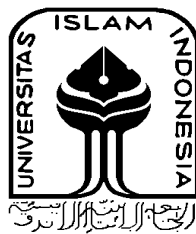


**IDENTIFIKASI DAN PEMODELAN SISTEM KENDALI
POSISI LINIER BERDASARKAN TRAJECTORY RESPONS**

TUGAS AKHIR

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



Disusun Oleh :

Nama : ULLUM AKHYA AL ANSHORI
No. Mahasiswa : 18525076
NIRM : 1802130044

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

2025

PERNYATAAN KEASLIAN

Dengan ini saya menyatakan bahwa karya tulis ilmiah yang saya buat merupakan karya sendiri bukan hasil plagiarisme dari karya tulis yang dibuat oleh orang lain. Semua referensi dan kutipan yang saya tulis pada karya tulis ini saya cantumkan sitasi dan sumber pustakanya. Apabila dikemudian hari saya dianggap melakukan pelanggaran hak kekayaan intelektual dan yang saya tulis pada karya ilmiah ini tidak benar, Maka saya bersedia menerima sanksi dan hukuman yang berlaku.

Yogyakarta, 4 Juli 2025



Ullum Akhya Al Anshori

LEMBAR PENGESAHAN DOSEN PEMBIMBING

IDENTIFIKASI DAN PEMODELAN SISTEM KENDALI
POSISI LINIER BERDASARKAN TRAJECTORY RESPONS

TUGAS AKHIR

Disusun Oleh :

Nama : ULLUM AKHYA AL ANSHORI
No. Mahasiswa : 18525076
NIRM : 1802130044

Yogyakarta, 11 Juni 2025

Pembimbing I,



Purtojo, S.T., M.Sc

Pembimbing II,



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

LEMBAR PENGESAHAN DOSEN PENGUJI

IDENTIFIKASI DAN PEMODELAN SISTEM KENDALI POSISI LINIER BERDASARKAN TRAJECTORY RESPONS

TUGAS AKHIR

Disusun Oleh :

Nama : ULLUM AKHYA AL ANSHORI

No. Mahasiswa : 18525076

NIRM : 1802130044

Tim Penguji

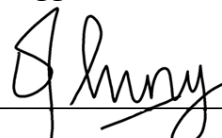
Purtojo, S.T., M.Sc.

Ketua


Tanggal : 03/07/2025

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

Anggota I


Tanggal : 30/06/2025

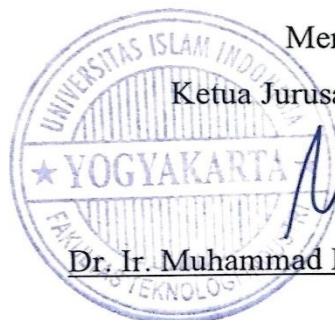
Yustiasih Purwaningrum, S.T., M.T.

Anggota II


Tanggal : 01/07/2025

Mengetahui

Ketua Jurusan Teknik Mesin



Dr. Ir. Muhammad Khaidh, S.T., M.T., IPP

HALAMAN PERSEMBAHAN

Alhamdulillah segala puji bagi Allah Subhanahu Wa Ta'ala atas segala berkat, rahmat dan karunia-NYA memberikan saya kemudahan dalam menyelesaikan tugas akhir saya yang berjudul “Identifikasi dan Pemodelan Sistem Kendali Posisi Linier Berdasarkan Trajectory Respons”. Shalawat serta salam tidak lupa selalu tercurahkan kepada baginda Nabi Muhammad Shallallahu ‘Alaihi Wasallam, manusia paling sempurna yang menjadi suri tauladan yang baik bagi umatnya. Dengan segala rasa syukur tugas akhir ini saya persembahkan kepada :

- Bapak dan ibu yang sangat saya cintai **Alm. Rustanto Nasori dan Dakoni**. Sebagai bukti atas amanah yang diberikan kepada saya dalam menempuh pendidikan S1. Terima kasih telah berjuang dalam mendidik dan membimbing saya sehingga saya dapat mencapai gelar sarjana. Semoga gelar sarjana ini dapat menjadi jalan saya untuk membahagiakan serta membanggakan kalian, dan segala jerih payah perjuangan kalian dapat menjadi pahala yang bermanfaat dan bernilai disisi Allah Subhanahu Wa Ta'ala.
- Kedua kakak yang sangat saya sayangi **Andika Agung Wicaksana dan Umu Fiqri Hanipah** Terima kasih atas segala bimbingan dan dukungan yang selalu kalian berikan. Semoga aku bisa menjadi apa yang kalian harapkan. Jangan pernah lelah dan bosan untuk selalu memberikan nasihat kepadaku.
- Pasangan saya yang sangat saya sayangi **Khopipah Muhtar**. Terima kasih atas kesabarannya selalu menemani, memberi dukungan dan selalu mendengarkan keluh kesah penulis dalam proses mengerjakan tugas akhir.

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)

Allah itu baik takdirnya tak pernah salah

(Anonymous)

There is no rest in this world, perhaps in the next

(Anonymous)

KATA PENGANTAR ATAU UCAPAN TERIMA KASIH

Alhamdulillahirabbil'alamin, puji syukur selalu terpanjatkan kepada Allah Subhanahu Wa Ta'ala atas berkat, rahmat dan hidayah-Nya laporan tugas akhir yang berjudul "Identifikasi dan Pemodelan Sistem Kendali Posisi Linier Satu Aksis Berdasarkan Trajectory Respons Terhadap Input Step" dapat diselesaikan dengan sebaik-baiknya. Tidak lupa shalawat serta salam selalu tercurahkan pada junjungan besar Nabi Muhammad Shallallahu 'Alaihi Wasallam, semoga kita termasuk golongan manusia yang mendapatkan syafaatnya, aamiin ya rabbal alamin.

Laporan tugas akhir ini disusun sebagai bentuk pertanggungjawaban penulis setelah melakukan semua kegiatan akademik di Teknik Mesin, Fakultas Teknologi Industri, Universitas Islam Indonesia. Penulis berusaha sebaik mungkin dalam menyusun laporan tugas akhir ini agar mudah dipahami dan menarik bagi pembaca.

Penulis mengucapkan terima kasih kepada pihak-pihak yang selalu memberikan dukungan serta bantuan baik secara moril maupun materil sehingga tugas akhir ini dapat terselesaikan dengan baik. Khususnya kepada :

1. Allah Subhanahu Wa Ta'ala yang telah memberikan kemudahan dan petunjuk dalam setiap langkah penulis untuk menyelesaikan laporan tugas akhir ini.
2. Nabi Muhammad Shallallahu 'Alaihi Wasallam yang telah memberikan suri tauladan yang baik.
3. Kedua orangtua, kedua kakak dan partner saya yang selalu memberikan semangat, kasih sayang, dukungan dan kepercayaan untuk menyelesaikan pendidikan di UII.
4. Bapak Purtojo, S.T., M.Sc selaku dosen pembimbing yang senantiasa memberikan arahan dengan sabar dan membagikan ilmunya dalam membantu menyelesaikan tugas akhir yang saya kerjakan.
5. Bapak Ir. Donny Suryawan, S.T., M.Eng., IPP sebagai dosen pembimbing yang senantiasa membantu memberikan masukan dan membagikan ilmunya untuk menyelesaikan tugas akhir ini.

Penulis merasa masih banyak kekurangan dalam menyusun laporan ini dan berharap kepada para pembaca dan semua pihak untuk dapat memberikan masukan. Semoga laporan ini dapat memberikan manfaat bagi pembaca dan diri penulis sendiri.

Yogyakarta, 11 Juni 2025

Penulis,

A handwritten signature in black ink, appearing to be 'Ullum' followed by a stylized flourish.

Ullum Akhya Al Anshori

IDENTIFIKASI DAN PEMODELAN SISTEM KENDALI POSISI LINIER BERDASARKAN TRAJECTORY RESPONS

ULLUM AKHYA AL ANSHORI

ABSTRAK

Pada penelitian ini dilakukan identifikasi dan pemodelan sistem kendali posisi linier. Objek yang dikendalikan berupa aktuator linier dua aksis yang terdiri dari motor DC yang terhubung dengan linier *ball screw*. Identifikasi karakteristik objek kendali dilakukan dengan metode *Nominal characteristic trajectory following* (NCTF) untuk mendapatkan model matematika berupa fungsi transfer dari sistem kendali posisi linier. Kinerja sistem dievaluasi dengan membandingkan gerak objek kendali melalui simulasi dan gerak objek kendali secara eksperimen. Gerak objek kendali *point to point* (PTP) dianalisis pada gerak satu aksis dengan konfigurasi *close-loop*. Gerak objek kendali *countinous motion* (CM) dianalisis pada gerak objek kendali dua aksis dengan target membentuk belah ketupat pada konfigurasi *open-loop* dan *close-loop*. Selisih gerak objek kendali antara simulasi dan eksperimen dinyatakan sebagai *error*. Dalam penelitian ini masih terdapat *error* antara gerak objek kendali simulasi dan eksperimen sistem kendali posisi PTP dan CM dalam konfigurasi *open-loop* maupun *close-loop*.

Kata kunci : *NCTF, PTP, CM, open-loop, close-loop*

IDENTIFICATION AND MODELING OF LINEAR POSITION CONTROL SYSTEM BASED ON TRAJECTORY RESPONSE

ULLUM AKHYA AL ANSHORI

ABSTRACT

This study focuses on the identification and modeling of a linear position control system. The controlled object consists of a two-axis linear actuator system utilizing DC motors coupled with linear ball screws. The control object's characteristics were identified using the Nominal Characteristic Trajectory Following (NCTF) method to obtain a mathematical model in the form of a transfer function for the linear position control system. System performance was evaluated by comparing the actuator's motion through simulation with experimental results. Point-to-point (PTP) motion was analyzed in single-axis operation under closed-loop configuration, while continuous motion (CM) was examined in two-axis operation tracing a diamond-shaped trajectory in both open-loop and closed-loop configurations. The discrepancy between simulated and experimental actuator motion was quantified as error. The study revealed persistent errors between simulated and experimental results for both PTP and CM position control systems in both open-loop and closed-loop configurations.

Keyword : NCTF, PTP, CM, open-loop, close-loop

DAFTAR ISI

Halaman Judul	i
Lembar Pengesahan Dosen Pembimbing	iii
Lembar Pengesahan Dosen Penguji	iv
Halaman Persembahan	v
Halaman Motto	vi
Kata Pengantar atau Ucapan Terima Kasih	vii
Abstrak	ix
Daftar Tabel	xiii
Daftar Gambar	xiv
Daftar Notasi	xvi
Bab 1 Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian atau Perancangan	4
1.6 Sistematika Penulisan	4
Bab 2 Tinjauan Pustaka	5
2.1 Kajian Pustaka	5
2.2 Dasar Teori	6
2.2.1 Sistem Kendali	6
2.2.2 Konfigurasi <i>Open-loop</i>	7
2.2.3 Konfigurasi <i>Close-loop</i>	9
2.2.4 Konsep NCTF	10
2.2.5 Ekperimen <i>Open-loop</i>	12
2.2.6 Pembuatan NCT	13
Bab 3 Metode Penelitian	15
3.1 Alur Penelitian	15
3.2 Peralatan	16
3.2.1 Perangkat Keras	16
3.2.2 Perangkat Lunak	19

3.2.3	Perancangan Perangkat Keras	19
Bab 4	Hasil dan Pembahasan.....	24
4.1	Identifikasi Sistem Kendali Posisi Linier	24
4.1.1	Pemeriksaan Resolusi Encoder Linier.....	24
4.1.2	Kalibrasi Nilai Tegangan Driver L298N dengan Nilai PWM.....	25
4.1.3	Perangkaian Blok Diagram Eksperimen <i>Open-loop</i>	27
4.1.4	Konstruksi NCT.....	29
4.1.5	Fungsi Transfer.....	31
4.2	Kinerja Sistem Kendali Posisi PTP	34
4.2.1	Eksperimen Satu Aksis <i>Close-loop</i>	34
4.3	Kinerja Sistem Kendali Posisi CM	40
4.3.1	Simulasi Belah Ketupat	40
4.3.2	Eksperimen <i>Open-loop</i> Belah Ketupat.....	44
4.3.3	Eksperimen <i>Close-Loop</i> Belah Ketupat	46
Bab 5	Penutup.....	51
5.1	Kesimpulan	51
5.2	Saran atau Penelitian Selanjutnya.....	51
Daftar Pustaka		53

DAFTAR TABEL

Tabel 3-1 Tabel fungsi pin pada Driver L298N	22
Tabel 3-2 Konfigurasi warna kabel linier enkoder RLS LM10IC001AA10K....	22
Tabel 4-1 Kalibrasi nilai tegangan.....	26
Tabel 4-2 Konfigurasi arah putaran motor DC.....	28
Tabel 4-3 Parameter NCT sumbu X.....	30
Tabel 4-4 Parameter NCT sumbu Y	31
Tabel 4-5 Nilai <i>error</i> sumbu X.....	39
Tabel 4-6 Nilai <i>error</i> sumbu Y	39
Tabel 4-7 Hasil pengukuran simulasi belah ketupat.....	43
Tabel 4-8 Perbandingan simulasi dan eksperimen <i>open-loop</i> sumbu X	46
Tabel 4-9 Perbandingan simulasi dan eksperimen <i>open-loop</i> sumbu Y	46
Tabel 4-10 Perbandingan simulasi dan eksperimen <i>close-loop</i> sumbu X.....	49
Tabel 4-11 Perbandingan simulasi dan eksperimen <i>close-loop</i> sumbu Y	50

DAFTAR GAMBAR

Gambar 2-1 Sistem kendali sederhana	6
Gambar 2-2 Sistem <i>open-loop</i>	7
Gambar 2-3 Sistem <i>close loop</i>	10
Gambar 2-4 Struktur sistem kendali NCTF	11
Gambar 2-5 NCT dan pergerakan objek kendali	11
Gambar 2-6 <i>Input stepwise</i> dan respons objek kendali	12
Gambar 2-7 <i>Nominal characteristic trajectory</i> (NCT)	13
Gambar 3-1 Diagram alir penelitian	15
Gambar 3-2 Arduino Mega 2560	16
Gambar 3-3 Driver L298N	17
Gambar 3-4 Maxon Motor DC	17
Gambar 3-5 Linier <i>ball-screw</i>	17
Gambar 3-6 Enkoder Linier	18
Gambar 3-7 Posisi Enkoder pada linier <i>ball-screw</i>	18
Gambar 3-8 Kabel <i>jumper</i>	18
Gambar 3-9 <i>Project board</i>	19
Gambar 3-10 <i>Power supply</i>	19
Gambar 3-11 Hasil rangkaian perangkat keras	20
Gambar 3-12 Aksis pada aktuator	20
Gambar 3-13 Skema rangkaian koneksi perangkat keras	21
Gambar 3-14 Rangkaian proses alur kerja	23
Gambar 4-1 Blok diagram pembacaan enkoder linier	24
Gambar 4-2 Hasil pembacaan enkoder linier pada Scilab sumbu X	25
Gambar 4-3 Hasil pembacaan enkoder linier pada Scilab sumbu Y	25
Gambar 4-4 Blok diagram pengukuran tegangan	26
Gambar 4-5 Blok diagram eksperimen <i>open-loop</i> sumbu X	27
Gambar 4-6 Blok diagram eksperimen <i>open-loop</i> sumbu Y	27
Gambar 4-7 Respons kecepatan eksperimen <i>open-loop</i> sumbu X	29
Gambar 4-8 Respons kecepatan eksperimen <i>open-loop</i> sumbu Y	29
Gambar 4-9 <i>Nominal Characteristic Trajectory</i> (NCT) sumbu X	30

Gambar 4-10 <i>Nominal Characteristic Trajectory</i> (NCT) sumbu Y	31
Gambar 4-11 Blok diagram kecepatan eksperimen dan simulasi sumbu X	32
Gambar 4-12 Blok diagram kecepatan eksperimen dan simulasi sumbu Y	33
Gambar 4-13 Respons kecepatan eksperimen dan simulasi sumbu X	33
Gambar 4-14 Respons kecepatan eksperimen dan simulasi sumbu Y	34
Gambar 4-15 Rangkaian blok diagram satu aksis <i>close-loop</i> sumbu X	35
Gambar 4-16 Rangkaian blok diagram satu aksis <i>close-loop</i> sumbu Y	35
Gambar 4-17 Respons posisi eksperimen <i>close-loop input</i> 30 mm sumbu X	36
Gambar 4-18 Respons posisi eksperimen <i>close-loop input</i> 40 mm sumbu X	37
Gambar 4-19 Respons posisi eksperimen <i>close-loop input</i> 50 mm sumbu X	37
Gambar 4-20 Respons posisi eksperimen <i>close-loop input</i> 30 mm sumbu Y	38
Gambar 4-21 Respons posisi eksperimen <i>close-loop input</i> 40 mm sumbu Y	38
Gambar 4-22 Respons posisi eksperimen <i>close-loop input</i> 50 mm sumbu Y	39
Gambar 4-23 Blok diagram simulasi belah ketupat	41
Gambar 4-24 <i>Signal builder</i> sumbu X	41
Gambar 4-25 <i>Signal builder</i> sumbu Y	42
Gambar 4-26 Visualisasi simulasi belah ketupat	43
Gambar 4-27 Rangkaian blok diagram simulasi dan eksperimen <i>open-loop</i> belah ketupat	44
Gambar 4-28 Visualisasi simulasi dan eksperimen <i>open-loop</i> belah ketupat	45
Gambar 4-29 Rangkaian blok diagram simulasi dan eksperimen <i>close-loop</i> belah ketupat	48
Gambar 4-30 Visualisasi simulasi dan eksperimen <i>close-loop</i> belah ketupat	49

DAFTAR NOTASI

NCTF	: <i>Nominal Characteristic Trajectory Following</i>	
NCT	: <i>Nominal Characteristic Trajectory</i>	
PTP	: <i>Point to Point</i>	
CM	: <i>Countinous Motion</i>	
CNC	: <i>Computer Numerical Control</i>	
PWM	: <i>Pulse Width Modulation</i>	
PID	: <i>Propotional Integral Derivative</i>	
PI	: <i>Propotional Integral</i>	
PV	: <i>Propotional Velocity</i>	
e	: <i>Error</i>	
$\dot{e}$	: <i>Error-rate</i>	
m	: <i>Gradien pada NCT</i>	
h	: <i>Kecepatan maksimum objek kendali</i>	(mm/s)
U	: <i>Input pada objek kendali</i>	(V)
K	: <i>Parameter sederhana dari objek kendali</i>	
α	: <i>Parameter sederhana dari objek kendali</i>	
A	: <i>Area deselerasi pada objek kendali</i>	
AC	: <i>Alternating Current</i>	
DC	: <i>Direct Current</i>	

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Sistem kendali menjadi bagian penting dari otomasi modern di bidang industri. Sistem kendali dapat mengatasi keterbatasan suatu sistem mekanik dan mengatur kinerja perangkat yang membutuhkan ketelitian dan tingkat akurasi yang tinggi agar sistem dapat bekerja sesuai dengan yang diinginkan. Sistem kendali posisi adalah salah satu sistem kendali yang banyak digunakan mulai dari mesin 3D *print*, mesin CNC (*computer numerical control*) hingga sistem robotika.

Pada sistem kendali posisi objek yang dikendalikan bisa berupa aktuator, aktuator yang banyak digunakan adalah motor elektrik. Umumnya sistem kendali posisi harus memenuhi beberapa kriteria seperti respon yang cepat, akurasi yang tinggi, *overshoot* yang kecil dan *robustness*. Metode pengendalian berperan untuk mengatur kecepatan dan ketepatan aktuator agar dapat berpindah menuju posisi yang diinginkan dan bergerak sesuai dengan perintah (*input*) yang diberikan. Batas kemampuan *controller-card*, aktuator dan enkoder menentukan performa dari metode pengendalian. *Controller-card* mengirim sinyal menuju aktuator sehingga terjadi gerakan pada aktuator, perubahan gerakan pada aktuator akan dibaca oleh enkoder secara *realtime*. Tingkat ketelitian enkoder sangat dibutuhkan agar nilai hasil simulasi tidak berbeda jauh dengan nilai hasil eksperimen.

Perancangan sistem kendali posisi linier menggunakan metode konvensional, yaitu dengan menghitung fungsi transfer dari sebuah objek kendali. Metode konvensional terbilang cukup rumit dan membutuhkan pengetahuan yang mendalam dalam ilmu pengendalian. Dalam proses mendapatkan fungsi transfer objek kendali, perlu dilakukan identifikasi sifat mekanis pada objek kendali. Sebagai contoh pada objek kendali motor DC, dibutuhkan identifikasi sifat mekanis motor DC. Secara umum sifat mekanis motor DC dapat diketahui berdasarkan *datasheet* yang disediakan oleh perusahaan pembuat motor DC. Permasalahan terjadi pada saat perusahaan pembuat motor DC tidak menyediakan *datasheet*. Permasalahan lain dapat terjadi ketika motor DC telah digunakan dalam

waktu lama. Hal ini dapat menyebabkan perubahan kondisi pada motor DC sehingga sifat mekanisnya berbeda dengan yang tertera pada *datasheet*. Oleh sebab itu, metode pengendalian *nominal characteristic trajectory following* (NCTF) telah diusulkan. Sistem kendali NCTF pertama kali diperkenalkan oleh (Wahyudi et al., 2003) sebagai *controller* dengan metode perancangan sederhana yang diaplikasikan pada mekanisme *rotary* untuk sistem kendali posisi *point to point* (PTP). Perancangan NCTF *controller* sangat sederhana dan mudah dimengerti. NCTF *controller* tidak membutuhkan model pasti dari parameter objek kendali. Parameter objek kendali didapatkan dari eksperimen *open-loop* sederhana yang menampilkan respons kecepatan dan perpindahan objek kendali pada saat deselerasi. Data respons kecepatan dan perpindahan objek kendali dikonstruksi menjadi bentuk lintasan (*trajectory*). *Trajectory* respons ini dikenal sebagai *nominal characteristic trajectory* (NCT). Parameter yang ada pada NCT digunakan untuk mendapatkan fungsi transfer dari objek kendali (Purtojo, 2017).

Selanjutnya (Sato & Maeda, 2009) mengembangkan *countinous motion* (CM) NCTF sebagai *controller* performa tinggi dengan struktur sistem kendali yang sederhana. CM NCTF *controller* diaplikasikan pada sistem kendali posisi *countinous motion* (CM). Penelitian (Chong & Sato, 2010) dilakukan pada mekanisme *non-contact* membuktikan bahwa proses mendesain NCTF *controller* tidak bergantung pada parameter *friction*. Pada tahun 2017 CM NCTF *controller* telah diusulkan oleh (Hasim et al., 2017) sebagai *controller* praktis untuk meningkatkan kinerja pada sistem kendali posisi kontinyu menggunakan mekanisme *ball-screw* X-Y yang digerakan oleh motor AC. Penelitian (Hasim et al., 2017) menjelaskan keunggulan motor AC dalam efektifitas, kestabilan dan sistem operasi yang lebih halus dibanding dengan motor DC. Namun, motor AC lebih sulit dikendalikan pada sistem kendali posisi dan sering terjadi gangguan akibat dari karakteristik nonlinier yang ada pada motor AC. Motor DC lebih unggul dalam segi kompleksitas dan pengendalian yang sederhana. Motor DC dapat diimplementasikan dengan *driver* sederhana yaitu dengan kendali sinyal PWM dan enkoder.

Pada penelitian ini penulis merancang sistem kendali posisi menggunakan mekanisme *ball-screw* X-Y dengan penggerak motor DC. Metode NCTF

digunakan untuk mendapatkan nilai fungsi transfer dari sistem kendali posisi. Kinerja sistem kendali posisi dievaluasi pada sistem kendali posisi PTP dan sistem kendali posisi CM. Penelitian ini berfokus kepada menganalisis nilai *error* atau perbandingan nilai antara simulasi dan eksperimen pada konfigurasi *open-loop* dan *close-loop*. Hasil penelitian ini dapat digunakan sebagai dasar untuk mendesain CM NCTF *controller* pada mekanisme *ball-screw* X-Y dengan penggerak motor DC. Penelitian ini diharapkan dapat menjadi dasar pada pengembangan mesin CNC *high performance* yang mengimplementasikan motor DC sebagai sistem penggerak.

1.2 Rumusan Masalah

Seperti yang telah dijelaskan pada latar belakang. Perancangan sistem kendali posisi linier membutuhkan identifikasi dan pemodelan. Sehingga perlu dilakukan penelitian untuk mengidentifikasi dan memodelkan sistem kendali posisi linier. Bagaimana cara mengidentifikasi karakteristik objek kendali? Jika telah mengetahui karakteristik objek kendali, selanjutnya adalah bagaimana cara merancang model sistem kendali posisi linier?

1.3 Batasan Masalah

Kecepatan perpindahan objek kendali digunakan untuk memverifikasi fungsi transfer agar dapat dinyatakan sudah mewakili karakteristik objek kendali. Parameter percepatan dari objek kendali tidak menjadi bahasan pada penelitian ini. Pada penelitian ini pergerakan dua aksis dilakukan. Gerakan dua aksis ditargetkan mendekati bentuk bidang datar belah ketupat. Gerakan dua aksis dinyatakan sudah membentuk belah ketupat apabila secara visual gerakan tersebut sudah mendekati atau membentuk belah ketupat.

Penelitian ini menggunakan perangkat lunak yang bersifat *open-source* yaitu Scilab 6.1.1 untuk mendesain blok diagram. *Controller-card* Arduino mega 2560 digunakan sebagai perangkat keras yang bersifat *open-source* dan mudah didapatkan. Perangkat keras lain dalam penelitian ini seperti *motor-driver*, aktuator dan enkoder terdiri satu jenis.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

1. Untuk mendapatkan model matematika berupa fungsi transfer dari sistem kendali posisi linier.
2. Untuk mengevaluasi kinerja sistem kendali posisi linier berdasarkan perubahan posisi pada simulasi fungsi transfer terhadap perubahan secara eksperimen.

1.5 Manfaat Penelitian atau Perancangan

Hasil penelitian ini dapat digunakan sebagai dasar untuk merancang CM NCTF *controller* pada sistem kendali posisi yang nantinya diharapkan dapat dilakukan pengembangan pada mesin CNC *High performance* dengan implementasi motor DC pada penelitian selanjutnya .

1.6 Sistematika Penulisan

Laporan penelitian ini terdiri dari lima bab. Bab pertama berisi mengenai mengapa penelitian ini dilakukan, ruang lingkup penelitian, dan tujuan dari penelitian. Bab kedua berisi mengenai teori dan penelitian terdahulu yang menjadi referensi penelitian ini. Bab ketiga membahas mengenai proses penelitian yang dilakukan, mulai dari alur penelitian, peralatan yang digunakan, proses perancangan. Bab keempat berisi hasil penelitian dan analisis terhadap hasil penelitian. Terakhir, bab kelima membahas mengenai kesimpulan penelitian dan saran.

BAB 2

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Pada penelitian (Akmeliawati & Wijaya, 2009) telah dijelaskan tentang teori konsep dasar dalam proses membangun NCTF *controller*. NCTF *controller* terdiri dari NCT dan *compensator*. Dalam membangun NCTF *controller* eksperimen *open-loop* sederhana dilakukan. Respons objek kendali dari eksperimen *open-loop* digunakan untuk mengkonstruksi NCT. *Compensator* digunakan untuk memberikan aksi pada objek kendali agar dapat bergerak menuju NCT dan tetap berada pada NCT. Penelitian ini juga membahas tentang perbandingan antara NCTF *controller* dengan PID (*proportional integral derivative*) *controller* pada sistem kendali PTP. NCTF *controller* menggunakan *compensator* PI (*propotional integral*). Hasil dari penelitian ini NCTF *controller* lebih efektif dengan nilai *overshoot* yang kecil dan *settling time* yang cepat dibandingkan dengan *controller* PID *controller*. NCTF *controller* menghasilkan *overshoot* 2.7% dan *settling time* 0.0449 detik sedangkan PID *controller* menghasilkan *overshoot* 75.9% dan *settling time* 0.1931 detik.

Pada penelitian (Hasim et al., 2017) dijelaskan tentang kinerja pengendalian posisi menggunakan NCTF *controller* dengan PI *compensator* pada sistem kendali posisi CM. Kinerja NCTF *controller* dievaluasi dengan cara membandingkan dengan kinerja PID *controller*. Objek kendali pada penelitian ini adalah motor AC yang menggerakkan *ball screw* pada sumbu X dan sumbu Y. Masing-masing *controller* ditargetkan untuk mengendalikan pergerakan membentuk lingkaran. Total *error* pada NCTF *controller* sebesar 26.74 μm sedangkan pada PID *controller* sebesar 60.09 μm . Hasil dari penelitian ini menunjukkan performa kinerja NCTF *controller* lebih efektif dalam mengatasi nilai *error* daripada PID *controller*.

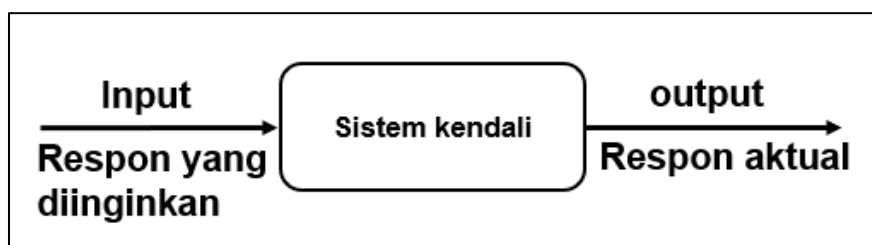
Pada penelitian (Jalani, 2011) dijelaskan tentang aplikasi NCTF *controller* pada sistem kendali posisi menggunakan mekanisme *rack and pinion* yang terhubung dengan motor DC sebagai objek kendali. Kinerja NCTF *controller*

dibandingkan dengan kinerja *propotional velocity* (PV) *controller* untuk mengetahui efektifitas kedua *controller* tersebut. Berdasarkan gerak objek kendali 50 mm *overshoot* pada NCTF *controller* menunjukkan hasil yang mendekati 0 untuk simulasi dan eksperimen, simulasi sebesar 0 % dan eksperimen sebesar 0.026 %. *Overshoot* pada simulasi PV *controller* sebesar 7.323 % sedangkan eksperimen sebesar 10.194 %. *Rise time* pada PV *controller* menunjukkan hasil yang lebih baik daripada NCTF *controller* dan untuk *settling time* keduanya memiliki nilai yang hampir sama. Pada simulasi kedua *controller* mendapatkan nilai nol untuk *steady state error* namun pada eksperimen NCTF *controller* menunjukkan hasil yang lebih baik. Kesimpulan dari penelitian ini kinerja NCTF *controller* lebih efektif digunakan pada sistem kendali posisi daripada PV *controller*.

2.2 Dasar Teori

2.2.1 Sistem Kendali

Sistem kendali (*control system*) adalah suatu sistem yang dirancang untuk mengendalikan objek kendali untuk mendapatkan keluaran (*output*) yang diinginkan dengan memberi masukan (*input*) tertentu (Hudati et al., 2021). Sistem kendali bertugas untuk mengatur variabel-variabel dalam sistem agar tetap berada pada nilai yang ditentukan meskipun terdapat gangguan eksternal. Gambar 2-1 menunjukkan skema sederhana dari sistem kendali.



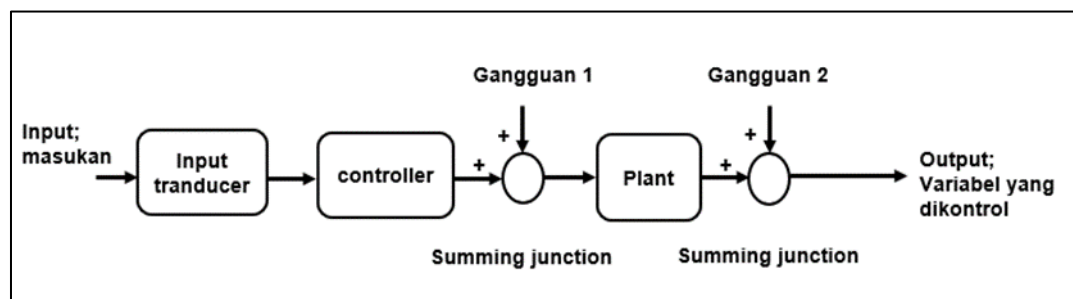
Gambar 2-1 Sistem kendali sederhana

Terdapat beberapa alasan utama digunakannya sistem kendali yaitu efisiensi energi, pengendalian yang akurat, kemampuan mengatasi gangguan dan fleksibilitas bentuk *input*. Sistem kendali terbagi menjadi dua bentuk utama yaitu

sistem *open-loop* dan sistem *close-loop*. Sistem kendali terdiri dari beberapa bagian penting yang bekerja secara terintegrasi untuk memastikan bahwa sistem dapat berfungsi sesuai dengan tujuan pengendalian. Komponen utama pada sistem kendali meliputi *input*, *controller*, *plant*, *output*, aktuator, sensor dan umpan balik (*feedback*).

2.2.2 Konfigurasi *Open-loop*

Sistem *open-loop* pada sistem kendali mangacu pada sistem dimana sinyal kendali *input* diberikan tanpa adanya umpan balik dari *output* sistem. Dalam sistem ini, diasumsikan bahwa sistem akan merespon sesuai dengan instruksi yang diberikan tanpa adanya penyesuaian terhadap perbedaan antara *output* yang diinginkan dengan *output* aktual. Alur kerja sistem *open-loop* dapat dipahami dengan ilustrasi pada Gambar 2-2. Masukan (*input*) diterima oleh *input tranducer*, selanjutnya masukan diteruskan pada sebuah *controller*. *Controller* menerima masukan tersebut kemudian akan diproses pada *plant* untuk menghasilkan keluaran (*output*) sesuai dengan masukan yang diberikan. Sistem *open-loop* hanya bergantung pada *input* yang sudah ditentukan dan tidak mempertimbangkan perubahan atau gangguan yang dapat terjadi pada sistem.



Gambar 2-2 Sistem *open-loop*

Input

Secara umum, *input* adalah variabel yang mempengaruhi operasi sistem kendali dan menjadi titik awal untuk proses pengendalian. *Input* pada sistem kendali merujuk pada sinyal atau data yang diberikan pada sistem kendali, *input* tersebut diproses dalam sistem untuk menghasilkan *output* yang diinginkan. *Input*

sangat penting dalam sistem kendali karena menjadi dasar untuk perbandingan dengan nilai *output* aktual. Sistem menggunakan data perbedaan antara nilai *input* dan *output* aktual untuk menghasilkan sinyal kendali yang dapat mengoreksi *error* dan memaksa sistem kembali menuju nilai yang diinginkan.

Input Transducer

Input transducer adalah perangkat yang berfungsi untuk mengubah besaran fisik atau energi dari lingkungan menjadi sinyal listrik yang dapat dipahami oleh sistem kendali. *Input transducer* bertindak sebagai perantara antara sistem kendali dan proses atau objek yang dikendalikan, dengan cara mengkonversi variabel fisik seperti kecepatan dan posisi menjadi sinyal yang dapat diproses lebih lanjut. *Input transducer* memungkinkan sistem kendali untuk menerima informasi yang diperlukan dari lingkungan atau *plant* sehingga kendali dapat dilakukan secara tepat dan efektif.

Controller

Dalam sistem kendali *open-loop*, *controller* berperan untuk mengirim sinyal kendali ke *plant* atau proses yang dikendalikan tanpa memperhatikan umpan balik dari hasil *output*. *Controller* bekerja berdasarkan perhitungan atau instruksi yang telah ditentukan sebelumnya, menghasilkan sinyal kendali yang diteruskan ke *plant* (objek kendali). Karena tidak ada mekanisme umpan balik untuk memantau *output* yang dihasilkan *controller* tidak melakukan penyesuaian atau koreksi meskipun terjadi perubahan pada hasil *output*. Dengan begitu kendali dilakukan tanpa memperhitungkan kondisi aktual dari sistem yang sedang dikendalikan. Pada sistem kendali *close-loop*, *controller* melakukan penyesuaian berdasarkan umpan balik yang diterima oleh *output*.

Gangguan (disturbance)

Gangguan (*disturbance*) adalah faktor eksternal yang mempengaruhi kinerja sistem kendali dan menyebabkan perbedaan antara *output* yang diinginkan dengan *output* aktual. Gangguan ini bisa berupa gesekan, perubahan beban, perubahan tegangan, akselerasi dan deselerasi. Untuk mengatasi pengaruh

gangguan, sistem kendali perlu dirancang agar dapat mengurangi dampaknya. Gangguan harus dipertimbangkan pada sistem kendali agar sistem tetap stabil.

***Plant* atau Objek Kendali**

Plant dalam sistem kendali merujuk pada bagian atau objek yang dikendalikan, yaitu komponen fisik yang beroperasi sesuai dengan *input* yang diberikan oleh sistem kendali. Contoh *plant* dalam sistem kendali dapat berupa motor DC. *Output* yang dihasilkan oleh motor DC adalah hasil yang ingin dicapai oleh sistem tersebut, hasilnya dapat berupa kecepatan dan posisi sudut dari motor DC. *Plant* menjadi elemen utama yang perlu dikendalikan dalam suatu sistem, dimana peran *controller* adalah memberikan sinyal kendali yang tepat untuk memastikan *output* dari *plant* sesuai dengan tujuan yang telah ditetapkan.

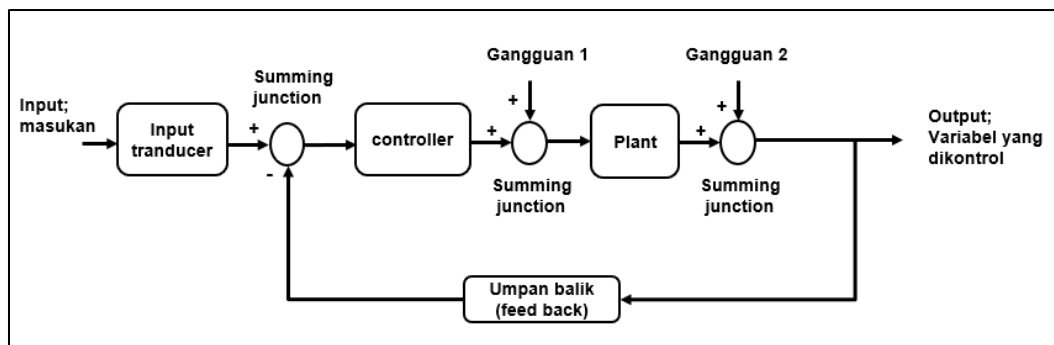
Output

Dalam sistem kendali, *output* merujuk pada variabel yang dihasilkan oleh *plant* atau proses yang sedang dikendalikan, yang merupakan hasil dari pengaruh sinyal kendali yang diberikan oleh *controller*. *Output* adalah nilai yang ingin dipertahankan atau dikendalikan agar tetap pada nilai yang diinginkan sesuai dengan tujuan sistem. Variabel *output* dapat berupa kecepatan dan posisi tergantung pada jenis sistem yang digunakan. Dalam sistem kendali nilai *output* yang terukur akan dibandingkan dengan nilai *input* yang ditetapkan. Jika terdapat perbedaan antara nilai *output* yang terukur dengan nilai *input* referensi yang diinginkan, *controller* akan berusaha mengurangi kesalahan (*error*) dengan menyesuaikan sinyal kendali yang diteruskan ke *plant*. Dalam sistem *open-loop*, *output* tidak digunakan untuk mengkoreksi atau mempengaruhi *input*, sementara pada sistem *close-loop*, *output* digunakan untuk memberikan umpan balik yang memungkinkan penyesuaian kendali secara dinamis.

2.2.3 Konfigurasi *Close-loop*

Dilihat pada gambar 2-3 terdapat perbedaan antara konfigurasi *open-loop* dan konfigurasi *close-loop*. Pada konfigurasi *close-loop* terdapat umpan balik (*feedback*) yang dapat mengatur kinerja sistem dan berpengaruh terhadap *input*.

Gangguan yang terjadi pada sistem *close-loop* dapat diatasi dengan adanya umpan balik yang dihasilkan dari perbandingan nilai *output* aktual dengan nilai *input* yang diinginkan. Perbedaan nilai *output* aktual dengan nilai *input* dinyatakan sebagai *error*. Nilai umpan balik atau *error* akan dikembalikan menuju *summing junction* yang selanjutnya akan diteruskan menuju *controller*. *Controller* akan menghasilkan sinyal kendali untuk mengurangi nilai *error* tersebut.



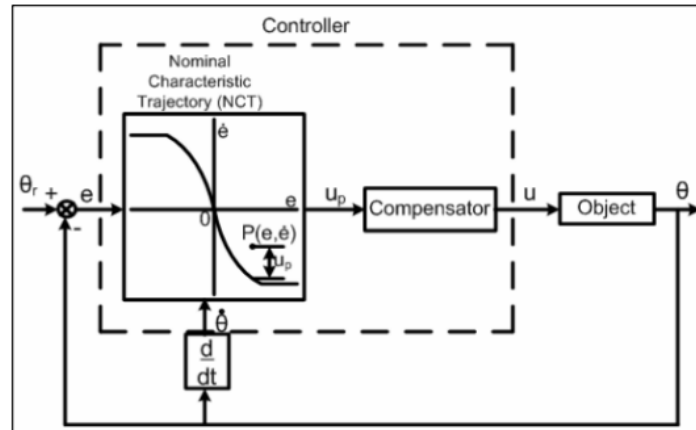
Gambar 2-3 Sistem *close loop*

2.2.4 Konsep NCTF

Struktur sistem kendali NCTF dapat dilihat pada Gambar 2-4. *Controller* terbagi menjadi dua bagian, yaitu *nominal characteristic trajectory* (NCT) dan *compensator*. Tujuan dari NCTF *controller* adalah membuat gerakan objek kendali mengikuti NCT kemudian berakhir pada titik *origin*-nya. NCT digunakan sebagai lintasan acuan gerakan objek kendali yang dibangun berdasarkan grafik yang didapatkan dari respon objek kendali. *Compensator* digunakan untuk mengolah data dari NCT kemudian dikirimkan ke objek kendali sehingga memaksa gerakan objek kendali mengikuti lintasan NCT (Purtojo, 2017).

Input dari *controller* adalah *error*, e , dan gerakan dari objek kendali, $\dot{x}$. Prinsip kerja *controller* adalah dengan membandingkan *input* gerak objek kendali dengan data *error-rate*, $\dot{e}$, yang didapat dari NCT. Perbedaan titik koordinat antara *error-rate* aktual dari pergerakan objek kendali dengan titik koordinat yang berada pada NCT adalah sinyal u_p . Sinyal u_p akan bernilai nol apabila objek kendali bergerak mengikuti NCT secara sempurna. Jika sinyal u_p tidak bernilai nol maka *compensator* akan memberikan aksi sampai nilai u_p menjadi nol atau dengan kata

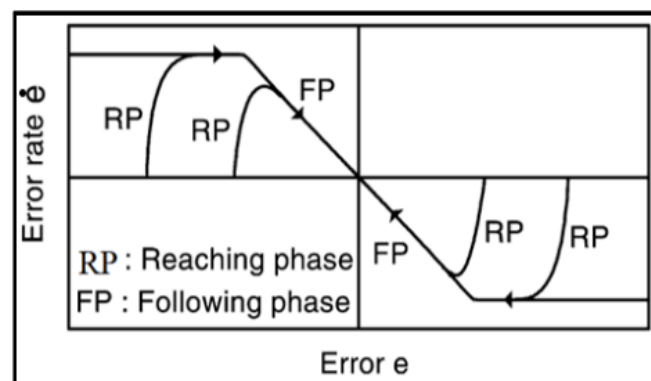
lain objek kendali sudah bergerak mengikuti NCT. Sinyal kendali u adalah *output* dari *controller* yang digunakan untuk menggerakkan objek kendali.



Gambar 2-4 Struktur sistem kendali NCTF

Sumber : (Purtojo, 2017)

Dalam proses pengendalian, *compensator* berfungsi untuk mengatur gerakan objek kendali agar dapat bergerak sesuai dengan lintasan NCT. Gerakan objek kendali pada lintasan NCT ditunjukkan pada Gambar 2-5, terdapat dua fase pada gerakan tersebut yaitu *reaching phase* dan *following phase*. Fase pertama, *reaching phase* adalah kondisi ketika objek kendali bergerak dari titik koordinat awal menuju lintasan NCT. Fase kedua, *following phase* adalah kondisi ketika objek kendali telah berada pada lintasan NCT dan bergerak mengikuti lintasan NCT kemudian berakhir pada titik *origin*.



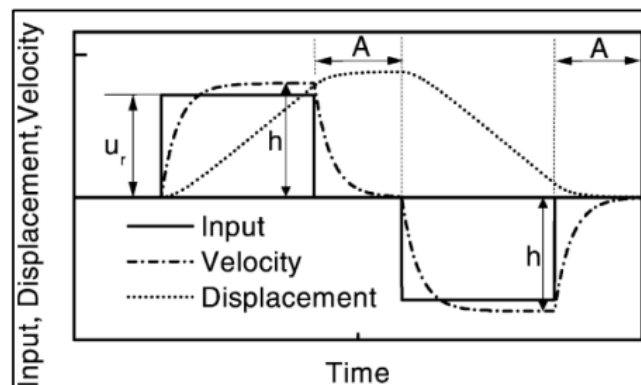
Gambar 2-5 NCT dan pergerakan objek kendali

Sumber : (Wahyudi et al., 2003)

Motor DC diasumsikan sebagai aktuator yang digunakan pada penurunan konsep sistem kendali NCTF. Karakteristik aktuator pada saat deselerasi atau perlambatan mempengaruhi keakuratan akhir operasi kendali posisi PTP. Karakteristik tersebut dapat digunakan untuk mendesain *controller*.

2.2.5 Ekperimen *Open-loop*

Eksperimen *open-loop* yang sederhana dan praktis dilakukan untuk mendapatkan karakteristik objek kendali. Eksperimen *open-loop* dilakukan dengan memberikan *input* sinyal *stepwise* pada objek kendali, dengan demikian data perpindahan dan kecepatan dapat diperoleh seperti yang diilustrasikan pada Gambar 2-6. Nilai *input* terhadap objek kendali disimbolkan dengan u_r , A merupakan area deselerisasi objek kendali yang berisikan data kecepatan maksimum menuju kecepatan nol. Kecepatan maksimum pada objek kendali disimbolkan dengan h .



Gambar 2-6 *Input stepwise* dan respons objek kendali

Sumber : (Purtojo, 2017)

Berdasarkan data respons objek kendali diperoleh kecepatan maksimum objek kendali dan data perlambatan atau deselerasi. Hal yang diperhatikan dalam sistem kendali posisi PTP adalah menghentikan objek kendali pada posisi tertentu. Dengan demikian karakteristik perlambatan digunakan. Karakteristik non-linieritas seperti gesekan dan saturasi pada sistem dapat diidentifikasi pada saat proses perlambatan. Sehingga data yang diperoleh dapat mewakili karakteristik sistem.

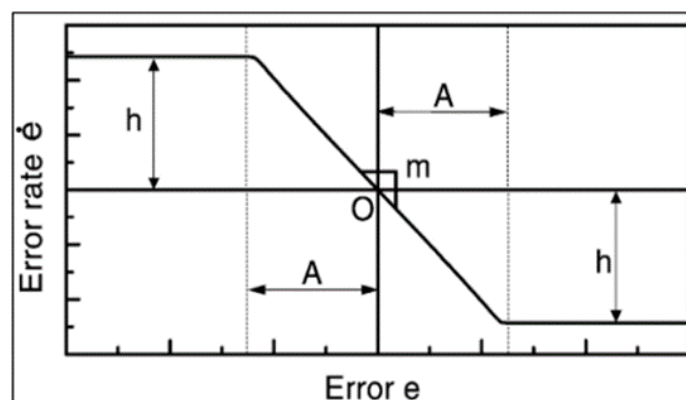
2.2.6 Pembuatan NCT

NCT dibangun melalui analisis grafik yang mencatat respon kecepatan objek kendali dan perpindahan objek kendali dalam wilayah A, sebagaimana ditampilkan dalam Gambar 2-7. Data kecepatan dan perpindahan pada saat memasuki proses perlambatan disimpan kemudian diolah sehingga dapat ditampilkan dalam bentuk lintasan NCT. Kemiringan (*gradien*) disekitar titik *origin* pada lintasan NCT dinyatakan dengan m . Dengan mengabaikan karakteristik non-linier seperti *friction* dapat diasumsikan bahwa objek kendali termasuk dalam sistem orde dua (Nise, 2015) dengan persamaan berikut ini :

$$\frac{\theta(s)}{U(s)} = K \frac{\alpha}{s(s+\alpha)} \quad (2.1)$$

Pada persamaan (2.1) $\theta(s)$ merupakan representasi dari perpindahan objek kendali, $U(s)$ adalah *input* yang diberikan pada objek kendali, kemudian K dan α merupakan model sederhana dari parameter objek kendali. Berdasarkan asumsi bahwa motor DC berfungsi sebagai aktuator yang menggerakkan sistem kendali model sederhana dari parameter objek kendali dapat digunakan.

Gradien m disekitar titik *origin* lintasan NCT berhubungan dengan model sederhana dari parameter objek kendali α . Berdasarkan NCT pada Gambar 2-7 diperoleh,



Gambar 2-7 Nominal characteristic trajectory (NCT)

Sumber : (Purtojo, 2017)

$$\frac{d\dot{e}}{de} = m \tag{2.2}$$

$$\frac{d\dot{e}}{de} = -\alpha \tag{2.3}$$

$$m = -\alpha \tag{2.4}$$

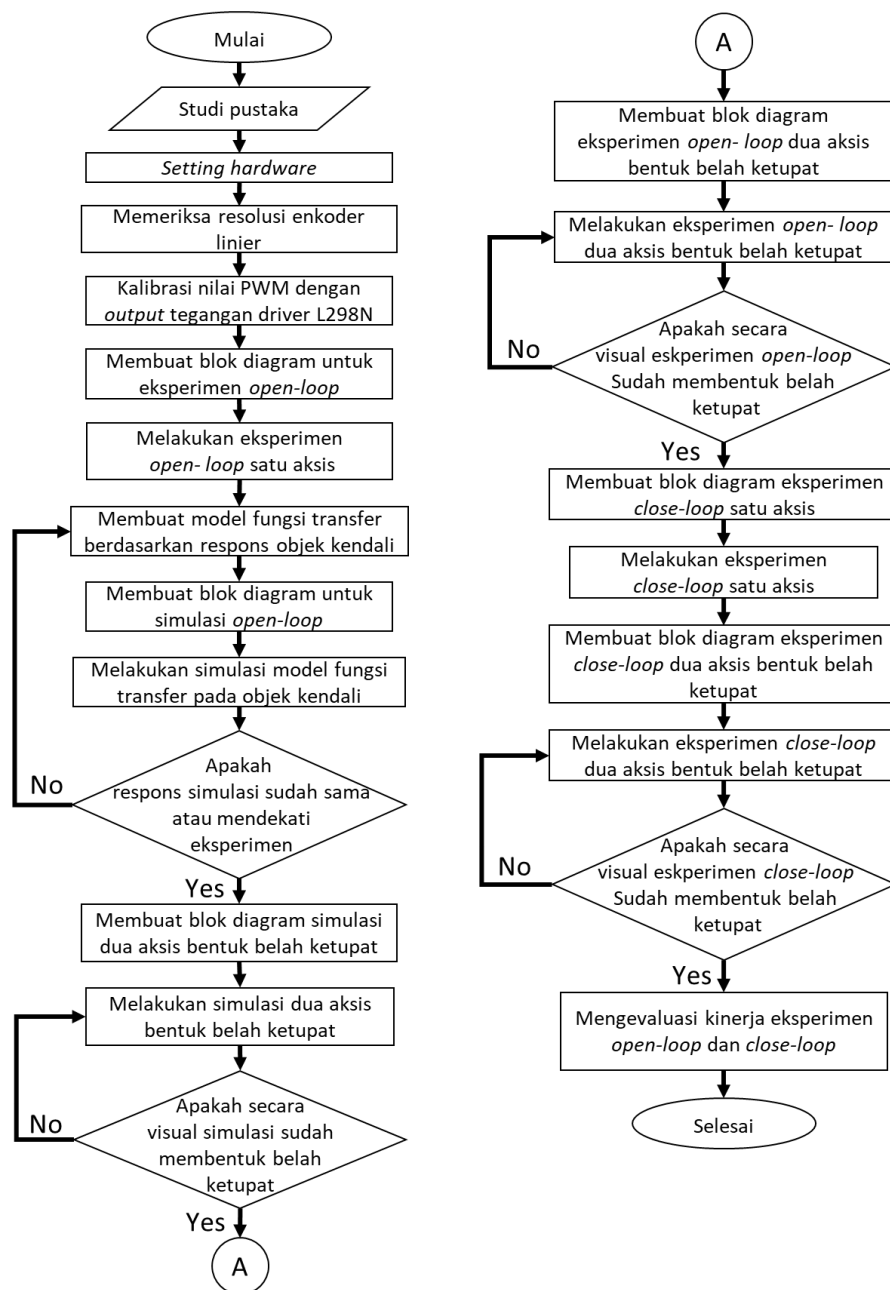
$$K = \frac{h}{u_r} \tag{2.5}$$

BAB 3

METODE PENELITIAN

3.1 Alur Penelitian

Diagram alir pada Gambar 3-1 menunjukkan tahapan-tahapan yang dilakukan pada penelitian ini.



Gambar 3-1 Diagram alir penelitian

3.2 Peralatan

Pada penelitian ini terdapat dua jenis peralatan yang digunakan yaitu, perangkat keras (*hardware*) dan perangkat lunak (*software*).

3.2.1 Perangkat Keras

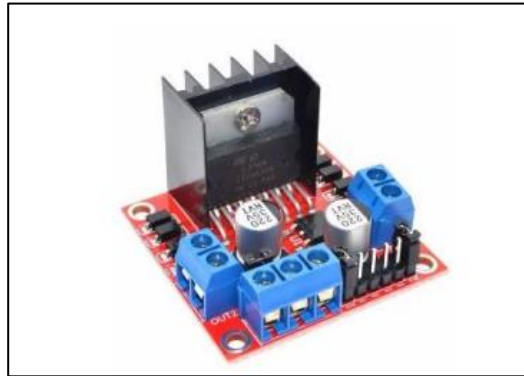
Pada penelitian ini perangkat keras yang digunakan adalah :

1. Arduino mega 2560 digunakan sebagai *controller-card*. (Gambar 3-2)
2. Motor Driver L298N digunakan sebagai pengatur gerakan motor DC. (Gambar 3-3)
3. Maxon DC motor tipe 126691 digunakan sebagai aktuator. (Gambar 3-4)
4. *Linear ball-screw* digunakan sebagai bagian aktuator yang merubah gerakan *rotary* dari motor DC menjadi gerakan linier. Pada penelitian ini digunakan dua *linear ball-screw* dengan panjang masing-masing 22 cm untuk sumbu X dan 11 cm untuk sumbu Y. (Gambar 3-5)
5. RLS LM10IC001AA10K12 adalah linier enkoder yang digunakan sebagai sensor untuk membaca perubahan posisi dari aktuator. (Gambar 3-6 dan Gambar 3-8)
6. Kabel *jumper* digunakan sebagai penghubung dari rangkaian perangkat keras. (Gambar 3-8)
7. *Project board* digunakan sebagai pencabangan antara kabel *jumper* dan perangkat keras. (Gambar 3-9)
8. *Power supply* (Input 220 VAC, Output 24 VDC) digunakan sebagai penyedia daya untuk aktuator dan motor *driver*. (Gambar 3-10)



Gambar 3-2 Arduino Mega 2560

Sumber:(<https://www.circuitstoday.com>)



Gambar 3-3 Driver L298N

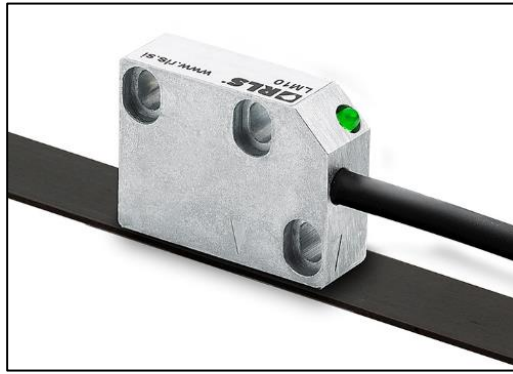
Sumber:(<https://www.circuitstoday.com>)



Gambar 3-4 Maxon Motor DC



Gambar 3-5 Linier *ball-screw*



Gambar 3-6 Enkoder Linier

Sumber:(<https://www.rls.si>)

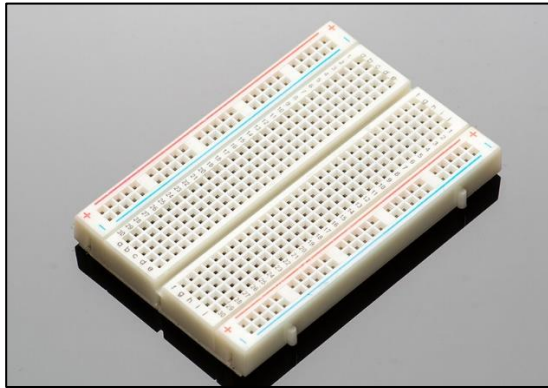


Gambar 3-7 Posisi Enkoder pada linier *ball-screw*



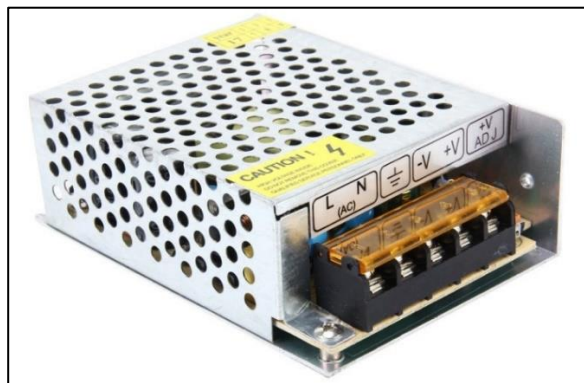
Gambar 3-8 Kabel *jumper*

Sumber:(<https://www.conrad.com>)



Gambar 3-9 Project board

Sumber:(<https://learn.adafruit.com>)



Gambar 3-10 Power supply

Sumber:(<https://www.melesoft.com>)

3.2.2 Perangkat Lunak

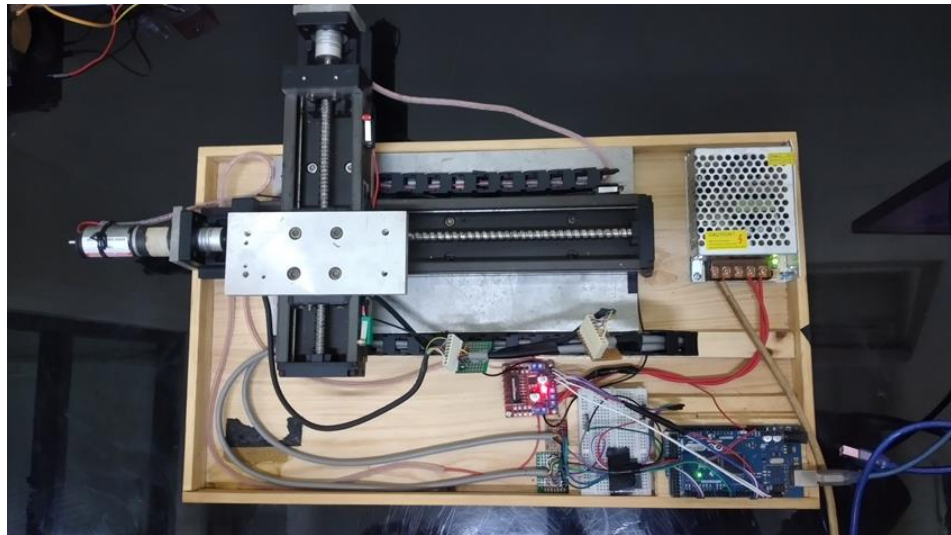
Berikut ini adalah perangkat lunak yang digunakan pada penelitian :

1. Arduino IDE (*Integrated Development Environment*) 2.3.2 digunakan untuk mengunggah *source-code* Arduino *toolbox* pada Arduino mega 2560.
2. Scilab 6.1.1 digunakan untuk membuat blok diagram.

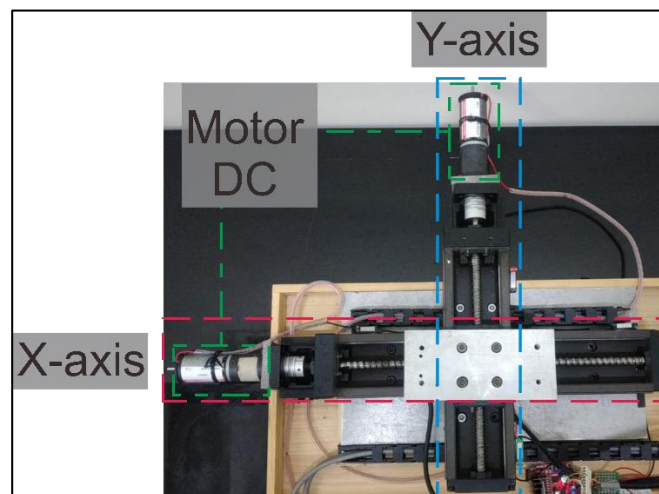
3.2.3 Perancangan Perangkat Keras

Gambar 3-11 menunjukkan hasil rangkaian perangkat keras. Dua motor DC digunakan untuk masing-masing aktuator yaitu aktuator sumbu X dan aktuator sumbu Y. Aktuator sumbu Y terpasang pada *work block* aktuator sumbu X. Motor DC digunakan sebagai aktuator yang menerima sinyal dari Driver L298N, putaran

motor DC diteruskan untuk menggerakkan *ball screw*. Gerakan *rotary* dari motor DC diubah menjadi gerakan linier melalui *ball screw*. Data perubahan posisi pada *ball screw* disimpan oleh linier enkoder yang terpasang pada *work block*. Gambar 3-12 menunjukkan keterangan aksis pada masing-masing aktuator.



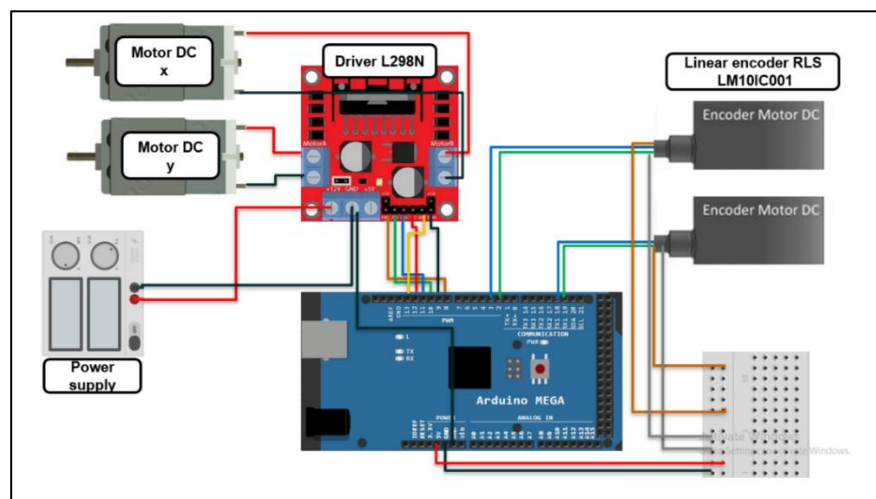
Gambar 3-11 Hasil rangkaian perangkat keras



Gambar 3-12 Aksis pada aktuator

Gambar 3-13 menunjukkan rangkaian koneksi antar perangkat keras. *Power supply* menyediakan tegangan sebesar 24 volt untuk menggerakkan motor DC terhubung melalui Driver L298N untuk mengatur kecepatan dan arah putaran motor DC. Fungsi masing-masing pin pada Driver L298N dapat dilihat pada Tabel

3-1. Pin 24 volt (positif) pada *power supply* terhubung dengan pin 12 volt (positif) pada Driver L298N dan pin *ground* (negatif) pada *power supply* terhubung dengan pin *ground* (negatif) pada Driver L298N. Terminal positif maupun negatif pada motor DC untuk sumbu X dan motor DC untuk sumbu Y berturut-turut terhubung dengan pin motor A dan pin motor B pada Driver L298N. Pin EN A dan pin EN B pada Driver L298N dihubungkan dengan pin PWM 8 dan 9 pada Arduino mega 2560 untuk mengendalikan kecepatan motor DC. Pengaturan arah putaran motor DC dilakukan dengan menghubungkan pin In 1, In 2, In 3 dan In 4 pada Driver L298N dengan pin digital 10, 11, 12, 13 pada Arduino mega 2560. Pada penelitian ini menggunakan dua linier enkoder yang terhubung dengan Arduino mega 2560 sebagai sensor untuk membaca perubahan posisi dan kecepatan motor DC. Secara berturut-turut pin 2, 3, 19 dan 18 dihubungkan dengan pin inkremental A dan B pada masing-masing enkoder. Secara berurut pin 5 volt dan pin *ground* pada Arduino mega 2560 terhubung dengan sinyal 5 volt dan *ground* pada masing-masing enkoder. Keterangan warna kabel pada linier enkoder dapat dilihat pada Tabel 3-2 yang didapatkan berdasarkan *datasheet* LM10D01_17 yang tersedia pada situs pembuat enkoder.



Gambar 3-13 Skema rangkaian koneksi perangkat keras

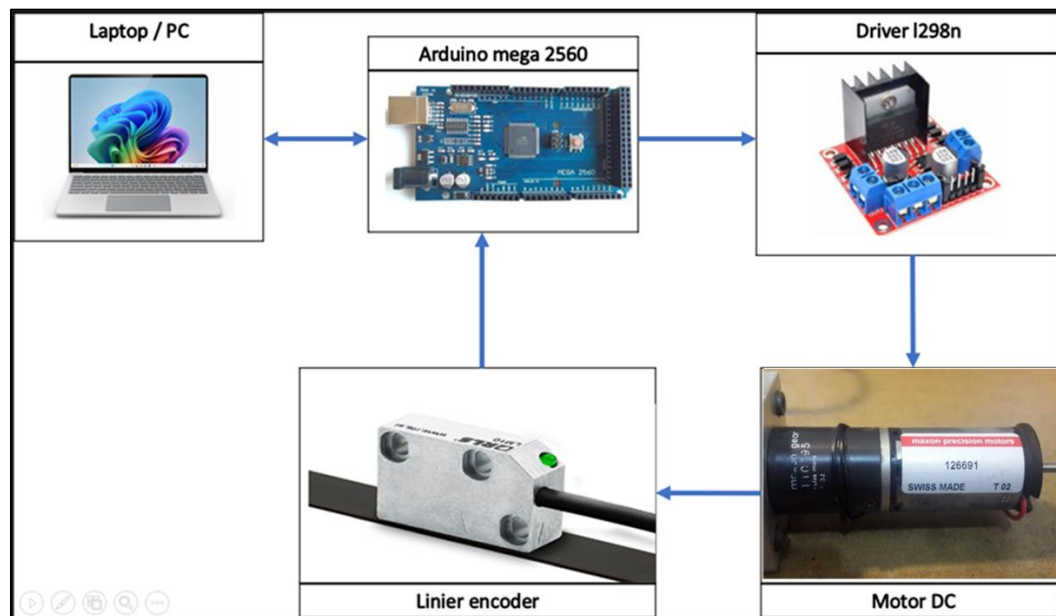
Tabel 3-1 Tabel fungsi pin pada Driver L298N

Pin	Fungsi
Pin 12 volt	Memberi tegangan pada motor yang terhubung sebesar 3.2 volt – 35 volt
Ground (GND)	Pin GND digunakan untuk menghubungkan rangkaian ke sistem ground
Motor A out 1 dan out 2	Menghubungkan driver dengan kutub motor DC yang terhubung
Motor B out 3 dan out 4	Menghubungkan driver dengan kutub motor DC yang terhubung
In 1	Mengendalikan arah putaran Motor A Counter Clockwise
In 2	Mengendalikan arah putaran Motor A Clockwise
In 3	Mengendalikan arah putaran Motor B Counter Clockwise
In 4	Mengendalikan arah putaran Motor B Clockwise
EN A	Mengendalikan kecepatan Motor A
EN B	Mengendalikan kecepatan Motor B

Tabel 3-2 Konfigurasi warna kabel linier enkoder RLS LM10IC001AA10K

Funtion	signal	colour (option F)
Power	5 V	Brown
	0 V	White
Incremental signal	A	Green
	A-	Yellow
	B	Blue
	B-	Red
Reference signals	Z	Pink
	Z-	Gray
Alarm	E	Violet
	E-	Black

Rangkaian proses alur kerja ditunjukkan Gambar 3-14. Arduino mega 2560 digunakan sebagai penjematan komunikasi antara laptop dengan Driver L298N dan enkoder. Laptop mengirimkan sinyal ke Driver L298N melalui Arduino mega 2560, selanjutnya Driver L298N akan meneruskan sinyal yang diterima untuk menggerakkan motor DC. Gerakan motor DC menyebabkan *work block* ikut bergerak. Perubahan posisi pada *work block* akan di proses oleh enkoder sehingga menghasilkan sinyal inkremental. Enkoder mengirimkan sinyal inkremental menuju Arduino mega 2560. Arduino mega 2560 memproses sinyal tersebut kemudian mengirimkannya ke laptop.



Gambar 3-14 Rangkaian proses alur kerja

BAB 4

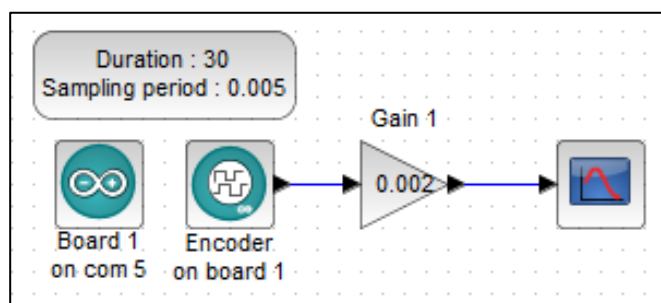
HASIL DAN PEMBAHASAN

4.1 Identifikasi Sistem Kendali Posisi Linier

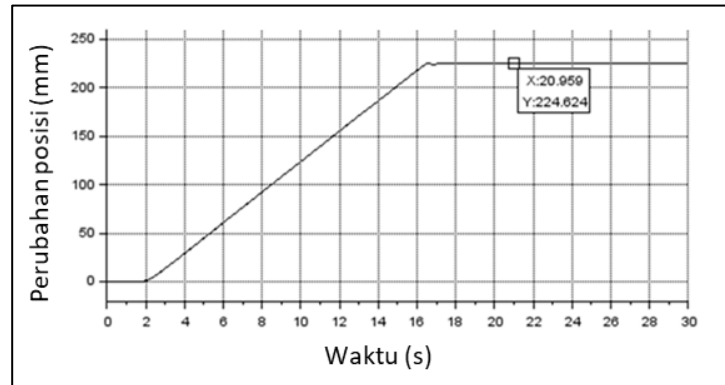
Identifikasi pada sistem kendali posisi linier dilakukan untuk mendapatkan fungsi transfer dari sistem kendali posisi linier.

4.1.1 Pemeriksaan Resolusi Enkoder Linier

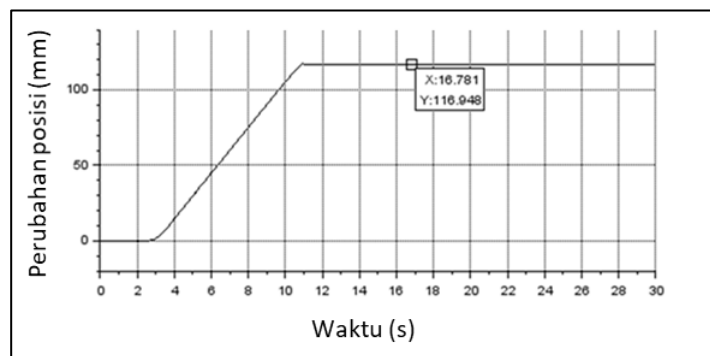
Enkoder linier digunakan untuk membaca perubahan posisi pada aktuator linier. Pemeriksaan enkoder linier dilakukan untuk menyesuaikan pembacaan satuan yang diinginkan. Resolusi enkoder linier sebesar $0.5 \mu m$. Gambar 4-1 menunjukkan blok diagram untuk pembacaan enkoder linier. Blok gain 1 pada Gambar 4-1 bertujuan untuk mengubah satuan dari μm menjadi mm. Gambar 4-2 menunjukkan hasil pengukuran panjang lintasan aktuator sumbu X sebesar 224.62 mm. Gambar 4-3 menunjukkan hasil pengukuran panjang lintasan aktuator sumbu Y sebesar 116.94 mm. Hasil pembacaan enkoder linier sudah sesuai dengan spesifikasi aktuator yang telah dijelaskan pada bagian 3.2.1.



Gambar 4-1 Blok diagram pembacaan enkoder linier



Gambar 4-2 Hasil pembacaan enkoder linier pada Scilab sumbu X



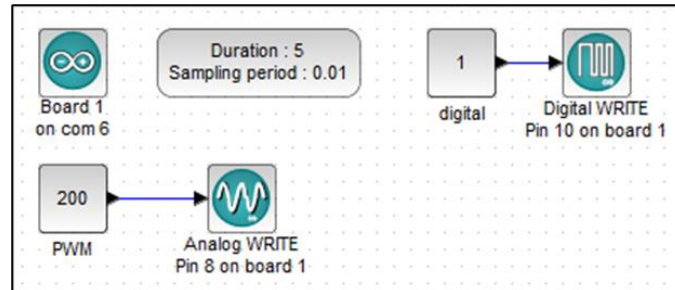
Gambar 4-3 Hasil pembacaan enkoder linier pada Scilab sumbu Y

4.1.2 Kalibrasi Nilai Tegangan Driver L298N dengan Nilai PWM

Motor DC dapat bergerak apabila diberi tegangan. Driver L298N dapat mengatur besar tegangan yang masuk ke motor DC. Oleh sebab itu, *input* pada objek kendali motor DC adalah tegangan yang keluar dari Driver L298N. *Controller-card* Arduino mega 2560 tidak dapat secara langsung mengirim sinyal tegangan menuju Driver L298N melainkan mengirim sinyal PWM (*pulse width modulation*). Sinyal PWM yang diterima oleh Driver L298N akan dikonversi menjadi tegangan yang akan diteruskan ke motor DC. Agar dapat mengetahui besar tegangan yang masuk ke motor DC, maka perlu dilakukan pengukuran antara sinyal PWM pada Arduino mega 2560 dengan besar tegangan yang keluar dari Driver L298N.

Gambar 4-4 menunjukkan blok diagram pada Scilab yang digunakan untuk pengukuran tegangan. Blok PWM pada Gambar 4-4 adalah besar nilai PWM yang diberikan Arduino mega 2560 ke pin EN A dan EN B pada Driver L298N. Blok

digital berisi nilai yang akan mengaktifkan pin In 1 maupun In 3 pada Driver L298N. Pengukuran dilakukan dengan mengukur tegangan yang keluar pada pin Driver L298N yang terhubung dengan motor DC menggunakan multimeter. Hasil kalibrasi tegangan dengan nilai PWM dapat dilihat pada Tabel 4-1.



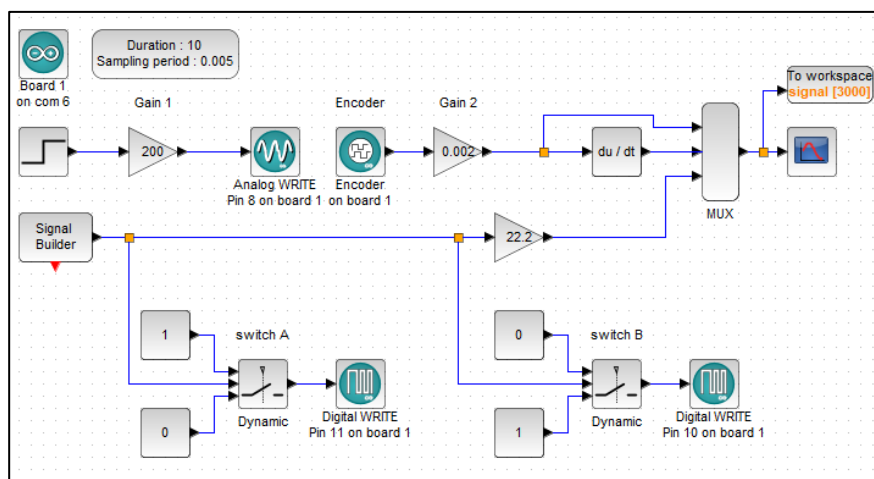
Gambar 4-4 Blok diagram pengukuran tegangan

Tabel 4-1 Kalibrasi nilai tegangan

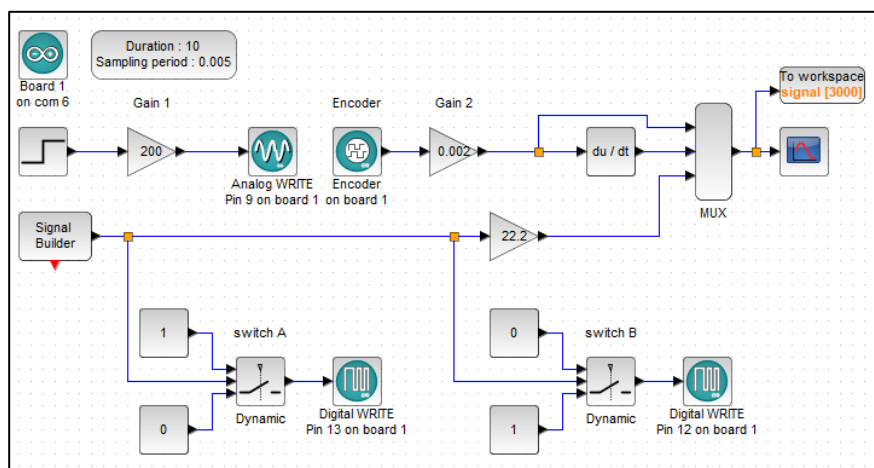
PWM	Tegangan (V)
26	14.9
35	17.3
50	19.2
65	20.2
80	20.8
95	21.2
110	21.4
125	21.7
145	21.8
155	21.9
170	22
185	22.1
200	22.2
215	22.3
229	22.3
240	22.4
255	22.4

4.1.3 Perangkaian Blok Diagram Eksperimen *Open-loop*

Gambar 4-5 dan Gambar 4-6 berturut- turut menunjukkan rangkaian blok diagram pada Scilab untuk eksperimen *open-loop* pada sumbu X dan sumbu Y. Blok *input stepwise* berisi nilai 1 yang akan masuk menuju blok *Gain* 1 untuk mewakili nilai PWM yang akan diterima oleh blok *Analog WRITE* pin 8. Nilai PWM yang digunakan adalah 200. Blok *Analog WRITE* pin 8 terhubung dengan pin EN A pada Driver L298N yang akan mengatur kecepatan motor DC pada sumbu X. *Input stepwise* pada sumbu Y terhubung dengan *Analog WRITE* pin 9 yang terhubung dengan pin EN B pada Driver L298N.



Gambar 4-5 Blok diagram eksperimen *open-loop* sumbu X



Gambar 4-6 Blok diagram eksperimen *open-loop* sumbu Y

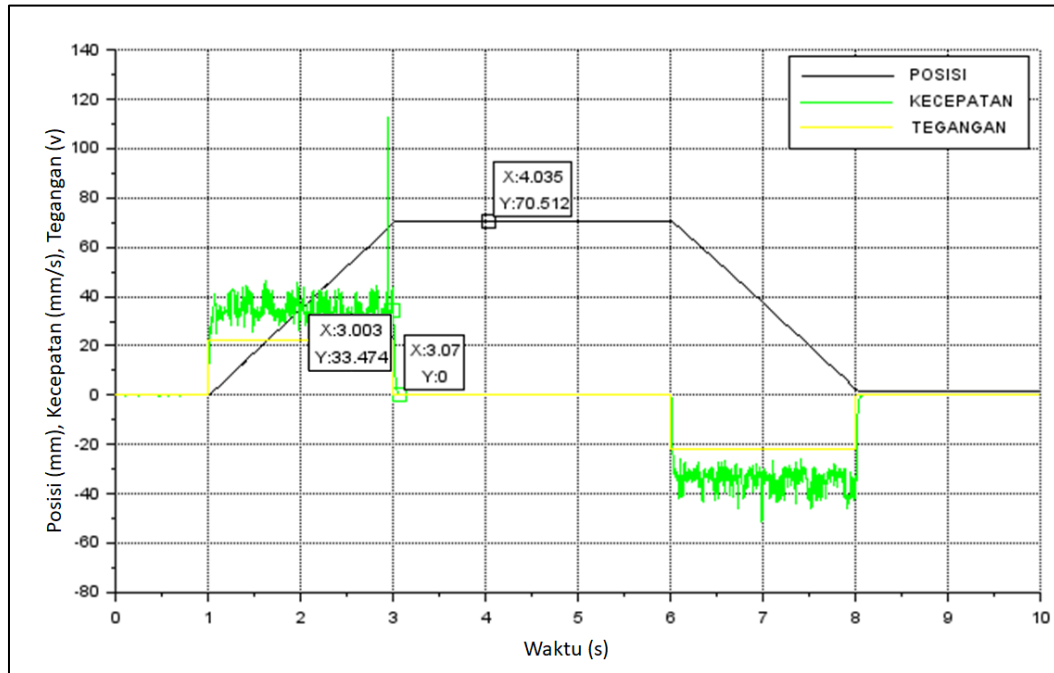
Pengaturan arah putaran motor DC menggunakan blok diagram *switch* yang terhubung dengan blok diagram *Digital WRITE*. Dengan menggunakan logika *high and low switch* mengatur *output* yang keluar bergantian antara nilai 1 dan 0. Sinyal yang keluar dari *signal builder* adalah 1, 0 dan -1, sinyal ini akan mempengaruhi *output* yang keluar dari *switch*. Tabel 4-2 menunjukkan konfigurasi putaran motor DC.

Tabel 4-2 Konfigurasi arah putaran motor DC

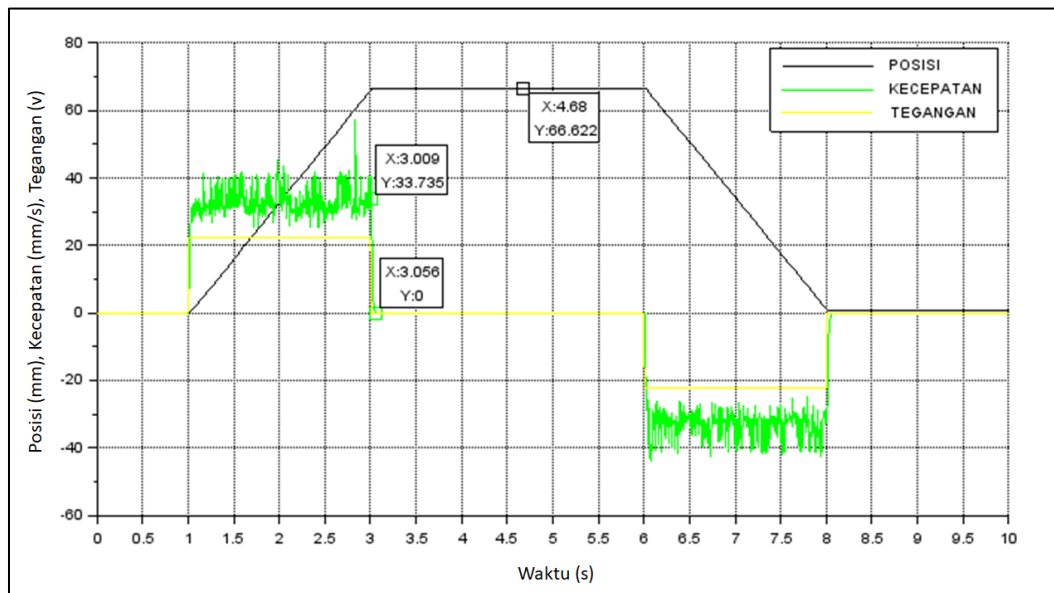
Signal Builder	Switch A	Switch B	Putaran
1	1	0	Clockwise (CW)
0	0	0	Berhenti
-1	0	1	Counter clockwise (CCW)

Blok enkoder pada Gambar 4-5 dan Gambar 4-6 digunakan untuk menyimpan nilai perubahan posisi yang terbaca oleh linier enkoder. Linier enkoder memiliki resolusi sebesar $0.5 \mu m$. Blok *Gain 2* digunakan untuk mengubah satuan pada enkoder menjadi *mm*.

Gambar 4-7 dan Gambar 4-8 menunjukkan grafik kecepatan, posisi dan tegangan. Kurva berwarna hijau merupakan kurva kecepatan dengan satuan *mm/s* dan kurva berwarna hitam merupakan kurva posisi dengan satuan *mm*. sedangkan kurva berwarna merah adalah tegangan dengan satuan *volt*.



Gambar 4-7 Respons kecepatan eksperimen *open-loop* sumbu X



Gambar 4-8 Respons kecepatan eksperimen *open-loop* sumbu Y

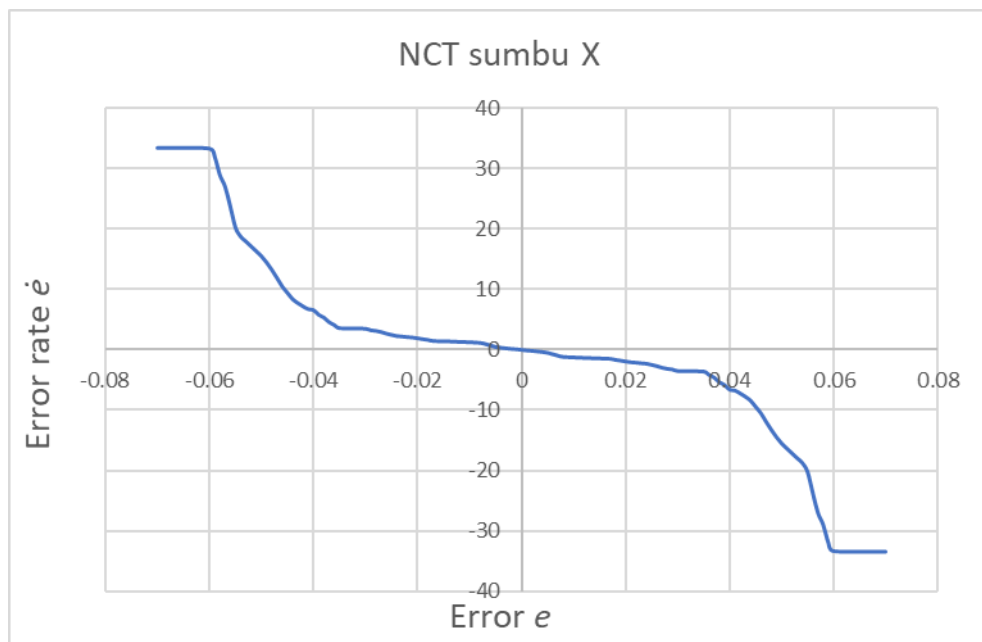
4.1.4 Konstruksi NCT

Pembentukan NCT didasarkan pada data perlambatan atau deselerisasi yang diperoleh melalui eksperimen *open-loop*. Eksperimen *open-loop*

disimulasikan dalam waktu 10 detik dengan menggunakan sampling periode sebesar 0.005 detik (Lihat Gambar 4-5 dan Gambar 4-6).

NCT pada sumbu X dikonstruksi dengan mengambil data eksperimen *open-loop* sumbu X pada detik ke 3 sampai detik ke 3.07. Dari data tersebut kemudian dibalik dimulai dari detik ke 3.07 sampai detik ke 3 (Lihat Gambar 4-7). Sehingga menghasilkan lintasan NCT yang dapat dilihat pada Gambar 4-9.

Berdasarkan penjelasan pada bagian 2.2.2 parameter NCT sumbu X dapat dilihat pada Tabel 4-3. Parameter NCT tersebut diperoleh berdasarkan data yang terdapat pada NCT sumbu X.



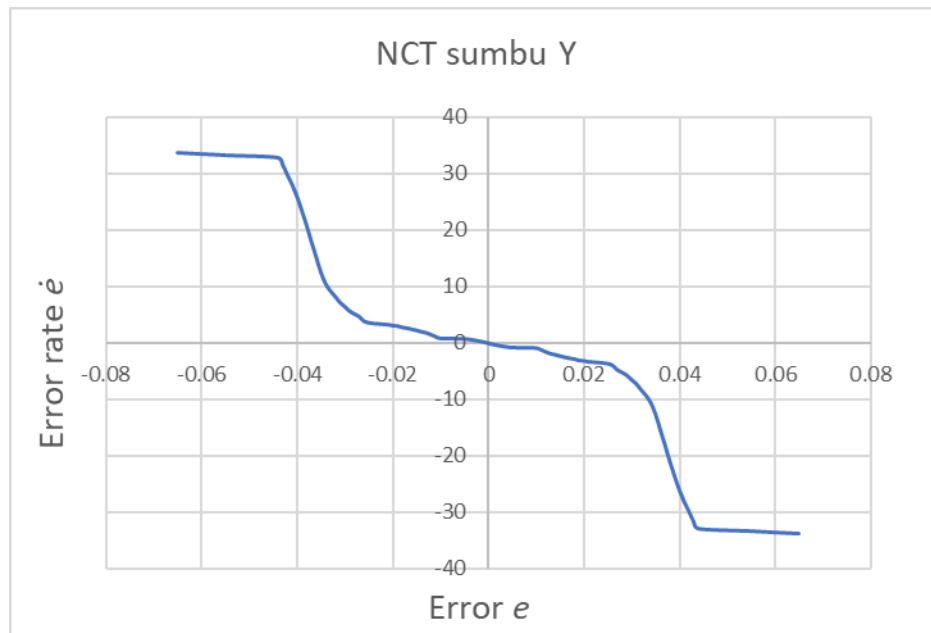
Gambar 4-9 Nominal Characteristic Trajectory (NCT) sumbu X

Tabel 4-3 Parameter NCT sumbu X

m_x	h_x (mm/s)	α_x	U_x (v)	K_x	" $K_x * \alpha_x$ "
-85.714	33.47	85.714	22.2	1.507	129.17

NCT pada sumbu Y dikonstruksi dengan mengambil data eksperimen *open-loop* sumbu Y pada detik ke 2.99 sampai detik ke 3.055. Dari data tersebut kemudian dibalik dimulai dari detik ke 3.055 sampai detik ke 2.99 (Lihat Gambar 4-8). Hasil konstruksi NCT sumbu Y dapat dilihat pada Gambar 4-10. Parameter

NCT sumbu Y diperoleh berdasarkan data NCT yang terdapat pada NCT sumbu X. Parameter NCT sumbu Y dapat dilihat pada Tabel 4-4.



Gambar 4-10 Nominal Characteristic Trajectory (NCT) sumbu Y

Tabel 4-4 Parameter NCT sumbu Y

m_y	h_y (mm/s)	α_y	U_y (v)	K_y	" $K_y * \alpha_y$ "
-179	33.73	179	22.2	1.519	271.9

4.1.5 Fungsi Transfer

Perhitungan fungsi transfer pada objek kendali menggunakan persamaan (2.1). Dengan menggunakan parameter NCT maka nilai fungsi transfer dapat diketahui. Pada sumbu X nilai fungsi transfer dihitung dengan memasukan parameter NCT pada Tabel 4-3 ke persamaan (2.1). Sehingga fungsi transfer pada sumbu X dapat dinyatakan dengan.

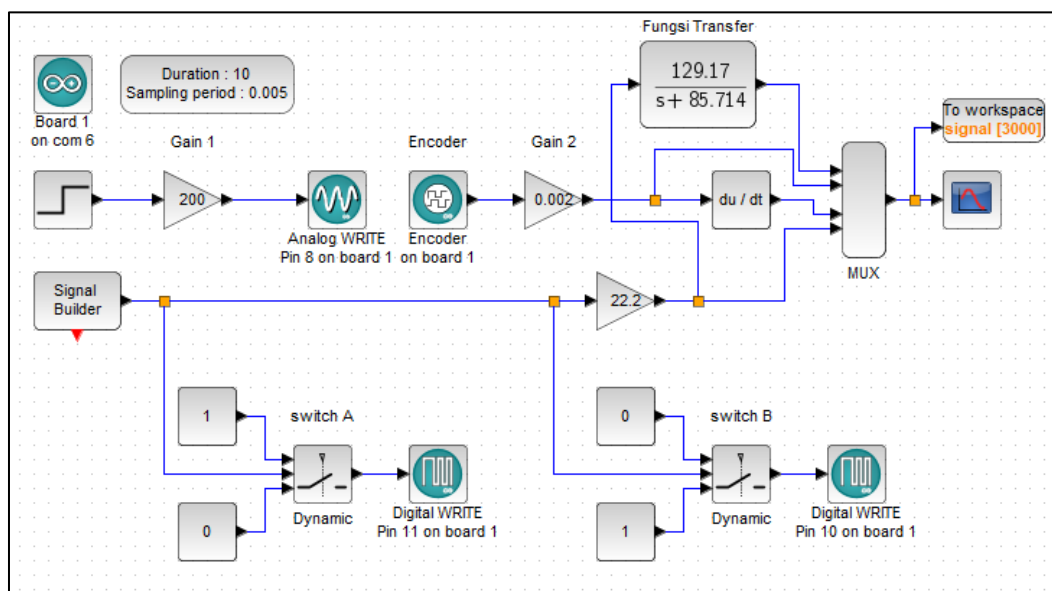
$$\frac{\theta(s)}{U(s)} = \frac{129.17}{s(s+85.714)} \quad (4.1)$$

Perhitungan fungsi transfer pada sumbu y dilakukan dengan memasukkan parameter NCT pada Tabel 4-4 ke persamaan (2.1). Sehingga fungsi transfer pada sumbu Y adalah

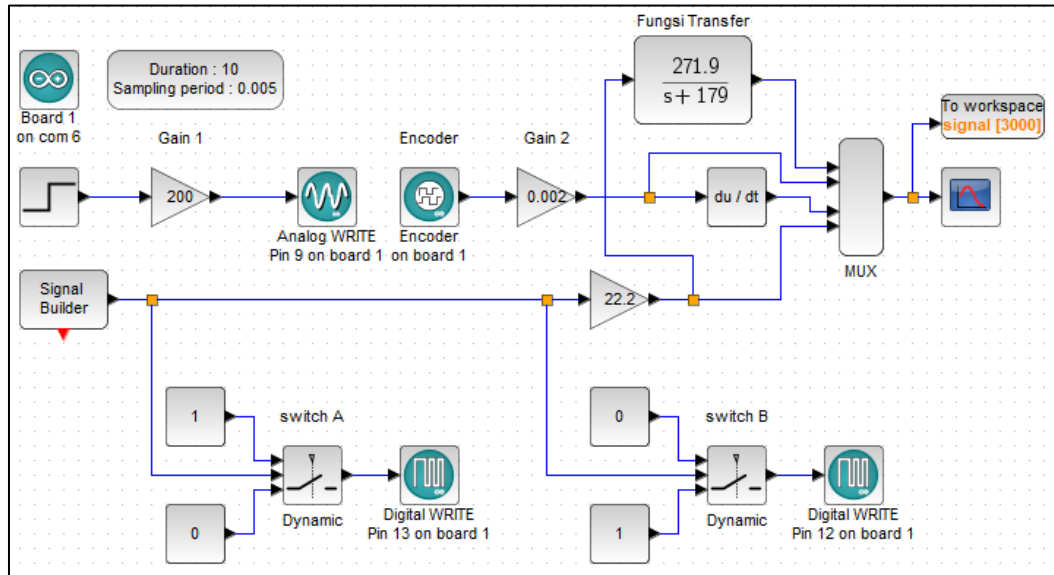
$$\frac{\theta(s)}{U(s)} = \frac{271.9}{s(s+179)} \quad (4.2)$$

Selanjutnya, setelah mendapatkan nilai fungsi transfer verifikasi dilakukan dengan mensimulasikan blok diagram seperti pada Gambar 4-11 untuk sumbu X dan Gambar 4-12 untuk sumbu Y. Verifikasi fungsi transfer dilakukan untuk mengetahui bahwa fungsi transfer sudah dapat mewakili karakteristik dari sistem kendali. Fungsi transfer dapat dinyatakan sudah mewakili karakteristik objek kendali apabila kecepatan simulasi sudah sama atau mendekati kecepatan eksperimen.

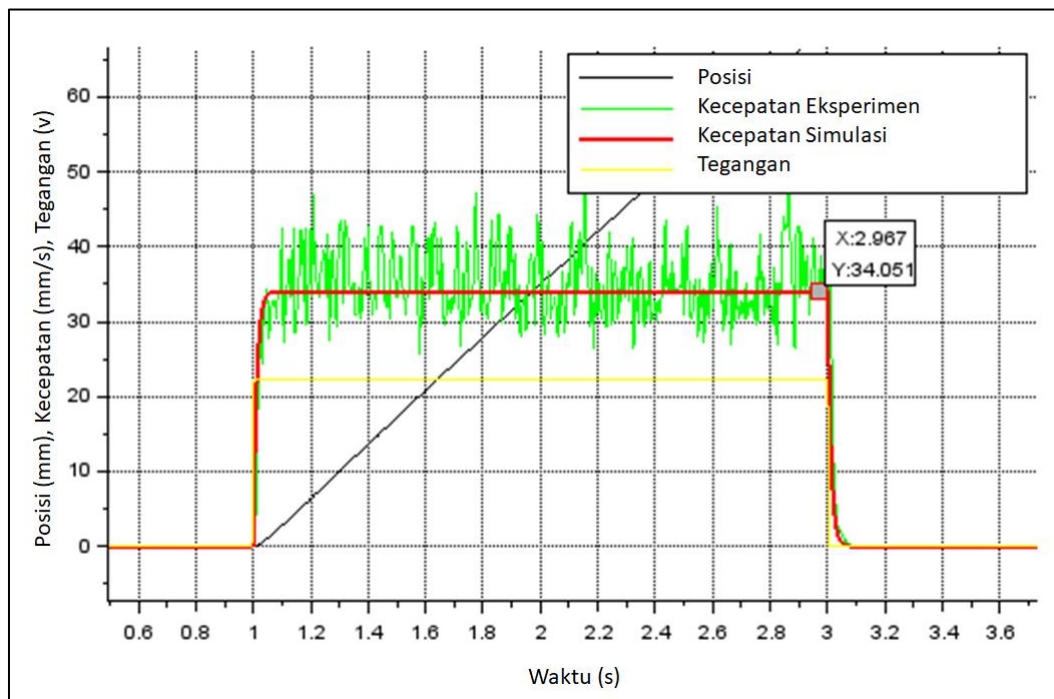
Gambar 4-13 dan Gambar 4-14 berturut-turut menunjukkan hasil simulasi fungsi transfer sumbu X dengan nilai $h_x = 34.051$ mm/s dan sumbu Y dengan nilai $h_y = 33.722$ mm/s.



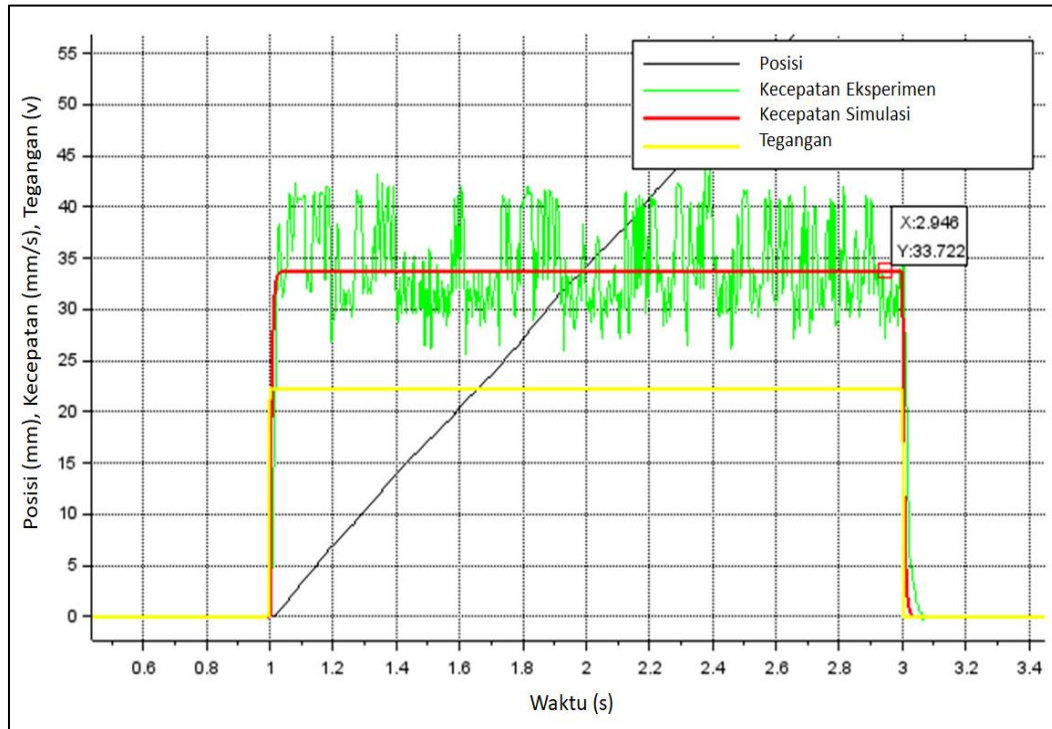
Gambar 4-11 Blok diagram kecepatan eksperimen dan simulasi sumbu X



Gambar 4-12 Blok diagram kecepatan eksperimen dan simulasi sumbu Y



Gambar 4-13 Respons kecepatan eksperimen dan simulasi sumbu X



Gambar 4-14 Respons kecepatan eksperimen dan simulasi sumbu Y

Berdasarkan respons kecepatan yang tersaji pada Gambar 4-13 dan Gambar 4-14 dapat dinyatakan bahwa fungsi transfer dapat mewakili sistem kendali posisi linier.

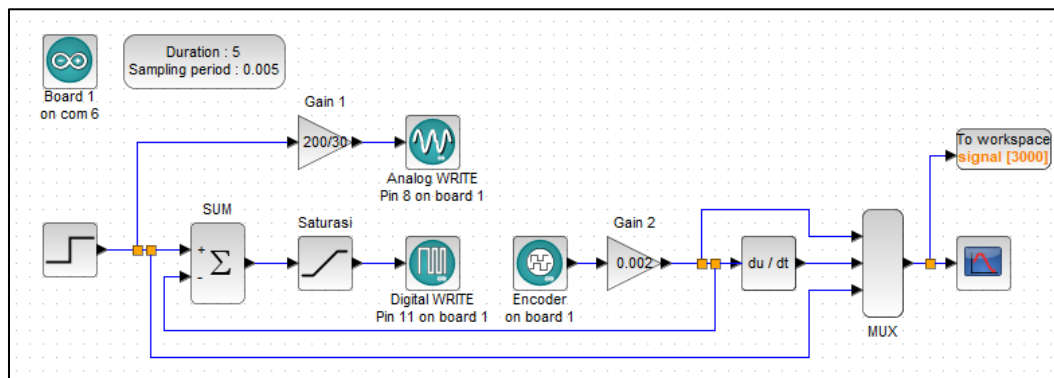
4.2 Kinerja Sistem Kendali Posisi PTP

Eksperimen satu aksis dengan konfigurasi *close-loop* dilakukan untuk mengetahui kinerja sistem kendali posisi PTP. Kinerja sistem kendali posisi PTP dievaluasi melalui eksperimen gerak objek kendali satu aksis untuk mengetahui nilai *error* pada masing-masing sumbu yaitu sumbu X dan sumbu Y. Eksperimen dilakukan dengan memberi *input* posisi pada masing-masing sumbu.

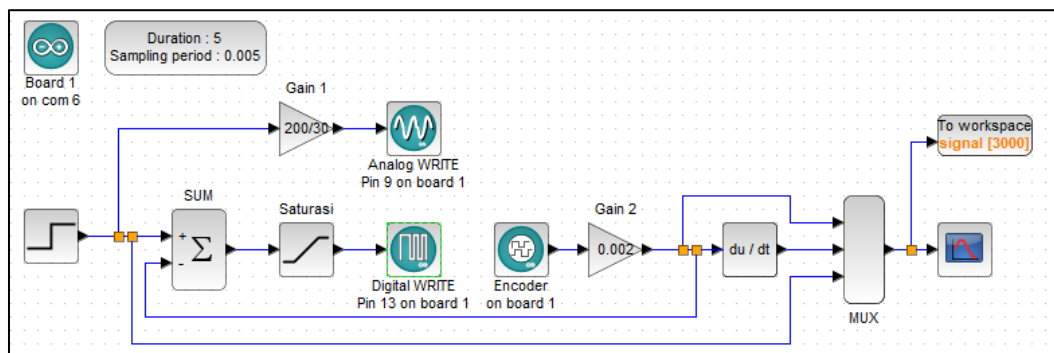
4.2.1 Eksperimen Satu Aksis *Close-loop*

Pemodelan blok diagram *close-loop* untuk gerakan satu aksis pada masing-masing aksis dirancang untuk mengetahui respons objek kendali terhadap *input* variabel posisi. Gambar 4-15 dan Gambar 4-16 berturut-turut menunjukkan rangkaian blok diagram *close-loop* satu aksis sumbu X dan sumbu Y. Pada

dasarnya rangkaian blok diagram untuk sumbu X dan sumbu Y memiliki alur yang sama perbedaan terdapat pada blok diagram *Analog WRITE* dan *Digital WRITE*. Pada sumbu X *Analog WRITE* terhubung dengan pin EN A dan *Digital WRITE* terhubung dengan pin In 2 sedangkan pada sumbu Y pin yang terhubung adalah pin EN B dan pin In 4. Blok *Gain 1* berisi nilai yang mengkonversi *input* posisi menjadi nilai PWM sehingga nilai yang masuk pada blok *Analog WRITE* menjadi nilai PWM sebesar 200. Blok SUM berfungsi untuk mengurangi nilai *input* dengan umpan balik pada enkoder. *Input* akan bernilai nol ketika nilai umpan balik enkoder sama dengan nilai *input* yang diberikan. Pin digital yang terhubung dengan *Digital WRITE* hanya dapat membaca nilai masukan 1 dan 0, sehingga digunakan blok saturasi untuk membatasi nilai maksimal dan minimal yang masuk menuju *Digital WRITE*.

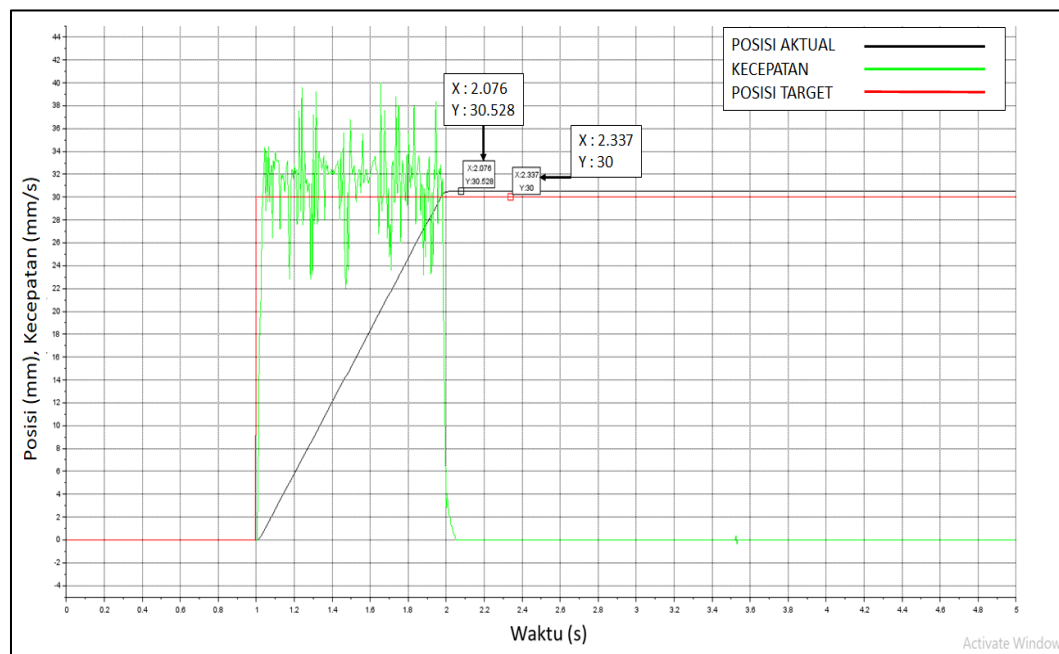


Gambar 4-15 Rangkaian blok diagram satu aksis *close-loop* sumbu X

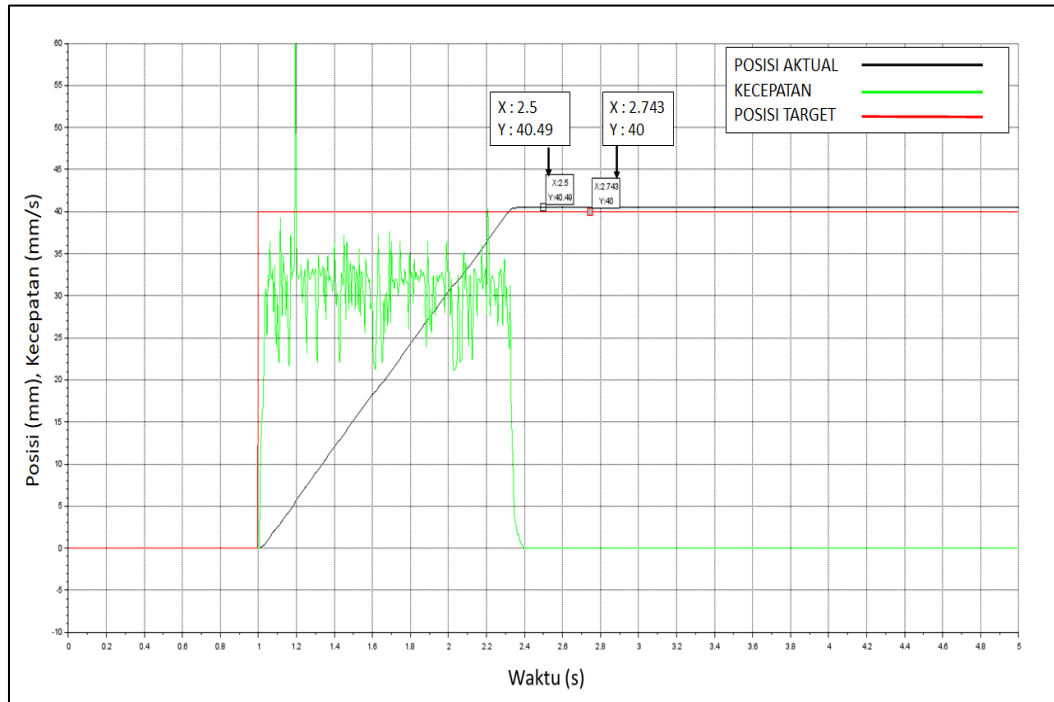


Gambar 4-16 Rangkaian blok diagram satu aksis *close-loop* sumbu Y

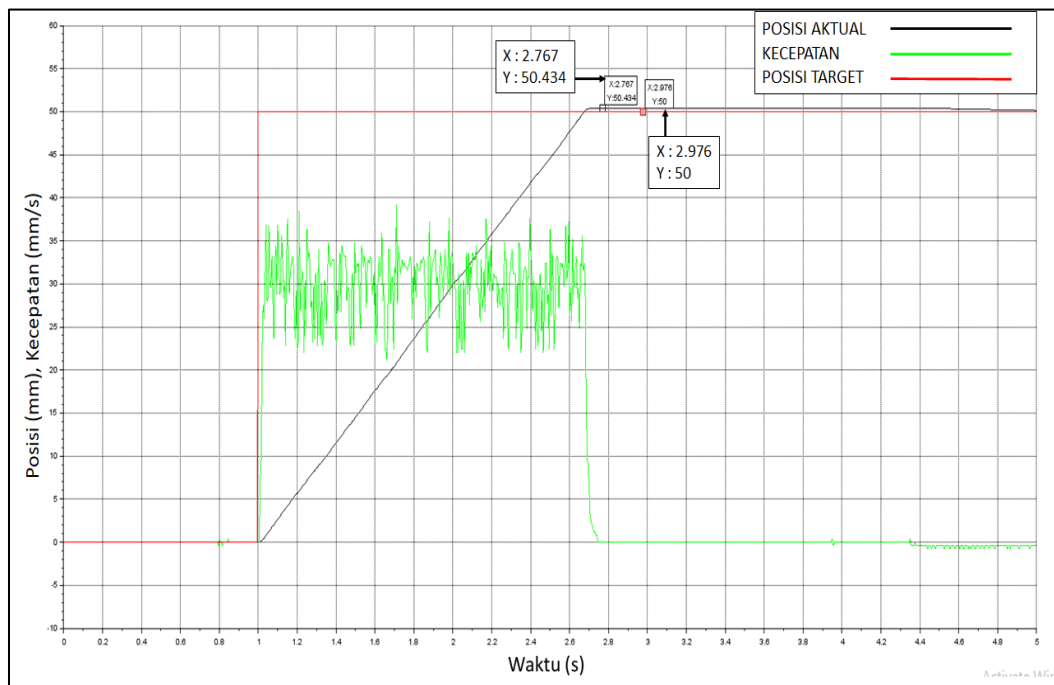
Eksperimen satu aksis *close-loop* dilakukan 3 kali untuk masing-masing sumbu dengan variasi *input* posisi sebesar 30 mm, 40 mm dan 50 mm. Eksperimen dilakukan satu kali untuk setiap variasi *input* posisi. Gambar 4-17, Gambar 4-18 dan Gambar 4-19 berurut menunjukkan respons posisi sumbu X untuk *input* 30 mm, 40 mm dan 50 mm. Pada Gambar 4-20, Gambar 4-21 dan Gambar 4-22 berurut menunjukkan respons posisi sumbu Y untuk *input* 30 mm, 40 mm dan 50 mm. Tabel 4-5 menunjukkan nilai *error* pada sumbu X dan Tabel 4-6 menunjukkan nilai *error* pada sumbu Y.



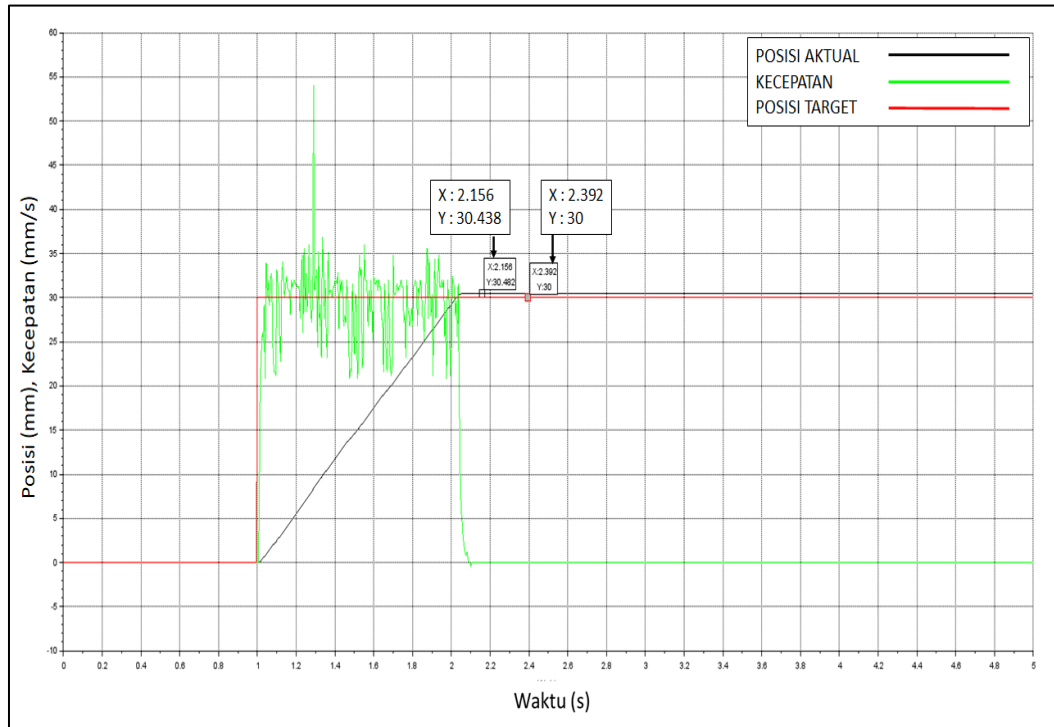
Gambar 4-17 Respons posisi eksperimen *close-loop input* 30 mm sumbu X



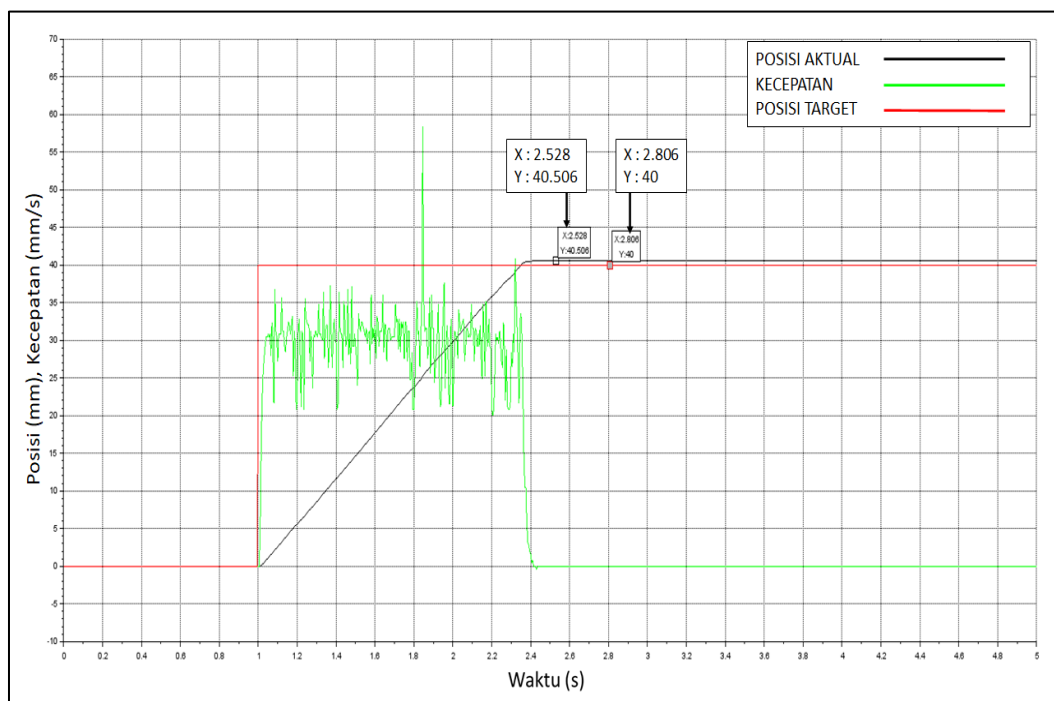
Gambar 4-18 Respons posisi eksperimen *close-loop input* 40 mm sumbu X



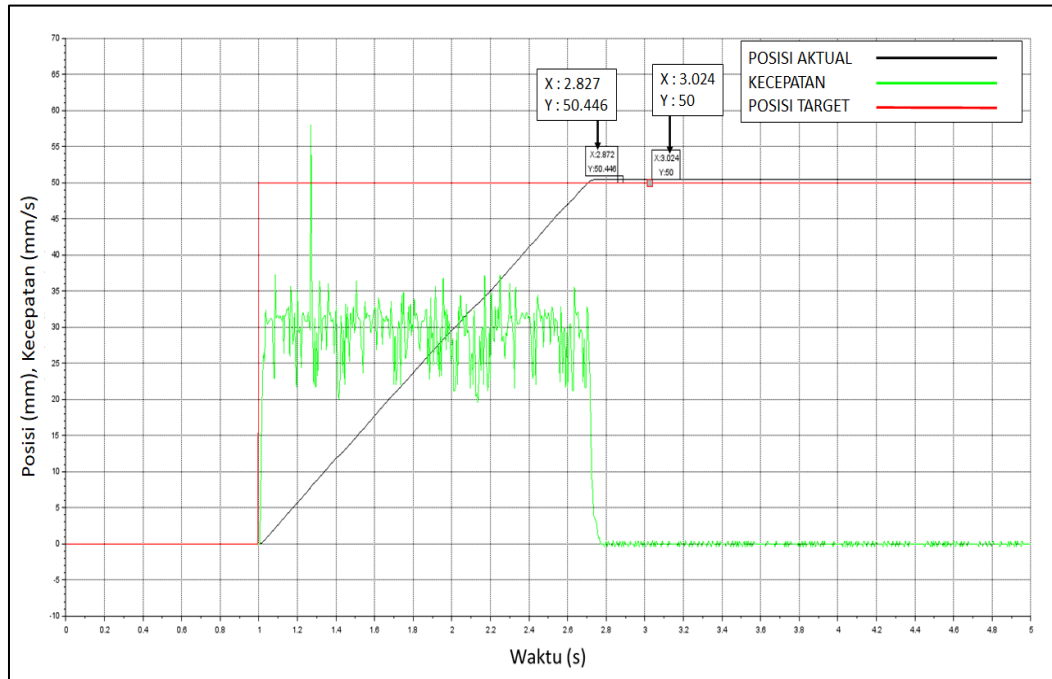
Gambar 4-19 Respons posisi eksperimen *close-loop input* 50 mm sumbu X



Gambar 4-20 Respons posisi eksperimen *close-loop input* 30 mm sumbu Y



Gambar 4-21 Respons posisi eksperimen *close-loop input* 40 mm sumbu Y



Gambar 4-22 Respons posisi eksperimen *close-loop* input 50 mm sumbu Y

Tabel 4-5 Nilai *error* sumbu X

Sumbu X			
Input posisi (mm)	Hasil eksperimen (mm)	Error	
		(mm)	%
30	30.528	0.528	1.76
40	40.490	0.490	1.22
50	50.434	0.434	0.86

Tabel 4-6 Nilai *error* sumbu Y

Sumbu Y			
Input posisi (mm)	Hasil eksperimen (mm)	Error	
		(mm)	%
30	30.482	0.482	1.42
40	40.506	0.506	1.26
50	50.446	0.446	0.89

Berdasarkan data yang tersaji pada Tabel 4-5 nilai *error* sumbu X berada pada 0.434 mm hingga yang terbesar pada 0.528 mm. *Error* terkecil berada pada nilai *input* 50 mm dan terbesar berada pada *input* 30 mm. Presentase *error* terkecil yaitu 0.86 % dan *error* terbesar yaitu 1.76 %. Pada Tabel 4-6 dapat dilihat nilai *error* sumbu Y berada pada 0.446 mm hingga yang terbesar berada pada 0,506 mm. *Error* terkecil berada pada *input* 50 mm dan *error* terbesar berada pada *input* 40 mm berdasarkan nilai *error* dalam satuan mm. Berdasarkan presentase *error* terkecil yaitu 0.89 % pada *input* 50 mm dan terbesar yaitu 1.42 % pada *input* 30 mm.

4.3 Kinerja Sistem Kendali Posisi CM

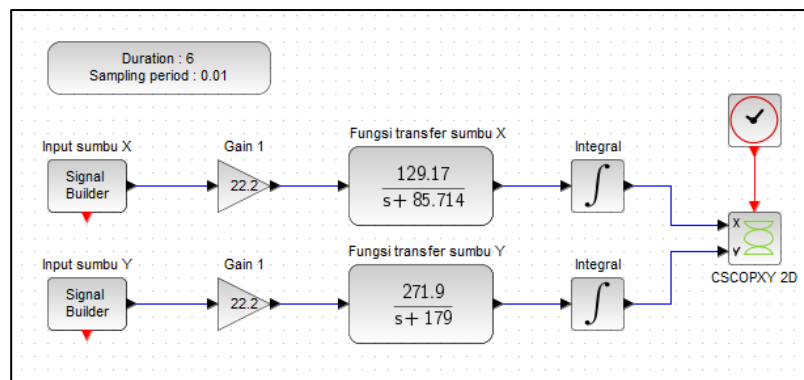
Kinerja sistem kendali posisi CM dievaluasi berdasarkan gerak objek kendali dua aksis yang membentuk bidang datar belah ketupat. Simulasi *open-loop* dilakukan untuk mendapat bentuk belah ketupat, gerak objek kendali simulasi adalah hasil ideal yang ingin dicapai dalam eksperimen *open-loop* dan *close-loop*.

Dalam menganalisis bentuk belah ketupat dilakukan pengukuran pada empat area yaitu area A, B, C dan D seperti yang dapat dilihat pada Gambar 4-26. Area A merupakan titik awal dan titik akhir pergerakan. Gerak objek kendali dimulai dari titik A berurut menuju titik B, C dan D kemudian kembali menuju titik akhir area A. masing-masing titik pada area yang sudah ditentukan akan dibandingkan nilainya antara hasil simulasi dengan eksperimen *open-loop* maupun *close-loop*. Selisih nilai antara simulasi dan eksperimen akan tercatat sebagai nilai *error*. Nilai *error* negatif menunjukkan gerak objek kendali eksperimen belum mencapai titik posisi gerak objek kendali simulasi. Nilai *error* positif menunjukkan gerak objek kendali eksperimen telah melebihi atau melewati titik posisi gerak objek kendali simulasi.

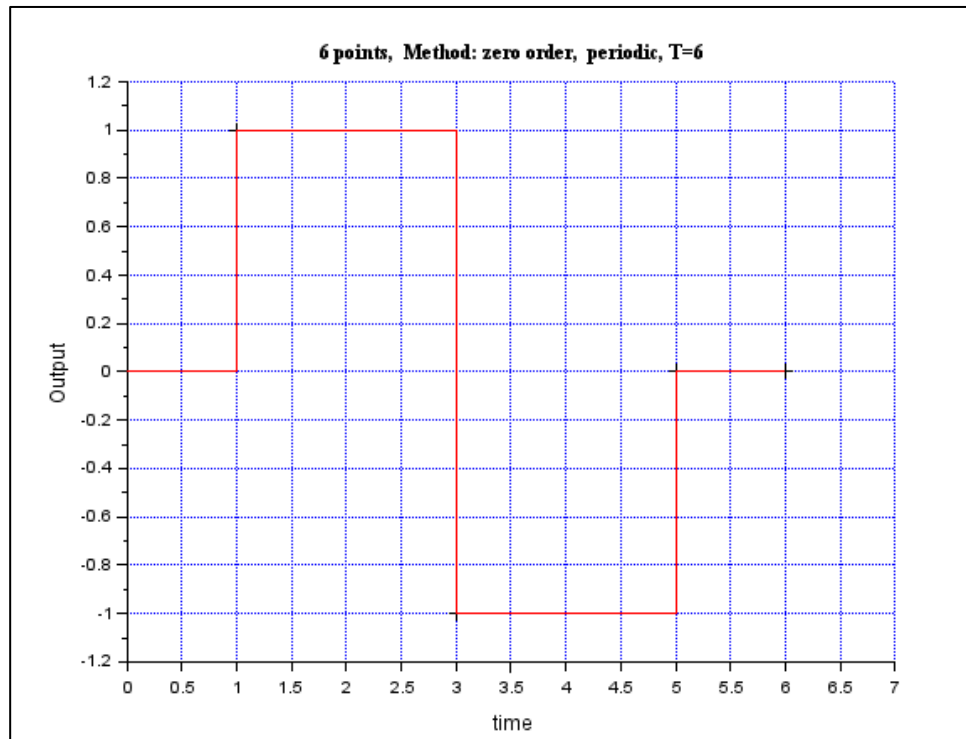
4.3.1 Simulasi Belah Ketupat

Gambar 4-23 menunjukkan rangkaian blok diagram simulasi *open-loop* bidang datar belah ketupat. Simulasi dilakukan dengan menggunakan *input signal builder*, *signal builder* pada sumbu X berbeda dengan *signal builder* pada sumbu Y. Gambar 4-24 menunjukkan *signal builder* pada sumbu X dan Gambar 4-25

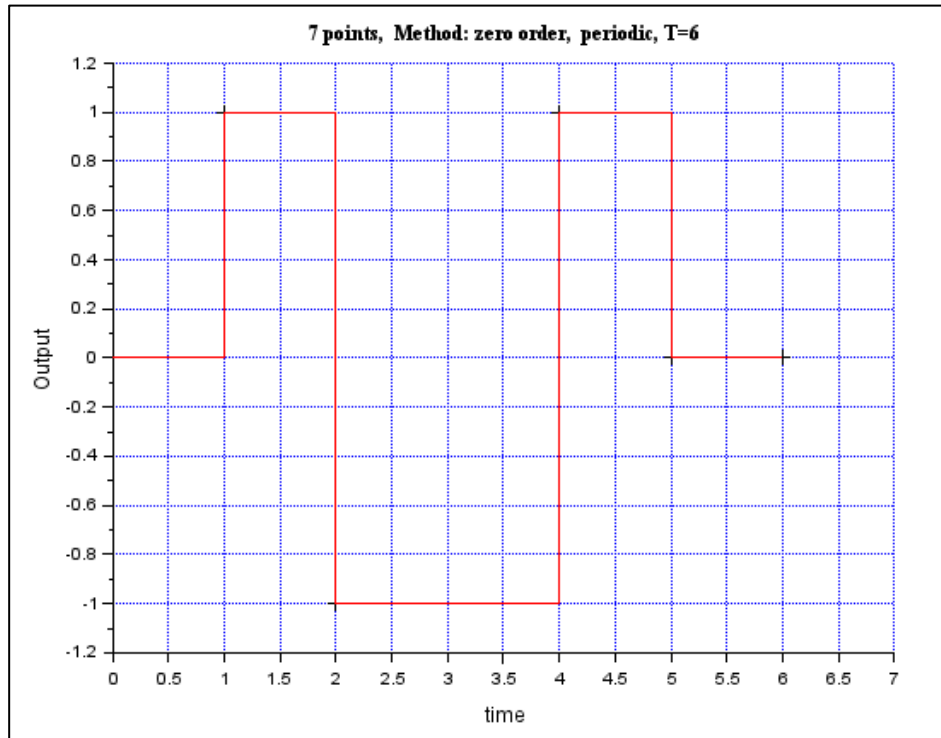
menunjukkan *signal builder* pada sumbu Y. Blok Gain 1 pada Gambar 4-23 merepresentasikan tegangan pada sistem kendali. Sinyal tegangan akan diterima oleh blok fungsi transfer *output* dari blok fungsi transfer berupa kecepatan. Blok integral digunakan untuk mendapatkan *output* posisi. Blok CSCOPXY 2D digunakan untuk memvisualisasikan pergerakan *actuator* pada sumbu X dan sumbu Y.



Gambar 4-23 Blok diagram simulasi belah ketupat

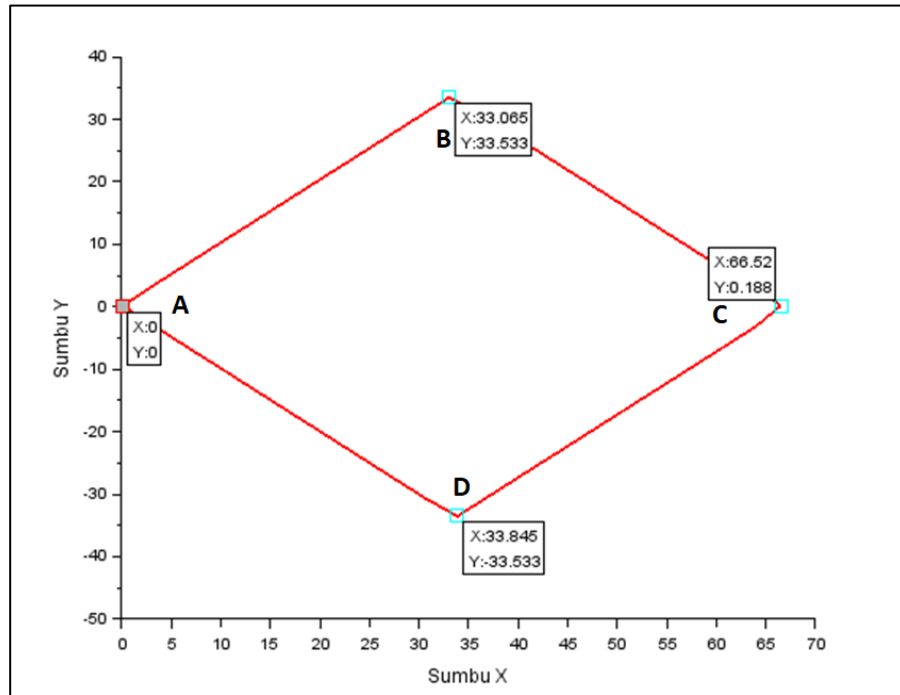


Gambar 4-24 Signal builder sumbu X



Gambar 4-25 Signal builder sumbu Y

Visualisasi gerak objek kendali simulasi *open-loop* belah ketupat dapat dilihat pada gambar 4-26. Pengukuran dilakukan pada area A, B, C dan D pada bidang datar belah ketupat seperti yang ditunjukkan Gambar 4-26. Pada area tersebut diukur gerak objek kendali masing-masing sumbu. Area A merupakan titik awal dan titik akhir pada gerak objek kendali. Pada area A pengukuran dilakukan dua kali ketika berada pada titik awal dan titik akhir, dengan asumsi titik awal memiliki nilai yang selalu sama yaitu 0. Hasil pengukuran dapat dilihat pada tabel 4-7. .



Gambar 4-26 Visualisasi simulasi belah ketupat

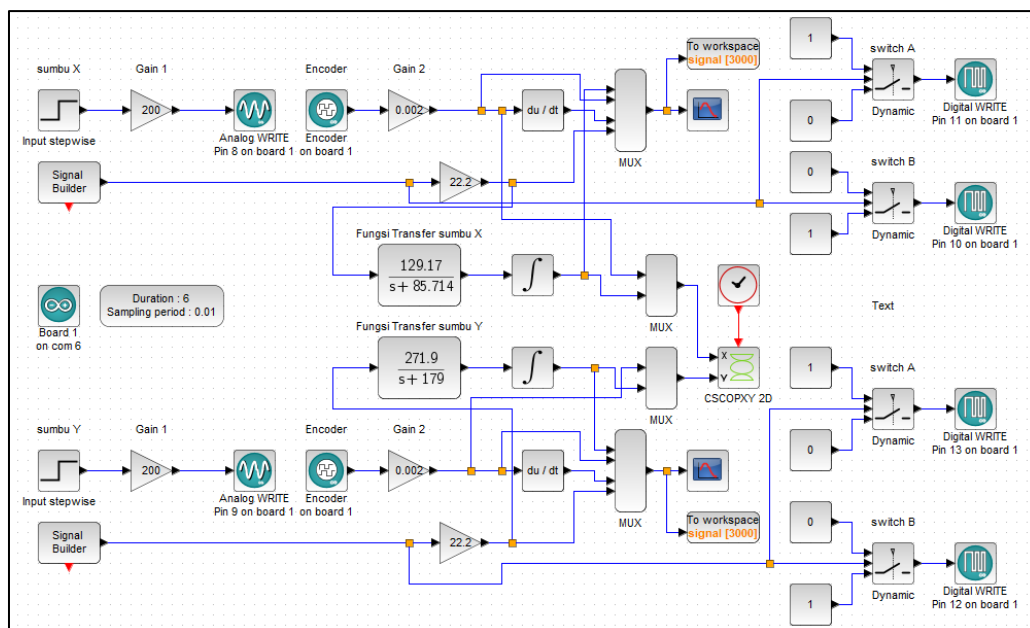
Tabel 4-7 Hasil pengukuran simulasi belah ketupat

Area belah ketupat	Sumbu (mm)	
	X	Y
A (titik awal)	0	0
B	33.065	33.533
C	66.520	0.188
D	33.845	-33.533
A (titik akhir)	0	0

Gerak objek kendali simulasi pada area A titik awal dan area A titik akhir memiliki nilai yang sama yaitu 0. Hal ini menyatakan bahwa gerak objek kendali simulasi dapat kembali ketitik awal secara sempurna. Pada area B sumbu X terukur pada posisi 33.065 mm dan sumbu Y pada posisi 33.533 mm. Area C sumbu X terukur pada posisi 66.520 mm dan sumbu Y pada posisi 0.188 mm. Area D sumbu X terukur pada posisi 33.845 mm dan sumbu Y pada posisi -33.533.

4.3.2 Eksperimen *Open-loop* Belah Ketupat

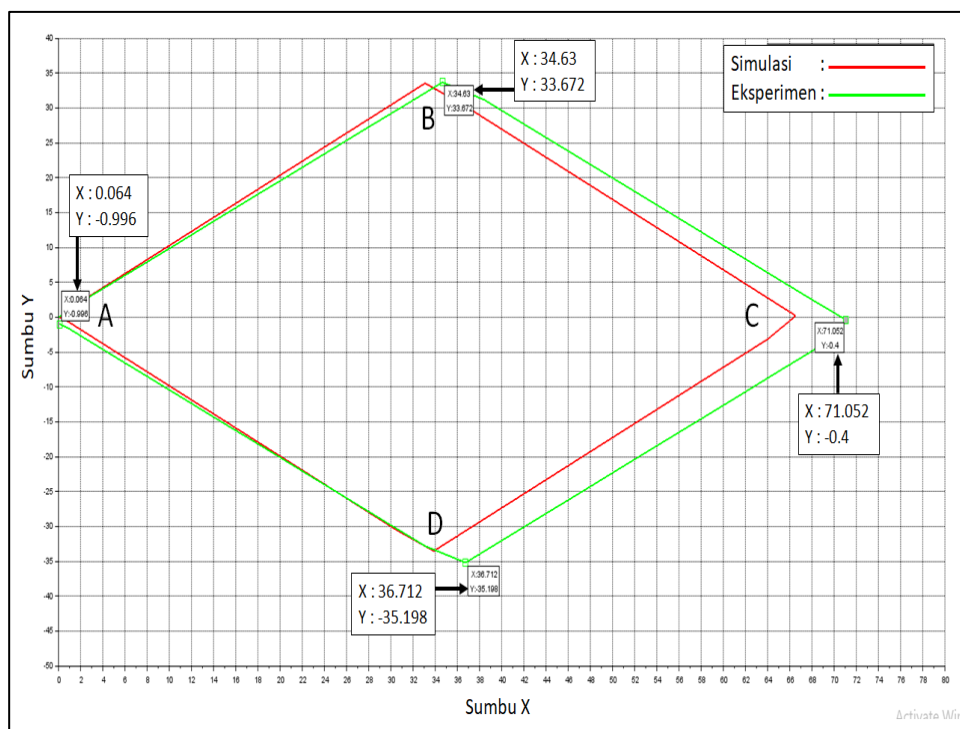
Eksperimen dilakukan dengan memberikan *input stepwise* untuk mengatur kecepatan dan *input signal builder* untuk mengatur arah putaran pada objek kendali. Parameter *signal builder* yang digunakan memiliki nilai yang sama seperti parameter *signal builder* pada simulasi *open-loop*. Aktuator bergerak secara kontinyu dari titik awal menuju titik akhir. Lintasan gerakan aktuator disimpan dan divisualisasikan pada blok diagram CSCOPXY 2D dengan target pergerakan membentuk bidang datar belah ketupat. Hasil dari gerak objek kendali simulasi dibandingkan dengan eksperimen untuk mengetahui nilai *error*. Gambar 4-27 menunjukkan rangkaian blok diagram gerak objek kendali simulasi dan eksperimen.



Gambar 4-27 Rangkaian blok diagram simulasi dan eksperimen *open-loop* belah ketupat

Visualisasi gerak objek kendali *open-loop* simulasi dengan gerak objek kendali eksperimen dapat dilihat pada Gambar 4-28. Gerak objek kendali eksperimen panjang setiap sisi belah ketupat dipengaruhi oleh sinyal yang dihasilkan oleh blok diagram *signal builder*. Sehingga penulis hanya dapat mengatur waktu pergerakan aktuator untuk membentuk belah ketupat tanpa memperhitungkan panjang sisi yang diinginkan.

Gerak objek kendali simulasi merupakan gerakan ideal yang ingin dicapai. Perbedaan ukuran dianalisis pada area A, B, C dan D seperti yang dapat dilihat pada Gambar 4-28. Posisi area A titik awal pada gerak objek kendali simulasi dan eksperimen diasumsikan memiliki nilai yang sama yaitu 0. Gerak objek kendali eksperimen *open-loop* area A titik akhir pada sumbu X terukur 0.064 mm dan sumbu Y 0.996 mm. Area B untuk sumbu X terukur pada posisi 34.63 mm dan sumbu Y 33.672 mm. Area C untuk sumbu X terukur pada posisi 71.052 mm dan sumbu Y -0.4 mm. Area D untuk sumbu X terukur pada posisi 36.712 mm dan sumbu Y -35.198 mm. Tabel 4-8 dan Tabel 4-9 menunjukkan nilai *error* pada setiap area belah ketupat untuk sumbu X dan sumbu Y.



Gambar 4-28 Visualisasi simulasi dan eksperimen *open-loop* belah ketupat

Tabel 4-8 Perbandingan simulasi dan eksperimen *open-loop* sumbu X

Area Belah Ketupat	Sumbu X (mm)		<i>Error</i>
	Simulasi	Eksperimen	
A (Titik awal)	0	0	0
B	33.065	34.630	1.565
C	66.520	71.052	4.532
D	33.845	36.712	-2.867
A (Titik akhir)	0	0.064	-0.064

Tabel 4-9 Perbandingan simulasi dan eksperimen *open-loop* sumbu Y

Area Belah Ketupat	Sumbu Y (mm)		<i>Error</i> (mm)
	Simulasi	Eksperimen	
A (Titik awal)	0	0	0
B	33.533	33.672	0.139
C	0.188	-0.4	0.588
D	-33.533	-35.198	0.43
A (Titik akhir)	0	-0.996	-0.996

Nilai *error* sumbu X terbesar berada pada area C yaitu sebesar 4.532 mm dan sumbu Y berada pada titik akhir area A yaitu sebesar -0.996 mm. Nilai *error* terkecil sumbu X berada pada titik akhir area A yaitu sebesar -0.064 mm dan pada sumbu Y berada pada area B yaitu sebesar 0.139 mm. Secara visual lintasan pada area B menuju area C dan lintasan area C menuju area D terdapat selisih yang cukup besar.

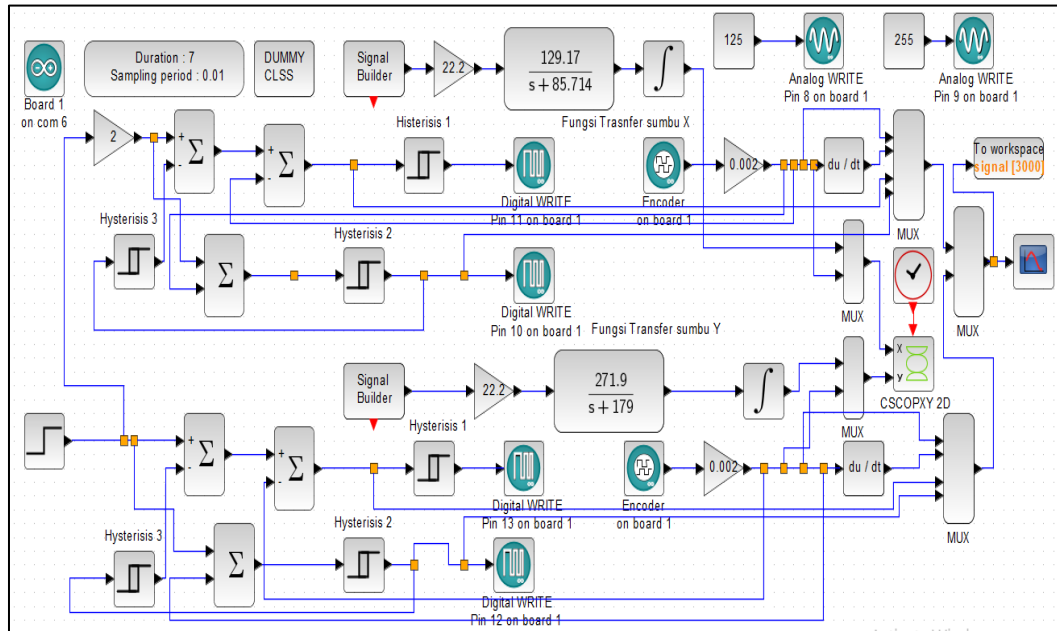
4.3.3 Eksperimen *Close-Loop* Belah Ketupat

Gambar 4-29 menunjukkan rangkaian blok diagram *close-loop* bidang datar belah ketupat melalui simulasi dan secara eksperimen. Aktuator sumbu X dan sumbu Y digerakan secara bersamaan agar dapat membentuk bidang datar belah ketupat. Pada sumbu X pergerakan aktuator terbagi menjadi dua fase, pertama

aktuator bergerak menuju posisi yang ditentukan kemudian fase kedua aktuator kembali ke posisi awal. Aktuator pada sumbu X bergerak dari posisi 0 mm menuju posisi 66 mm dengan putaran CW, kemudian bergerak kembali ke posisi 0 dengan putaran CCW. Pergerakan pada aktuator sumbu Y terbagi menjadi tiga fase. Pertama aktuator bergerak dari posisi 0 ke posisi 33 mm dengan putaran CW. Kedua putaran aktuator dibalik menjadi CCW dan bergerak sejauh 66 mm atau bergerak sejauh -33 mm dari posisi awal. Fase terakhir aktuator bergerak kembali menuju posisi awal atau 0 dengan putaran CW. Pada blok *stepwise* berisi nilai posisi sebesar 33 mm. Blok *Hysteresis* akan mengirim sinyal 1 atau 0 menuju blok *Digital WRITE* untuk mengatur arah putaran motor DC. Sinyal yang keluar dari blok *Hysteresis* dipengaruhi oleh sinyal *output* dari blok *SUM*. Blok *SUM* berfungsi untuk menjumlahkan atau mengurangi 2 *input* yang masuk dengan mengeluarkan satu *output* hasil perhitungan 2 *input* tersebut.

Ketika linier enkoder berada pada posisi 0 mm blok *Hysteresis* 1 pada sumbu X mengirimkan sinyal 1 ke blok *Digital WRITE* 11 yang akan menggerakkan aktuator sumbu X untuk fase pertama. Selanjutnya, ketika linier encoder berada pada posisi 66 mm fase kedua dijalankan. Blok *Hysteresis* 2 mengirimkan sinyal 1 ke blok *Digital WRITE* 10. Blok *Hysteresis* 3 berfungsi untuk memutus sinyal *input* sehingga tidak terjadi perulangan pada pergerakan.

Pada sumbu Y ketika linier enkoder berada pada posisi 0 mm blok *Hysteresis* 1 akan mengirim sinyal ke blok *Digital WRITE* 13 untuk gerakan aktuator fase pertama. Blok *Hysteresis* 2 akan menggerakkan aktuator pada fase kedua dengan mengirim sinyal 1 ke blok *Digital WRITE* 12 ketika enkoder berada pada posisi 33 mm. Pada fase ketiga blok *Hysteresis* 3 akan aktif yang akan mempengaruhi *input stepwise* sehingga blok *Hysteresis* 2 akan kembali mengirim sinyal 1 pada *Digital WRITE* 13 yang akan menggerakkan aktuator menuju posisi awal. Perbedaan nilai *input* ketika *Hysteresis* 3 aktif akan memutus sinyal *input* sehingga tidak terjadi perulangan.

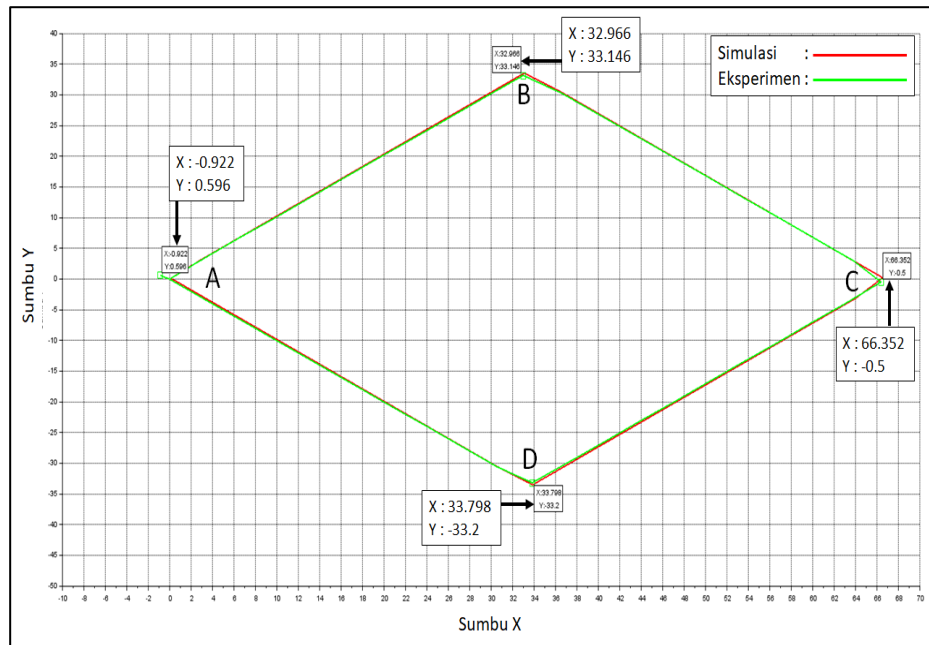


Gambar 4-29 Rangkaian blok diagram simulasi dan eksperimen *close-loop* belah ketupat

Berbeda dengan gerak objek kendali *open-loop* gerak objek kendali *close-loop* memiliki umpan balik sehingga antara *input* dan *output* memiliki variabel yang sama. *Input* yang diberikan adalah perubahan posisi dari aktuator. *Input* yang diberikan sesuai atau mendekati nilai yang didapat dari gerak objek kendali simulasi. Visualisasi pergerakan bentuk belah ketupat hasil simulasi dan eksperimen *close-loop* dapat dilihat pada Gambar 4-30. Garis merah menunjukkan visualisasi dari simulasi dan garis hijau menunjukkan visualisasi dari eksperimen *close-loop*.

Nilai *error* pada setiap area dapat dilihat pada Tabel 4-10 untuk sumbu X dan 4-11 untuk sumbu Y. Pergerakan eksperimen *close-loop* area A titik akhir pada sumbu X terukur -0.922 mm dan sumbu Y 0.596 mm. Area B untuk sumbu X terukur pada posisi 32.966 mm dan sumbu Y 33.146 mm. Area C untuk sumbu X terukur pada posisi 66.352 mm dan sumbu Y -0.5 mm. Area D untuk sumbu X terukur pada posisi 33.798 mm dan sumbu Y -33.2 mm. Nilai *error* sumbu X terbesar berada pada titik akhir area A yaitu sebesar 0.92 mm dan sumbu Y berada pada area C yaitu sebesar 0.688 mm. Nilai *error* terkecil sumbu X berada pada area

D yaitu sebesar 0.047 mm dan pada sumbu Y berada pada area D yaitu sebesar -0.333 mm.



Gambar 4-30 Visualisasi simulasi dan eksperimen *close-loop* belah ketupat

Tabel 4-10 Perbandingan simulasi dan eksperimen *close-loop* sumbu X

Area Belah Ketupat	Sumbu X (mm)		Error
	Simulasi	Eksperimen	
A (Titik awal)	0	0	0
B	33.065	32.966	-0.099
C	66.520	66.352	-0.168
D	33.845	33.798	0.047
A (Titik akhir)	0	-0.922	0.922

Tabel 4-11 Perbandingan simulasi dan eksperimen *close-loop* sumbu Y

Area Belah Ketupat	Sumbu Y (mm)		<i>Error</i> (mm)
	Simulasi	Eksperimen	
A (Titik awal)	0	0	0
B	33.533	33.146	-0.387
C	0.188	-0.5	0.688
D	-33.533	-33.2	-0.333
A (Titik akhir)	0	0.596	0.596

Berdasarkan data *error* dan visual eksperimen *close-loop* lebih mendekati hasil dari simulasi dibandingkan dengan eksperimen *open-loop*. Hal ini karena dalam konfigurasi *close-loop* terdapat umpan balik yang dapat mengetahui perubahan posisi dan pergerakan didasarkan pada umpan balik tersebut. Pada penelitian ini umpan balik hanya menentukan posisi yang ingin dicapai tanpa menambahkan koreksi *error* yang terjadi. Penambahan *controller* untuk mengoreksi *error* yang terjadi dapat dilakukan pada penelitian yang selanjutnya.

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan didapatkan kesimpulan sebagai berikut:

1. Model matematika berupa fungsi transfer untuk sistem kendali linier berhasil didapatkan dengan fungsi transfer $\frac{\theta(s)}{U(s)} = \frac{129.17}{s(s+85.714)}$ untuk sumbu X dan $\frac{\theta(s)}{U(s)} = \frac{271.9}{s(s+179)}$ untuk sumbu Y.
2. Evaluasi kinerja sistem kendali posisi telah dilakukan melalui simulasi dan secara eksperimen. Berdasarkan data *error* dan secara visual gerak objek kendali *close-loop* lebih baik daripada *open-loop*. Gerak objek kendali eksperimen *close-loop* sudah mendekati gerak objek kendali simulasi namun masih terdapat *error* yang cukup besar. Hal ini terjadi karena penelitian ini belum mengimplementasikan *controller* untuk menghilangkan atau meminimalisir *error*.

5.2 Saran atau Penelitian selanjutnya

Pada proses eksekusi eksperimen objek kendali menggunakan *software* Scilab, kerap kali *software* Scilab tidak merespon. Masalah ini kemungkinan terjadi karena *software* Scilab terlalu berat untuk dijalankan oleh perangkat komputer. Oleh karena itu dibutuhkan perangkat komputer yang *compatible* agar *software* Scilab dapat berjalan dengan baik.

Penelitian yang dilakukan penulis menghasilkan respon posisi pada gerak objek kendali yang membentuk bidang datar belah ketupat. Gerak objek kendali dilakukan dengan konfigurasi *open-loop* dan *close-loop*. Hasil gerak objek kendali bidang datar belah ketupat masih memiliki perbedaan (*error*) antara gerak objek kendali simulasi dan eksperimen. *Error* yang terjadi dapat dihilangkan atau diperkecil menggunakan *controller*. Beberapa *controller* dapat diaplikasikan untuk

penelitian lebih lanjut seperti PID (*proportional integral derivative*), *fuzzy*, *state space*, dan sebagainya.

DAFTAR PUSTAKA

- Akmeliawati, R., & Wijaya, A. A. (2009). *NCTF Control Method of Two Mass System for PTP Positioning*.
- Chong, S. H., & Sato, K. (2010). *Practical and robust control for precision motion: AR-CM NCTF control of a linear motion mechanism with friction characteristics*. 34, 286–300. <https://doi.org/10.1049/iet-cta.2014.0544>
- Hasim, N., Chong, S. H., Hee, W. K., & Ibrahim, Z. (2017). Contouring control of ball screw mechanism using a practical control method. *IOP Conference Series: Materials Science and Engineering*, 210, 012053. <https://doi.org/10.1088/1757-899X/210/1/012053>
- Hudati, I., Aji, A. P., & Nurrahma, S. (2021). Kendali Posisi Motor DC dengan Menggunakan Kendali PID. *Jurnal Listrik, Instrumentasi dan Elektronika Terapan (JuLIET)*, 2(2). <https://doi.org/10.22146/juliet.v2i2.71148>
- Jalani, N. H. (2011). *Design and Implementation of Nominal Characteristic Trajectory Following (NCTF) Controller for Linear Motion Servo System*. <https://doi.org/10.13140/2.1.3455.2323>
- Nise, N. S. (2015). *Control systems engineering* (Seventh edition). Wiley.
- Purtojo, P. (2017). *Sistem Kendali NCTF untuk Operasi Kendali Posisi Titik-ke-Titik*. Universitas Islam Indonesia.
- Sato, K., & Maeda, G. J. (2009). A practical control method for precision motion—Improvement of NCTF control method for continuous motion control. *Precision Engineering*, 33(2), 175–186. <https://doi.org/10.1016/j.precisioneng.2008.05.006>

Wahyudi, wahyudi, Sato, K., & Akira, S. (2003). *Characteristics of practical control for point-to-point (PTP) positioning systems: Effect of design parameters and actuator saturation on positioning performance*. 27, 157–169.