : 1,5 detik
- *Error Steady State* : 3,48 rad
- *Percent Overshoot* : 221,6 %
- *Peak time* : 0,5 detik



Gambar 4-8 Grafik respon Pendulum tanpa kendali diskrit atau konstanta “0”

Respon pada grafik di atas menunjukkan ayunan terjadi oleh pendulum ketika Motor DC yang diberikan *input* masukan sebesar 1,57 radian. Pada grafik di atas didapatkan data:

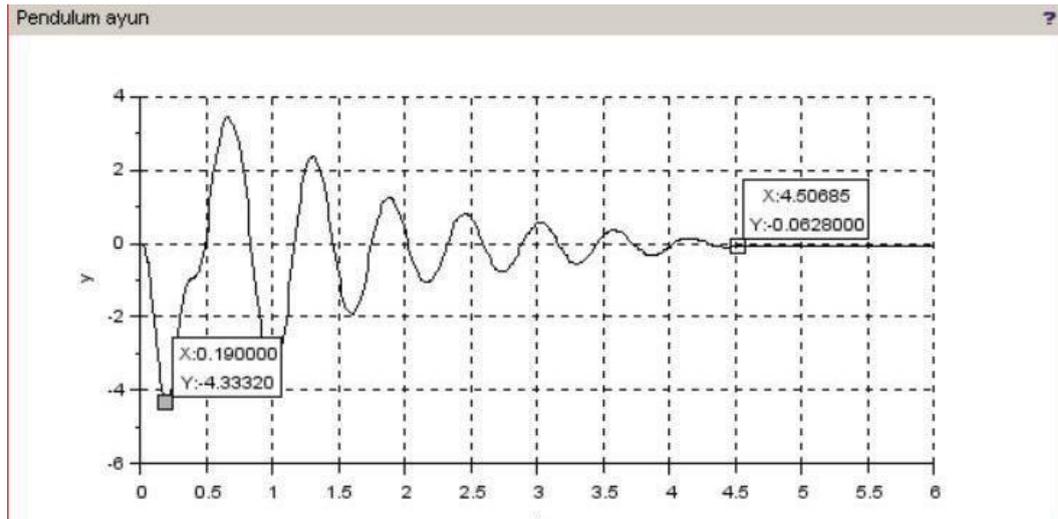
- *Peak time* : 0,23 detik
- *Settling time* : 3,82 detik
- *Decay ratio* : 0,53



Gambar 4-9 Grafik respon Motor DC tanpa kendali diskrit atau konstanta “0”

Respon di atas menunjukkan pergerakan untuk motor DC tanpa kendali diskrit. *Input* yang diberikan adalah 3,14 radian. Garis hitam pada hasil di atas menunjukkan nilai *input* masukan pada alat *rotary pendulum*, sedangkan garis hijau menunjukkan respon aktual dari motor DC. Dari grafik di atas didapatkan data:

- *Rise time* : 0,5 detik
- *Settling time* : 2,3 detik
- *Error steady state* : 1,05 rad
- *Percent overshoot* : 33,4 %
- *Peak time* : 0,5 detik



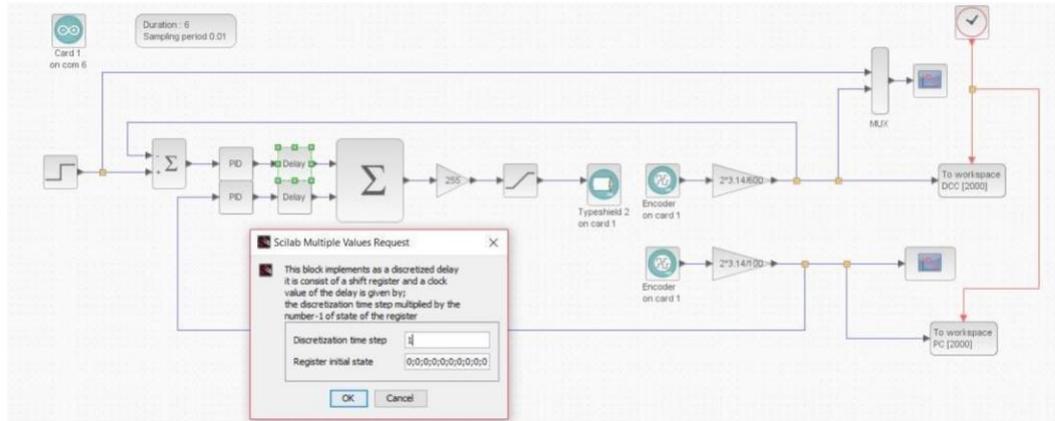
Gambar 4-10 Grafik respon *pendulum* tanpa kendali diskrit atau konstanta “0”

Respon di atas menunjukkan ayunan yang terjadi pada pendulum ketika motor DC diberikan *input* sebesar 3,14 radian. Dari grafik di atas tersebut didapatkan data:

- *Peak time* : 0,19 detik
- *Settling time* : 4,5 detik
- *Decay ratio* : 0,72

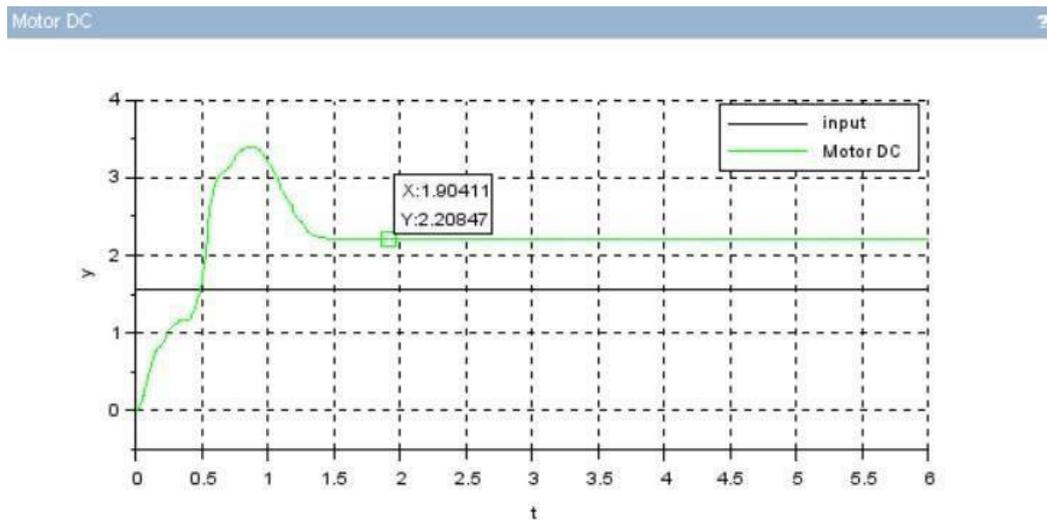
#### 4.2.2 Pengujian Kendali Diskrit *Manual Tuning*

Selanjutnya adalah melakukan pencarian nilai konstanta kendali diskrit pada pendulum ayun menggunakan metode *manual tuning*. Pada pengujian ini dilakukan percobaan dengan dua nilai *input* yaitu 1,57 rad dan 3,14 rad dan tiga nilai konstanta yaitu konstanta “0”, konstanta “1” dan konstanta “2”, tiap konstanta diuji sebanyak lima kali. Dari kelima kali percobaan itu hasil akan ditampilkan di tabel dan grafik, sedangkan di bawah akan ditampilkan contoh hasil dari masing masing percobaan. Konfigurasi untuk Xcos pengujian kali ini ditunjukkan pada gambar 4-11 di bawah ini.



Gambar 4-11 Konfigurasi pengujian kendali diskrit

Dari hasil kendali dengan konfigurasi diskrit tersebut didapat grafik yang ditunjukkan pada Gambar untuk respon motor DC dan Gambar untuk respon dari pendulum ayun untuk masing-masing konstanta dengan lima kali percobaan.

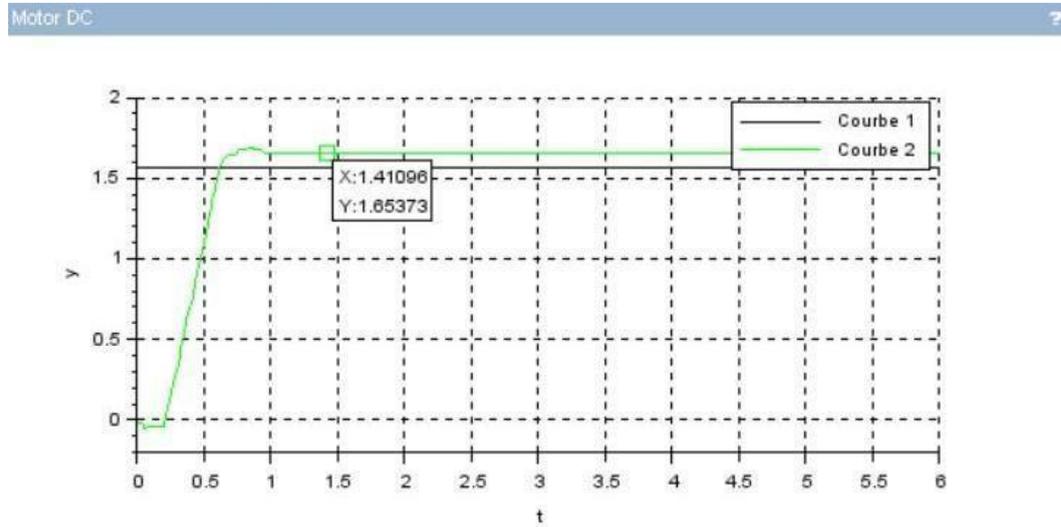


Gambar 4-12 Grafik respon Motor DC konstanta “1”

Dari gambar 4-12 di atas didapatkan data hasil respon pergerakan Motor DC dengan kendali diskrit yang diberi *input* 1,57 radian dilakukan dengan ketiga konstanta yaitu konstanta “0”, konstanta “1” dan konstanta “2”, untuk konstanta “1” didapat data sebagai berikut:

*Rise time* : 0,8 detik  
*Settling time* : 1,5 detik

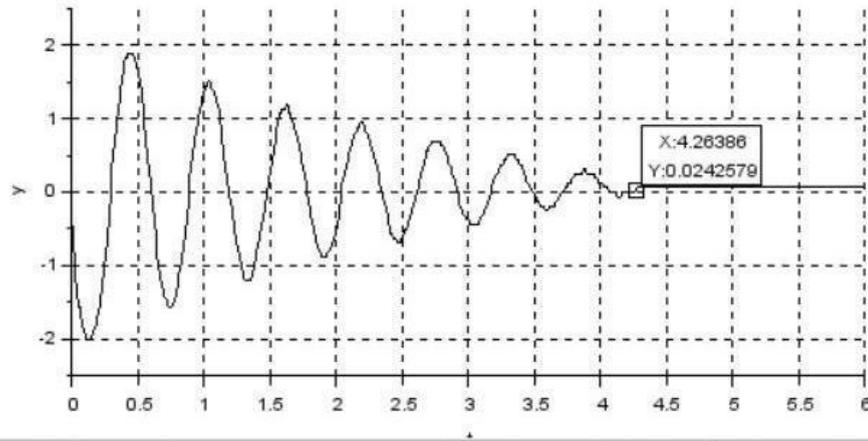
*Error steady state* : 0,63 rad  
*Percent overshoot* : 40 %  
*Peak time* : 0,5 detik



Gambar 4-13 Grafik respon Motor DC konstanta “1”

Kemudian untuk konstanta “1” pada pergerakan Motor DC didapat data sebagai berikut:

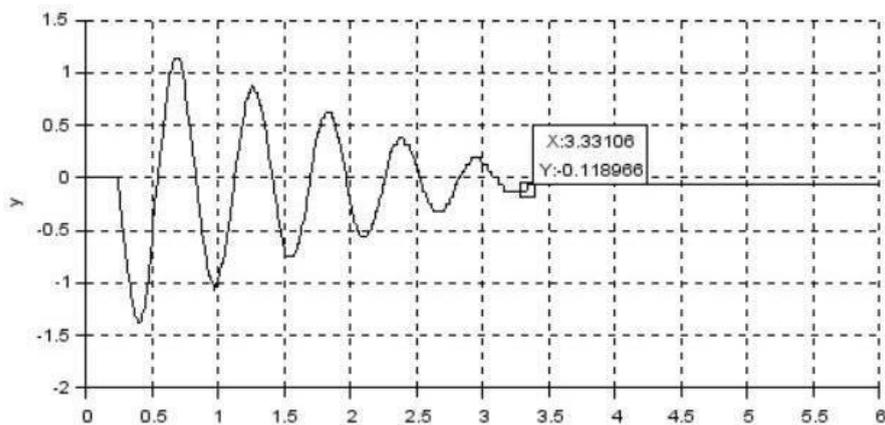
*Rise time* : 0,7 detik  
*Settling time* : 1 detik  
*Error steady state* : 0,08 rad  
*Percent overshoot* : 5 %  
*Peak time* : 0,7 detik



Gambar 4-14 Grafik respon Pendulum ayun konstanta “1”

Dari grafik di atas telah didapatkan rata-rata data respon dari pergerakan *pendulum* ayun dengan kendali diskrit pada konstanta “1” sebagai berikut:

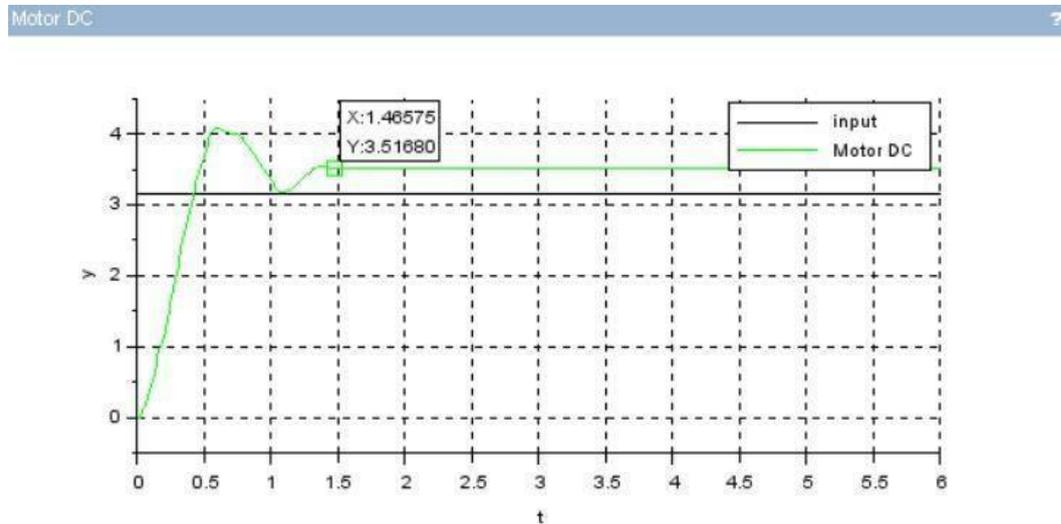
- Peak time* : 0,2 detik
- Settling time* : 4,2 detik
- Decay ratio* : 0,8



Gambar 4-15 Grafik respon Pendulum ayun konstanta “2”

Kemudian dari konstanta “2” diperoleh respon pergerakan *pendulum* ayun sebagai berikut:

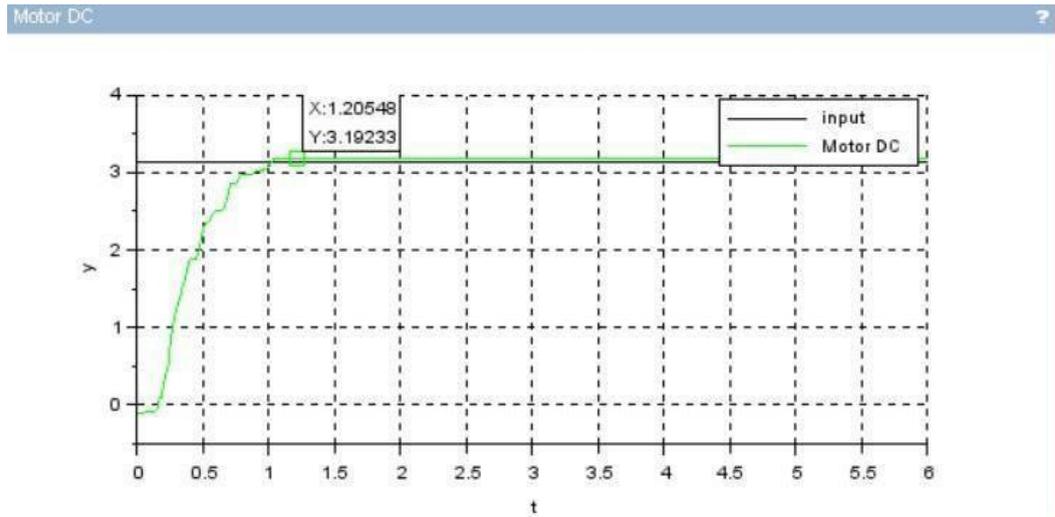
*Peak time* : 0,4 detik  
*Settling time* : 3,33 detik  
*Decay ratio* : 0,58



Gambar 4-16 Grafik respon Motor DC konstanta “1”

Grafik di atas menunjukkan data hasil respon pergerakan Motor DC dengan kendali diskrit dengan *input* 3,14 radian yang dilakukan dengan ketiga konstanta yaitu konstanta “0”, konstanta “1” dan konstanta “2”, untuk konstanta “1” didapat data sebagai berikut:

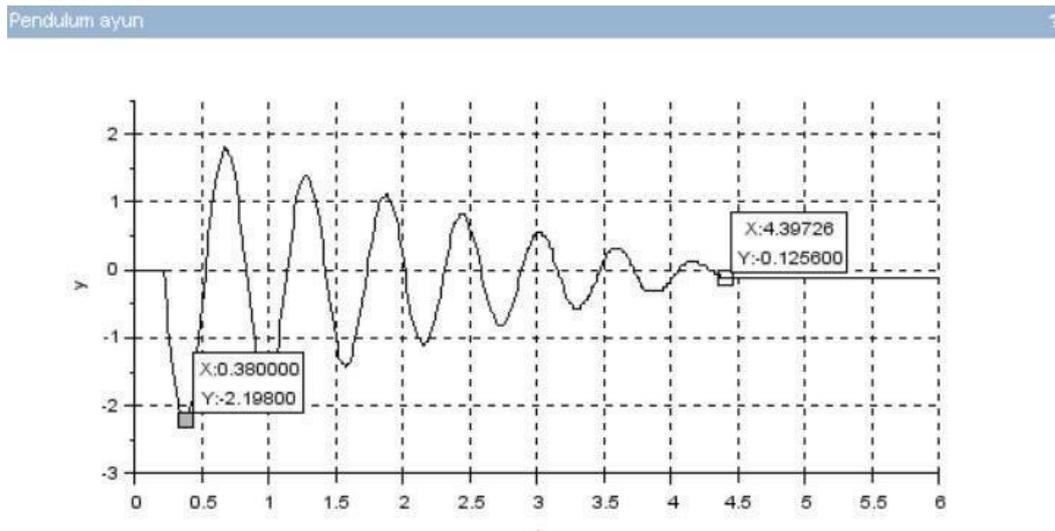
*Rise time* : 0,6 detik  
*Settling time* : 1,46 detik  
*Error steady state* : 0,37 rad  
*Percent overshoot* : 11 %  
*Peak time* : 0,6 detik



Gambar 4-17 Grafik respon Motor DC konstanta “2”

Kemudian untuk konstanta “2” pada pergerakan Motor DC didapat data sebagai berikut:

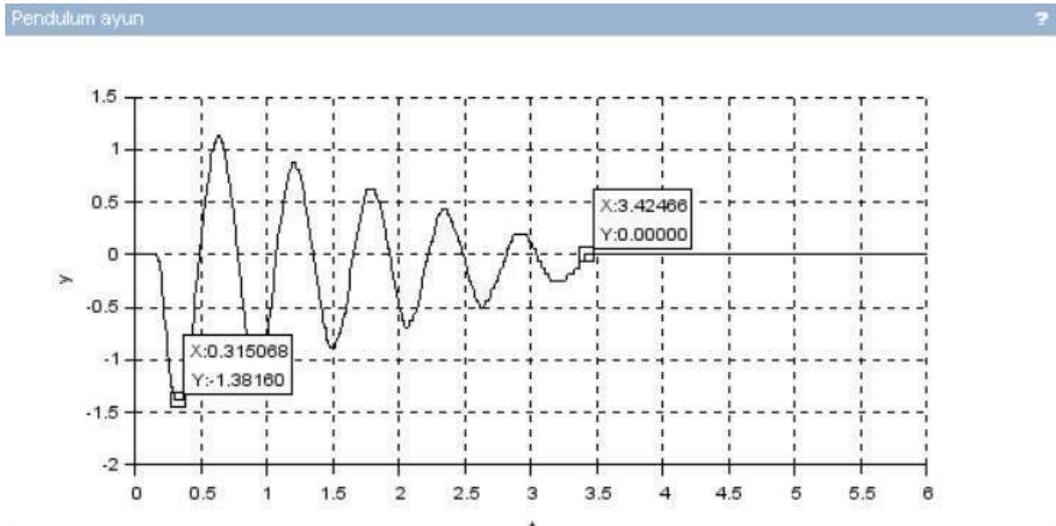
<i>Rise time</i>	: 1 detik
<i>Settling time</i>	: 1,2 detik
<i>Error steady state</i>	: 0,05 rad
<i>Percent overshoot</i>	: 0,01 %
<i>Peak time</i>	: 0,9 detik



Gambar 4-18 Grafik respon pendulum ayun konstanta “1”

Dari grafik di atas telah didapatkan data respon dari pergerakan *pendulum* ayun dengan kendali diskrit pada konstanta “1” sebagai berikut:

*Peak time* : 0,38 detik  
*Settling time* : 4,39 detik  
*Decay ratio* : 0,72



Gambar 4-19 Grafik respon pendulum ayun konstanta “2”

Kemudian dari konstanta “2” diperoleh respon pergerakan *pendulum* ayun sebagai berikut:

*Peak time* : 0,3 detik  
*Settling time* : 3,42 detik  
*Decay ratio* : 0,66

### 4.3 Pembahasan

Dari hasil pengujian, dilakukan dua uji coba untuk mengendalikan ayunan dari *rotary pendulum*. Kedua pengujian tersebut adalah, pengujian tanpa kendali diskrit dan pengujian dengan kendali diskrit dengan metode *manual tuning*, yang dimana pada pengujian dengan kendali diskrit dengan metode *manual tuning* dilakukan menggunakan dua konstanta yaitu konstanta “1” dan konstanta “0”. Selanjutnya dari pengujian tanpa kendali diskrit dan pengujian dengan kendali diskrit menggunakan *manual tuning* dibandingkan sehingga dari hasilnya diketahui mana yang lebih baik dalam performanya. Tabel 4-2, Tabel 4-3 dan Tabel

4-4, Tabel 4-5 menunjukkan perbandingan dari respon motor DC dan respon *pendulum* ayun.

Tabel 4-2 Hasil *steady state* dari alat uji dengan *input* 1,57 radian

	Konstanta		
Percobaan	0	1	2
1	3,51	2,06	1,64
2	3,29	2,28	1,69
3	3,51	2,20	1,66
4	3,39	2,36	1,65
5	3,50	2,30	1,60

Tabel 4-2 menunjukkan hasil dari respon motor DC dengan *input* 1,57 radian tanpa kendali diskrit atau dengan konstanta “0” dan diberikan kendali diskrit metode *manual tuning* dengan dua konstanta yaitu konstanta “1” dan konstanta “2”.

Tabel 4-3 Hasil *settling time* dari alat uji dengan *input* 1,57 radian

	Konstanta		
Percobaan	0	1	2
1	3,75	4,04	3,26
2	3,75	3,66	3,33
3	3,84	3,80	3,63
4	4,28	4,26	3,58
5	3,82	4,28	3,54

Tabel 4-3 menunjukkan hasil yang didapat dari respon *pendulum* ayun dengan *input* 1,57 radian tanpa kendali diskrit atau konstanta “0” dan diberikan kendali diskrit menggunakan metode *manual tuning* dengan dua konstanta yaitu konstanta “1” dan konstanta “2”.

Tabel 4-4 Hasil *steady state* dari alat uji dengan *input* 3,14 radian

	Konstanta		
Percobaan	0	1	2
1	3,99	3,50	3,19
2	4,12	3,51	3,28
3	4,19	3,51	3,14
4	4,27	3,51	3,23
5	4,38	3,53	3,22

Tabel 4-4 menunjukkan hasil dari respon motor DC dengan *input* 3,14 radian tanpa kendali diskrit atau dengan konstanta “0” dan diberikan kendali diskrit metode *manual tuning* dengan dua konstanta yaitu konstanta “1” dan konstanta “2”.

Tabel 4-5 Hasil *settling time* dari alat uji dengan *input* 3,14 radian

	Konstanta		
Percobaan	0	1	2
1	4,05	4,04	3,06
2	4,48	4,65	3,42
3	4,68	4,39	3,83
4	4,49	4,57	3,43
5	4,50	4,01	3,63

Tabel 4-5 menunjukkan hasil gabungan dari respon *pendulum* ayun antara kedua metode tersebut pada *input* 3,14 radian. Hal yang diperhatikan pada pengujian ini adalah berapa lama waktu yang dibutuhkan untuk *pendulum* berhenti berayun dan seberapa besar ayunan yang terjadi pada *pendulum*.

Dari tabel-tabel tersebut diketahui dengan nilai yang diinput untuk nilai *input step* 1,57 radian dan 3,14 radian, bahwa respon dari alat *rotary pendulum* memerlukan kendali agar hasil keluaran sesuai dengan nilai yang telah diinput.

Ketika diberikan kendali diskrit dengan metode *manual tuning*, grafik hasil dari uji *input* 1,57 radian pada tabel 4-2 menunjukkan rata-rata respon oleh alat uji

menunjukkan adanya perbaikan pada hasilnya. Hal ini ditunjukkan dari respon motor DC mulai mendekati nilai keluaran yang diinginkan. Namun, masih terdapat *overshoot* yang ditunjukkan pada grafik, ketika diberikan konfigurasi kendali diskrit metode *manual tuning* konstanta “0”. Sedangkan, ketika objek kendali *rotary pendulum* dikendalikan dengan kendali diskrit metode *manual tuning* konstanta “1” semakin terlihat adanya perbaikan namun masih terdapat *overshoot*. Kemudian dilakukan lagi pengujian dengan kendali diskrit metode *manual tuning* dengan konstanta yang “2” terlihat perubahan pada hasil respon dari motor DC dimana respon semakin mendekati hasil yang diinginkan dan tidak adanya *overshoot* yang terjadi pada metode ini.

Begitu juga pada percobaan selanjutnya menggunakan *input* 3,14 radian, hasil yang diberikan sudah membaik namun masih terdapat *overshoot*, ketika objek kendali dikendalikan dengan kendali diskrit metode *manual tuning* dengan konstanta “1” kembali terlihat adanya perbaikan namun masih terdapat *overshoot*. Selanjutnya dilakukan lagi pengujian menggunakan konstanta yang “2” terlihat perubahan pada respon dari alat uji yang dimana semakin mendekati hasil yang diinginkan dan tidak adanya *overshoot* yang terjadi pada metode ini.

Dari kedua hasil percobaan diatas dapat dilihat bahwa hasil yang didapat dan perbaikan yang didapat sebanding, baik untuk *input* 1,57 radian maupun 3,14 radian. Dengan menggunakan konstanta yang “2” perbaikan semakin meningkat dan mendekati *input* yang diinginkan.

Dari Tabel 4-3 di atas ketika alat tidak diberi suatu kendali hanya diberi *input* 1,57 radian, *pendulum* berayun sangat besar dan membutuhkan waktu apabila di rata-rata 3,88 detik sebelum *pendulum* benar-benar berhenti. Menggunakan kendali diskrit metode *manual tuning* pada konstanta “1” menunjukkan bahwa ayunan *pendulum* tidak sebesar ayunan yang terdapat ketika tidak diberikan kendali, tetapi pada konstanta “1” ini masih membutuhkan waktu yang sedikit lebih lama agar *pendulum* berhenti berayun. Pada pengaplikasian metode kendali diskrit *manual tuning* konstanta “2”, ayunan dari *pendulum* terlihat semakin kecil dan waktu yang diperlukan untuk berhenti semakin singkat dibanding metode sebelumnya dan konstanta sebelumnya.

Berikut juga dengan percobaan pada tabel 4-5, ketika alat tidak diberikan kendali atau konstanta “0”, hanya diberikan *input* 3,14, *pendulum* berayun sangat besar dan membutuhkan waktu rata-rata 4,44 detik sampai *pendulum* benar-benar berhenti. Kemudian digunakan kendali diskrit metode *manual tuning* dengan konstanta “1” menunjukkan adanya sedikit perbaikan sampai benar-benar berhenti. Selanjutnya konstanta diganti dengan konstanta “2” terlihat adanya perubahan yang cukup besar dalam waktu *pendulum* ayun benar-benar berhenti.

Hasil ini masih bisa dikembangkan kembali sehingga bisa mendapatkan hasil yang lebih baik lagi. Seperti dengan mengembangkan pengendalian dengan cara menggabungkan kendali diskrit dengan metode kendali lain seperti NCTF maupun Fuzzy. Kemudian cara lainnya dengan menggunakan perangkat lunak Matlab untuk mencari nilai konstanta diskrit. Hal ini dikarenakan pada perangkat lunak Matlab, terdapat fungsi *auto tuning* sehingga bisa mendapatkan konstanta diskrit sesuai dengan respon dari alat sesuai dengan yang diinginkan.

## BAB 5

### PENUTUP

#### 5.1 Kesimpulan

Kesimpulan dari penelitian ini adalah:

1. Objek kendali *rotary pendulum* mempunyai dua karakteristik objek kendali. Pada motor DC memiliki karakteristik *critical damped* dan *pendulum* memiliki karakteristik *underdamped*. Model matematis objek kendali dari *rotary pendulum* telah mampu disusun sesuai dengan karakteristik tersebut.
2. Simulasi pada objek kendali sudah dilakukan dengan perancangan blok-blok fungsi pada Xcos, setelah melakukan simulasi telah diketahui karakteristik pada objek kendali *rotary pendulum*.
3. Sistem kendali waktu diskrit menggunakan blok fungsi pada Xcos untuk objek kendali *rotary pendulum* telah berhasil dirancang. Skema sistem kendali yang dirancang dapat menunjukkan hasil respon dengan dua metode yaitu tanpa kendali diskrit dan dengan menggunakan kendali waktu diskrit metode *manual tuning* dengan dua konstanta yang berbeda.
4. Penggunaan metode *manual tuning* dengan konstanta “0” sudah dapat memberikan keluaran nilai hasil yang sudah mendekati nilai yang diinginkan. Tetapi dari hasil yang sudah didapat masih kurang maksimal, untuk mencari hasil yang lebih diinginkan kembali melakukan *manual tuning* dengan dua konstanta lagi yaitu konstanta “1” dan konstanta “2”. Penggunaan konstanta “1” ini memiliki hasil yang semakin mendekati nilai yang diinginkan tetapi masih kurang maksimal. Kemudian pada konstanta “2” ini memiliki hasil yang lebih baik dibanding kedua konstanta sebelumnya. Hal ini dikarenakan setiap model objek kendali memiliki kerumitan dan spesifikasi yang berbeda pada setiap objeknya. Metode perumusan konstanta Diskrit ini digunakan *tuning* konstanta secara *manual*.

## **5.2 Saran atau Penelitian Selanjutnya**

1. Penelitian yang selanjutnya dapat menggabungkan dua metode kendali yang berbeda, seperti penggunaan metode kendali Diskrit dengan kendali yang ditambah dengan Fuzzy atau metode kendali lainnya.
2. Penelitian selanjutnya bisa melakukan percobaan lagi mengenai penataan kabel agar objek kendali bisa berputar  $360^0$  tanpa terhalang oleh kabel.

## DAFTAR PUSTAKA

- Anyakrawati, A. (2015). Implementasi Model Reference Adaptive Systems (MRAS) untuk Kestabilan pada Rotary Inverted Pendulum. *Jurnal Seminar Hasil Aretasiwi Anyakrawati*.
- Apkarian, J., Paul Karam, Q., & Michel Lévis, Q. (2012). *student Workbook Inverted Pendulum experiment for LabVIEW™ users Standardized for ABET \* Evaluation Criteria*. <http://www.quanser.com>
- Banks, J., Carson, J., Nelson, B. L., & Nicol, D. (1984). Discrete-event system simulation (Fourth edition). *Prentice Hall International Series in Industrial and Systems Engineering*, 1–161.
- Junaidi, & Prabowo, Y. D. (2018). Project Sistem Kendali Elektronik Berbasis Arduino. In *CV Anugrah Utama Raharja*.
- Muntari, & Nurhadi, H. (2013). Desain Sistem Kendali Rotary Pendulum Dengan Sliding-PID. *JURNAL TEKNIK POMITS Vol. 2, No. 2*.
- Nusantoro, G. D., Muslimin, M. A., Purwanto, & Indra, R. (2012). Rancang Bangun Rotary Inverted Pendulum (RIP) dengan Menggunakan Kontrol PID. *Jurnal EECCIS*.
- Pasirulloh, M. A., & Suryani, E. (2017). *PEMODELAN DAN SIMULASI SISTEM INDUSTRI MANUFAKTUR MENGGUNAKAN METODE SIMULASI HYBRID*.
- Raihana, F. (2019). *Encoder dan Decoder*. Universitas Pembangunan Jaya.
- Septerina, D. (2019). Rancang Bangun Conveyor Pada Alat Pengisi Minuman Otomatis Dengan Kecepatan Putaran Motor DC (Direct Current) Pada PLC (Programmable Logic Controller). *Journal of Chemical Information and Modeling*, 53(9), 1689–1699.
- Solihin, M. I., Wahyudi, & Legowo, A. (2010). *Fuzzy-tuned PID Anti-swing Control of Automatic Gantry Crane*.
- Sutejo, H. R. (2020). PEMODELAN DAN PEMBUATAN PROTOTIPE SISTEM KENDALI POSISI ROTARY INVERTED PENDULUM. *Universitas Islam Indonesia*.

Wahyudi, A. E. (2015). Perancangan Dan Pembuatan Alat Ukur Jarak Digital Berbasis Arduino Menggunakan Sensor Rotary Encoder. *Karya Ilmiah*, 1–31.

# LAMPIRAN

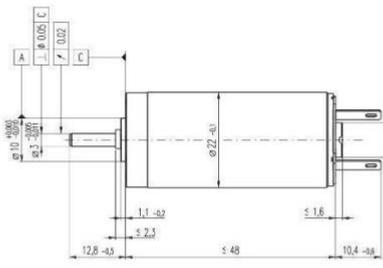
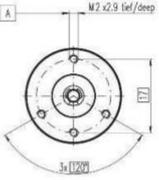
## Lampiran 1

### Datasheet Motor DC Maxon S2322

maxon special program

### S 2322 Ø22 mm, Graphite Brushes, 6 Watt

**NRND** See page 13  
Not recommended for New Design

**M 1:1**

Stock program

Standard program

Special program (on request)

Part Numbers	
Winding number	-11.225-200 (Insert winding number)
2322. ...	980 981 990 982 983 985 987

Motor Data	Winding number	980	981	990	982	983	985	987
<b>Values at nominal voltage</b>								
1 Nominal voltage	V	12	15	18	18	24	36	48
2 No load speed	rpm	7010	7000	7530	6690	7190	7240	6370
3 No load current	mA	42.2	33.3	30.6	25.8	21.3	14.3	8.88
4 Nominal speed	rpm	4300	4310	4840	3970	4490	4500	3580
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	0.654	0.583	0.520	0.420	0.279	0.184
7 Stall torque	mNm	32.9	33.5	35.9	31.4	34.1	33.4	28.8
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
<b>Characteristics</b>								
10 Terminal resistance	Ω	5.61	8.71	10.9	14.1	21.6	49.7	117
11 Terminal inductance	mH	0.492	0.790	0.995	1.26	1.97	4.43	10.2
12 Torque constant	mNm/A	15.4	19.4	21.8	24.6	30.7	46.1	69.9
13 Speed constant	rpm/V	622	491	437	389	311	207	137
14 Speed/torque gradient	rpm/mNm	227	220	219	223	219	224	228
15 Mechanical time constant	ms	13.9	13.5	13.3	13.3	13.0	12.9	12.8
16 Rotor inertia	gcm <sup>2</sup>	5.84	5.86	5.79	5.70	5.68	5.50	5.37

**Specifications**

**Thermal data**

17 Thermal resistance housing-ambient 14.1 KW

18 Thermal resistance winding-housing 3.6 KW

19 Thermal time constant winding 9.24 s

20 Thermal time constant motor 649 s

21 Ambient temperature -20...+100°C

22 Max. permissible winding temperature +125°C

**Mechanical data (ball bearings)**

23 Max. permissible speed 9500 rpm

24 Axial play 0.05 - 0.15 mm

25 Radial play 0.025 mm

26 Max. axial load (dynamic) 2.8 N

27 Max. force for press fits (static) 64 N

28 Max. radial loading, 5 mm from flange 14 N

**Other specifications**

29 Number of pole pairs 1

30 Number of commutator segments 7

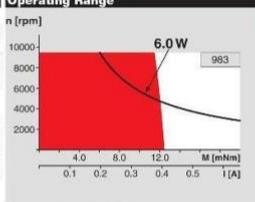
31 Weight of motor 92 g

Values listed in the table are nominal. Explanation of the figures on page 71.

**Option**

Sleeve bearings in place of ball bearings

**Operating Range**



**Comments**

■ **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.

□ **Short term operation**  
The motor may be briefly overloaded (recurring).

— **Assigned power rating**

**maxon Modular System** Overview on page 20 - 25

- Planetary Gearhead Ø22 mm 0.5 - 1.0 Nm Page 250
- Planetary Gearhead Ø22 mm 0.5 - 2.0 Nm Page 251
- Planetary Gearhead Ø26 mm 0.5 - 2.0 Nm Page 382

364 maxon special program
May 2013 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, A', B, B'	T: Totem pole output N: NPN open collector 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		5: A, A' 6: B, B', Z, Z'	V: Voltage output L: Line driver output		

**■ Specifications**

Item	Shaft Type/hollow Shaft Type/Blind Hollow Shaft Type Ø40mm Incremental Rotary Encoder	
Resolution (PPR) <sup>*1</sup>	*1, *2, *5, 10, *12, 15, 20, 23, 25, 30, 35, 40, 45, 50, 60, 75, 100, 120, 150, 180, 200, 240, 250, 256, 300, 360, 400, 500, 512, 600, 800, 1000, 1024, 1200, 1500, 1800, 2000, 2048, 2500, 3000, 3600, 5000	
Output phase	A, B, Z phase (line driver A, A', B, B', Z, Z' phase)	
Phase difference of output	Phase difference between A and B: $\frac{T}{2} \pm \frac{T}{8}$ (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=
	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=
Response time (Rise/Fall)	Totem pole output	Max. 1µs (cable length: 2m, I sink = 20mA)
	NPN open collector output	
	Voltage output	
	Line driver output	
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 <sup>2</sup> (4×10 <sup>-6</sup> kg·m <sup>2</sup> )
	Shaft loading	Radial: max. 2kgf, Thrust: max. 1kgf
Max. allowable revolution <sup>**2</sup>	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, Ø6mm 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, A', B, 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.  
(Max. response revolution (rpm) =  $\frac{\text{Max. response frequency}}{\text{Resolution}} \times 60 \text{ sec}$ )

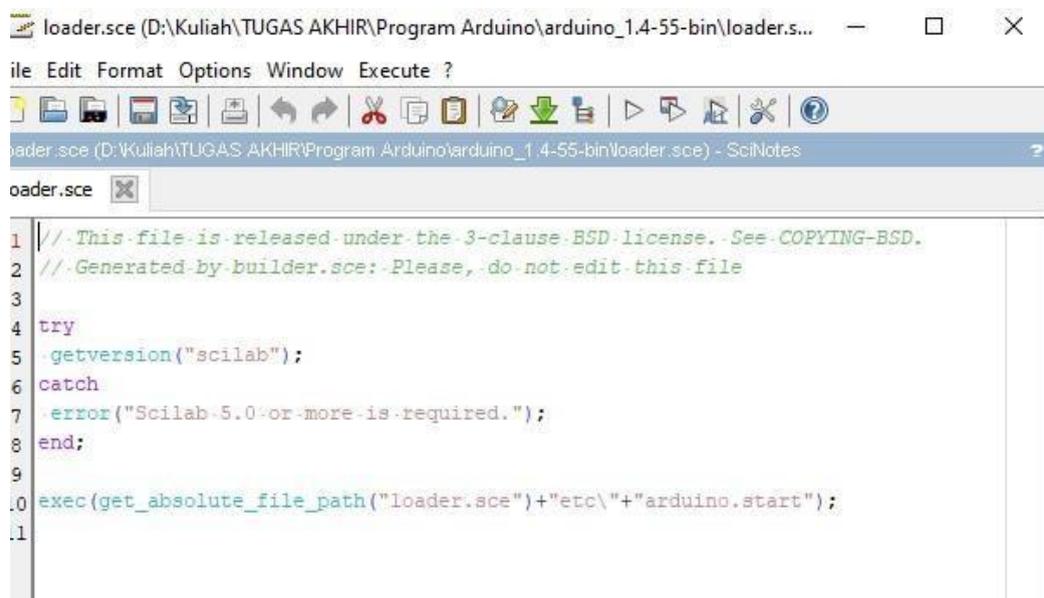
×Environment resistance is rated at no freezing or condensation.

**Autonics**

H-24

### Lampiran 3

#### Loader Scilab



```
loader.sce (D:\Kuliah\TUGAS AKHIR\Program Arduino\arduino_1.4-55-bin\loader.sce) - SciNotes
loader.sce (D:\Kuliah\TUGAS AKHIR\Program Arduino\arduino_1.4-55-bin\loader.sce) - SciNotes
loader.sce

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

