



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TM 184835**

# **PEMODELAN DAN ANALISIS DINAMIKA LENGAN ROBOT 3 DOF MENGGUNAKAN KONTROL PID**

**MIFTAHUL FARID MARDIANSAH**  
0211154000095

Dosen Pembimbing  
Dr.Eng. **UNGGUL WASIWITONO, S.T., M.Eng.Sc**

**DEPARTEMEN TEKNIK MESIN**  
Fakultas Teknologi Industri dan Rekayasa Sistem  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020





**TUGAS AKHIR - TM184835**

# **PEMODELAN DAN ANALISIS DINAMIKA LENGAN ROBOT 3 DOF MENGGUNAKAN KONTROL PID**

**MIFTAHUL FARID MARDIANSAH  
0211154000095**

**Dosen Pembimbing  
Dr.Eng. Unggul Wasiwitono, S.T., M.Eng.Sc**

**DEPARTEMEN TEKNIK MESIN  
FAKULTAS TEKNOLOGI INDUSTRI DAN REKAYASA SISTEM  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA - 2020**





**FINAL PROJECT - TM 184835**

**MODELLING AND DYNAMICS ANALYSIS 3 DOF ROBOTIC ARM  
MANIPULATOR USING PID CONTROL**

**MIFTAHUL FARID MARDIANSAH  
0211154000095**

**Academic Advisor  
Dr.Eng. Unggul Wasiwitono, S.T., M.Eng.Sc**

**MECHANICAL ENGINEERING DEPARTMENT  
Faculty of Industrial Technology and System Engineering  
Sepuluh Nopember of Intitute Technology  
Surabaya 2020**



**PEMODELAN DAN ANALISIS DINAMIKA LENGAN ROBOT 3 DOF  
MENGUNAKAN KONTROL PID**

**TUGAS AKHIR**

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Teknik  
pada  
Program Studi S-1 Departemen Teknik Mesin  
Fakultas Teknologi Industri dan Rekayasa Sistem  
Institut Teknologi Sepuluh Nopember

Oleh:

**Miftahul Farid Mardiansah**  
NRP. 0211154000095

Disetujui oleh Tim Penguji Tugas Akhir:

1. Dr.Eng. Unggul Wasiwitonon, S.T., M.Eng.Sc ..... (Pembimbing)  
NIP. 197805102001121001
2. Prof. Ir. I Nyoman Sutantra, M.Sc., Ph.D. .... (Penguji I)  
NIP. 195106051978031002
3. Dr.Eng. Yohanes, S.T., M.Sc ..... (Penguji II)  
NIP. 198006272012121003
4. Dr. Latifah Nurahmi, S.T., M.Sc ..... (Penguji III)  
NIP. 1986201712037



**SURABAYA**  
**AGUSTUS, 2020**





# PEMODELAN DAN ANALISIS DINAMIKA LENGAN ROBOT 3 DOF MENGGUNAKAN KONTROL PID

**Nama Mahasiswa** : Miftahul Farid Mardiansah  
**NRP** : 0211154000095  
**Department** : Teknik Mesin  
**Pembimbing** : Dr. Eng Unggul Wasiwitono, ST.,  
M.Eng.Sc

## ABSTRAK

Penelitian tentang lengan robot terus dilakukan seiring dengan perkembangan robot di dunia industri. Pada lengan robot 3 sumbu yang memiliki 3 sumbu rotari membuat lengan robot sangat fleksibel dalam bergerak dengan ruang kerja yang luas namun menempati ruang yang lebih sempit. Untuk mengatur kecepatan yang tinggi dan akurasi yang baik secara bersamaan diperlukan analisis dinamika yang tepat. Dinamika umum robot tergantung pada dua faktor yaitu sistem mekanik dan sistem kontrol robot.

Pada penelitian ini dianalisa dinamika lengan robot yang memiliki 3 derajat kebebasan. Simulasi dinamis menggunakan Simscape Multibody. Sistem kendali yang digunakan pada lengan robot adalah kontrol PID. *Trajectory* berupa setengah lingkaran, lingkaran, dan bebas digunakan untuk membandingkan dinamika lengan robot dengan sistem kendali yang berbeda.

Hasil pergerakan lengan robot menggunakan *automated tuning* PID lebih presisi daripada metode ziegler-nichols. Pergerakan lengan robot menggunakan metode ziegler-nichols mengalami osilasi di awal pergerakan, sedangkan *automated tuning* PID bergerak sesuai dengan referensi *trajectory* yang diberikan. Parameter  $K_p$ ,  $K_i$ , dan  $K_d$  dengan respon paling bagus ditunjukkan oleh sistem kendali PID menggunakan *automated tuning* pada *trajectory* bebas sendi ke-2, dengan nilai  $K_p = 1227.9766$ ,  $K_i = 7283.8264$ , dan  $K_d = 5.2512$  didapatkan besar

*settling time* = 0.0484 detik, *rise time* = 0.402 detik, dan persen *overshoot* = 10%.

**Kata Kunci:** *Automated Tuning, Kontrol PID, Lengan Robot, Trajectory, Ziegler-Nichols.*

# **MODELLING AND DYNAMIC ANALYSIS 3 DOF ROBOTIC ARM MANIPULATOR USING PID CONTROLLER**

**Student Name** : Miftahul Farid Mardiansah  
**Student ID Number** : 0211154000095  
**Department** : Mechanical Engineering  
**Advisor Lecturer** : Dr. Eng Unggul Wasiwitono,  
ST., M.Eng.Sc

## **ABSTRACT**

*Research on robotic arms continues in line with the development of robots in the industrial world. The 3-axis robot arm which has 3 rotary axes makes the robot arm very flexible in moving with a large workspace but occupies a narrower space. To regulate high speed and good accuracy at the same time a precise dynamic analysis is required. The general dynamics of a robot depends on two factors, namely the mechanical system and the robot control system.*

*In this study, the dynamics of a robot arm which has 3 degrees of freedom were analyzed. Dynamic simulation using Simscape Multibody. The control system used in the robot arm is PID control. Trajectory in the form of a semicircle, circle, and free is used to compare the dynamics of the robot arm with different control systems.*

*The result of the movement of the robot arm using Automated tuning PID is more precise than the ziegler-nichols method. The movement of the robot arm using the ziegler-nichols method oscillates at the beginning of the movement, while the automated tuning PID moves according to the given trajectory reference. The  $K_p$ ,  $K_i$ , and  $K_d$  parameters with the best response are shown by the PID control system using automated tuning on the 2nd joint-free trajectory, with a value of  $K_p = 1227.9766$ ,  $K_i =$*

7283.8264, and  $K_d = 5.2512$ , the settling time is obtained = 0.0484 seconds, rise time = 0.402 seconds, and percent overshoot = 10%.

***Key words: Automated Tuning, PID Controller, Robotic Arm, Trajectory, Ziegler-Nichols.***

## KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Allah SWT, karena berkat limpahan rahmat dan hidayah-Nya penulis dapat menyelesaikan tugas akhir dengan judul “PEMODELAN DAN ANALISIS DINAMIKA LENGAN ROBOT 3 DOF MENGGUNAKAN KONTROL PID” di tengah pandemi COVID-19 dalam keadaan sehat. Tujuan dari penulisan Tugas Akhir ini adalah untuk memenuhi persyaratan kelulusan program Studi Strata I Departemen Teknik Mesin, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Dalam proses penyusunan Tugas Akhir ini banyak pihak yang telah memberikan dukungan dan bantuan kepada penulis, hingga akhirnya Tugas Akhir ini dapat terselesaikan dengan baik. Oleh karena itu, penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam penyusunan Tugas Akhir ini, di antaranya:

1. Keluarga tersayang, Bapak Sirjo, Ibu Sumina, Ayah Suratno, Ibu Indah Rahayu, Ririn, Suhirman, Khusnul Khotimah, Ahmad Teddy Firmansyah, dan Wafiq Annadlho yang telah memberikan doa dan dukungan tiada henti hingga dapat menyelesaikan Tugas Akhir ini.
2. Bapak Dr. Eng. Unggul Wasiwitono, S.T., M.Eng.Sc, selaku dosen pembimbing yang telah memberikan bimbingan dan arahan selama penyusunan Tugas Akhir ini.
3. Prof. Ir. I Nyoman Sutantra, M.Sc., Ph.D., Dr.Eng. Yohanes, S.T., M.Sc, dan Dr. Latifah Nurahmi, S.T., M.Sc selaku dosen penguji yang telah memberikan kritik dan saran untuk Tugas Akhir ini.
4. Sahabat sekaligus keluarga kontrakan “Star Family” yaitu Prasetya Faiz Pambudi, Aldhien Yogha Pratama, Marga Firnanda, Hafidh Naufaldi Satriawan, Andreas Tommy Irawan, Agung Budi Darmawan, Sakti Priyanto, Fauzan

Alfindi, Rezky Dwi Anugerah, Adi Akbar Rafsanjani, Reno Hari Pratikto, Roni Setiawan, Aulia Bukhaira, Teduh Adhi Prakoso, Aufar Rafdi dan Bramantya Adi Nugraha yang telah menjadi keluarga dalam menghadapi dunia perkuliahan di Teknik Mesin ITS.

5. Teman-teman angkatan M58 pada khususnya dan teman-teman teknik mesin pada umumnya.

Dengan segala keterbatasan kemampuan dan pengetahuan penulis, tidak menutup kemungkinan Tugas Akhir ini jauh dari sempurna. Oleh karena itu, penulis bersedia menerima kritik dan saran dari berbagai pihak untuk penyempurnaan lebih lanjut. Semoga hasil penulisan Tugas Akhir ini dapat bermanfaat bagi semua pihak.

Surabaya, 13 Agustus 2020

Penulis

# DAFTAR ISI

<b>ABSTRAK .....</b>	<b>i</b>
<b>KATA PENGANTAR .....</b>	<b>ii</b>
<b>DAFTAR ISI.....</b>	<b>iii</b>
<b>DAFTAR GAMBAR.....</b>	<b>v</b>
<b>DAFTAR TABEL.....</b>	<b>viii</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah .....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penulisan .....	4
<b>BAB II TINJAUAN PUSAKA.....</b>	<b>5</b>
2.1 Hasil Penelitian Terdahulu .....	5
2.2 Teori Dasar.....	8
2.2.1 Sistem Lengan Robot .....	9
2.2.1.1 DH Parameter .....	10
2.2.1.2 Inverse Kinematics .....	10
2.2.2 Persamaan Dinamis Lengan Robot.....	11
2.2.3 Aktuator pada Sendi .....	13
2.2.4 Sistem Kendali PID .....	16
2.2.4.1 Kontrol Proporsional .....	17
2.2.4.2 Kontrol Integratif.....	18
2.2.4.3 Kontrol Derivatif .....	18
2.2.4.4 Metode Ziegler-Nichols.....	18
2.2.4.5 Automated Tuning PID .....	21
2.2.5 Motor Arus Searah .....	22
2.2.6 Trajectory Planning .....	23
2.2.6.1 Joint Space Trajectory Planning .....	23
2.2.6.2 Cartesian Space Trajectory Planning .....	23

<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>25</b>
3.1 Metode yang Digunakan.....	25
3.2 Diagram Alir Penelitian.....	26
3.3 Spesifikasi Model Lengan Robot.....	27
3.4 Parameter Motor DC .....	28
3.5 Desain Kendali PID.....	29
<b>BAB IV PEMBAHASAN.....</b>	<b>31</b>
4.1 Deskripsi Umum .....	31
4.2 Analisa Trajectory Setengah Lingkaran.....	32
4.2.1 Kontrol PID menggunakan Advance Tuning PID .....	33
4.2.1.1 Kecepatan Sudut pada tiap Sendi.....	38
4.2.1.2 Percepatan Sudut pada tiap Sendi .....	41
4.2.1.3 Torsi pada tiap Sendi .....	44
4.2.2 Kontrol PID menggunakan Metode Ziegler-Nichols .....	47
4.2.2.1 Kecepatan Sudut pada tiap Sendi.....	52
4.2.2.2 Percepatan Sudut pada tiap Sendi .....	55
4.2.2.3 Torsi pada tiap Sendi .....	58
4.3 Analisa Trajectory Lingkaran.....	61
4.3.1 Kontrol PID menggunakan Advance Tuning PID .....	61
4.3.1.1 Kecepatan Sudut pada tiap Sendi.....	66
4.3.1.2 Percepatan Sudut pada tiap Sendi .....	69
4.3.1.3 Torsi pada tiap Sendi .....	72
4.3.2 Kontrol PID menggunakan Metode Ziegler-Nichols .....	75
4.3.2.1 Kecepatan Sudut pada tiap Sendi.....	80
4.3.2.2 Percepatan Sudut pada tiap Sendi .....	83
4.3.2.3 Torsi pada tiap Sendi .....	86
4.4 Analisa Trajectory Bebas .....	89
4.4.1 Kontrol PID menggunakan Advance Tuning PID .....	89
4.4.1.1 Kecepatan Sudut pada tiap Sendi.....	93
4.4.1.2 Percepatan Sudut pada tiap Sendi .....	96
4.4.1.3 Torsi pada tiap Sendi .....	99



4.4.2 Kontrol PID menggunakan Metode Ziegler-Nichols .....	102
<b>BAB V KESIMPULAN DAN SARAN.....</b>	<b>103</b>
<b>DAFTAR PUSTAKA</b>	

*(Halaman ini sengaja dikosongkan)*

## DAFTAR GAMBAR

Gambar 1.1 Lengan robot untuk manufaktur .....	1
Gambar 2.1 Pemodelan lengan robot.....	5
Gambar 2.2 Model Fisik Penelitian Muhammad Bilal dkk .....	6
Gambar 2.3 Step Respon pada Tiap Motor.....	8
Gambar 2.4 Sistem lengan robot sederhana.....	9
Gambar 2.5. Kontrol PID pada sistem.....	17
Gambar 2.6 Kurva Ziegler-Nichols Tipe 1 .....	19
Gambar 2.7 Kurva Ziegler-Nichols Tipe 2 .....	20
Gambar 2.8 PID <i>Tuner</i> pada Matlab-Simulink .....	21
Gambar 2.9 Pemodelan motor DC .....	22
Gambar 3.1: Diagram Alir Penelitian .....	26
Gambar 3.2 Gambar 3 Dimensi Lengan Robot 3 Derajat Kebebasan .....	27
Gambar 4.1: Pemodelan lengan robot menggunakan simulink...	31
Gambar 4.2. Grafik <i>Trajectory</i> Setengah Lingkaran menggunakan <i>Automated Tuning</i> PID.....	34
Gambar 4.3 Grafik Sendi 1 Menggunakan <i>Automated Tuning</i> PID pada Setengah Lingkaran .....	35
Gambar 4.4 Grafik sendi 2 Menggunakan <i>Automated Tuning</i> PID pada Setengah Lingkaran .....	36
Gambar 4.5 Gerakan sendi 3 Menggunakan <i>Automated Tuning</i> PID pada Setengah Lingkaran .....	37
Gambar 4.6 Kecepatan sudut sendi 1 Menggunakan <i>Automated</i> <i>Tuning</i> PID pada Setengah Lingkaran.....	38
Gambar 4.7 Kecepatan sudut sendi 2 Menggunakan <i>Automated</i> <i>Tuning</i> PID pada Setengah Lingkaran.....	39

Gambar 4.8 Kecepatan sudut sendi 3 Menggunakan <i>Automated Tuning</i> PID pada Setengah Lingkaran .....	40
Gambar 4.9 Percepatan Sudut Sendi 1 menggunakan <i>Automated Tuning</i> PID pada Setengah Lingkaran .....	41
Gambar 4.10 Percepatan Sudut Sendi 2 menggunakan <i>Automated Tuning</i> PID pada Setengah Lingkaran .....	42
Gambar 4.11 Percepatan Sudut Sendi 3 menggunakan <i>Automated Tuning</i> PID pada Setengah Lingkaran .....	43
Gambar 4.12 Torsi Sendi 1 menggunakan <i>Automated Tuning</i> PID pada Setengah Lingkaran .....	44
Gambar 4.13 Torsi Sendi 2 menggunakan <i>Automated Tuning</i> PID pada Setengah Lingkaran .....	45
Gambar 4.14 Torsi Sendi 3 menggunakan <i>Automated Tuning</i> PID pada Setengah Lingkaran .....	46
Gambar 4.15: Grafik <i>Trajectory</i> Setengah Lingkaran menggunakan Metode Ziegler-Nichols .....	48
Gambar 4.16: Grafik Sendi 1 menggunakan Metode Ziegler-Nichols pada Setengah Lingkaran .....	49
Gambar 4.17: Grafik Sendi 2 menggunakan Metode Ziegler-Nichols pada Setengah Lingkaran .....	50
Gambar 4.18: Grafik Sendi 3 menggunakan Metode Ziegler-Nichols pada Setengah Lingkaran .....	51
Gambar 4.19 Kecepatan sudut sendi 1 Metode Ziegler-Nichols pada Setengah Lingkaran .....	52
Gambar 4.20 Kecepatan sudut sendi 2 Metode Ziegler-Nichols pada Setengah Lingkaran .....	53
Gambar 4.21 Kecepatan sudut sendi 3 Metode Ziegler-Nichols pada Setengah Lingkaran .....	54
Gambar 4.22 Percepatan sudut sendi 1 Metode Ziegler-Nichols pada Setengah Lingkaran .....	55

Gambar 4.23 Percepatan sudut sendi 2 Metode Ziegler-Nichols pada Setengah Lingkaran .....	56
Gambar 4.24 Percepatan sudut sendi 3 Metode Ziegler-Nichols pada Setengah Lingkaran .....	57
Gambar 4.25 Torsi Sendi 1 menggunakan Metode Ziegler-Nichols pada Setengah Lingkaran .....	58
Gambar 4.26 Torsi Sendi 2 menggunakan Metode Ziegler-Nichols pada Setengah Lingkaran .....	59
Gambar 4.27 Torsi Sendi 3 menggunakan Metode Ziegler-Nichols pada Setengah Lingkaran .....	60
Gambar 4.28 Grafik <i>Trajectory</i> Lingkaran menggunakan <i>Automated Tuning</i> PID .....	62
Gambar 4.29 Grafik Sendi 1 menggunakan <i>Automated Tuning</i> PID pada Lingkaran .....	63
Gambar 4.30 Grafik Sendi 2 menggunakan <i>Automated Tuning</i> PID pada Lingkaran .....	64
Gambar 4.31: Grafik Sendi 3 menggunakan <i>Automated Tuning</i> PID pada Lingkaran .....	65
Gambar 4.32 Kecepatan sudut sendi 1 Menggunakan <i>Automated Tuning</i> PID pada Lingkaran .....	66
Gambar 4.33 Kecepatan sudut sendi 2 Menggunakan <i>Automated Tuning</i> PID pada Lingkaran .....	67
Gambar 4.34 Kecepatan sudut sendi 3 Menggunakan <i>Automated Tuning</i> PID pada Lingkaran .....	68
Gambar 4.35 Percepatan Sudut Sendi 1 menggunakan <i>Automated Tuning</i> PID pada Lingkaran .....	69
Gambar 4.36 Percepatan Sudut Sendi 2 menggunakan <i>Automated Tuning</i> PID pada Lingkaran .....	70
Gambar 4.37 Percepatan Sudut Sendi 3 menggunakan <i>Automated Tuning</i> PID pada Lingkaran .....	71
Gambar 4.38 Torsi Sendi 1 menggunakan <i>Automated Tuning</i>	

PID pada Lingkaran .....	72
Gambar 4.39 Torsi Sendi 2 menggunakan <i>Automated Tuning</i>	
PID pada Lingkaran .....	73
Gambar 4.40 Torsi Sendi 3 menggunakan <i>Automated Tuning</i>	
PID pada Lingkaran .....	74
Gambar 4.41 Grafik <i>Trajectory</i> Lingkaran menggunakan Metode Ziegler-Nichols .....	76
Gambar 4.42 Gerakan Sendi 1 menggunakan Metode Ziegler-Nichols pada Lingkaran.....	77
Gambar 4.43 Gerakan Sendi 2 menggunakan Metode Ziegler-Nichols pada Lingkaran.....	78
Gambar 4.44 Gerakan Sendi 3 menggunakan Metode Ziegler-Nichols pada Lingkaran.....	79
Gambar 4.45 Kecepatan sudut sendi 1 Metode Ziegler-Nichols pada Lingkaran.....	80
Gambar 4.46 Kecepatan sudut sendi 2 Metode Ziegler-Nichols pada Lingkaran.....	81
Gambar 4.47 Kecepatan sudut sendi 3 Metode Ziegler-Nichols pada Lingkaran.....	82
Gambar 4.48 Percepatan sudut sendi 1 Metode Ziegler-Nichols pada Lingkaran.....	83
Gambar 4.49 Percepatan sudut sendi 2 Metode Ziegler-Nichols pada Lingkaran.....	84
Gambar 4.50 Percepatan sudut sendi 3 Metode Ziegler-Nichols pada Lingkaran.....	85
Gambar 4.51 Torsi Sendi 1 menggunakan Metode Ziegler-Nichols pada Lingkaran.....	86
Gambar 4.52 Torsi Sendi 2 menggunakan Metode Ziegler-Nichols pada Lingkaran.....	87
Gambar 4.53 Torsi Sendi 3 menggunakan Metode Ziegler-Nichols pada Lingkaran.....	88

Gambar 4.54 Grafik Sendi 1 menggunakan <i>Automated Tuning</i> PID pada <i>Trajectory</i> Bebas .....	90
Gambar 4.55 Grafik Sendi 2 menggunakan <i>Automated Tuning</i> PID pada <i>Trajectory</i> Bebas .....	91
Gambar 4.56 Grafik Sendi 3 menggunakan <i>Automated Tuning</i> PID pada <i>Trajectory</i> Bebas .....	92
Gambar 4.57 Kecepatan sudut sendi 1 Menggunakan <i>Automated Tuning</i> PID pada <i>Trajectory</i> Bebas .....	93
Gambar 4.58 Kecepatan sudut sendi 2 Menggunakan <i>Automated Tuning</i> PID pada <i>Trajectory</i> Bebas .....	94
Gambar 4.59 Kecepatan sudut sendi 3 Menggunakan <i>Automated Tuning</i> PID pada <i>Trajectory</i> Bebas .....	95
Gambar 4.60 Percepatan Sudut Sendi 1 menggunakan <i>Automated Tuning</i> PID pada <i>Trajectory</i> Bebas .....	96
Gambar 4.61 Percepatan Sudut Sendi 2 menggunakan <i>Automated Tuning</i> PID pada <i>Trajectory</i> Bebas .....	97
Gambar 4.62 Percepatan Sudut Sendi 3 menggunakan <i>Automated Tuning</i> PID pada <i>Trajectory</i> Bebas .....	98
Gambar 4.63 Torsi Sendi 1 menggunakan <i>Automated Tuning</i> PID pada <i>Trajectory</i> Bebas.....	99
Gambar 4.64 Torsi Sendi 2 menggunakan <i>Automated Tuning</i> PID pada <i>Trajectory</i> Bebas.....	100
Gambar 4.65 Torsi Sendi 3 menggunakan <i>Automated Tuning</i> PID pada <i>Trajectory</i> Bebas.....	101

*(Halaman ini sengaja dikosongkan)*



## DAFTAR TABEL

Tabel 2.1: Nilai PID yang diperoleh tiap sendi.....	6
Tabel 2.2: Perbandingan performa kontrol PID dengan Fuzzy Logic Control .....	7
Tabel 2.3: Parameter Denavit dan Hartenberg Lengan Robot 3 DOF.....	10
Tabel 2.4 Metode Ziegler-Nichols tipe 1.....	19
Tabel 2.5 Metode Ziegler-Nichols tipe 2.....	20
Tabel 3.1: Parameter Permodelan Lengan Robot .....	28
Tabel 3.2: Parameter DC motor.....	29
Tabel 4.1: Tabel Tuning PID pada <i>Trajectory Setengah</i> Lingkaran dengan <i>Automated Tuning</i> .....	33
Tabel 4.2: Tabel Tuning PID Metode Ziegler-Nichols Setengah Lingkaran .....	47
Tabel 4.3 Tabel Tuning PID pada Trajectory Lingkaran dengan <i>Automated Tuning</i> .....	61
Tabel 4.4 Tabel Tuning PID Metode Ziegler-Nichols Lingkaran .....	75
Tabel 4.5: Tabel Tuning PID pada Trajectory Bebas dengan <i>Automated Tuning</i> .....	89
Tabel 4.6 : Tabel Tuning PID Metode Ziegler-Nichols Lingkaran .....	102

*(Halaman ini sengaja dikosongkan)*

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Berdasarkan data hasil survei *International Federation of Robotics* (IFR) yaitu organisasi nirlaba profesional yang didirikan pada tahun 1987 untuk mempromosikan, memperkuat dan melindungi industri robotika di seluruh dunia mencatat bahwa penjualan robot setiap tahunnya meningkat. Sejak tahun 2010, permintaan untuk robot industri meningkat pesat karena tren yang sedang berlangsung menuju otomatisasi dan perbaikan teknis inovatif yang berlanjut di robot industri. Antara tahun 2012 sampai dengan tahun 2017, rata-rata peningkatan penjualan robot di dunia adalah 19%. Pada tahun 2017 tercatat bahwa penjualan robot industri mencapai 381.335 unit.



Gambar 1.1: Lengan robot untuk manufaktur

Robot industri sampai saat ini masih dioperasikan karena fleksibel, mudah dioperasikan, tingkat kepresisian yang tinggi, dan

mampu menyelesaikan pekerjaan yang rumit. Robot industri yang sering dipakai dalam manufaktur adalah lengan robot. Penelitian tentang lengan robot terus dilakukan seiring dengan perkembangan robot di dunia industri. Dengan adanya 3 sumbu rotari dan 3 sumbu translasi membuat lengan robot sangat fleksibel dalam bergerak dengan ruang kerja yang luas namun menempati ruang yang lebih sempit. Dengan motor yang kuat, lengan robot mampu dibebani cukup tinggi sebagai alat loading atau handling. Fleksibilitas gerakannya membuat robot ini cocok untuk tugas rumit seperti manufaktur. Gambar 1.1 menunjukkan aplikasi robot dalam bidang manufaktur. Robot industri saat ini dibuat sangat berat untuk mencapai kekuatan tinggi yang meningkatkan akurasi gerakan robot. Namun berat ini membatasi kecepatan robot dan energi yang dibutuhkan untuk menggerakkan sistem. Kebutuhan untuk mencapai kecepatan yang lebih tinggi dan kinerja sistem yang lebih baik membuatnya perlu untuk mempertimbangkan generasi baru manipulator ringan sebagai alternatif. Manipulator yang ringan membutuhkan lebih sedikit energi untuk bergerak dan memiliki kemampuan muatan yang lebih besar dan kemampuan *manuver* yang lebih besar.

Manipulator lengan robot adalah robot yang paling umum digunakan dalam aplikasi industri. Manipulator lengan robot terdiri dari serangkaian tautan, dihubungkan dengan sambungan, yang dapat dipindahkan secara translasi atau rotasi. Sebagian dari sendi yang digerakkan akan berpengaruh dari alas ke *end effector*. Lengan manipulator hampir mirip dengan tangan manusia, dan *end-effector* dapat dirancang untuk melakukan tugas yang diinginkan. Agar berhasil memenuhi tugas yang diinginkan, kinematika dan dinamika sebuah manipulator harus dipelajari terlebih dulu, dan kemudian pengendali juga perlu dirancang dan diimplementasikan dalam sistem mekanis manipulator. Berbagai

pengendali telah dirancang dan diterapkan di manipulator robot. Berbagai jenis pengontrol ini dalam hal kinerja akan dievaluasi. Pada sistem kontrol lengan robot berfungsi agar lengan robot dapat bergerak sesuai dengan perintah yang diberikan. Sistem gerak lengan robot ini menggunakan metode *inverse kinematic* dalam melakukan pergerakan lengan robot. Metode yang digunakan untuk sistem kendali lengan robot digunakan kendali *Propositional Integral Derivatif* (PID). PID merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut. Pengontrol PID adalah pengontrol konvensional yang banyak dipakai dalam dunia industri. Alasan menggunakan metode PID pada lengan robot karena metode ini mudah untuk dioperasikan, mudah dipahami, dan banyak dipakai dalam dunia industri.

Kinerja dinamis adalah faktor penting dalam penanganan robot industri, terutama hal yang paling penting dalam analisis dinamis adalah ketika kecepatan tinggi dan akurasi yang baik diperlukan secara bersamaan. Perilaku robot industri biasanya rumit dikarenakan oleh karakteristik dinamis seperti interaksi antara sendi, efek *nonlinier* seperti *Coriolis* dan gaya *sentrifugal*, dan berbagai inersia sendi. Analisis dinamika lengan robot dapat dilihat dari kecepatan sudut, percepatan sudut, dan torsi saat lengan robot bergerak. Oleh karena itu, pada penelitian ini berfokus pada analisis dinamika dan kendali lengan robot menggunakan mode kontrol PID.

## **1.2. Rumusan Permasalahan**

Berdasarkan permasalahan dari latar belakang maka didapat rumusan masalah pada tugas akhir ini adalah sebagai berikut:

1. Bagaimana model dinamis lengan robot yang bergerak mengikuti perencanaan *trajectory* ?
2. Bagaimana respon dinamis lengan robot menggunakan sistem kontrol PID ?

### **1.3. Batasan Masalah**

Penelitian ini memiliki beberapa batasan masalah sebagai berikut:

1. Proses simulasi dilakukan dengan bantuan software Matlab-Simulink
2. Lengan robot memiliki 3 derajat kebebasan
3. Motor DC dimodelkan dengan *transfer function*
4. Lengan robot dianggap rigid
5. Gaya gesek, getaran, dan deformasi yang terjadi saat lengan robot bergerak diabaikan
6. Penelitian ini berfokus pada pengendali dan respon pengendali

### **1.4. Tujuan Penelitian**

Tujuan dari tugas akhir ini adalah sebagai berikut:

1. Membuat model lengan robot yang bergerak mengikuti perencanaan *trajectory*.
2. Menganalisa respon dinamis lengan robot menggunakan sistem kontrol PID

### **1.5. Manfaat Penelitian**

Manfaat dari penulisan tugas akhir ini adalah:

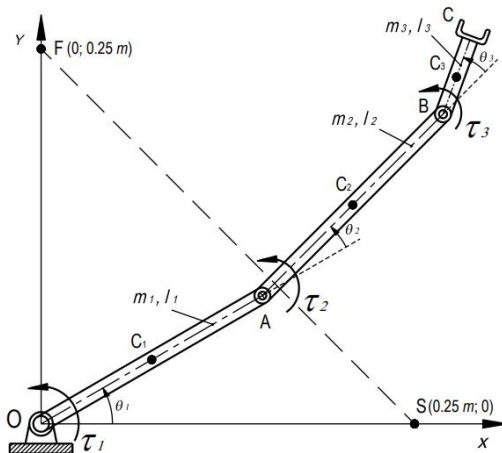
1. Mengembangkan penelitian dari lengan robot.
2. Dapat digunakan sebagai pembelajaran sistem kendali.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Hasil Penelitian Terdahulu

Penelitian yang pertama dilakukan Thien Van Nguyen dkk dengan judul *Controlling the Motion of a Planar 3-DOF Manipulator Using PID Controllers*. penelitian ini menyajikan metode untuk memodelkan manipulator 3-DOF dengan kontroler PID di Matlab-SimMechanics dan cara untuk mengetahui parameter optimal pada kontroler PID termasuk: *gain proporsional, gain integral, dan gain derivatif* menggunakan algoritma genetika (GA). Untuk tujuan itu, manipulator 3-DOF akan disimulasikan untuk melakukan tugas yang diberikan. Hasil yang dicapai dari model menunjukkan bahwa kesulitan dalam menyetel parameter pengontrol PID dapat diselesaikan menggunakan GA, sementara SimMechanics dianggap sebagai alat eksperimen virtual yang baik untuk mensimulasikan mekanisme.

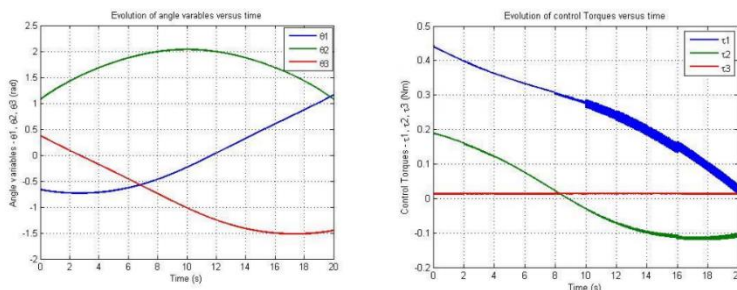


Gambar 2.1: Pemodelan lengan robot

Sendi ke	Kp	Ki	Kd
1	19.4350	21.245	-0.79
2	0.81	0.195	-0.285
3	0.016	0.007	-0.002

Tabel 2.1: Nilai PID yang diperoleh tiap sendi

SimMechanics didasarkan pada lingkungan Simulink dalam perangkat lunak Matlab dan dapat digunakan untuk memodelkan dan menganalisis sistem mekanis. Selain itu, animasi dari operasi mekanisme ditampilkan di layar selama analisis gerakan. Berkat karakteristik itu, para peneliti dapat menganalisis dan menyesuaikan dengan lebih mudah operasi mekanisme sesuai dengan tujuannya sendiri. Selain itu, properti ini juga berguna ketika peneliti ingin memverifikasi pemenuhan tugas yang diberikan mekanisme. Selain itu, dengan menghapus dan menambahkan blok dalam diagram SimMechanics, struktur sistem mekanik dapat dengan mudah diubah, maka sangat nyaman untuk robotika bidang desain.



Gambar 2.2 Perubahan Sudut dan Torsi terhadap Waktu pada saat Lengan Robot Bergerak oleh Thien Van Nguyen dkk

Penelitian yang kedua berjudul *Performance Analysis Of PID, PD And Fuzzy Controllers For Position Control Of 3-DOF Robot Manipulator*. Penelitian ini menyajikan analisis komparatif

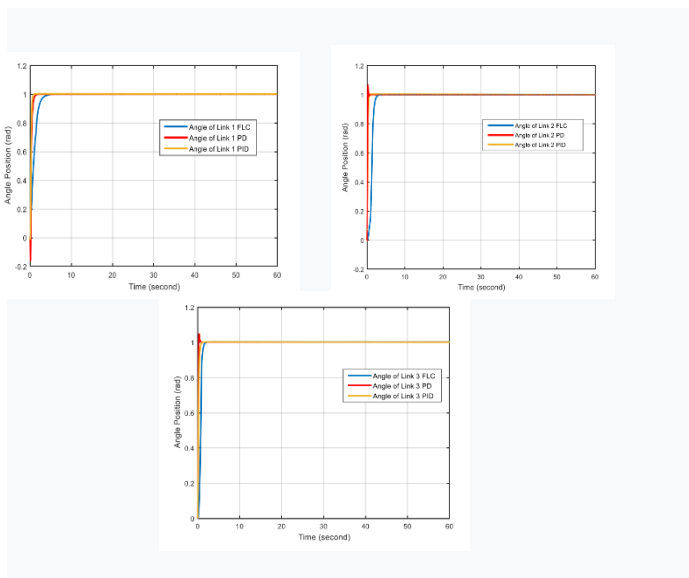


dari tiga pengendali pada robot 3-DOF manipulator. Derivatif integral proporsional (PID), derivatif proporsional (PD) dan Kontroler fuzzy logic (FL) dirancang dan diterapkan pada setiap sendi robot oleh simulasi. Kinerja setiap metode kontrol dinilai menggunakan transien dan karakteristik respons keadaan tunak. Perbandingan hasil yang diperoleh, PID dan PD tampil lebih baik dalam hal Rise Time dan Settling Time, sementara FLC bagus untuk mengurangi overshoot.

Tabel 2.2: Perbandingan Kontrol PID, PD, dan *Fuzzy Logic*

Variabel	PID	PI	FLC
<b>Kp Link 1</b>	90	76.599	20
<b>Kp Link 2</b>	100	205	67
<b>Kp Link 3</b>	75	60.795	65
<b>Kd Link 1</b>	90	21.999	9.275
<b>Kd Link 2</b>	15	13.799	13.275
<b>Kd Link 3</b>	15	8.549	11.975
<b>Ki Link 1</b>	1	-	-
<b>Ki Link 2</b>	1	-	-
<b>Ki Link 3</b>	1	-	-

Secara umum, setelah diamati bahwa ketiga pengontrol mampu melacak *set-point* secara efektif. FLC memiliki kinerja yang lebih baik dalam hal *overshoot* jika dibandingkan dengan PD dan PID. Keuntungan menggunakan PD dan PID, bahwa PD dan PID menunjukkan peningkatan kinerja jika dibandingkan dengan FLC. Pada sistem kontrol kedua PID dan FLC dapat bekerja secara efektif.



Gambar 2.3: Perubahan Sudut menggunakan Kontrol PID, PD, dan FLC pada tiap Sendi

Dalam penelitian ini, tiga kontrol posisi berbeda dengan metodologi telah dirancang dan dianalisis pada Manipulator robot 3-DOF. Pengendali PID, PD, dan FLC diterapkan pada setiap tautan robot manipulator dan perbandingan kinerja dibuat menggunakan karakteristik *transient* dan *steady state*. Hasilnya menunjukkan bahwa ketiga pengontrol mampu lacak *setpoint* dengan kesalahan kondisi stabil yang dapat diabaikan. Kontroler PID dan PD memberikan kinerja yang lebih baik dalam hal waktu naik dan waktu penyelesaian sementara FLC menghasilkan penurunan overshoot.

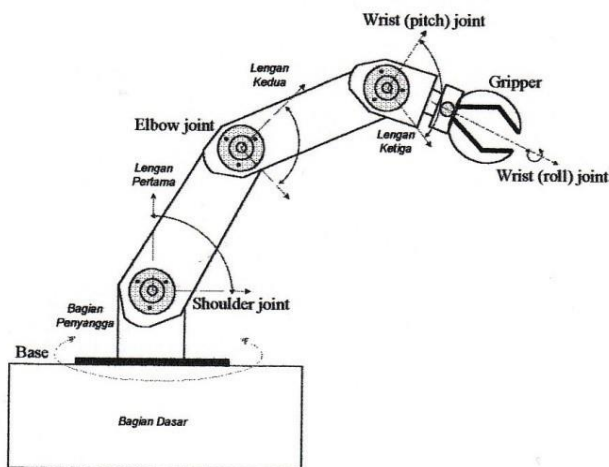
## 2.2 Teori Dasar

Pada bab ini akan diuraikan beberapa teori pendukung yang menjadi dasar keilmuan yang digunakan dalam penelitian. Teori-teori pendukung tersebut, mayoritas berdasarkan penelitian-

penelitian terdahulu. Hal ini berguna untuk menjelaskan bahwa dalam penelitian ini tetap mengacu pada teori yang sampai saat ini masih digunakan.

### 2.2.1. Sistem Lengan Robot

Sistem lengan robot memiliki sedikit atau banyak derajat kebebasan tergantung dari konfigurasinya. Diagram skematik dari sistem lengan robot yang sederhana ditunjukkan dalam. Pada sistem lengan robot ini, jika gaya memegang terlalu kecil, lengan robot akan menjatuhkan obyek mekaniknya, dan jika gayanya terlalu besar, maka lengan robot tersebut mungkin merusak atau menghancurkan obyek tersebut. Jadi lengan robot tersebut harus memiliki alat perasa-raba dan alat perasa-gaya. Alat ukur ketegangan semi konduktor seringkali digunakan untuk alat perasa-gaya tersebut.



Gambar 2.4 : Sistem lengan robot sederhana

Sistem lengan robot dikontrol oleh kontrol digital. Untuk kontrol computer yang memuaskan, harus mempunyai model matematika yang tepat dari sistem lengan robot tersebut. Karena kebanyakan sistem lengan robot melibatkan banyak ruas dan

hubungan yang diperpanjang, maka bukanlah suatu hal yang sederhana untuk menentukan ruas mana yang harus diputar dan hubungan mana yang harus diperpanjang untuk membawa lengan ke posisi dan orientasi yang dikehendaki. Untuk menghitung lokasi dan orientasi dari lengan, perlu diketahui jumlah rotasi dan gerakan linear yang terlibat dalam berbagai komponen dari sistem lengan robot (Ogata, 1996).

### 2.2.1.1 Denavit-Hartenberg Parameter

DH parameter adalah suatu cara untuk merepresentasikan analisa hubungan gerak rotasi dan translasi antara lengan-lengan yang terhubung dalam suatu manipulator diperkenalkan oleh Denavit dan Hartenberg (1955). Prinsip dasar (D-H ) adalah dengan melakukan tranformasi koordinat antara dua link yang berdekatan dan hasilnya adalah matriks (4x4). Variable D-H yaitu :  $\theta$ ,  $a$ ,  $d$ ,  $\alpha$  . Dimana  $a$  sebagai panjang link,  $\alpha$  sebagai sudut twist antar sendi,  $d$  sebagai jarak offset, dan  $\theta$  sebagai sudut joint (Singh, 2017).

DH parameter yang digunakan pada lengan robot 3 DOF adalah sebagai berikut

Tabel 2.3: Parameter Denavit dan Hartenberg Lengan Robot 3 DOF

Sambungan	$\theta_i$ (rad)	$a_i$ (m)	$d_i$ (m)	$\alpha$ (rad)	Joint Limits (deg)
1	1	0.105	0.173	$\pi/2$	[-180, 180]
2	2	0.7	0	0	[-90, 90]
3	3	0.31	0	0	[-270, 90]

### 2.2.1.2 Invers Kinematic

Kinematika mundur atau inverse kinematics adalah metode yang digunakan untuk mengubah posisi *end effector* pada sumbu X, sumbu Y, dan sumbu Z menjadi sudut yang harus

digerakkan oleh sendi. Invers kinematik mengacu pada penggunaan persamaan kinematika robot untuk menentukan parameter bersama yang memberikan posisi yang diinginkan pada posisi akhir atau efektor. Spesifikasi pergerakan robot sehingga posisi akhir atau efektor mencapai titik yang diinginkan adalah dikenal sebagai perencanaan gerak (Niku, 2011).

*Inverse kinematik* mengubah rencana gerak menjadi nilai sudut yang harus diberikan bagi aktuator atau penggerak dalam pergerakan robot. Sudut yang diperoleh dari perhitungan kinematika mundur dijadikan sebagai masukan untuk menggerakkan sendi *revolute* pada simulasi matlab simulink. Secara garis besar metode *inverse kinematic* akan mencari nilai-nilai parameter yang harus diberikan kepada setiap aktuator untuk mencapai tujuan akhir. Untuk mendapatkan nilai-nilai parameter tersebut, robot harus mengetahui terlebih dahulu manipulator yang dimilikinya, baik ukuran maupun jumlah aktuator serta derajat kebebasan yang ada. Kemudian, robot harus ditanamkan rumus-rumus yang didapat dari berbagai model perhitungan, baik dari segi analisa grafik langsung maupun menggunakan metode-metode dari berbagai peneliti.

### **2.2.2 Persamaan Dinamis Lengan Robot**

Model matematika untuk lengan robot manipulator sangat penting untuk pengembangan sistem. Untuk mengendalikan pergerakan lengan, analisis manipulator robot harus melibatkan torsi dari sendi lengan robot. Perkembangan model matematika untuk kontrol torsi pada sendi dari manipulator harus melibatkan deskripsi dinamis lengkap dari seluruh sistem yang terdiri dari link dan aktuator (Tsai, 1999).

Analisis dinamika lengan robot mempelajari hubungan antara sendi torsi/gaya yang diterapkan oleh aktuator dan posisi, kecepatan dan percepatan robot lengan sehubungan dengan waktu. Dinamika manipulator robot sangat kompleks dan faktor nonlinier yang membuat sulit mengendalikan secara akurat. Persamaan dinamis dari manipulator robot adalah biasanya diwakili oleh

Persamaan Diferensial Non-Linear digabungkan persamaan yang berasal dari Lagrangians (Lafmejani & Zaradipour, 2014).

Persamaan Dinamis dari lengan robot dideskripsikan sebagai berikut

$$M(q) \ddot{q} + C(q, \dot{q}) + G(q) = \tau \quad (2.1)$$

dimana  $\tau$  adalah torsi aktuator,  $M(q)$  adalah matriks inersia,  $C(q, \dot{q})$  adalah matriks gaya sentripetal/koriolis, dan  $G(q)$  adalah gravitasi. Variabel  $q$  adalah vektor dari sudut sendi untuk sambungan *revolute*.

Persamaan matriks inersia lengan robot 3 DOF sebagai berikut

$$M = \begin{bmatrix} M_{11} & 0 & 0 \\ 0 & M_{22} & 0 \\ 0 & 0 & M_{33} \end{bmatrix} \quad (2.2)$$

$$\begin{aligned} M_{11} &= \frac{1}{2} m_1 l_1^2 + \frac{1}{3} m_2 l_2^2 \cos^2 \theta_2 + \frac{1}{3} m_3 l_3^2 \cos(\theta_2 + \theta_3)^2 + \\ &\quad m_3 l_2^2 \cos^2 \theta_2 + m_3 l_2 l_3 \cos(\theta_2 + \theta_3) \cos \theta_2 \\ M_{22} &= \frac{1}{3} m_2 l_2^2 + \frac{1}{3} m_3 l_2^2 + m_3 l_2 l_3 \cos \theta_3 \\ M_{23} &= \frac{1}{3} m_3 l_3^2 + m_3 l_3^2 + \frac{1}{3} m_3 l_2 l_3 \cos \theta_3 \\ M_{32} &= \frac{1}{3} m_3 l_3^2 + m_3 l_3^2 + \frac{1}{3} m_3 l_2 l_3 \cos \theta_3 \\ M_{33} &= \frac{1}{3} m_3 l_3^2 \end{aligned}$$

Persamaan gaya sentripetal/koriolis lengan robot 3 DOF sebagai berikut

$$C = \begin{bmatrix} C1 \\ C2 \\ C3 \end{bmatrix} \quad (2.3)$$

$$\begin{aligned}
C_1 &= \left[ -\frac{4}{3} m_2 l_2^2 \sin 2\theta_2 - \frac{1}{3} m_3 l_3^2 \sin 2(\theta_2 + \theta_3) \right] \dot{\theta}_2 \dot{\theta}_1 + \\
&\quad \left[ -\frac{1}{3} m_3 l_3^2 \sin 2(\theta_2 + \theta_3) - m_3 l_2 l_3 \sin(\theta_2 + \theta_3) \right] \dot{\theta}_3 \dot{\theta}_1 \\
C_2 &= \left[ -m_3 l_2 l_3 \sin \theta_3 \right] \dot{\theta}_2 \dot{\theta}_1 + \left[ -\frac{1}{2} m_3 l_2 l_3 \sin \theta_3 \right] \dot{\theta}_3^2 + \\
&\quad \left[ -\frac{1}{6} m_2 l_2^2 \sin 2\theta_2 + \frac{1}{6} m_3 l_3^2 \sin 2(\theta_2 + \theta_3) + \right. \\
&\quad \left. -\frac{1}{2} m_3 l_2^2 \sin 2\theta_2 + \frac{1}{2} m_3 l_2 l_3 \sin(2\theta_2 + \theta_3) \right] \dot{\theta}_1^2 \\
C_3 &= \left[ \frac{1}{2} m_3 l_2 l_3 \sin \theta_3 \right] \dot{\theta}_1^2 + \\
&\quad \left[ \frac{1}{6} m_2 l_2^2 \sin 2(\theta_2 + \theta_3) + \frac{1}{2} m_3 l_2 l_3 \cos \theta_2 \sin(\theta_2 + \right. \\
&\quad \left. \theta_3) \right] \dot{\theta}_3^2
\end{aligned}$$

Persamaan gravitasi lengan robot 3 DOF sebagai berikut

$$G = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix} \quad (2.4)$$

$$\begin{aligned}
G_1 &= 0 \\
G_2 &= \frac{1}{2} m_2 g l_2 \cos \theta_2 + \frac{1}{2} m_3 g l_3 \cos(\theta_2 + \theta_3) + \frac{1}{2} \\
&\quad m_3 g l_3 \cos \theta_2 \\
G_3 &= \frac{1}{2} m_3 g l_3 \cos(\theta_2 + \theta_3)
\end{aligned}$$

### 2.2.3 Aktuator Pada Sendi

Dinamika dari lengan robot digerakkan oleh motor arus searah (motor DC). Motor arus searah (DC) dapat memberikan torsi awal yang tinggi dan juga dimungkinkan untuk mendapatkan kontrol kecepatan pada rentang yang luas. Metode kontrol kecepatan biasanya lebih sederhana dan lebih murah daripada motor arus bolak-balik. Motor arus searah memegang status yang sangat penting dalam sistem kontrol otomatis penggerak listrik.

Dibandingkan dengan motor arus bolak-balik, kinerja kontrol kecepatan motor arus searah jauh lebih baik. Motor DC digunakan untuk peralatan elektronik yang membutuhkan berbagai pengaturan kecepatan dengan kepresisian tinggi dan telah banyak digunakan dalam peralatan mesin kontrol numerik yang terkomputerisasi dan kontrol proses. Kecepatan motor DC dapat diatur hingga tingkat yang sangat besar untuk memberikan kemudahan kontrol dan kinerja tinggi. Torsi pada motor DC dideskripsikan sebagai berikut

$$K_t i = \tau_m \quad (2.5)$$

$$K_t i = J_m \frac{d^2 \theta_m}{dt^2} + B_m \frac{d\theta_m}{dt} \quad (2.6)$$

dari persamaan tersebut diubah menjadi Transformasi Laplace, sehingga menjadi

$$J_m \omega_m s(s) + B_m \omega_m(s) = K_t I(s) \quad (2.7)$$

dari persamaan elektronika didapatkan

$$V_{in} = R_i i + L_a \frac{di}{dt} + K_e \frac{d\theta_m}{dt} \quad (2.8)$$

dari persamaan diubah menjadi transformasi laplace

$$V_{in}(s) = R I(s) + L_a s I(s) + K_e \omega(s) \quad (2.9)$$

dimana  $\omega$  adalah kecepatan sudut,  $J_m$  adalah inersia motor,  $R_a$  adalah resistansi aktuator,  $L_a$  adalah induktansi aktuator,  $K_t$  adalah konstanta torsi,  $K_e$  adalah konstanta *back* EMF,  $B_m$  adalah koefisien redaman motor, dan  $\omega_m = \dot{\theta}_m$  adalah kecepatan motor (Agbaraji, 2017).

Subsistem mekanis dari aktuator dan lengan robot terhubung di aktuator dan menghubungkan roda gigi. Torsi total pada motor sebagai berikut



$$\Sigma \text{ torque motor} = J_m \ddot{\theta}_m \quad (2.10)$$

$$\tau - B_m \dot{\theta}_m - r_m F = J_m \ddot{\theta}_m \quad (2.11)$$

$$\tau = B_m \dot{\theta}_m + r_m F + J_m \ddot{\theta}_m \quad (2.12)$$

total torsi pada lengan robot adalah

$$\Sigma \text{ torque link} = J_1 \ddot{\theta} \quad (2.13)$$

$$r_1 F - B_1 \dot{q} = J_1 \ddot{q} \quad (2.14)$$

$$F = \frac{J_1 + B_1 \dot{q}}{r_1} \quad (2.15)$$

jika disubstitusikan persamaan 2.10 dan persamaan 2.13 menjadi

$$\tau = J_m \ddot{\theta}_m + B_m \dot{\theta}_m + r_m r_1 (J_1 \ddot{q} + B_1 \dot{q}) \quad (2.16)$$

Kinematika gear untuk gear motor dan tautan gigi adalah sebagai berikut

$$\frac{r_m}{r_l} = \frac{N_m}{N_l} = \frac{\dot{q}}{\dot{\theta}_m} = \frac{q}{\theta_m} \quad (2.17)$$

dimana  $r_m$  adalah jari-jari roda gigi motor,  $r_l$  adalah jarak link gear,  $N_m$  adalah jumlah gigi motor,  $N_l$  adalah jumlah gigi link gear,  $F$  adalah kekuatan kontak,  $J_1$  adalah inersia pada link,  $B_1$  adalah koefisien redaman *link*.

Posisi sudut pada link berasal dari posisi motor dan rasio gear motor dan gear link, sebagaimana

$$\theta_m = \left( \frac{r_l}{r_m} \right) q \quad (2.18)$$

$$\dot{\theta}_m = \left( \frac{r_l}{r_m} \right) \dot{q} \quad (2.19)$$

$$\ddot{\theta}_m = \left( \frac{r_l}{r_m} \right) \ddot{q} \quad (2.20)$$

dengan substitusi dari persamaan 2.18 2.19 2.20 didapatkan persamaan

$$\tau = \left( \frac{r_l}{r_m} J_m + \frac{r_l}{r_m} J_1 \right) \ddot{q} + \left( \frac{r_l}{r_m} B_m + \frac{r_l}{r_m} B_1 \right) \dot{q} \quad (2.21)$$

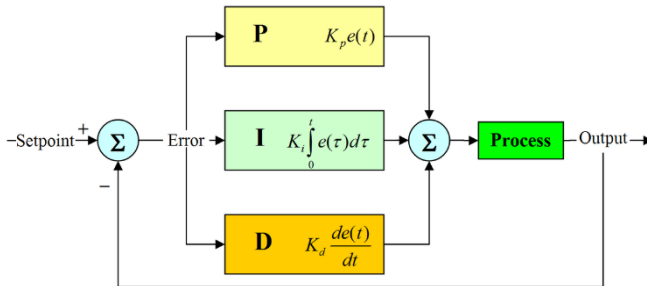
jika dikombinasikan antara persamaan subsistem mekanik dan elektrik dari aktuator, didapatkan persamaan dinamis

$$\left( \frac{r_l}{r_m} J_m + \frac{r_l}{r_m} J_1 \right) \ddot{q} + \left( \frac{r_l}{r_m} B_m + \frac{r_l}{r_m} B_1 \right) \dot{q} = K_d I \quad (2.22)$$

$$V_{in} = RI + L_a sI + \frac{r_l}{r_m} K_e \dot{q} \quad (2.23)$$

#### 2.2.4 Sistem Kendali PID

PID (*Proportional-Integral-Derivative controller*) merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut. Pengontrol PID adalah pengontrol konvensional yang banyak dipakai dalam dunia industri. Kendali PID memiliki beberapa kelebihan diantaranya adalah mudah dirancang, memiliki harga yang murah, perawatan yang tidak mahal, serta tidak memerlukan keahlian khusus bagi operator. PID blok diagram ditunjukkan pada 2.5



Gambar 2.5. Kontrol PID pada sistem

Komponen kontrol PID ini terdiri dari tiga jenis yaitu Proportional, Integratif dan Derivatif. Ketiganya dapat dipakai bersamaan maupun sendiri-sendiri tergantung dari respon yang kita inginkan terhadap suatu plant. PID adalah salah satu controller untuk sistem linier dan juga cukup mudah untuk penggunaannya. Maka dari itu penggunaan PID controller dirasa cukup untuk memberikan dampak pada alat penelitian kali ini.

Sistem kendali PID merupakan sistem kendali close loop yang cukup sederhana dan memiliki performa yang bagus. Namun kendali ini tidak dapat bekerja dengan baik apabila terjadi ketidakpastian dan ketidaklinieran pada sistem. Meskipun demikian sistem kendali PID dapat dikombinasikan dengan sistem kendali lain seperti *fuzzy control*, *adaptive control*, *sliding mode control*, dan *robust control* untuk menghasilkan performa yang lebih baik.

#### 2.2.4.1 Kontrol Proporsional

$K_p$  adalah Konstanta Proporsional.  $K_p$  berlaku sebagai *Gain* (penguat) saja tanpa memberikan efek dinamik kepada kinerja kontroler. Penggunaan kontrol P memiliki berbagai keterbatasan karena sifat kontrol yang tidak dinamik ini. Walaupun demikian dalam aplikasi-aplikasi dasar yang sederhana kontrol P ini cukup mampu untuk memperbaiki respon transien khususnya *rise time* dan *settling time*. Pengontrol proporsional memiliki

keluaran yang sebanding/proporsional dengan besarnya sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya).

#### **2.2.4.2 Kontrol Integratif**

Pengontrol Integral berfungsi menghasilkan respon sistem yang memiliki kesalahan keadaan nol (*Error Steady State* = 0). Jika sebuah pengontrol tidak memiliki unsur integrator, pengontrol proporsional tidak mampu menjamin keluaran sistem dengan kesalahan keadaan nilainya nol. Kontrol I dapat memperbaiki sekaligus menghilangkan respon steady-state, namun pemilihan  $K_i$  yang tidak tepat dapat menyebabkan respon transien yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem. Pemilihan  $K_i$  yang sangat tinggi justru dapat menyebabkan output berosilasi karena menambah orde sistem. Keluaran pengontrol ini merupakan hasil penjumlahan yang terus menerus dari perubahan masukannya. Jika sinyal kesalahan tidak mengalami perubahan, maka keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Sinyal keluaran pengontrol integral merupakan luas bidang yang dibentuk oleh kurva kesalahan / error.

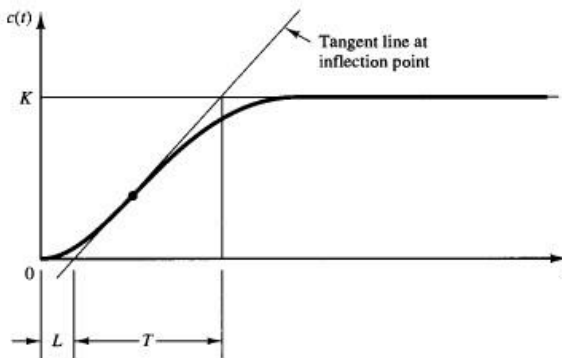
#### **2.2.4.3 Kontrol Derivatif**

Keluaran pengontrol diferensial memiliki sifat seperti halnya suatu operasi derivatif. Perubahan yang mendadak pada masukan pengontrol akan mengakibatkan perubahan yang sangat besar dan cepat. Ketika masukannya tidak mengalami perubahan. Dengan sifat ini ia dapat digunakan untuk memperbaiki respon transien dengan memprediksi error yang akan terjadi. Kontrol Derivative hanya berubah saat ada perubahan error sehingga saat error statis kontrol ini tidak akan bereaksi, hal ini pula yang menyebabkan kontroler derivatif tidak dapat dipakai sendiri.

#### **2.2.4.4 Metode Ziegler-Nichols**

Salah satu metoda mencari parameter-parameter PID adalah dengan metoda Ziegler-Nichols. Metoda ini memiliki 2 tipe,

yaitu tipe 1 (*open loop*) dan tipe 2 (*close loop*). Metode ke-1 didasarkan pada respon *plant* terhadap masukan dalam sistem *open loop*. *Plant* yang tidak mempunyai integrator, menghasilkan kurva tanggapan terhadap masukan tangga seperti kurva huruf S pada Gambar. Kurva tanggapan *plant* digunakan untuk mencari waktu tunda  $L$  dan konstanta waktu  $T$ .



Gambar 2.6 kurva S Ziegler-Nichols Tipe 1

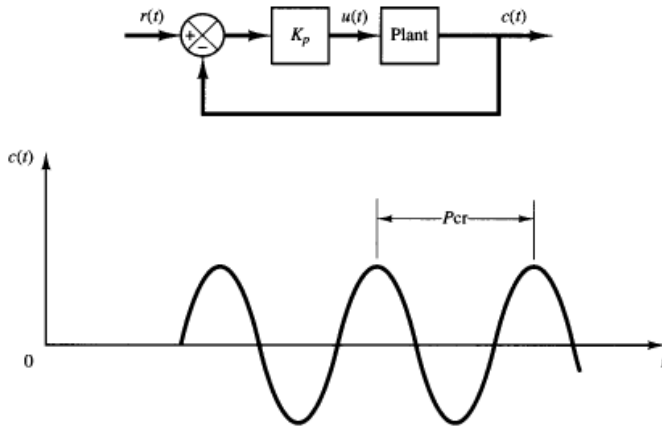
Parameter-parameter yang didapat dari kurva reaksi digunakan untuk menentukan parameter-parameter pengendali PID berdasarkan tetapan empiris Ziegler-Nichols. Rumus-rumus untuk parameter pengendali menggunakan metode kurva reaksi ditabelkan pada Tabel

Tabel 2.4 Metode Ziegler-Nichols tipe 1

Pengendali	$K_p$	$T_i$	$T_d$
P	$T/L$	$\sim$	0
PI	$0,9T/L$	$L/0,3$	0
PID	$1,2T/L$	$2L$	$0,5L$

Pada metode ke-2 dilakukan dalam *close loop*, dimana masukan referensi yang digunakan adalah fungsi tangga (step). Pengendali pada metode ini hanya pengendali proporsional.  $K_p$ ,

dinaikkan dari 0 hingga nilai kritis  $K_p$ , sehingga diperoleh keluaran yang terus-menerus berosilasi dengan amplitudo yang sama. Nilai kritis  $K_p$  ini disebut sebagai *ultimated gain*.



Gambar 2.7 Kurva Ziegler-Nichols Tipe 2

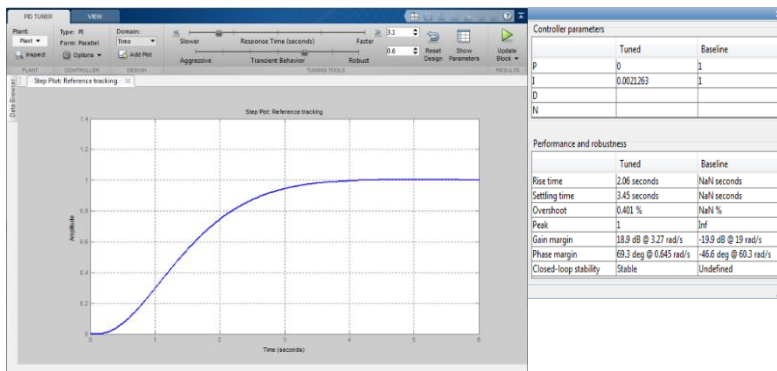
Dari metode ini akan diperoleh nilai *critical gain*  $K_{cr}$  dan periode kritis  $P_{cr}$ . Nilai periode kritis  $P_{cr}$  diperoleh setelah keluaran sistem mencapai kondisi yang terus menerus berosilasi. Konstanta-konstanta pengendali disesuaikan dengan tetapan empiris Ziegler-Nichols pada Tabel 2.5

Tabel 2.5 Metode Ziegler-Nichols tipe 2

Pengendali	$K_p$	$T_i$	$T_d$
P	$0,5K_{cr}$	~	0
PI	$0,45K_{cr}$	$P_{cr}/1,2$	0
PID	$0,6K_{cr}$	$0,5P_{cr}$	$0,125P_{cr}$

### 2.2.4.5 Automated Tuning PID

*Automated tuning* PID pada Matlab-Simulink menggunakan kondisi awal yang ditentukan dalam model Simulink sebagai titik operasi. Desain kontroler *Automated tuning* PID untuk *plant* yang telah dilinierisasi. *Automated Tuning* PID pada Matlab-Simulink secara otomatis membuat linier pada titik operasi yang ditentukan oleh model kondisi awal.

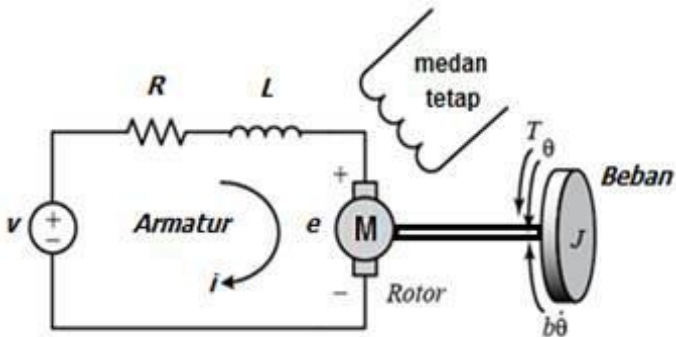


Gambar 2.8 PID *Tuner* pada Matlab-Simulink

Secara otomatis, *Automated tuning* PID menghitung model linier dari *plant* di model yang digunakan. *Automated tuning* PID menganggap instalasi sebagai kombinasi semua blok antara output dan input pengontrol PID. Dengan demikian, instalasi mencakup semua blok dalam *loop* kontrol, selain dari pengontrol itu sendiri. *PID Tuner* dianggap sebagai *plant* semua blok dalam *loop* antara *output* dan *input* blok Kontroler PID. Blok di *plant* yang digunakan dapat mencakup *nonlinier*. Karena penyetelan otomatis memerlukan model linier, *PID Tuner* menghitung perkiraan linierisasi instalasi pada model yang digunakan. Model linierisasi adalah sebuah pendekatan untuk sistem *nonlinear*, yang berlaku di daerah kecil di sekitar titik operasi dari sistem.

### 2.2.5 Motor Arus Searah

Motor arus searah (DC) dapat memberikan torsi awal yang tinggi dan juga dimungkinkan untuk mendapatkan kontrol kecepatan pada rentang yang luas. Metode kontrol kecepatan biasanya lebih sederhana dan lebih murah daripada motor arus bolak-balik. Motor arus searah memegang status yang sangat penting dalam sistem kontrol otomatis penggerak listrik. Dibandingkan dengan motor arus bolak-balik, kinerja kontrol kecepatan motor arus searah jauh lebih baik.



Gambar 2.9 Pemodelan motor DC

Motor DC digunakan untuk peralatan elektronik yang membutuhkan berbagai pengaturan kecepatan dengan kepresisian tinggi dan telah banyak digunakan dalam peralatan mesin kontrol numerik yang terkomputerisasi dan kontrol proses. Kecepatan motor DC dapat diatur hingga tingkat yang sangat besar untuk memberikan kemudahan kontrol dan kinerja tinggi. Pengaplikasian motor DC biasanya pada kendaraan listrik, robot industri, dan peralatan elektronik yang lain.



## 2.2.6 Perencanaan Trajectory

### 2.2.6.1 Joint Space Trajectory Planning

Perencanaan *trajectory* dengan secara langsung menentukan konfigurasi atau sudut masing-masing sendi dari titik akhir pergerakan robot disebut dengan *joint space trajectory planning*. Perencanaan pada joint space lebih sederhana dan cepat, dikarenakan tidak diperlukan *path planning* dan perhitungan *inverse kinematics* pada setiap perpindahan path point. Dengan pemanfaatan perencanaan ini, maka titik akhir akan dicapai setepat mungkin sesuai dengan acuan yang diberikan.

Perencanaan banyak diaplikasikan pada robot atau mesin yang orientasi gerakannya titik ke titik seperti mesin bor. Namun dalam aplikasinya ditambahkan perhitungan *inverse kinematics* diawal untuk mengkonversi koordinat awal dan titik akhir *end effector* menjadi konfigurasi sudut awal dan akhir sendi. Kelemahan *trajectory planning* ini adalah posisi *end effector* tidak secara langsung terkontrol dan tidak dapat melewati halang rintang pada *workspace*.

### 2.2.6.2 Cartesian Trajectory Planning

*Trajectory planning* yang terbaik dilakukan dengan menentukan secara pasti posisi dan orientasi *end effector* pada semua path points. Perencanaan *trajectory* seperti ini dikenal dengan *cartesian trajectory planning*. Untuk mencapai *trajectory* ini, maka setiap waktu sampling sistem harus memperoleh data sudut setiap sendi lalu mengkonversikan ke posisi orientasi *end effector* menggunakan *forward kinematics*. Setelah itu dihitung nilai kesalahan *trajectory* yang terjadi, dan akan ditentukan posisi berikutnya. Posisi yang didapat dikonversikan lagi menggunakan *invers kinematics* untuk kemudian diubah menjadi konfigurasi sudut.

*Cartesian trajectory planning* baik digunakan pada robot manipulator yang fungsinya mengharuskan *end effector* bergerak

mengikuti jalur lintasan tertentu seperti mesin pengecatan namun komputasinya yang cukup lama membuat *trajectory* ini jarang dipilih (Nurhakim, 2010).

## BAB III

### METODOLOGI

#### 3.1 Metode yang Digunakan

Proses perancangan tugas akhir memerlukan metode yang terstruktur agar tujuan dari penelitian ini tercapai. Berikut merupakan langkah-langkah dari penelitian ini, yaitu

1. Studi Literatur

Pada tahap studi literatur dilakukan kajian secara teoritis mengenai pemahaman dasar mengenai lengan robot, analisa dinamis, kontrol PID, rangkaian elektronika dan software yang mendukung untuk penyelesaian permasalahan dalam penelitian ini. Studi pustaka yang digunakan diperoleh dari buku, jurnal, internet atau penelitian sebelumnya.

2. Perumusan Masalah dan Penetapan Tujuan Penelitian

Perumusan masalah dalam penelitian ini untuk mengidentifikasi permasalahan guna menjawab latar belakang dari penelitian ini. Perumusan masalah penelitian ini adalah bagaimana cara membuat desain dan kontrol pada 3 DOF manipulator lengan robot serta analisis dinamikanya. Sedangkan tujuan penelitian ditetapkan untuk menjawab permasalahan dari penelitian ini, yaitu untuk membuat desain dan kontrol pada 3 DOF manipulator lengan robot serta analisis dinamikanya.

3. Membuat Desain dan Simulasi Lengan Robot

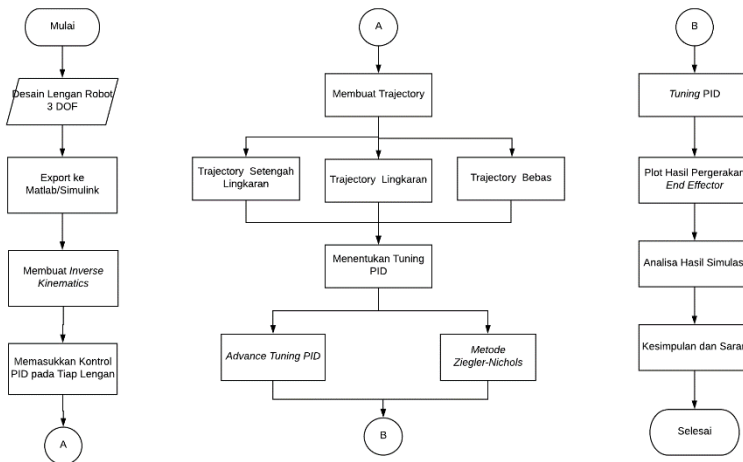
Penulis membuat model 3 dimensi dalam proses desain lengan robot. Pembuatan model lengan robot dimulai dengan pembuatan alas, *link* 1, *link* 2, dan *link* 3. Bagian-bagian dari lengan robot kemudian dirakit dan dipastikan agar setiap sambungan terhubung sesuai dengan perencanaan. Bentuk lengan robot kemudian diubah ke dalam blok diagram kemudian dipasangkan dengan kendali PID. Proses simulasi menggunakan software Matlab-Simulink.

4. Analisa Hasil Simulasi

Menganalisa hasil simulasi berupa respon kendali PID dan menganalisa pergerakan *end effector* pada lengan robot sesuai *trajectory* yang telah direncanakan.

### 3.2 Diagram Alir Penelitian

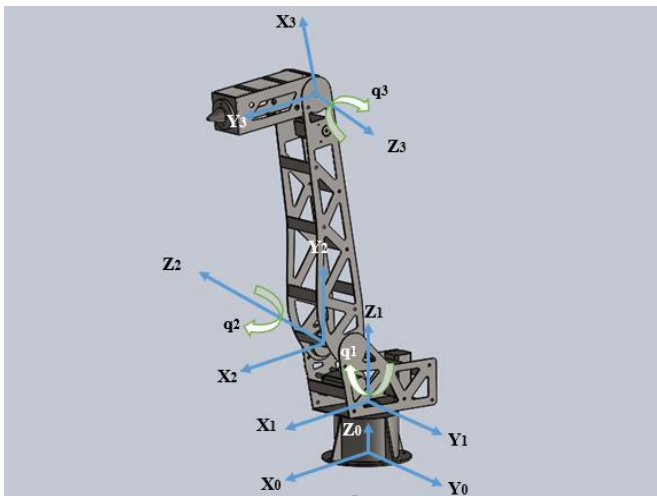
Penulisan penelitian ini dilakukan untuk membuat model dan menganalisis dinamika 3 DOF lengan robot. Dari proses analisa, simulasi hingga pembuatan alat dilakukan beberapa tahapan sebagai berikut:



Gambar 3.1 Diagram Alir Penelitian

### 3.3 Spesifikasi Model Lengan Robot

Proses desain model 3 Dimensi dan proses simulasi permodelan lengan robot yang akan dianalisa dilakukan menggunakan software Solidwork dan Matlab Simulink. Lengan robot yang dianalisa adalah lengan robot yang memiliki tiga derajat kebebasan dengan tiga sendi *revolute*. Gerakan pada sendi *revolute* yaitu gerakan yang hanya bisa berputar. Setiap sendi yang berputar diatur oleh motor DC. Gambar 3.1 menunjukkan model 3D lengan robot yang akan dianalisis.



Gambar 3.2 Gambar 3 Dimensi Lengan Robot 3 Derajat Kebebasan

Tabel 3.1 menunjukkan parameter-parameter lengan robot yang akan digunakan pada penelitian ini. Massa yang didapatkan, diambil dari data software Solidwork. Data tersebut nantinya akan digunakan pada pemodelan dan analisis dinamika lengan robot. Material yang digunakan adalah besi dengan kode ASTM A36 Steel. Material ini biasanya digunakan sebagai bahan konstruksi secara umum.

Tabel 3.1: Parameter Permodelan Lengan Robot

Komponen	Besaran (satuan)	Panjang (satuan)
Massa Alas	4,063 kg	115,3 mm
Massa Link 1	3,272 kg	177,53 mm
Massa Link 2	3,777 kg	700 mm
Massa End Effector	0,404 kg	310 mm

### 3.4 Parameter Motor DC

Dalam pembuatan model motor DC dibutuhkan sebuah validasi untuk menyesuaikan pemodelan sistem yang disimulasikan dengan realita saat motor DC bergerak. Namun kendala yang ditemui adalah memperoleh data-data spesifikasi/parameter motor secara lengkap. Produsen motor biasanya hanya memberikan informasi dalam bentuk grafik antara torsi dengan arus motor, torsi dengan tegangan, torsi dengan kecepatan dan beberapa informasi yang bersifat visual. Sehingga hampir tidak mungkin melakukan pemodelan motor secara ideal dalam desain sistem kontrol. Kecuali motor yang digunakan merupakan hasil desain sendiri. Akibatnya, pada kebanyakan proses perancangan banyak dilakukan asumsi-asumsi yang bersifat menyederhanakan model matematik. Berikut ini adalah parameter-parameter yang digunakan

Tabel 3.2: Parameter DC motor

Parameter	Simbol	Sendi 1	Sendi 2	Sendi 3	Satuan
Inersia motor	J	0.7	0.5	0.1	Kgm <sup>2</sup>
Koefisien redaman	B	0.1	0.1	0.1	N m/(rad/s)
Resistansi	R	3.6	3.6	3.6	Ohm
Induktansi	L	$12 \times 10^{-6}$	$12 \times 10^{-6}$	$12 \times 10^{-6}$	H
Kuat arus	i	1	1	1	A
Konstanta torsi	K <sub>m</sub>	0.1	0.1	0.1	N.m/A
Konstanta back EMF	K <sub>e</sub>	3.922	3.922	3.922	V.s/rad

### 3.5 Desain Kendali PID

Peneliti menggunakan kontrol PID untuk mempercepat respon gerakan dari referensi gerak yang diberikan. Kontrol ini adalah merupakan substansi yang paling penting dalam distribusi system control dan juga sering diintegrasikan dengan logika, urutan fungsi-fungsi, *selector*, dan blok fungsi yang sederhana yang digunakan untuk membuat sistem yang lebih kompleks.

Setiap aktuator akan dikontrol oleh kontrol PID. Pada penelitian ini akan menggunakan *Automated tuning strategy*. *Automated tuning* PID mampu menyesuaikan parameter *gain* sistem dinamik secara cepat dan tepat untuk mendapatkan desain yang *robust* dengan respon waktu yang ideal serta dapat digunakan pada *single* dan *multiloop* PID *tuning methods*. Model khusus urutan kerja dengan PID tuner dengan mengikuti aturan-aturan sebagai berikut:

- 1) Aktifkan PID *tuner* untuk menghitung model linier dari model simulink dan desain awal.
- 2) *Tune* kontrol pada PID tuner dengan mengatur waktu respon dan PID *tuner* menghitung parameter-parameter *gain* sedemikian rupa sehingga sistemnya menjadi *robust* dan seimbang.

- 3) *Export* parameter-parameter kontrol desain kembali ke kontrol PID dan simulasikan *performace control* pada Simulink.

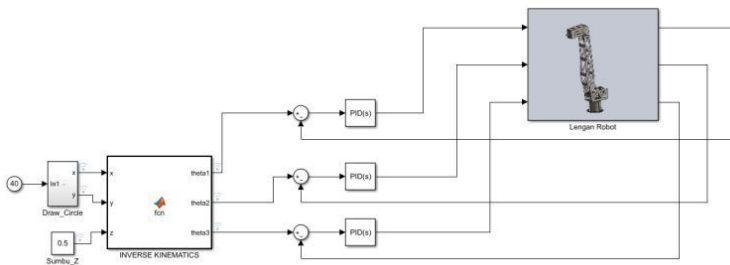


## BAB IV

### ANALISA DATA DAN PEMBAHASAN

#### 4.1 Deskripsi Umum

Lengan robot dengan 3 derajat kebebasan dimodelkan dalam bentuk 3 dimensi dalam software Soliworks. Pemodelan dimulai dengan pembuatan bagian-bagian dari lengan robot yang terdiri dari alas, link 1, link 2, dan *end effector*. Penggabungan bagian-bagian dari lengan robot selanjutnya diubah ke dalam bentuk blok diagram dalam SimMechanics/MATLAB. Dari blok diagram tersebut, kemudian disimulasikan ke dalam bentuk 3 dimensi pada software Simulink. Simulink adalah *graphical extension* MATLAB untuk memodelkan dan mensimulasikan sebuah sistem. Dalam Simulink, sistem digambarkan sebagai sebuah diagram blok, diantaranya *transfer function*, summing junction, didalamnya terdapat pula *virtual input* dan *output device* seperti *display* dan *oscilloscope*. Data dan informasi dari satu blok dikirim ke berbagai blok lainnya yang dihubungkan dengan garis.



Gambar 4.1: Pemodelan lengan robot menggunakan simulink

Dalam penelitian ini, input yang digunakan berasal *inverse kinematics*. *Inverse kinematics* mengubah besaran dari nilai posisi sumbu X, sumbu Y, dan sumbu Z dalam satuan meter menjadi sudut (*theta*) yang digunakan untuk memutar sendi 1, sendi 2, dan

sendi 3 dalam satuan radian. Variasi pada penelitian ini yaitu variasi *trajectory* yang terdiri dari *trajectory* lingkaran, setengah lingkaran dan bebas. Jari-jari dari *trajectory* lingkaran dan setengah lingkaran adalah 650 mm. Hasil yang didapatkan berupa nilai percepatan sudut ( $\text{rad/s}^2$ ), kecepatan sudut ( $\text{rad/s}$ ), dan torsi (Nm) pada masing-masing sendi lengan robot.

Sistem kontrol dan model motor DC dipasangkan pada simulasi lengan robot dalam bentuk diagram blok kemudian akan diperoleh nilai  $K_p$ ,  $K_d$ ,  $K_i$  yang sesuai dengan sistem pemodelan. Variasi tuning pada kontrol PID yang digunakan yaitu *automated tuning* PID dan Ziegler-Nichol. *Automated* PID tuning pada blok diagram adalah dengan menekan tombol *tune*. Kemudian program MATLAB akan mengidentifikasi *plant* yang dibuat dan memberikan parameter-parameter PID yang disesuaikan dengan *plant*. *Automated tuning* PID mampu menyesuaikan parameter *gain* sistem secara cepat untuk mendapatkan desain yang *robust* dengan respon waktu yang bagus, sehingga didapatkan nilai  $K_p$ ,  $K_i$ , dan  $K_d$  secara optimal. Kontrol PID akan mengontrol posisi, kecepatan sudut, dan percepatan sudut pada setiap sendi lengan robot. Tanggapan sistem dapat dilihat setelah sistem diberikan masukan sudut yang berbeda. Dari metode Ziegler-Nichol didapatkan nilai  $K_p$ ,  $K_i$ , dan  $K_d$  kemudian nilai tersebut dimasukkan ke dalam *plant*. Hasil pergerakan *end effector* dari *Automated tuning* dan Ziegler-Nichols kemudian dibandingkan, manakah salah satu dari metode tersebut yang menghasilkan *trajectory* yang presisi.

#### 4.2 Analisa Trajektori Setengah Lingkaran

Simulasi pada lengan robot yang dilakukan yaitu dengan membuat rumus dari setengah lingkaran terlebih dahulu. Rumus membuat setengah lingkaran ditulis dalam diagram blok matlab function pada simulink. Matlab *function* yang berisi rumus setengah lingkaran kemudian disambungkan dengan diagram blok *inverse kinematics*. Adapun rumus yang digunakan untuk membuat setengah lingkaran adalah sebagai berikut:

$$x = 0,65 \cos (\pi)$$

$$y = 0,65 \sin (\pi)$$

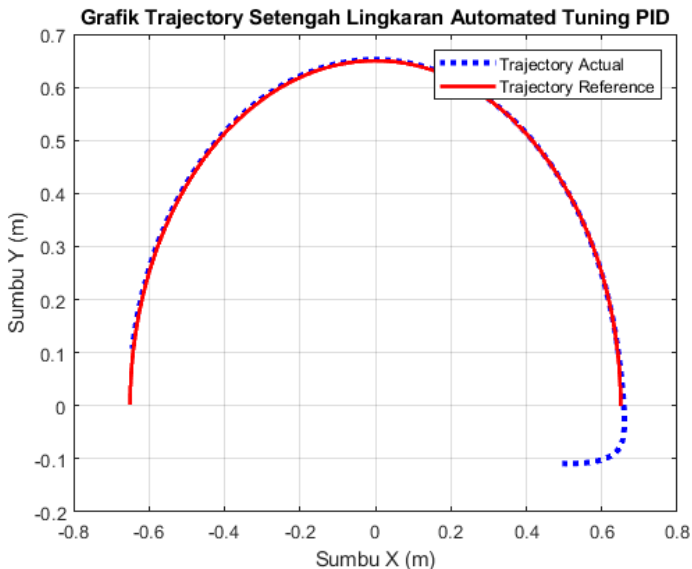
Selanjutnya, diagram blok *inverse kinematics* disambungkan dengan diagram blok motor DC dan lengan robot. Jari-jari yang ditentukan dalam simulasi ini adalah 0.65 meter. Lama proses simulasi dilakukan dalam waktu 10 detik. Dalam simulasi ini, dilakukan dua variasi *tuning* pada kontrol PID yaitu menggunakan *automated tuning* dan metode ziegler-nichols.

#### 4.2.1 Kontrol PID menggunakan *Automated Tuning* PID

*Automated* PID *tuning* pada blok diagram adalah dengan menekan tombol *tune*. Kemudian program MATLAB akan mengidentifikasi plant yang dibuat dan memberikan parameter-parameter PID yang disesuaikan dengan *plant*. *Automated* PID *tuning* mampu menyesuaikan parameter *gain* sistem secara cepat untuk mendapatkan desain yang *robust* dengan respon waktu yang bagus, sehingga didapatkan nilai Kp, Ki, dan Kd secara optimal. Hasil yang didapatkan dalam *automated tuning* PID sebagai berikut

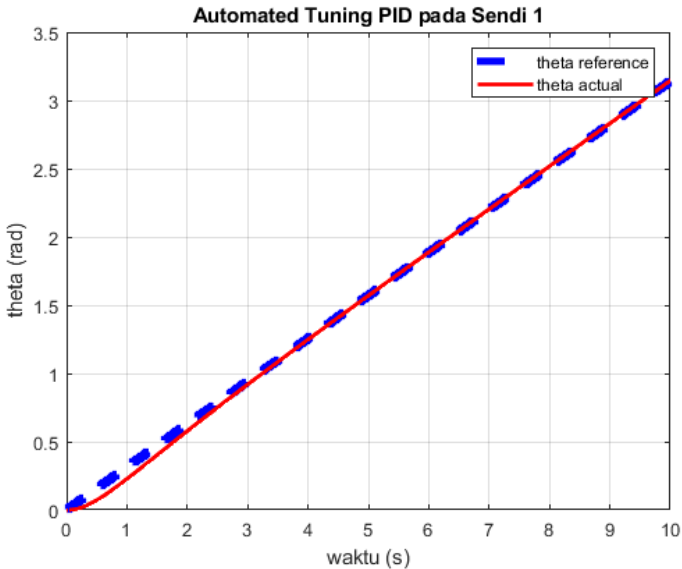
Tabel 4.1: Tabel Tuning PID pada *Trajectory* Setengah Lingkaran dengan *Automated Tuning*

	Nilai Kp	Nilai Ki	Nilai Kd
Sendi 1	111.4693	61.4583	7.0479
Sendi 2	106.8937	56.5723	0.5507
Sendi 3	141.3573	97.9261	1.7032



Gambar 4.2. Grafik *Trajectory* Setengah Lingkaran menggunakan *Automated Tuning PID*

Hasil yang didapatkan berupa grafik yang ditunjukkan pada gambar 4.2. Pada grafik tersebut menunjukkan gerakan dari *end effector* dan referensi *trajectory* yang diinginkan. Pada garis warna biru menunjukkan gerakan dari *end effector*, *end effector* bergerak dari kanan ke kiri. Titik awal pergerakan robot berada pada sumbu x sebesar 0,5 meter dan pada sumbu y sebesar -0,1 meter. Sedangkan garis warna merah menunjukkan referensi *trajectory* yang diinginkan. Pada grafik tersebut menunjukkan bahwa lengan robot bergerak mengikuti *trajectory* dengan presisi.



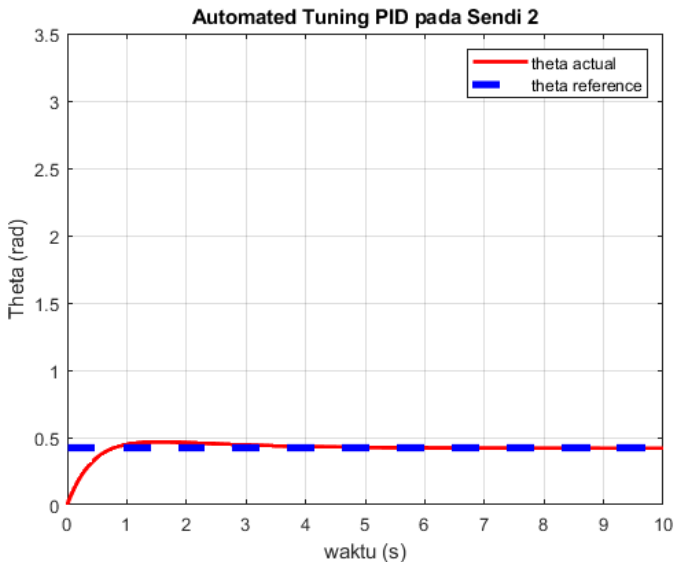
Gambar 4.3 Grafik Sendi 1 Menggunakan *Automated Tuning* PID pada Setengah Lingkaran

Hasil simulasi *Automated tuning* PID pada sendi 1 ditunjukkan pada gambar 4.3. Pada grafik tersebut menunjukkan sudut yang ditempuh oleh sendi 1. Sudut yang dicapai mulai dari nol sampai 3,14 radian atau setara dengan 180 derajat selama 10 detik. Grafik warna biru menunjukkan referensi sudut yang harus dicapai oleh sendi. Grafik warna merah menunjukkan sudut aktual yang digerakkan oleh sendi 1.

Gerakan aktual sendi 1 bergerak dari 0 radian ke 3,14 radian selama 10 detik. Gerakan aktual pada sendi 1 mengikuti sudut referensi pada detik ke-3 dengan stabil tanpa mengalami osilasi.

Gerakan sudut pada sendi 1 dipengaruhi oleh kontrol PID. Semakin baik respon yang diberikan PID maka semakin cepat dan presisi sendi 1 mengikuti sudut referensi yang diinginkan. Dari

nilai  $K_p$ ,  $K_i$ , dan  $K_d$  tersebut, didapatkan nilai *settling time* = 4.45 detik, *rise time* = 0.578 detik, persen *overshoot* = 10.2%.



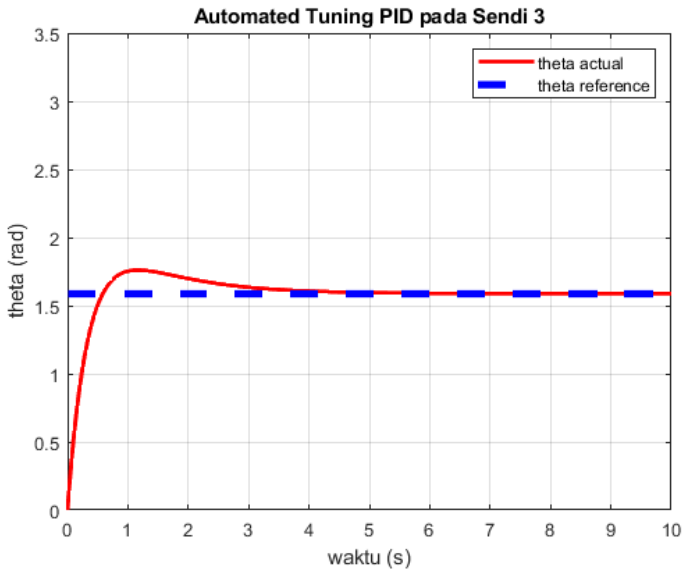
Gambar 4.4 Grafik sendi 2 Menggunakan *Automated Tuning* PID pada Setengah Lingkaran

Hasil simulasi *Automated tuning* PID pada sendi 2 ditunjukkan pada gambar 4.4. Pada grafik tersebut menunjukkan sudut yang ditempuh oleh sendi 2. Sudut yang harus dicapai adalah 0,42 radian selama 10 detik. Grafik warna biru menunjukkan referensi sudut yang harus dicapai oleh sendi. Grafik warna merah menunjukkan sudut aktual yang digerakkan oleh sendi 2.

Gerakan aktual sendi 2 bergerak dari 0 radian ke 0,48 radian pada detik ke-0,75. Gerakan selanjutnya mengalami penurunan hingga stabil pada nilai sudut yang harus dicapai sebesar 0,42 radian. Gerakan aktual pada sendi 2 mengikuti sudut referensi pada detik ke-3 tanpa mengalami osilasi.

Gerakan sudut pada sendi 2 dipengaruhi oleh kontrol PID. Semakin baik respon yang diberikan PID maka semakin cepat dan

presisi sendi 2 mengikuti sudut referensi yang diinginkan. Dari nilai  $K_p$ ,  $K_i$ , dan  $K_d$  tersebut, didapatkan nilai *settling time* = 4.54 detik, *rise time* = 0.553 detik, persen *overshoot* = 11%.



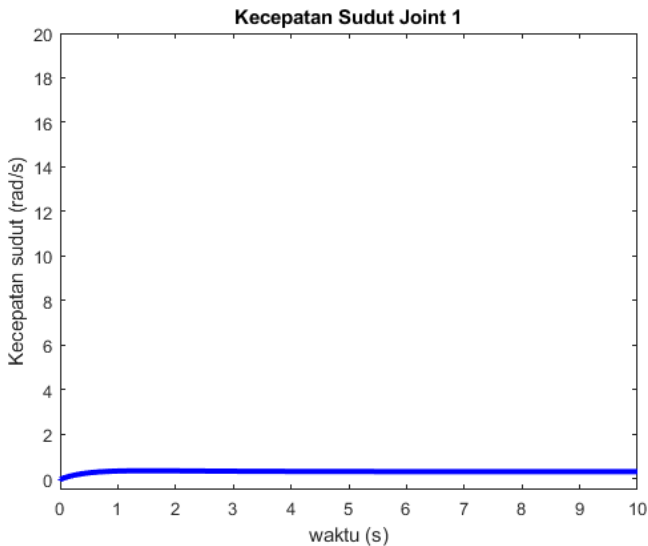
Gambar 4.5 Gerakan sendi 3 Menggunakan *Automated Tuning* PID pada Setengah Lingkaran

Hasil simulasi *Automated tuning* PID pada sendi 3 ditunjukkan pada gambar 4.5. Pada grafik tersebut menunjukkan sudut yang ditempuh oleh sendi 3. Sudut yang harus dicapai adalah 1,6 radian selama 10 detik. Grafik warna biru menunjukkan referensi sudut yang harus dicapai oleh sendi. Grafik warna merah menunjukkan sudut aktual yang digerakkan oleh sendi 3.

Gerakan aktual sendi 3 bergerak dari 0 radian ke 1,75 radian pada detik ke-0,5. Gerakan selanjutnya mengalami penurunan hingga stabil pada nilai sudut yang harus dicapai sebesar 1,6 radian. Gerakan aktual pada sendi 3 mengikuti sudut referensi pada detik ke-1 tanpa mengalami osilasi.

Gerakan sudut pada sendi 3 dipengaruhi oleh kontrol PID. Semakin baik respon yang diberikan PID maka semakin cepat dan presisi sendi 3 mengikuti sudut referensi yang diinginkan. Dari nilai  $K_p$ ,  $K_i$ , dan  $K_d$  tersebut, didapatkan nilai *settling time* = 3.46 detik, *rise time* = 0.432 detik, persen *overshoot* = 10.8%.

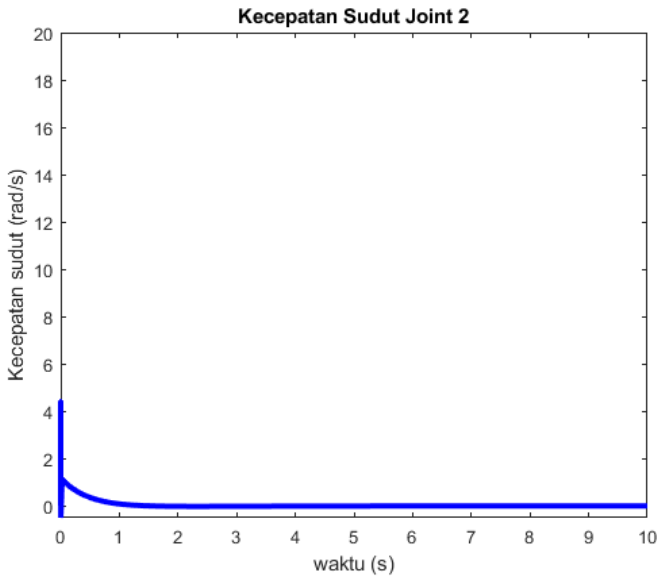
#### 4.2.1.1 Kecepatan Sudut pada tiap Sendi



Gambar 4.6 Kecepatan sudut sendi 1 Menggunakan *Automated Tuning* PID pada Setengah Lingkaran

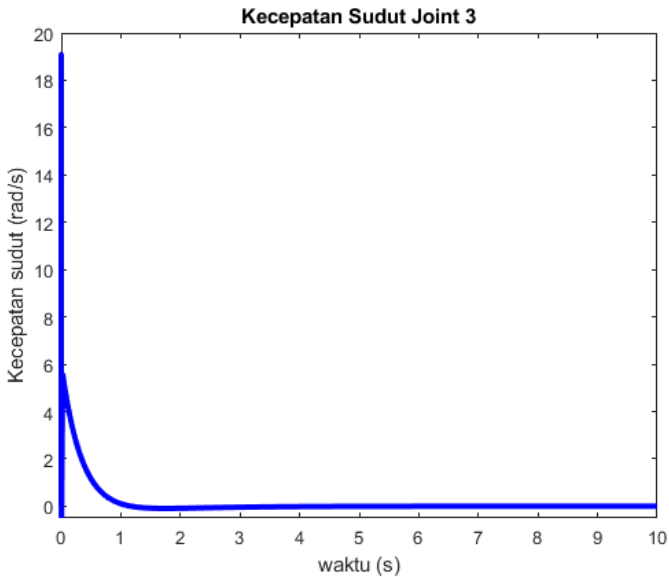
Hasil simulasi kecepatan sudut *Automated tuning* PID pada sendi 1 ditunjukkan pada gambar 4.6. Pada grafik tersebut menunjukkan kecepatan sudut yang ditempuh oleh sendi 1. Hasil simulasi pada kecepatan sudut didapatkan dari turunan sudut aktual sendi 1 terhadap waktu. Pada detik ke-0, kecepatan sudut turun kemudian naik mencapai 0,35 rad/s. Terjadi penurunan lagi pada detik ke-2 mencapai 0,32 rad/s. Kecepatan sudut stabil pada 0,32 rad/s pada detik ke-5 sampai detik ke-10.





Gambar 4.7 Kecepatan sudut sendi 2 Menggunakan *Automated Tuning* PID pada Setengah Lingkaran

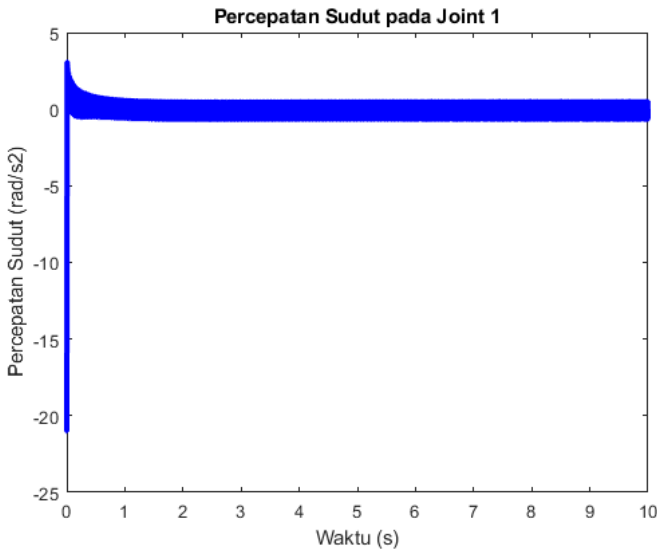
Hasil simulasi kecepatan sudut *Automated tuning* PID pada sendi 2 ditunjukkan pada gambar 4.7. Pada grafik tersebut menunjukkan kecepatan sudut yang ditempuh oleh sendi 2. Hasil simulasi pada kecepatan sudut didapatkan dari turunan sudut aktual sendi 2 terhadap waktu. Pada detik ke-0, mengalami kenaikan hingga mencapai 4,8 rad/s kemudian turun hingga mencapai 1 rad/s. Pada detik ke-1 mengalami pengurangan kecepatan sudut dari 1 rad/s hingga berhenti.



Gambar 4.8 Kecepatan sudut sendi 3 Menggunakan *Automated Tuning* PID pada Setengah Lingkaran

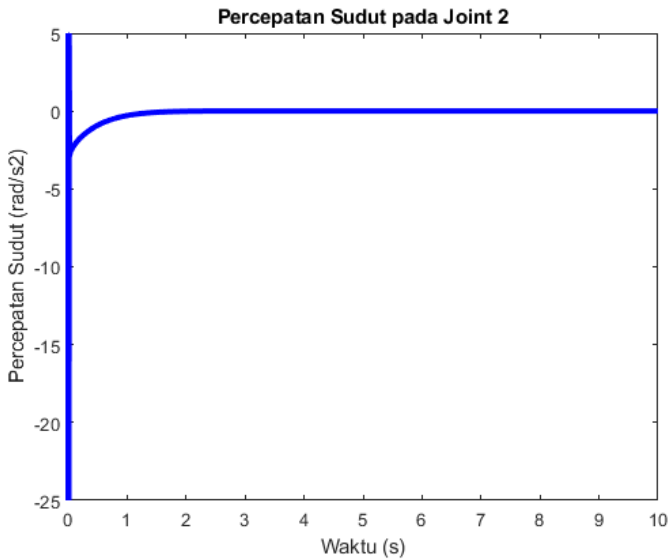
Hasil simulasi kecepatan sudut *Automated tuning* PID pada sendi 3 ditunjukkan pada gambar 4.8. Pada grafik tersebut menunjukkan kecepatan sudut yang ditempuh oleh sendi 3. Hasil simulasi pada kecepatan sudut didapatkan dari turunan sudut aktual sendi 3 terhadap waktu. Pada detik ke-0, mengalami penurunan hingga  $-2$  rad/s, kemudian mengalami kenaikan hingga mencapai  $17$  rad/s lalu turun lagi hingga mencapai  $5$  rad/s. Pada detik ke-1 mengalami pengurangan kecepatan sudut dari  $5$  rad/s hingga berhenti.

#### 4.2.1.2 Percepatan Sudut pada tiap Sendi



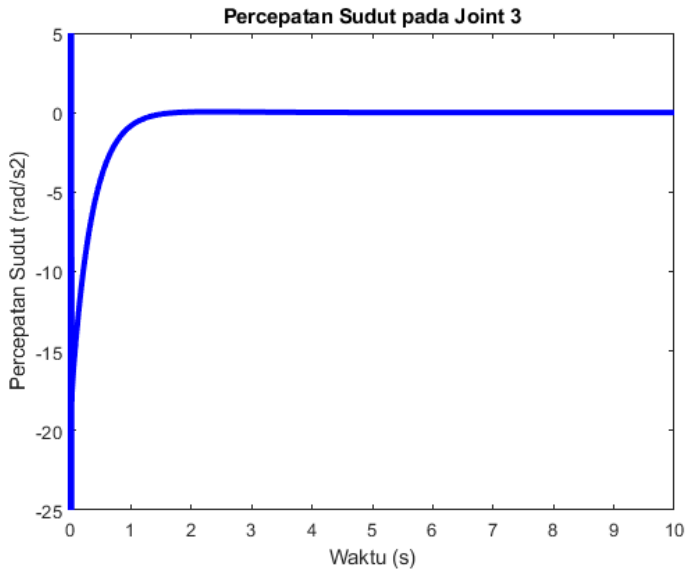
Gambar 4.9 Percepatan Sudut Sendi 1 menggunakan *Automated Tuning* PID pada Setengah Lingkaran

Hasil simulasi percepatan sudut *Automated tuning* PID pada sendi 1 ditunjukkan pada gambar 4.9. Pada grafik tersebut menunjukkan percepatan sudut yang ditempuh oleh sendi 1. Hasil simulasi pada percepatan sudut didapatkan dari turunan kecepatan sudut sendi 1 terhadap waktu. Pada detik ke-0, mengalami penurunan hingga  $-21 \text{ rad/s}^2$ , kemudian mengalami kenaikan hingga mencapai  $3 \text{ rad/s}^2$  lalu turun lagi hingga mencapai  $3 \text{ rad/s}^2$ . Pada detik ke-1 mengalami pengurangan kecepatan sudut dari  $3 \text{ rad/s}^2$  hingga mencapai  $0 \text{ rad/s}^2$  yang stabil selama 10 detik.



Gambar 4.10 Percepatan Sudut Sendi 2 menggunakan *Automated Tuning* PID pada Setengah Lingkaran

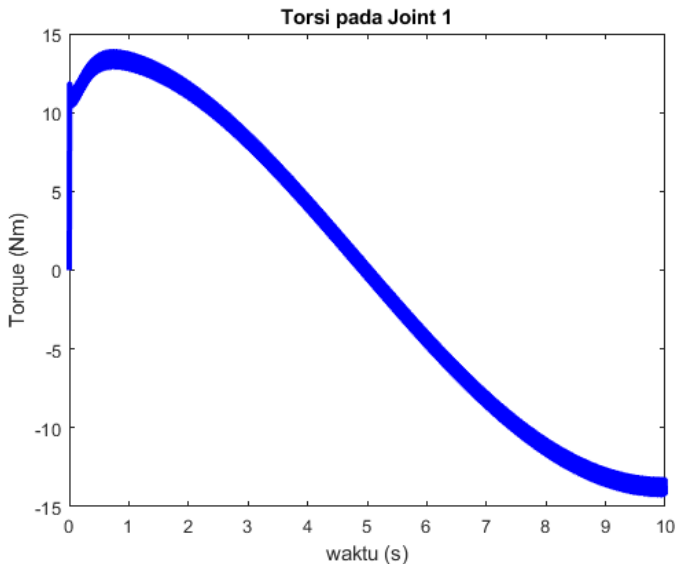
Hasil simulasi percepatan sudut *Automated tuning* PID pada sendi 2 ditunjukkan pada gambar 4.10. Pada grafik tersebut menunjukkan percepatan sudut yang ditempuh oleh sendi 2. Hasil simulasi pada percepatan sudut didapatkan dari turunan kecepatan sudut sendi 2 terhadap waktu. Pada detik ke-0, mengalami penurunan hingga  $-3 \text{ rad/s}^2$ , kemudian mengalami kenaikan hingga mencapai  $0 \text{ rad/s}^2$  yang stabil hingga detik ke-10.



Gambar 4.11 Percepatan Sudut Sendi 3 menggunakan *Automated Tuning* PID pada Setengah Lingkaran

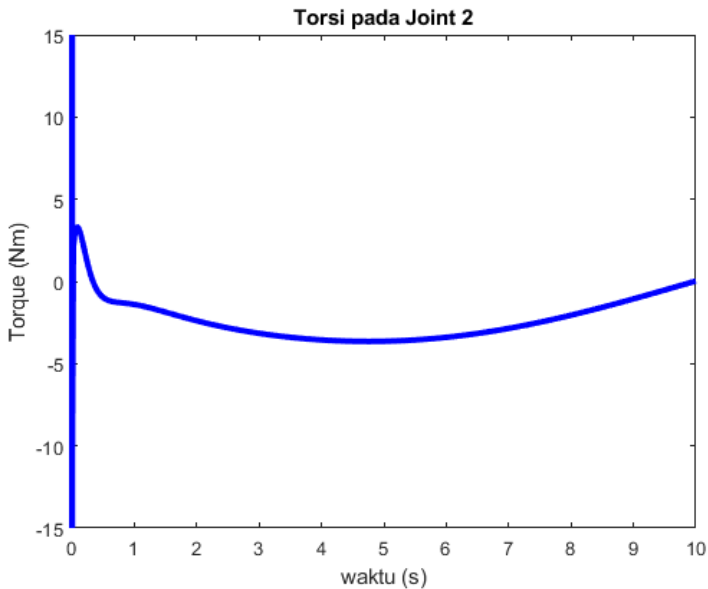
Hasil simulasi percepatan sudut *Automated tuning* PID pada sendi 3 ditunjukkan pada gambar 4.11. Pada grafik tersebut menunjukkan percepatan sudut yang ditempuh oleh sendi 3. Hasil simulasi pada percepatan sudut didapatkan dari turunan kecepatan sudut sendi 3 terhadap waktu. Pada detik ke-0, mengalami penurunan hingga  $-15 \text{ rad/s}^2$ , kemudian mengalami kenaikan hingga mencapai  $0 \text{ rad/s}^2$  yang stabil hingga detik ke-10.

### 4.2.1.3 Torsi pada tiap Sendi



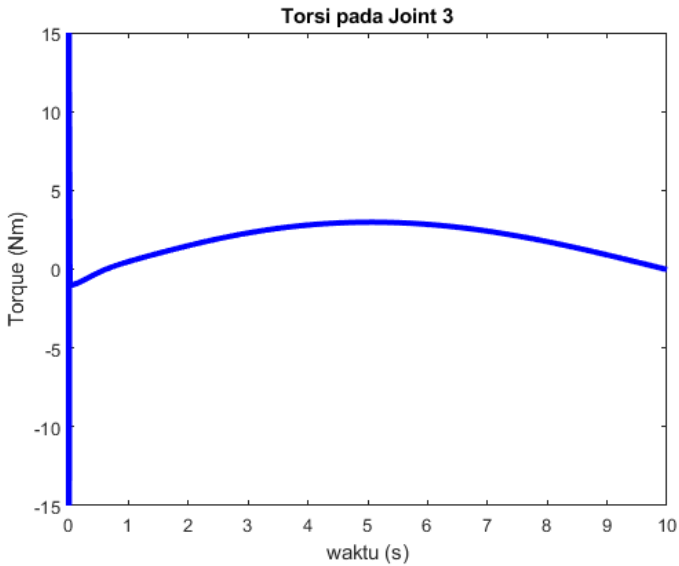
Gambar 4.12 Torsi Sendi 1 menggunakan *Automated Tuning* PID pada Setengah Lingkaran

Hasil simulasi torsi *Automated tuning* PID pada sendi 1 ditunjukkan pada gambar 4.12. Pada grafik tersebut menunjukkan torsi yang ditempuh oleh sendi 1. Hasil simulasi besarnya torsi dipengaruhi sudut, kecepatan sudut, dan percepatan sudut pada sendi 1. Pada detik ke-0, mengalami kenaikan hingga 12 Nm. Saat menuju detik ke-1, mengalami kenaikan lagi hingga mencapai 14,2 Nm. Setelah detik ke-1, mengalami penurunan hingga -14 Nm sampai detik ke-10.



Gambar 4.13 Torsi Sendi 2 menggunakan *Automated Tuning* PID pada Setengah Lingkaran

Hasil simulasi torsi *Automated tuning* PID pada sendi 2 ditunjukkan pada gambar 4.13. Pada grafik tersebut menunjukkan torsi yang ditempuh oleh sendi 2. Hasil simulasi besarnya torsi dipengaruhi sudut, kecepatan sudut, dan percepatan sudut pada sendi 2. Pada detik ke-0, mengalami kenaikan hingga 3,3 Nm. Pada detik ke-5 mengalami penurunan hingga -3,6 Nm, kemudian naik mencapai 0 Nm pada detik ke-10.



Gambar 4.14 Torsi Sendi 3 menggunakan *Automated Tuning* PID pada Setengah Lingkaran

Hasil simulasi torsi *Automated tuning* PID pada sendi 3 ditunjukkan pada gambar 4.14. Pada grafik tersebut menunjukkan torsi yang ditempuh oleh sendi 3. Hasil simulasi besarnya torsi dipengaruhi sudut, kecepatan sudut, dan percepatan sudut pada sendi 3. Pada detik ke-0, mengalami penurunan pada -1 Nm. Pada detik ke-5 mengalami kenaikan hingga 3 Nm, kemudian turun mencapai 0 Nm pada detik ke-10.



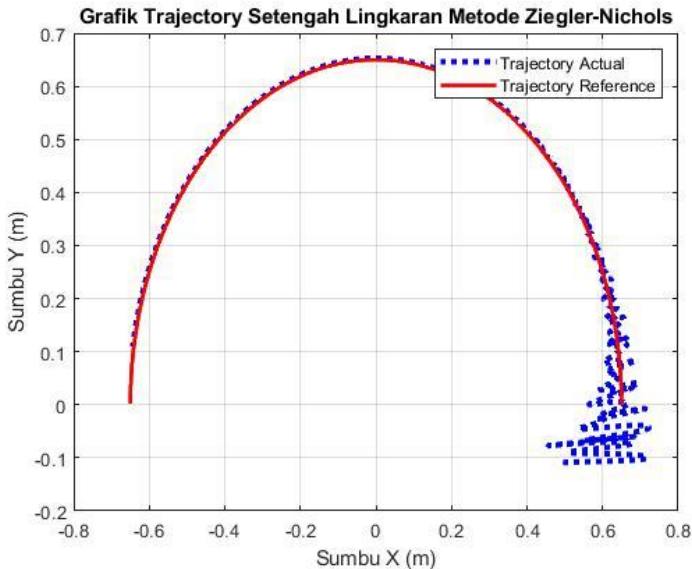
#### 4.2.2 Kontrol PID menggunakan Metode Ziegler-Nichols

Simulasi pergerakan lengan robot dengan kontrol PID dilakukan dengan metode ziegler-nichol. Metode ini digunakan untuk mendapatkan parameter-parameter seperti  $K_p$ ,  $K_i$ , dan  $K_d$ . Parameter tersebut selanjutnya dilakukan tuning pada diagram blok PID. Diagram blok PID kemudian disambungkan dengan motor DC dan lengan robot. Berikut parameter-parameter yang didapatkan saat simulasi menggunakan MATLAB.

Tabel 4.2: Tabel Tuning PID Metode Ziegler-Nichols Setengah Lingkaran

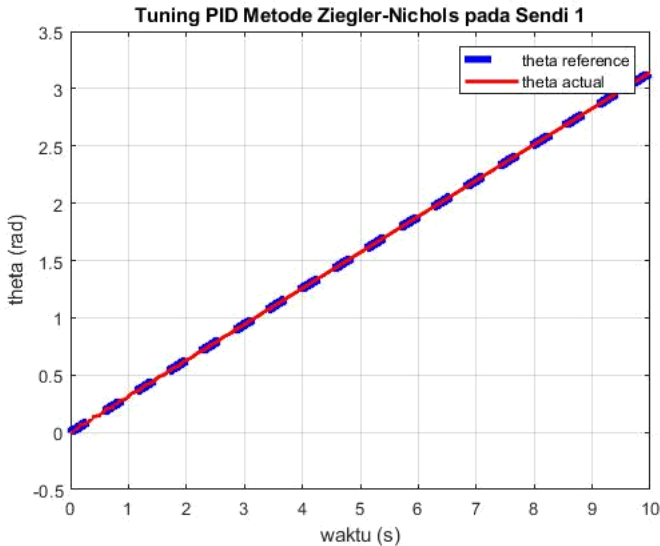
	Nilai $K_p$	Nilai $K_i$	Nilai $K_d$
Sendi 1	140.991	28795.2754	0.17258
Sendi 2	1014.2902	220727.3244	1.1652
Sendi 3	190.5715	37687.3572	0.24091

Parameter - parameter tersebut mempengaruhi tingkat kepresisian dari pergerakan sendi pada lengan robot terhadap trajectory yang diinginkan. Apabila parameter tersebut dimasukkan ke dalam blok diagram PID pada simulink, maka didapatkan grafik pergerakan end effector. Grafik pergerakan end effector ditunjukkan pada gambar 4.15.



Gambar 4.15: Grafik *Trajectory* Setengah Lingkaran menggunakan Metode Ziegler-Nichols

Grafik 4.15 menunjukkan gerakan dari *end effector* aktual dan referensi *trajectory* yang diinginkan. Garis warna biru menunjukkan gerakan aktual dari *end effector*, *end effector* bergerak dari kanan ke kiri. Titik awal pergerakan robot berada pada sumbu x sebesar 0,5 meter dan pada sumbu y sebesar -0,1 meter. Pergerakan lengan robot mengalami osilasi di awal, tetapi gerakan lengan robot bisa stabil dan mengikuti referensi *trajectory*. Garis warna merah menunjukkan referensi *trajectory* yang diinginkan. Pada grafik tersebut menunjukkan bahwa lengan robot bergerak mengikuti *trajectory*, namun tingkat kepresisiannya kurang baik diawal simulasi karena terjadi osilasi.

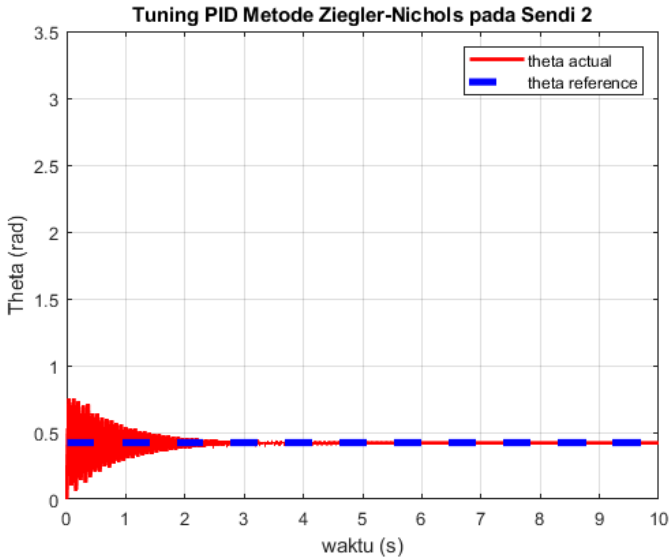


Gambar 4.16: Grafik Sendi 1 menggunakan Metode Ziegler-Nichols pada Setengah Lingkaran

Hasil simulasi tuning PID menggunakan metode Ziegler-Nichols pada sendi 1 ditunjukkan pada gambar 4.16. Pada grafik tersebut menunjukkan sudut yang ditempuh oleh sendi 1. Sudut yang dicapai mulai dari nol sampai 3.14 radian atau setara dengan 180 derajat selama 10 detik. Grafik warna biru menunjukkan referensi sudut yang harus dicapai oleh sendi. Grafik warna merah menunjukkan sudut aktual yang digerakkan oleh sendi 1.

Gerakan sudut pada sendi 1 dipengaruhi oleh tuning pada kontrol PID. Semakin baik respon yang diberikan PID maka semakin cepat dan presisi sendi 1 mengikuti sudut referensi yang diinginkan. Dari nilai  $K_p$ ,  $K_i$ , dan  $K_d$  tersebut, didapatkan nilai *settling time* = 3.85 detik, *rise time* = 0.0367 detik, persen *overshoot* = 89.8%. Besarnya *overshoot* mempengaruhi *steady state error* hingga menyebabkan respon tidak cepat stabil dan beresilasi. *Overshoot* yang besar disebabkan oleh nilai *gain*

integral ( $K_i$ ) yang tidak sesuai dengan *plant* sistem. Pada kasus Ziegler-Nichols tersebut, nilai  $K_i$  yang diberikan terlalu besar sehingga menyebabkan besarnya persen *overshoot*.

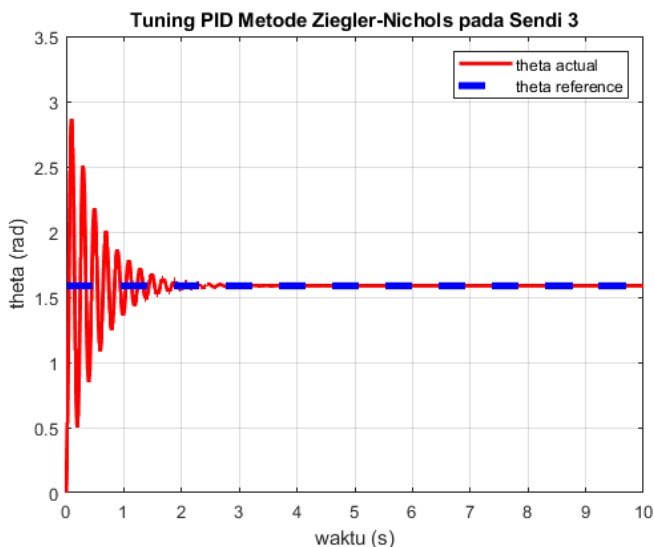


Gambar 4.17: Grafik Sendi 2 menggunakan Metode Ziegler-Nichols pada Setengah Lingkaran

Hasil simulasi tuning PID menggunakan metode ziegler-nichols pada sendi 2 ditunjukkan pada gambar 4.17. Pada grafik tersebut menunjukkan sudut yang ditempuh oleh sendi 2. Sudut yang harus dicapai adalah 0.42 radian selama 10 detik. Grafik warna biru menunjukkan referensi sudut yang harus dicapai oleh sendi. Grafik warna merah menunjukkan sudut aktual yang digerakkan oleh sendi 2.

Gerakan sudut pada sendi 2 dipengaruhi oleh tuning pada kontrol PID. Gerakan pada sendi 2 mengalami osilasi yang cukup besar dan rapat, osilasi ini terjadi di awal simulasi. Osilasi tersebut kembali stabil pada detik ke-7 dan selanjutnya bergerak mengikuti

referensi *trajectory*. Dari nilai  $K_p$ ,  $K_i$ , dan  $K_d$  menggunakan metode ziegler-nichols, didapatkan nilai *settling time* = 0.761 detik, *rise time* = 0.0126 detik, persen *overshoot* = 85.7%. Besarnya *overshoot* mempengaruhi *steady state error* hingga menyebabkan respon tidak cepat stabil dan berosilasi. *Overshoot* yang besar disebabkan oleh nilai *gain* integral ( $K_i$ ) yang tidak sesuai dengan *plant* sistem. Pada kasus Ziegler-Nichols tersebut, nilai  $K_i$  yang diberikan terlalu besar sehingga menyebabkan besarnya persen *overshoot*.



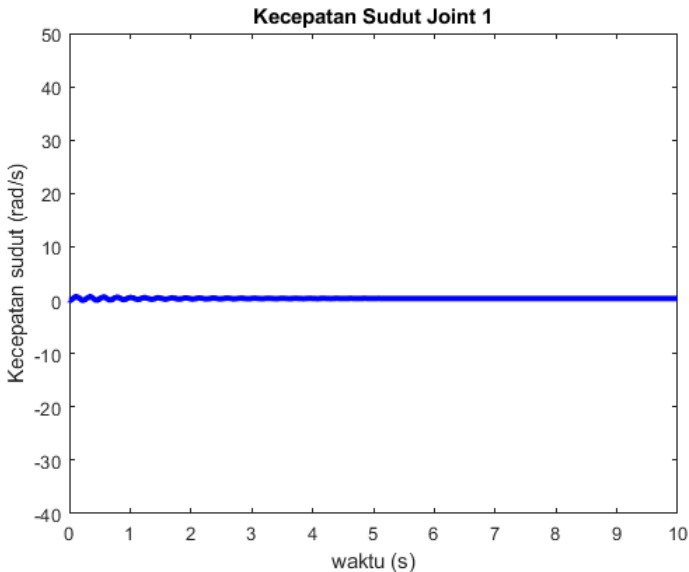
Gambar 4.18: Grafik Sendi 3 menggunakan Metode Ziegler-Nichols pada Setengah Lingkaran

Hasil simulasi tuning PID menggunakan metode ziegler-nichols pada sendi 3 ditunjukkan pada gambar 4.18 Pada grafik tersebut menunjukkan sudut yang ditempuh oleh sendi 3. Sudut yang harus dicapai adalah 1.6 radian selama 10 detik. Grafik warna biru menunjukkan referensi sudut yang harus dicapai oleh sendi.

Grafik warna merah menunjukkan sudut aktual yang digerakkan oleh sendi 3.

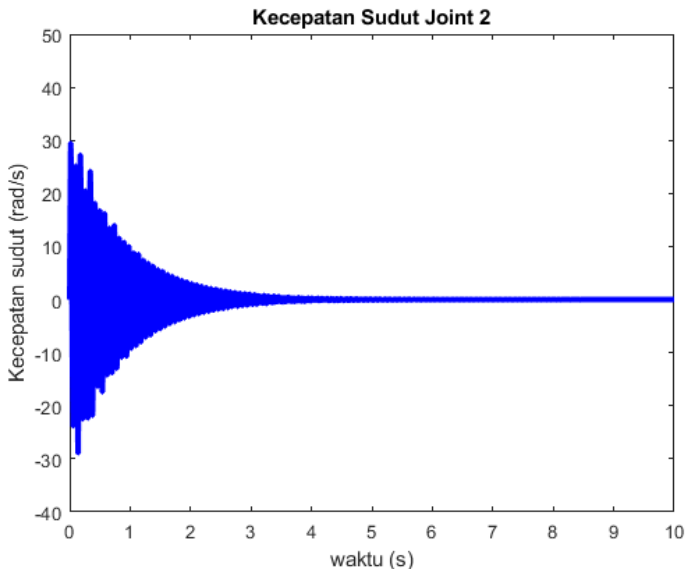
Gerakan sudut pada sendi 3 dipengaruhi oleh tuning pada kontrol PID. Gerakan pada sendi 3 mengalami osilasi yang cukup besar dan tidak rapat, osilasi ini terjadi di awal simulasi. Osilasi tersebut kembali stabil pada detik ke-3 dan selanjutnya bergerak mengikuti referensi *trajectory*. Dari nilai  $K_p$ ,  $K_i$ , dan  $K_d$  menggunakan metode ziegler-nichols, didapatkan nilai *settling time* = 1.99 detik, *rise time* = 0.0332 detik, persen *overshoot* = 83%. Besarnya *overshoot* mempengaruhi *steady state error* hingga menyebabkan respon tidak cepat stabil dan berosilasi. *Overshoot* yang besar disebabkan oleh nilai *gain* integral ( $K_i$ ) yang tidak sesuai dengan *plant* sistem. Pada kasus Ziegler-Nichols tersebut, nilai  $K_i$  yang diberikan terlalu besar sehingga menyebabkan besarnya persen *overshoot*.

#### 4.2.2.1 Kecepatan Sudut pada tiap Sendi



Gambar 4.19 Kecepatan sudut sendi 1 Metode Ziegler-Nichols pada Setengah Lingkaran

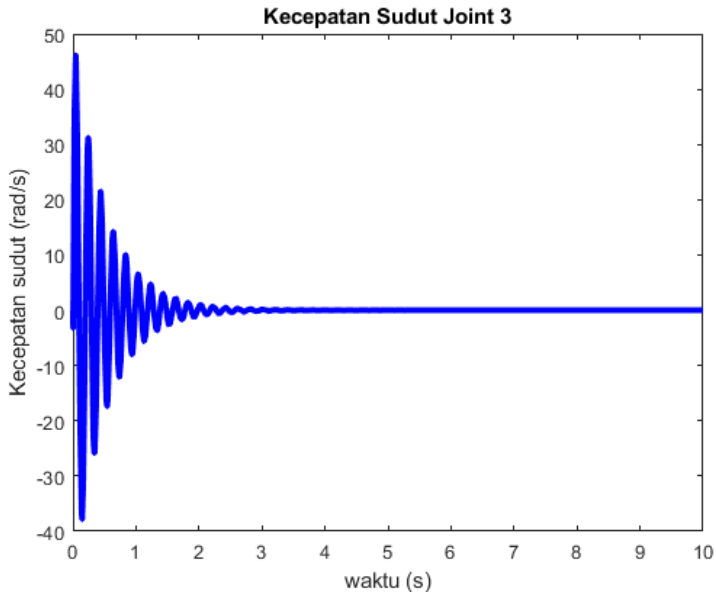
Hasil simulasi kecepatan sudut Metode Ziegler-Nichols pada sendi 1 ditunjukkan pada gambar 4.19. Pada grafik tersebut menunjukkan kecepatan sudut yang ditempuh oleh sendi 1. Hasil simulasi pada kecepatan sudut didapatkan dari turunan sudut aktual sendi 1 terhadap waktu. Pada detik ke-0 hingga detik ke-3 terjadi osilasi. Kecepatan sudut stabil pada 0 rad/s pada detik ke-3 sampai detik ke-10.



Gambar 4.20 Kecepatan sudut sendi 2 Metode Ziegler-Nichols pada Setengah Lingkaran

Hasil simulasi kecepatan sudut Metode Ziegler-Nichols pada sendi 2 ditunjukkan pada gambar 4.20. Pada grafik tersebut menunjukkan kecepatan sudut yang ditempuh oleh sendi 2. Hasil simulasi pada kecepatan sudut didapatkan dari turunan sudut aktual sendi 2 terhadap waktu. Pada detik ke-0 hingga detik ke-5 terjadi osilasi. Kecepatan sudut stabil pada 0 rad/s pada detik ke-5 sampai detik ke-10.

osilasi. Osilasi terjadi pada antara 30 rad/s dan -30 rad/s. Kecepatan sudut stabil pada 0 rad/s pada detik ke-5 sampai detik ke-10.

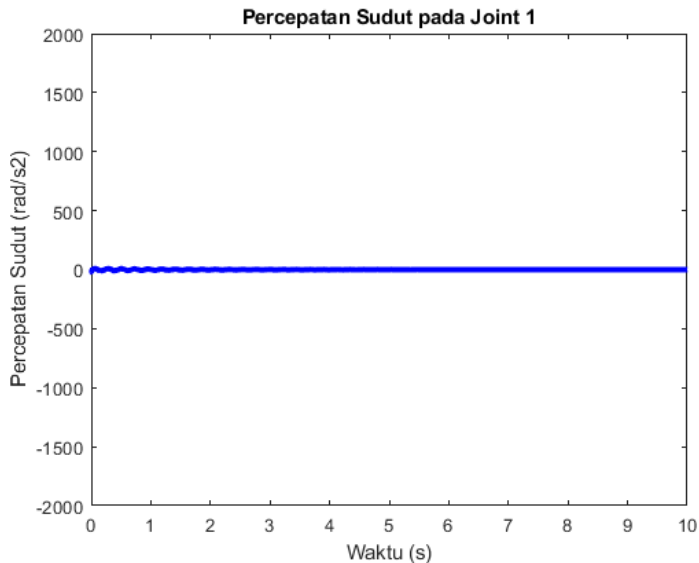


Gambar 4.21 Kecepatan sudut sendi 3 Metode Ziegler-Nichols pada Setengah Lingkaran

Hasil simulasi kecepatan sudut Metode Ziegler-Nichols pada sendi 3 ditunjukkan pada gambar 4.21. Pada grafik tersebut menunjukkan kecepatan sudut yang ditempuh oleh sendi 3. Hasil simulasi pada kecepatan sudut didapatkan dari turunan sudut aktual sendi 3 terhadap waktu. Pada detik ke-0 hingga detik ke-3 terjadi osilasi. Osilasi terjadi antara 48 rad/s dan -39 rad/s. Kecepatan sudut stabil pada 0 rad/s pada detik ke-3 sampai detik ke-10.

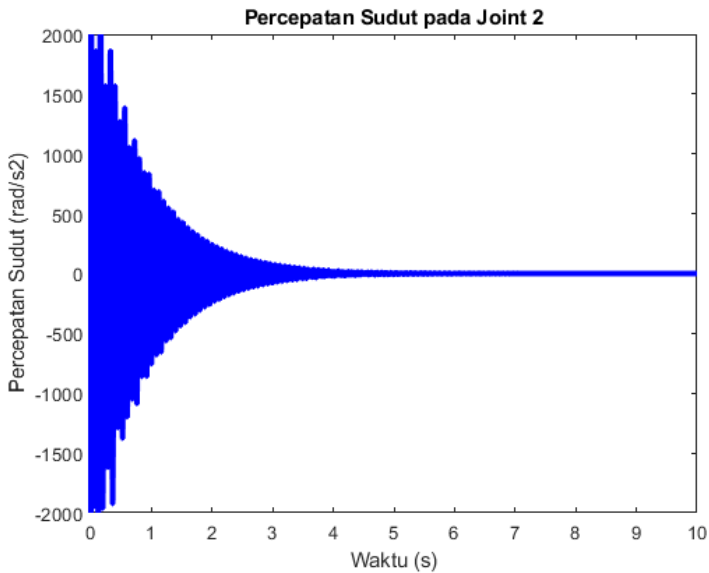


#### 4.2.2.2 Percepatan Sudut pada tiap Sendi



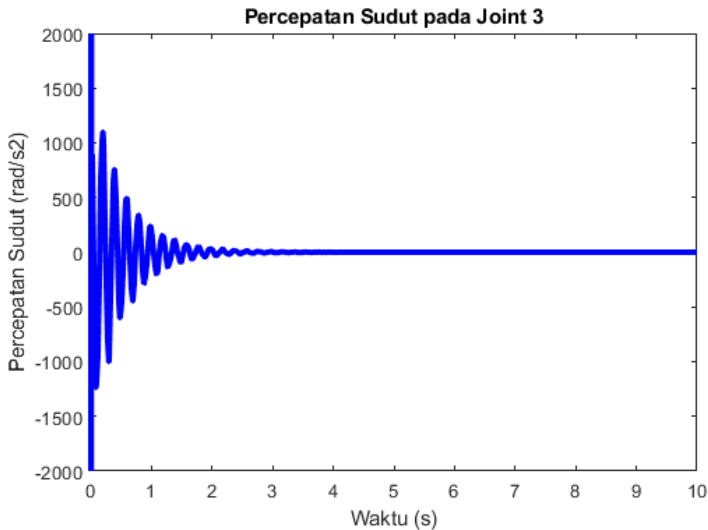
Gambar 4.22 Percepatan sudut sendi 1 Metode Ziegler-Nichols pada Setengah Lingkaran

Hasil simulasi percepatan sudut Metode Ziegler-Nichols pada sendi 1 ditunjukkan pada gambar 4.22. Pada grafik tersebut menunjukkan percepatan sudut yang ditempuh oleh sendi 1. Hasil simulasi pada percepatan sudut didapatkan dari turunan kecepatan sudut sendi 1 terhadap waktu. Pada detik ke-0 hingga detik ke-3 terjadi osilasi. Osilasi yang terjadi sangat kecil. Percepatan sudut stabil pada  $0 \text{ rad/s}^2$  pada detik ke-3 sampai detik ke-10.



Gambar 4.23 Percepatan sudut sendi 2 Metode Ziegler-Nichols pada Setengah Lingkaran

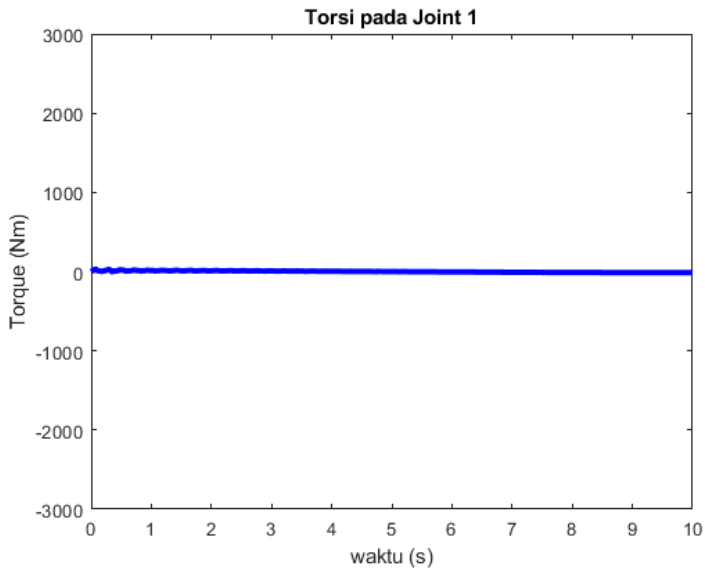
Hasil simulasi percepatan sudut Metode Ziegler-Nichols pada sendi 2 ditunjukkan pada gambar 4.23. Pada grafik tersebut menunjukkan percepatan sudut yang ditempuh oleh sendi 2. Hasil simulasi pada percepatan sudut didapatkan dari turunan kecepatan sudut sendi 2 terhadap waktu. Pada detik ke-0 hingga detik ke-5 terjadi osilasi. Osilasi terjadi antara  $2000 \text{ rad/s}^2$  hingga  $-2000 \text{ rad/s}^2$ . Percepatan sudut stabil pada  $0 \text{ rad/s}^2$  pada detik ke-5 sampai detik ke-10.



Gambar 4.24 Percepatan sudut sendi 3 Metode Ziegler-Nichols pada Setengah Lingkaran

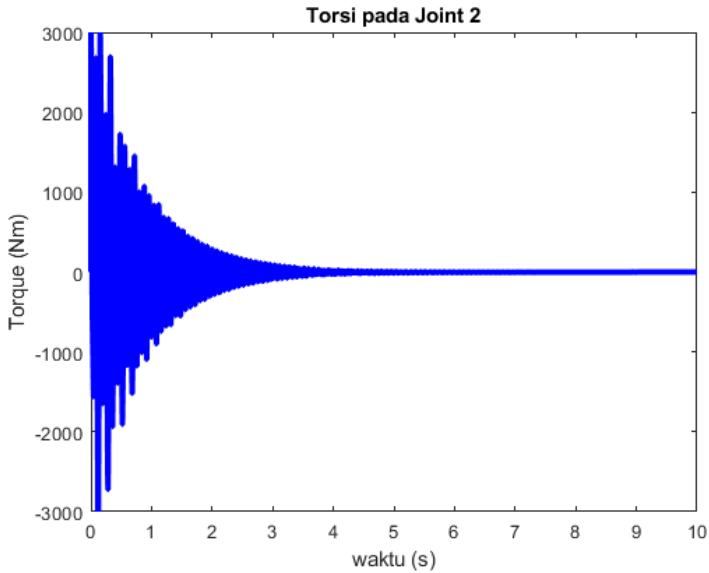
Hasil simulasi percepatan sudut Metode Ziegler-Nichols pada sendi 3 ditunjukkan pada gambar 4.24. Pada grafik tersebut menunjukkan percepatan sudut yang ditempuh oleh sendi 3. Hasil simulasi pada percepatan sudut didapatkan dari turunan kecepatan sudut sendi 3 terhadap waktu. Pada detik ke-0 hingga detik ke-3 terjadi osilasi. Osilasi terjadi antara  $1000 \text{ rad/s}^2$  dan  $-1000 \text{ rad/s}^2$ . Percepatan sudut stabil pada  $0 \text{ rad/s}^2$  pada detik ke-3 sampai detik ke-10.

### 4.2.2.3 Torsi pada tiap Sendi



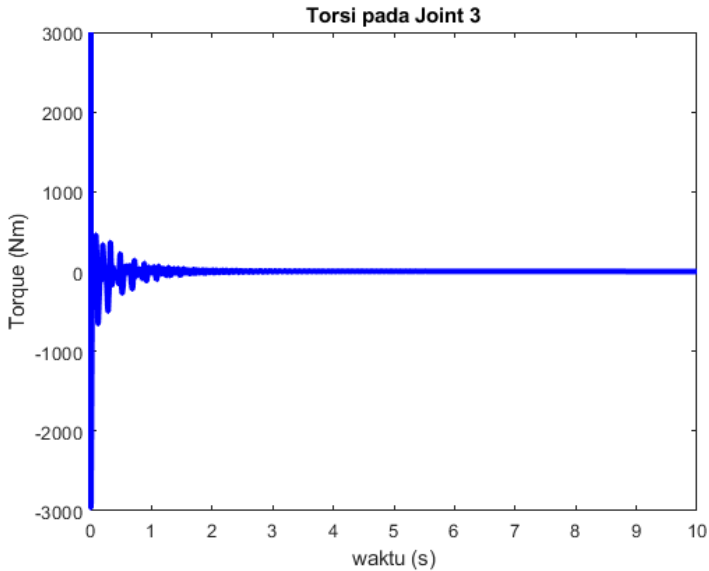
Gambar 4.25 Torsi Sendi 1 menggunakan Metode Ziegler-Nichols pada Setengah Lingkaran

Hasil simulasi torsi Metode Ziegler-Nichols pada sendi 1 ditunjukkan pada gambar 4.25. Pada grafik tersebut menunjukkan torsi yang ditempuh oleh sendi 1. Hasil simulasi besarnya torsi dipengaruhi sudut, kecepatan sudut, dan percepatan sudut pada sendi 1. Pada detik ke-0, mengalami kenaikan hingga 32 Nm, kemudian mengalami osilasi hingga detik ke-6. Pada detik ke-6 hingga detik ke-10 mengalami penurunan hingga -14 Nm.



Gambar 4.26 Torsi Sendi 2 menggunakan Metode Ziegler-Nichols pada Setengah Lingkaran

Hasil simulasi torsi Metode Ziegler-Nichols pada sendi 2 ditunjukkan pada gambar 4.26. Pada grafik tersebut menunjukkan torsi yang ditempuh oleh sendi 2. Hasil simulasi besarnya torsi dipengaruhi sudut, kecepatan sudut, dan percepatan sudut pada sendi 2. Pada detik ke-0 mengalami osilasi hingga detik ke-4. Osilasi terjadi antara 3000 Nm dan -3000 Nm. Pada detik ke-4 hingga detik ke-10 stabil pada 0 Nm.



Gambar 4.27 Torsi Sendi 3 menggunakan Metode Ziegler-Nichols pada Setengah Lingkaran

Hasil simulasi torsi Metode Ziegler-Nichols pada sendi 3 ditunjukkan pada gambar 4.27. Pada grafik tersebut menunjukkan torsi yang ditempuh oleh sendi 3. Hasil simulasi besarnya torsi dipengaruhi sudut, kecepatan sudut, dan percepatan sudut pada sendi 3. Pada detik ke-0, mengalami osilasi hingga detik ke-2. Osilasi terjadi antara 500 Nm dan -500 Nm. Pada detik ke-2 hingga detik ke-10 stabil pada 0 Nm.

### 4.3 Analisa Trajektori Lingkaran

Telah dilakukan simulasi pergerakan lengan robot dengan masukan yang berasal dari inverse kinematics. *End effector* disimulasikan untuk mengikuti lintasan melingkar berdasarkan implementasi dari masukan yang diberikan. Jari-jari untuk lintasan melingkar adalah 0.65 meter. Rumus membuat lingkaran ditulis dalam diagram blok matlab function pada simulink. Matlab function yang berisi rumus lingkaran kemudian disambungkan dengan diagram blok *inverse kinematics*. Adapun rumus yang digunakan untuk membuat setengah lingkaran adalah sebagai berikut

$$x = 0,65 \cos(2\pi)$$

$$y = 0,65 \sin(2\pi)$$

Lama proses simulasi dilakukan selama 10 detik. Dalam simulasi ini, dilakukan dua variasi *tuning* pada kontrol PID yaitu menggunakan *automated tuning* dan metode ziegler-nichols.

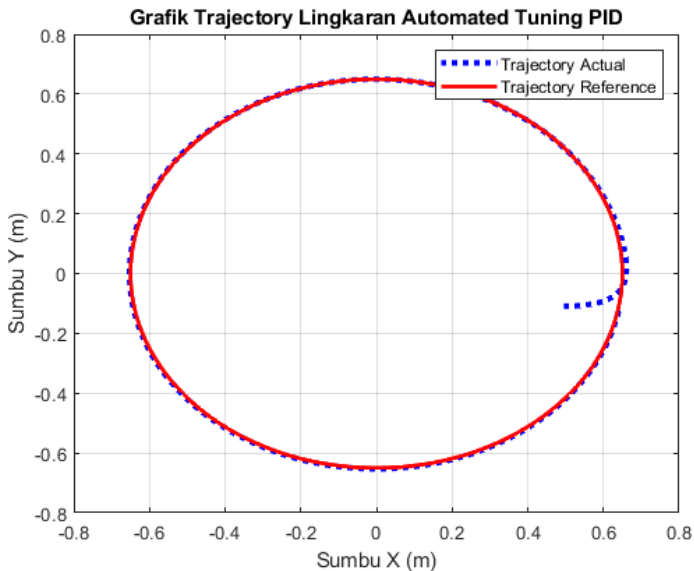
#### 4.3.1 Kontrol PID menggunakan Automated Tuning PID

*Automated PID tuning* pada blok diagram adalah dengan menekan tombol *tune*. Kemudian program MATLAB akan mengidentifikasi *plant* yang dibuat dan memberikan parameter-parameter PID yang disesuaikan dengan *plant*. *Automated tuning* PID mampu menyesuaikan parameter gain sistem secara cepat untuk mendapatkan desain yang robust dengan respon waktu yang bagus, sehingga didapatkan nilai Kp, Ki, dan Kd secara optimal. Hasil yang didapatkan dalam *automated tuning* PID yaitu

Tabel 4.3 Tabel Tuning PID pada *Trajectory* Lingkaran dengan *Automated Tuning*

	Nilai Kp	Nilai Ki	Nilai Kd
Sendi 1	305.765	455.1365	-5.8591

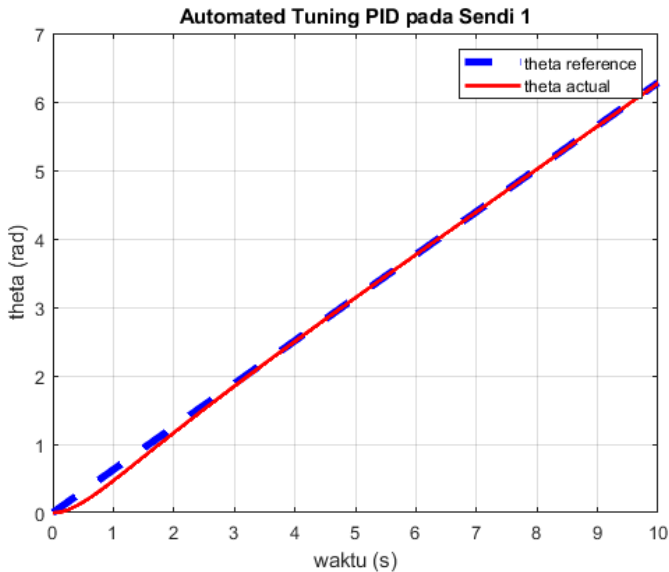
Sendi 2	721.3225	2601.5998	-5.3854
Sendi 3	179.8206	160.3367	-2.5176



Gambar 4.28 Grafik *Trajectory* Lingkaran menggunakan *Automated Tuning PID*

Hasil yang didapatkan berupa grafik yang ditunjukkan pada gambar 4.28. Pada grafik tersebut menunjukkan gerakan dari *end effector* dan referensi *trajectory* yang diinginkan. Pada garis warna biru menunjukkan gerakan dari *end effector*, *end effector* bergerak dari kanan ke kiri. Titik awal pergerakan lengan robot berada pada sumbu X sebesar 0,5 meter dan pada sumbu Y sebesar -0,1 meter. Titik akhir pergerakan lengan robot berada pada sumbu X sebesar 0.65 dan sumbu Y sebesar 0. Garis warna merah menunjukkan referensi *trajectory* yang diinginkan. Pada grafik tersebut menunjukkan bahwa lengan robot bergerak mengikuti *trajectory* dengan presisi.

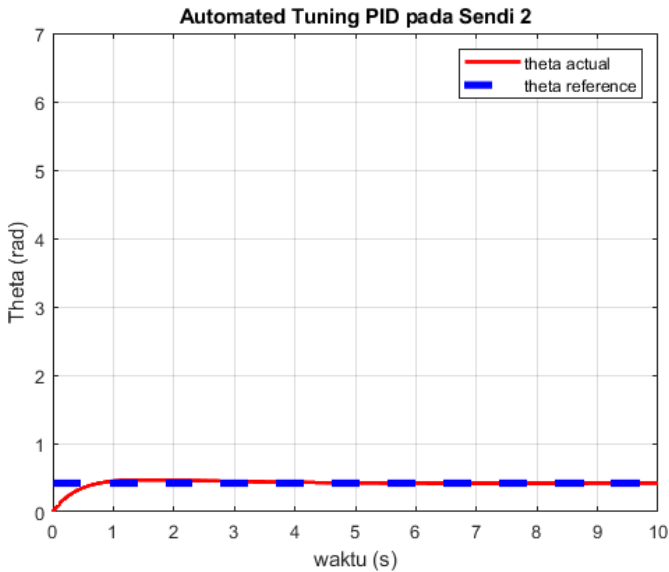




Gambar 4.29 Grafik Sendi 1 menggunakan *Automated Tuning* PID pada Lingkaran

Hasil simulasi *Automated tuning* PID pada sendi 1 ditunjukkan pada gambar 4.29. Pada grafik tersebut menunjukkan sudut yang ditempuh oleh sendi 1. Sudut yang dicapai mulai dari nol sampai 3.14 radian atau setara dengan 180 derajat selama 10 detik. Grafik warna biru menunjukkan referensi sudut yang harus dicapai oleh sendi. Grafik warna merah menunjukkan sudut aktual yang digerakkan oleh sendi 1.

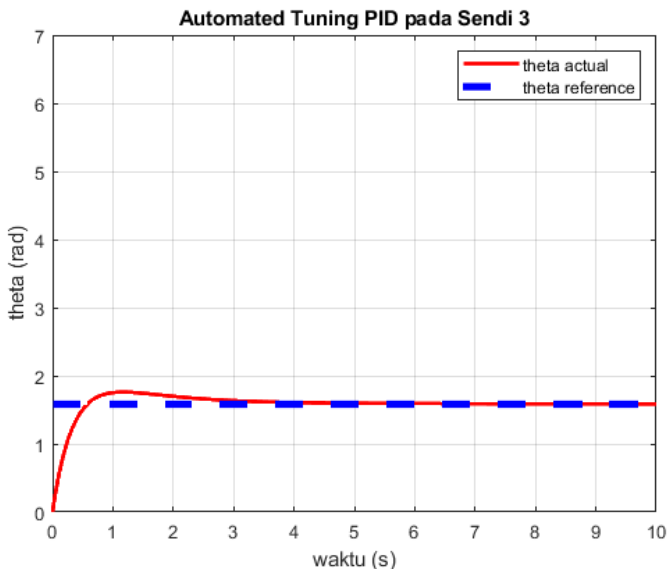
Gerakan sudut pada sendi 1 dipengaruhi oleh kontrol PID. Semakin baik respon yang diberikan PID maka semakin cepat dan presisi sendi 1 mengikuti sudut referensi yang diinginkan. Dari nilai  $K_p$ ,  $K_i$ , dan  $K_d$  tersebut, didapatkan nilai *settling time* = 1.56 detik, *rise time* = 0.188 detik, persen *overshoot* = 12.8%.



Gambar 4.30 Grafik Sendi 2 menggunakan *Automated Tuning* PID pada Lingkaran

Hasil simulasi *Automated tuning* PID pada sendi 2 ditunjukkan pada gambar 4.30. Pada grafik tersebut menunjukkan sudut yang ditempuh oleh sendi 2. Sudut yang harus dicapai adalah 0.42 radian selama 10 detik. Grafik warna biru menunjukkan referensi sudut yang harus dicapai oleh sendi. Grafik warna merah menunjukkan sudut aktual yang digerakkan oleh sendi 2.

Gerakan sudut pada sendi 2 dipengaruhi oleh kontrol PID. Semakin baik respon yang diberikan PID maka semakin cepat dan presisi sendi 2 mengikuti sudut referensi yang diinginkan. Dari nilai  $K_p$ ,  $K_i$ , dan  $K_d$  tersebut, didapatkan nilai *settling time* = 0.652 detik, *rise time* = 0.0723 detik, persen *overshoot* = 13%.

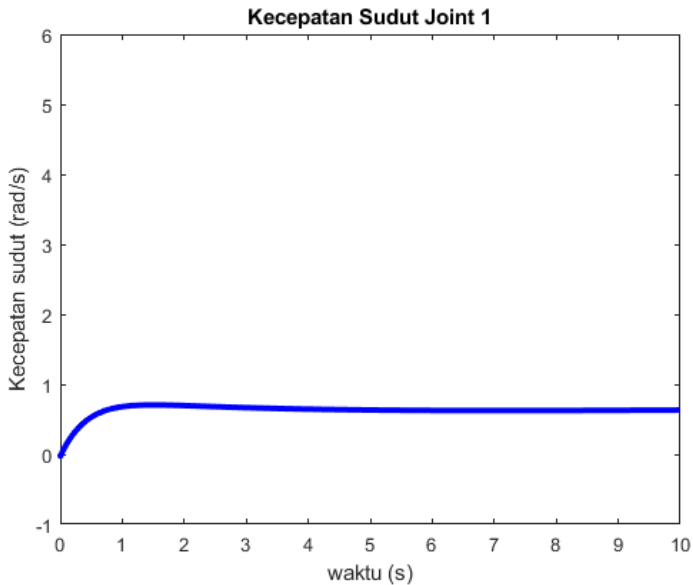


Gambar 4.31: Grafik Sendi 3 menggunakan *Automated Tuning* PID pada Lingkaran

Hasil simulasi *Automated tuning* PID pada sendi 3 ditunjukkan pada gambar 4.31. Pada grafik tersebut menunjukkan sudut yang ditempuh oleh sendi 3. Sudut yang harus dicapai adalah 1.6 radian selama 10 detik. Grafik warna biru menunjukkan referensi sudut yang harus dicapai oleh sendi. Grafik warna merah menunjukkan sudut aktual yang digerakkan oleh sendi 3.

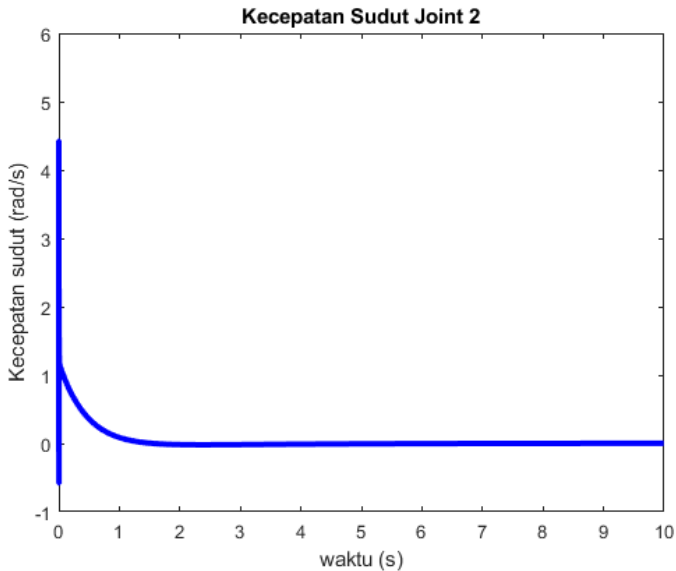
Gerakan sudut pada sendi 3 dipengaruhi oleh kontrol PID. Semakin baik respon yang diberikan PID maka semakin cepat dan presisi sendi 3 mengikuti sudut referensi yang diinginkan. Dari nilai  $K_p$ ,  $K_i$ , dan  $K_d$  tersebut, didapatkan nilai *settling time* = 2.67 detik, *rise time* = 0.318 detik, persen *overshoot* = 11.6%.

### 4.3.1.1 Kecepatan Sudut pada tiap Sendi



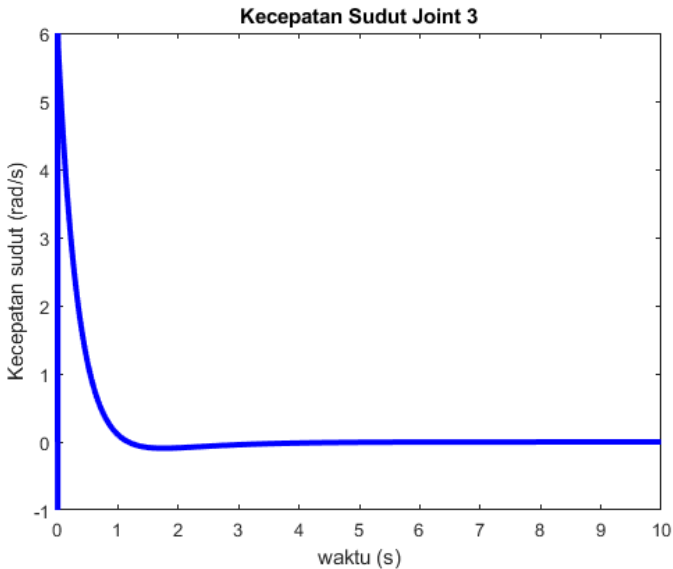
Gambar 4.32 Kecepatan sudut sendi 1 Menggunakan *Automated Tuning* PID pada Lingkaran

Hasil simulasi kecepatan sudut *Automated tuning* PID pada sendi 1 ditunjukkan pada gambar 4.32. Pada grafik tersebut menunjukkan kecepatan sudut yang ditempuh oleh sendi 1. Hasil simulasi pada kecepatan sudut didapatkan dari turunan sudut aktual sendi 1 terhadap waktu. Pada detik ke-0, kecepatan sudut mengalami kenaikan mencapai 0,8 rad/s. Terjadi penurunan lagi pada detik ke-1 mencapai 0,72 rad/s. Kecepatan sudut stabil pada 0,72 rad/s pada detik ke-2 sampai detik ke-10.



Gambar 4.33 Kecepatan sudut sendi 2 Menggunakan *Automated Tuning* PID pada Lingkaran

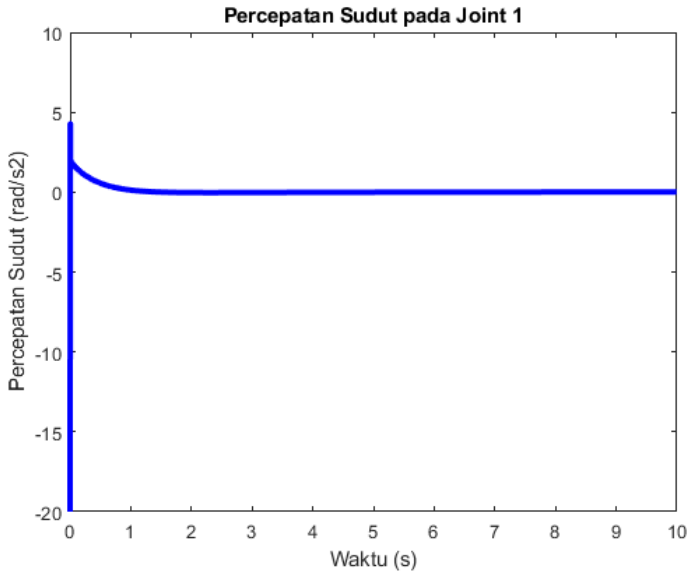
Hasil simulasi kecepatan sudut *Automated tuning* PID pada sendi 2 ditunjukkan pada gambar 4.33. Pada grafik tersebut menunjukkan kecepatan sudut yang ditempuh oleh sendi 2. Hasil simulasi pada kecepatan sudut didapatkan dari turunan sudut aktual sendi 2 terhadap waktu. Pada detik ke-0, kecepatan sudut turun kemudian naik mencapai 1 rad/s. Terjadi penurunan lagi pada detik ke-1 mencapai 0 rad/s. Kecepatan sudut stabil pada 0 rad/s pada detik ke-1 sampai detik ke-10.



Gambar 4.34 Kecepatan sudut sendi 3 Menggunakan *Automated Tuning* PID pada Lingkaran

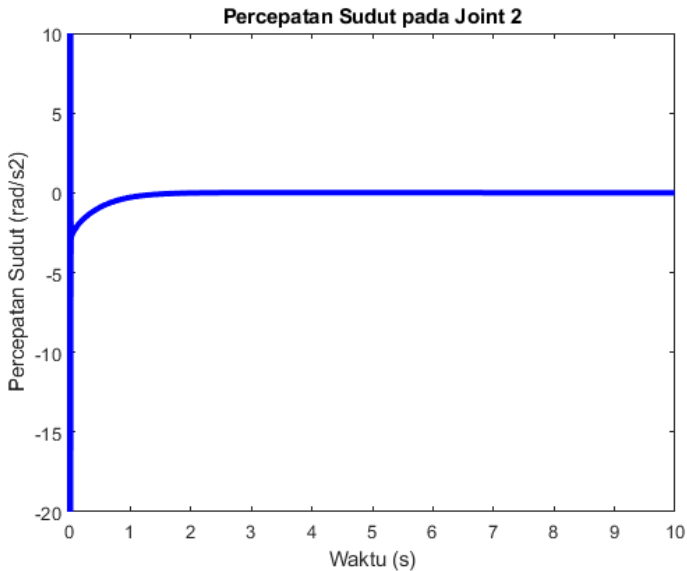
Hasil simulasi kecepatan sudut *Automated tuning* PID pada sendi 3 ditunjukkan pada gambar 4.34. Pada grafik tersebut menunjukkan kecepatan sudut yang ditempuh oleh sendi 3. Hasil simulasi pada kecepatan sudut didapatkan dari turunan sudut aktual sendi 3 terhadap waktu. Pada detik ke-0, kecepatan sudut turun kemudian naik mencapai 6 rad/s. Terjadi penurunan lagi pada detik ke-1 mencapai 0 rad/s. Kecepatan sudut stabil pada 0 rad/s pada detik ke-1 sampai detik ke-10.

#### 4.3.1.2 Percepatan Sudut pada tiap Sendi



Gambar 4.35 Percepatan Sudut Sendi 1 menggunakan *Automated Tuning* PID pada Lingkaran

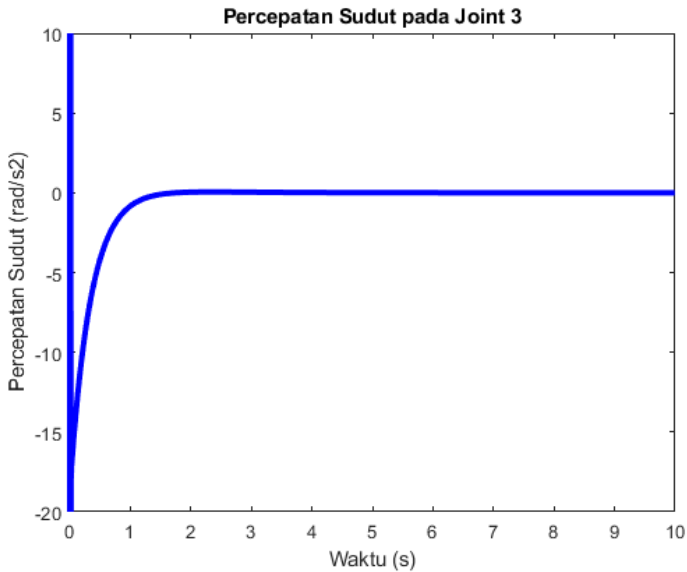
Hasil simulasi percepatan sudut *Automated tuning* PID pada sendi 1 ditunjukkan pada gambar 4.35. Pada grafik tersebut menunjukkan percepatan sudut yang ditempuh oleh sendi 1. Hasil simulasi pada percepatan sudut didapatkan dari turunan kecepatan sudut sendi 1 terhadap waktu. Pada detik ke-0, mengalami penurunan hingga -20  $\text{rad/s}^2$ , kemudian mengalami kenaikan hingga mencapai 4  $\text{rad/s}^2$ . Pada detik ke-1 mengalami pengurangan kecepatan sudut dari 4  $\text{rad/s}^2$  hingga mencapai 0  $\text{rad/s}^2$  yang stabil hingga detik ke-10.



Gambar 4.36 Percepatan Sudut Sendi 2 menggunakan *Automated Tuning* PID pada Lingkaran

Hasil simulasi percepatan sudut *Automated tuning* PID pada sendi 2 ditunjukkan pada gambar 4.36. Pada grafik tersebut menunjukkan percepatan sudut yang ditempuh oleh sendi 2. Hasil simulasi pada percepatan sudut didapatkan dari turunan kecepatan sudut sendi 2 terhadap waktu. Pada detik ke-0, mengalami penurunan hingga  $-3 \text{ rad/s}^2$ , kemudian mengalami kenaikan hingga mencapai  $0 \text{ rad/s}^2$  yang stabil hingga detik ke-10.

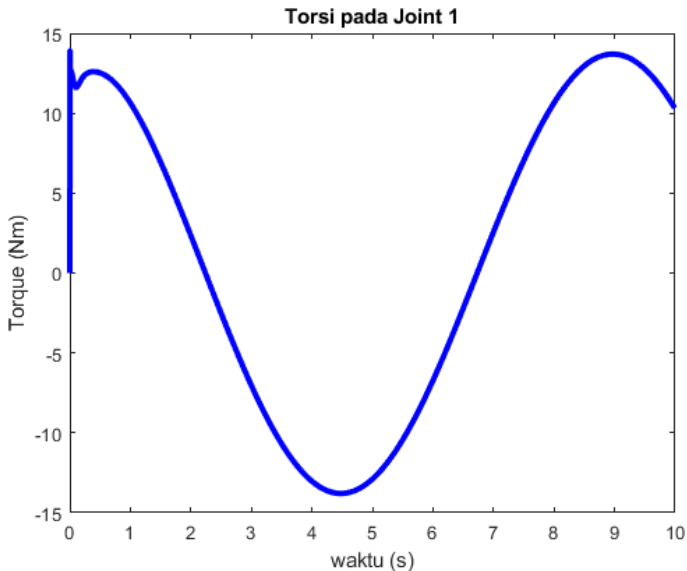




Gambar 4.37 Percepatan Sudut Sendi 3 menggunakan *Automated Tuning* PID pada Lingkaran

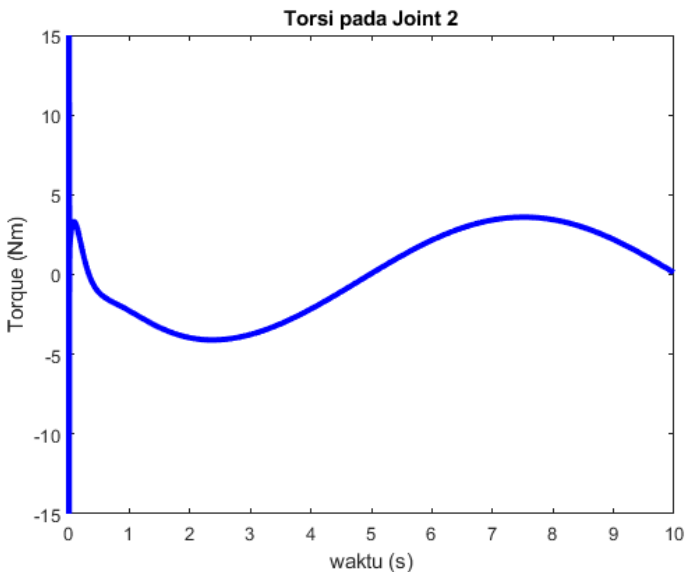
Hasil simulasi percepatan sudut *Automated tuning* PID pada sendi 3 ditunjukkan pada gambar 4.37. Pada grafik tersebut menunjukkan percepatan sudut yang ditempuh oleh sendi 3. Hasil simulasi pada percepatan sudut didapatkan dari turunan kecepatan sudut sendi 3 terhadap waktu. Pada detik ke-0, mengalami penurunan hingga  $-15 \text{ rad/s}^2$ , kemudian mengalami kenaikan mencapai  $0 \text{ rad/s}^2$  yang stabil hingga detik ke-10.

### 4.3.1.3 Torsi pada tiap Sendi



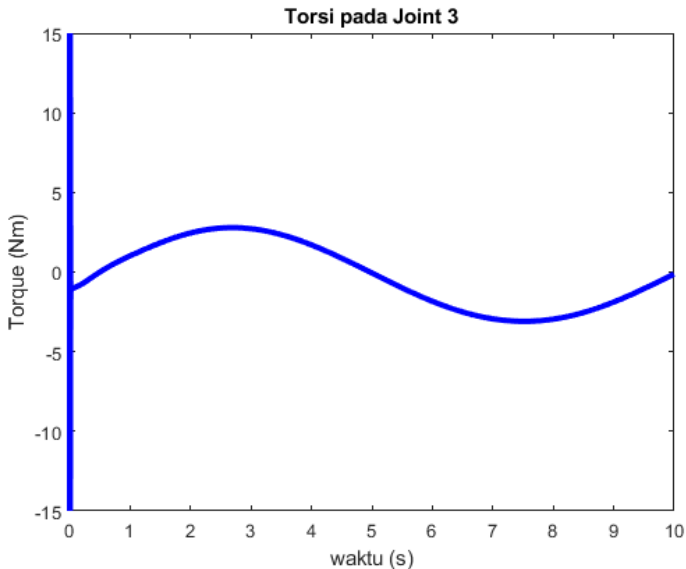
Gambar 4.38 Torsi Sendi 1 menggunakan *Automated Tuning* PID pada Lingkaran

Hasil simulasi torsi *Automated tuning* PID pada sendi 1 ditunjukkan pada gambar 4.38. Pada grafik tersebut menunjukkan torsi yang ditempuh oleh sendi 1. Hasil simulasi besarnya torsi dipengaruhi sudut, kecepatan sudut, dan percepatan sudut pada sendi 1. Grafik torsi sendi 1 seperti grafik cosinus. Pada detik ke-0, mengalami kenaikan hingga 14 Nm. Pada detik ke-4 hingga detik ke-5, terjadi penurunan maksimum mencapai -14 Nm. Pada detik ke-9, terjadi kenaikan maksimum mencapai 14 Nm.



Gambar 4.39 Torsi Sendi 2 menggunakan *Automated Tuning* PID pada Lingkaran

Hasil simulasi torsi *Automated tuning* PID pada sendi 2 ditunjukkan pada gambar 4.39. Pada grafik tersebut menunjukkan torsi yang ditempuh oleh sendi 2. Hasil simulasi besarnya torsi dipengaruhi sudut, kecepatan sudut, dan percepatan sudut pada sendi 2. Pada detik ke-0, mengalami kenaikan hingga 3,2 Nm. Pada detik ke-2 mengalami penurunan hingga -4 Nm, kemudian pada detik ke-7 mengalami kenaikan hingga 4 Nm. Pada detik ke-7 hingga detik ke-10 mengalami penurunan hingga 0 Nm.



Gambar 4.40 Torsi Sendi 3 menggunakan *Automated Tuning* PID pada Lingkaran

Hasil simulasi torsi *Automated tuning* PID pada sendi 3 ditunjukkan pada gambar 4.40. Pada grafik tersebut menunjukkan torsi yang ditempuh oleh sendi 3. Hasil simulasi besarnya torsi dipengaruhi sudut, kecepatan sudut, dan percepatan sudut pada sendi 3. Pada detik ke-0, mengalami penurunan hingga -1 Nm. Pada detik ke-2 mengalami penurunan hingga 2,78 Nm, kemudian pada detik ke-7 mengalami penurunan hingga -3,1 Nm. Pada detik ke-7 hingga detik ke-10 mengalami kenaikan hingga 0 Nm.

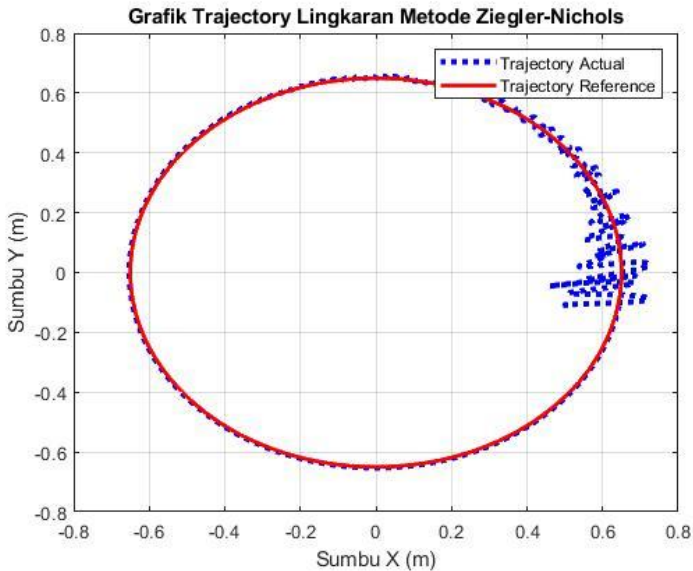
### 4.3.2 Kontrol PID menggunakan Metode Ziegler-Nichols

Simulasi pergerakan lengan robot dengan kontrol PID dilakukan dengan metode ziegler-nichol. Metode ini digunakan untuk mendapatkan parameter-parameter seperti Kp, Ki, dan Kd. Parameter tersebut selanjutnya dilakukan *tuning* pada diagram blok PID. Diagram blok PID kemudian disambungkan dengan motor DC dan lengan robot. Berikut parameter-parameter yang didapatkan saat simulasi menggunakan MATLAB.

Tabel 4.4 Tabel Tuning PID Metode Ziegler-Nichols Lingkaran

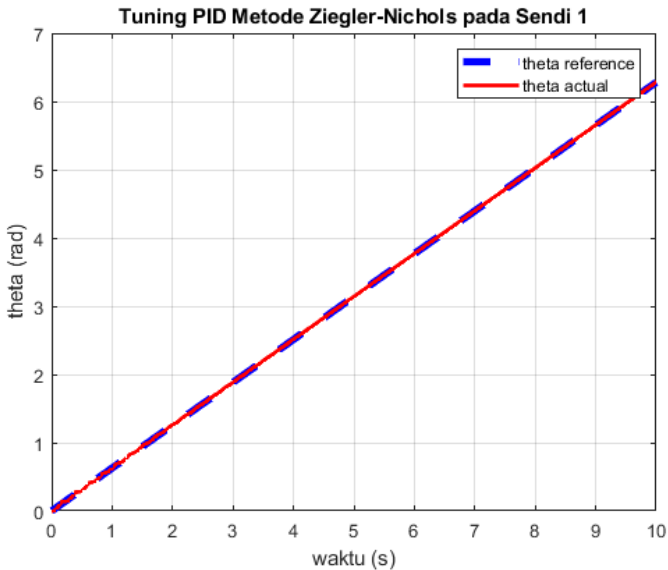
	Nilai Kp	Nilai Ki	Nilai Kd
Sendi 1	140.991	28795.2754	0.17258
Sendi 2	1014.2902	220727.3244	1.1652
Sendi 3	190.5715	37687.3572	0.24091

Parameter - parameter tersebut mempengaruhi tingkat kepresisian dari pergerakan sendi pada lengan robot terhadap *trajectory* yang diinginkan. Apabila parameter tersebut dimasukkan ke dalam blok diagram PID pada simulink, maka didapatkan grafik pergerakan end effector. Grafik pergerakan *end effector* ditunjukkan pada gambar 4.41.



Gambar 4.41 Grafik *Trajectory* Lingkaran menggunakan Metode Ziegler-Nichols

Grafik 4.41 menunjukkan gerakan dari *end effector* aktual dan referensi trajectory yang diinginkan. Garis warna biru menunjukkan gerakan aktual dari *end effector*, *end effector* bergerak dari kanan ke kiri. Titik awal pergerakan robot berada pada sumbu X sebesar 0,5 meter dan pada sumbu Y sebesar -0,1 meter, sedangkan titik akhir berada pada sumbu X sebesar 0 dan pada sumbu Y sebesar 0.65. Gerakan lengan robot berjalan dari koordinat awal kemudian berputar berlawanan dengan arah jarum jam. Pergerakan lengan robot mengalami osilasi di awal, tetapi gerakan lengan robot bisa stabil dan mengikuti referensi *trajectory*. Garis warna merah menunjukkan referensi trajectory yang diinginkan. Pada grafik tersebut menunjukkan bahwa lengan robot bergerak mengikuti *trajectory*, namun tingkat kepresisiannya kurang baik diawal simulasi.

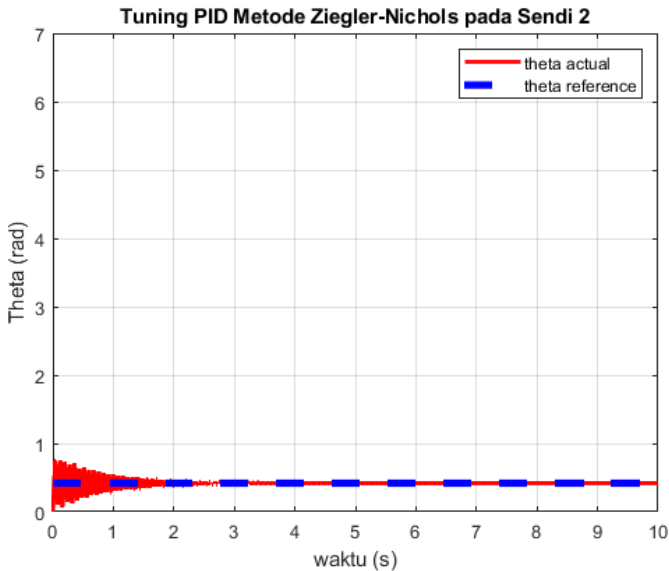


Gambar 4.42 Gerakan Sendi 1 menggunakan Metode Ziegler-Nichols pada Lingkaran

Hasil simulasi *tuning* PID menggunakan metode ziegler-nichols pada sendi 1 ditunjukkan pada gambar 4.42. Pada grafik tersebut menunjukkan sudut yang ditempuh oleh sendi 1. Sudut yang dicapai mulai dari nol sampai 6.28 radian atau setara dengan 360 derajat selama 10 detik. Grafik warna biru menunjukkan referensi sudut yang harus dicapai oleh sendi. Grafik warna merah menunjukkan sudut aktual yang digerakkan oleh sendi 1.

Gerakan sudut pada sendi 1 dipengaruhi oleh tuning pada kontrol PID. Semakin baik respon yang diberikan PID maka semakin cepat dan presisi sendi 1 mengikuti sudut referensi yang diinginkan. Dari nilai  $K_p$ ,  $K_i$ , dan  $K_d$  tersebut, didapatkan nilai *settling time* = 3.85 detik, *rise time* = 0.0367 detik, persen *overshoot* = 89.8%. Besarnya *overshoot* mempengaruhi *steady state error* hingga menyebabkan respon tidak cepat stabil dan

berosilasi. *Overshoot* yang besar disebabkan oleh nilai *gain* integral ( $K_i$ ) yang tidak sesuai dengan *plant* sistem. Pada kasus Ziegler-Nichols tersebut, nilai  $K_i$  yang diberikan terlalu besar sehingga menyebabkan besarnya persen *overshoot*.



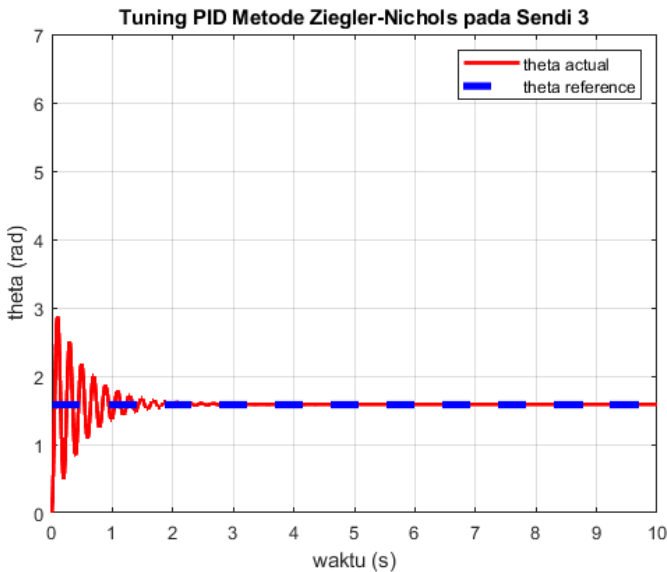
Gambar 4.43: Grafik Sendi 2 menggunakan Metode Ziegler-Nichols pada Lingkaran

Hasil simulasi *tuning* PID menggunakan metode ziegler-nichols pada sendi 2 ditunjukkan pada gambar 4.43. Pada grafik tersebut menunjukkan sudut yang ditempuh oleh sendi 2. Sudut yang harus dicapai adalah 0.42 radian selama 10 detik. Grafik warna biru menunjukkan referensi sudut yang harus dicapai oleh sendi. Grafik warna merah menunjukkan sudut aktual yang digerakkan oleh sendi 2.

Gerakan sudut pada sendi 2 dipengaruhi oleh *tuning* pada kontrol PID. Gerakan pada sendi 2 mengalami osilasi yang cukup besar dan rapat, osilasi ini terjadi di awal simulasi. Osilasi tersebut



kembali stabil pada detik ke-7 dan selanjutnya bergerak mengikuti referensi *trajectory*. Dari nilai  $K_p$ ,  $K_i$ , dan  $K_d$  menggunakan metode ziegler-nichols, didapatkan nilai *settling time* = 0.761 detik, *rise time* = 0.0126 detik, persen *overshoot* = 85.7%. Besarnya *overshoot* mempengaruhi *steady state error* hingga menyebabkan respon tidak cepat stabil dan berosilasi. *Overshoot* yang besar disebabkan oleh nilai *gain* integral ( $K_i$ ) yang tidak sesuai dengan *plant* sistem. Pada kasus Ziegler-Nichols tersebut, nilai  $K_i$  yang diberikan terlalu besar sehingga menyebabkan besarnya persen *overshoot*.



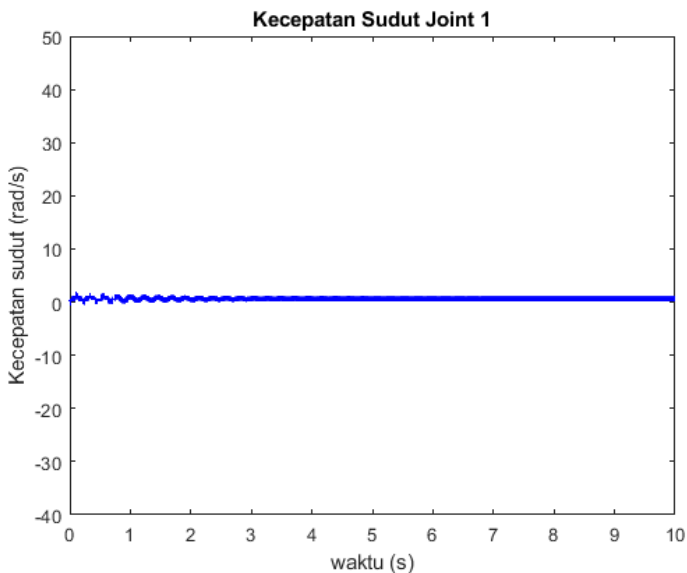
Gambar 4.44: Gerakan Sendi 3 menggunakan Metode Ziegler-Nichols pada Lingkaran

Hasil simulasi *tuning* PID menggunakan metode ziegler-nichols pada sendi 3 ditunjukkan pada gambar 4.2.8. Pada grafik tersebut menunjukkan sudut yang ditempuh oleh sendi 3. Sudut

yang harus dicapai adalah 1.6 radian selama 10 detik. Grafik warna biru menunjukkan referensi sudut yang harus dicapai oleh sendi. Grafik warna merah menunjukkan sudut aktual yang digerakkan oleh sendi 3.

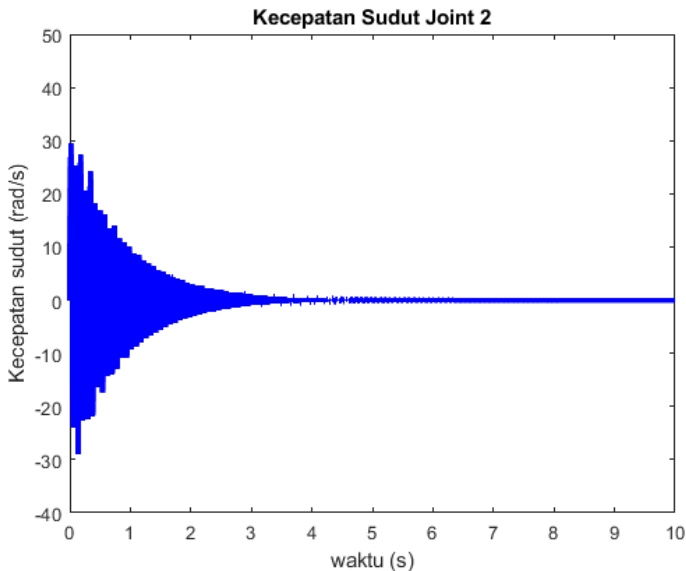
Gerakan sudut pada sendi 3 dipengaruhi oleh tuning pada kontrol PID. Gerakan pada sendi 3 mengalami osilasi yang cukup besar dan tidak rapat, osilasi ini terjadi di awal simulasi. Osilasi tersebut kembali stabil pada detik ke-3 dan selanjutnya bergerak mengikuti referensi *trajectory*. Dari nilai  $K_p$ ,  $K_i$ , dan  $K_d$  menggunakan metode ziegler-nichols, didapatkan nilai *settling time* = 1.99 detik, *rise time* = 0.0332 detik, persen *overshoot* = 83%. Besarnya *overshoot* mempengaruhi *steady state error* hingga menyebabkan respon tidak cepat stabil dan berosilasi. *Overshoot* yang besar disebabkan oleh nilai *gain* integral ( $K_i$ ) yang tidak sesuai dengan *plant* sistem. Pada kasus Ziegler-Nichols tersebut, nilai  $K_i$  yang diberikan terlalu besar sehingga menyebabkan besarnya persen *overshoot*.

#### 4.3.2.1 Kecepatan Sudut pada tiap Sendi



Gambar 4.45 Kecepatan sudut sendi 1 Metode Ziegler-Nichols pada Lingkaran

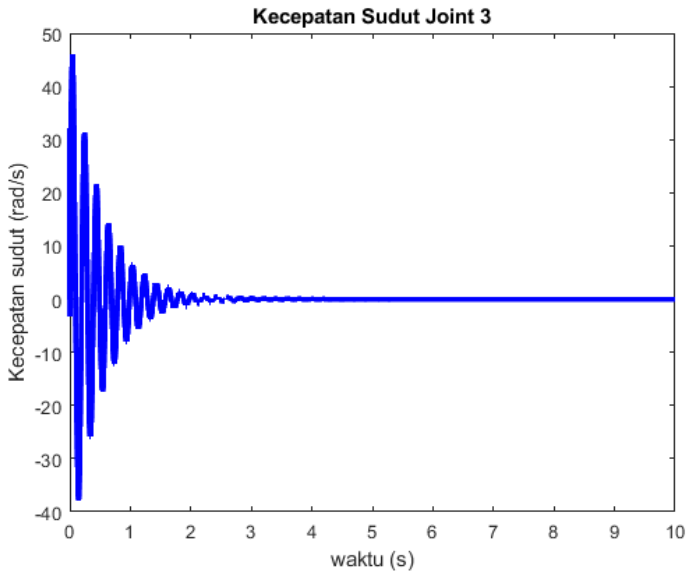
Hasil simulasi kecepatan sudut Metode Ziegler-Nichols pada sendi 1 ditunjukkan pada gambar 4.45. Pada grafik tersebut menunjukkan kecepatan sudut yang ditempuh oleh sendi 1. Hasil simulasi pada kecepatan sudut didapatkan dari turunan sudut aktual sendi 1 terhadap waktu. Pada detik ke-0 hingga detik ke-3 terjadi osilasi. Kecepatan sudut stabil pada 0 rad/s pada detik ke-3 sampai detik ke-10.



Gambar 4.46 Kecepatan sudut sendi 2 Metode Ziegler-Nichols pada Lingkaran

Hasil simulasi kecepatan sudut Metode Ziegler-Nichols pada sendi 2 ditunjukkan pada gambar 4.46. Pada grafik tersebut menunjukkan kecepatan sudut yang ditempuh oleh sendi 2. Hasil simulasi pada kecepatan sudut didapatkan dari turunan sudut aktual

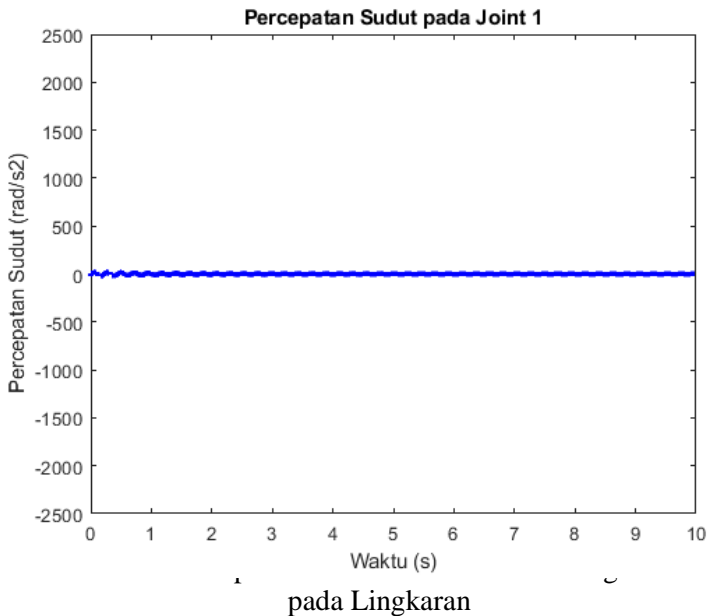
sendi 2 terhadap waktu. Pada detik ke-0 hingga detik ke-5 terjadi osilasi. Kecepatan sudut stabil pada 0 rad/s pada detik ke-5 sampai detik ke-10.



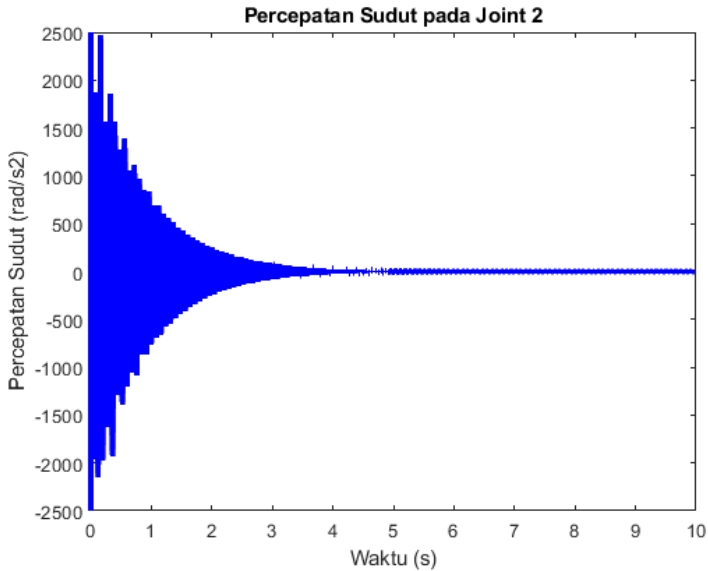
Gambar 4.47 Kecepatan sudut sendi 3 Metode Ziegler-Nichols pada Lingkaran

Hasil simulasi kecepatan sudut Metode Ziegler-Nichols pada sendi 3 ditunjukkan pada gambar 4.47. Pada grafik tersebut menunjukkan kecepatan sudut yang ditempuh oleh sendi 3. Hasil simulasi pada kecepatan sudut didapatkan dari turunan sudut aktual sendi 3 terhadap waktu. Pada detik ke-0 hingga detik ke-4 terjadi osilasi. Kecepatan sudut stabil pada 0 rad/s pada detik ke-4 sampai detik ke-10.

#### 4.3.2.2 Percepatan Sudut pada tiap Sendi

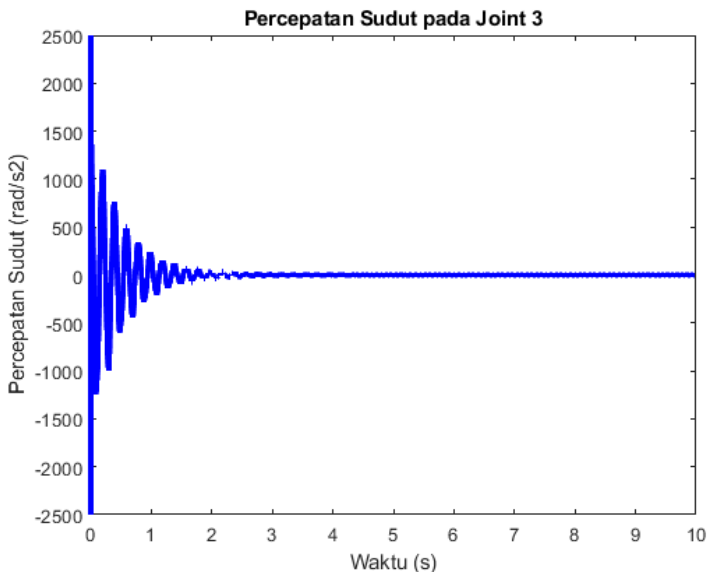


Hasil simulasi percepatan sudut Metode Ziegler-Nichols pada sendi 1 ditunjukkan pada gambar 4.48. Pada grafik tersebut menunjukkan percepatan sudut yang ditempuh oleh sendi 1. Hasil simulasi pada percepatan sudut didapatkan dari turunan kecepatan sudut sendi 1 terhadap waktu. Pada detik ke-0 hingga detik ke-8 terjadi osilasi. Percepatan sudut stabil pada 0 rad/s pada detik ke-8 sampai detik ke-10.



Gambar 4.49 Percepatan sudut sendi 2 Metode Ziegler-Nichols pada Lingkaran

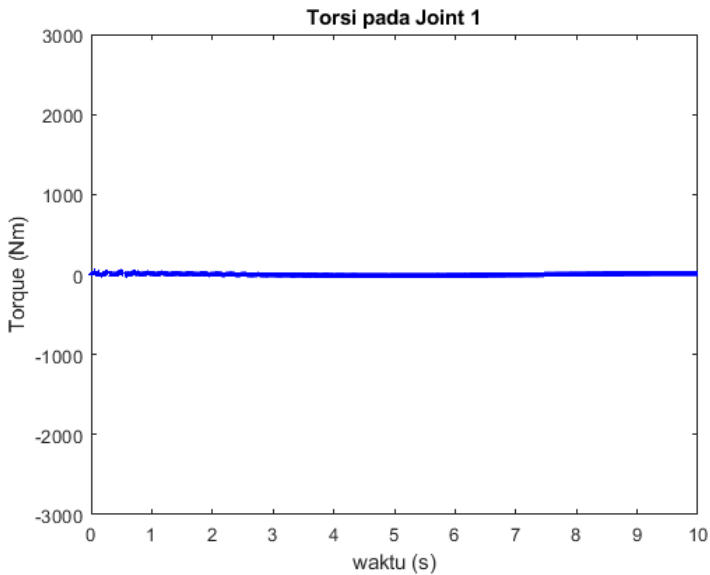
Hasil simulasi percepatan sudut Metode Ziegler-Nichols pada sendi 2 ditunjukkan pada gambar 4.49. Pada grafik tersebut menunjukkan percepatan sudut yang ditempuh oleh sendi 2. Hasil simulasi pada percepatan sudut didapatkan dari turunan kecepatan sudut sendi 2 terhadap waktu. Pada detik ke-0 hingga detik ke-5 terjadi osilasi. Percepatan sudut stabil pada 0 rad/s pada detik ke-5 sampai detik ke-10.



Gambar 4.50 Percepatan sudut sendi 3 Metode Ziegler-Nichols pada Lingkaran

Hasil simulasi percepatan sudut Metode Ziegler-Nichols pada sendi 3 ditunjukkan pada gambar 4.50. Pada grafik tersebut menunjukkan percepatan sudut yang ditempuh oleh sendi 3. Hasil simulasi pada percepatan sudut didapatkan dari turunan kecepatan sudut sendi 3 terhadap waktu. Pada detik ke-0 hingga detik ke-4 terjadi osilasi. Percepatan sudut stabil pada 0  $\text{rad/s}$  pada detik ke-4 sampai detik ke-10.

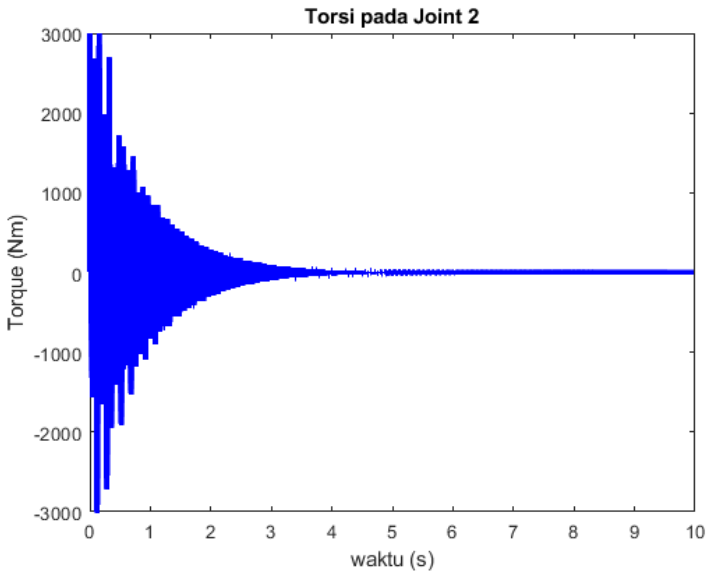
### 4.3.2.3 Torsi pada tiap Sendi



Gambar 4.51 Torsi Sendi 1 menggunakan Metode Ziegler-Nichols pada Lingkaran

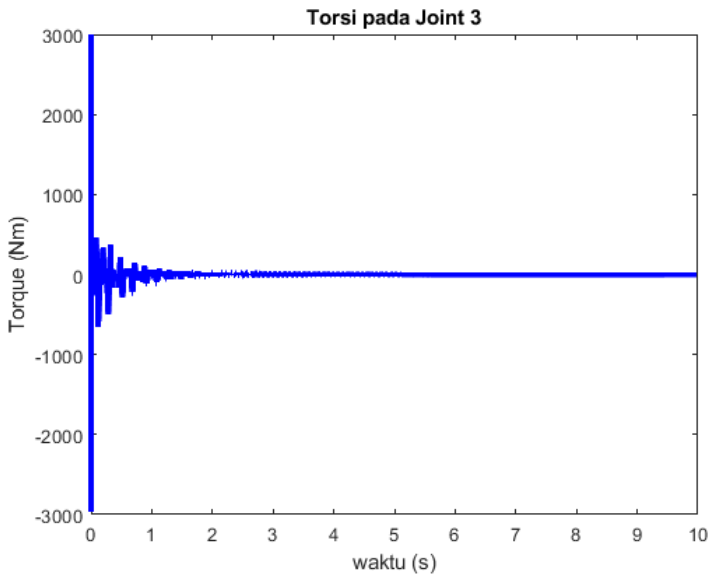
Hasil simulasi torsi Metode Ziegler-Nichols pada sendi 1 ditunjukkan pada gambar 4.51. Pada grafik tersebut menunjukkan torsi yang ditempuh oleh sendi 1. Hasil simulasi besarnya torsi dipengaruhi sudut, kecepatan sudut, dan percepatan sudut pada sendi 1. Pada detik ke-0 hingga detik ke-6 mengalami osilasi. Pada detik ke-7 stabil dan mengalami peningkatan hingga 10 Nm.





Gambar 4.52 Torsi Sendi 2 menggunakan Metode Ziegler-Nichols pada Lingkaran

Hasil simulasi torsi Metode Ziegler-Nichols pada sendi 2 ditunjukkan pada gambar 4.52. Pada grafik tersebut menunjukkan torsi yang ditempuh oleh sendi 2. Hasil simulasi besarnya torsi dipengaruhi sudut, kecepatan sudut, dan percepatan sudut pada sendi 2. Pada detik ke-0, mengalami osilasi hingga detik ke-5. Osilasi terjadi antara 3000 Nm dan -3000 Nm. Pada detik ke-5 hingga detik ke-10 stabil pada 0 Nm.



Gambar 4.53 Torsi Sendi 3 menggunakan Metode Ziegler-Nichols pada Lingkaran

Hasil simulasi torsi Metode Ziegler-Nichols pada sendi 3 ditunjukkan pada gambar 4.53. Pada grafik tersebut menunjukkan torsi yang ditempuh oleh sendi 3. Hasil simulasi besarnya torsi dipengaruhi sudut, kecepatan sudut, dan percepatan sudut pada sendi 3. Pada detik ke-0, mengalami osilasi hingga detik ke-3. Osilasi terjadi antara 500 Nm dan -500 Nm. Pada detik ke-3 hingga detik ke-10 stabil pada 0 Nm.

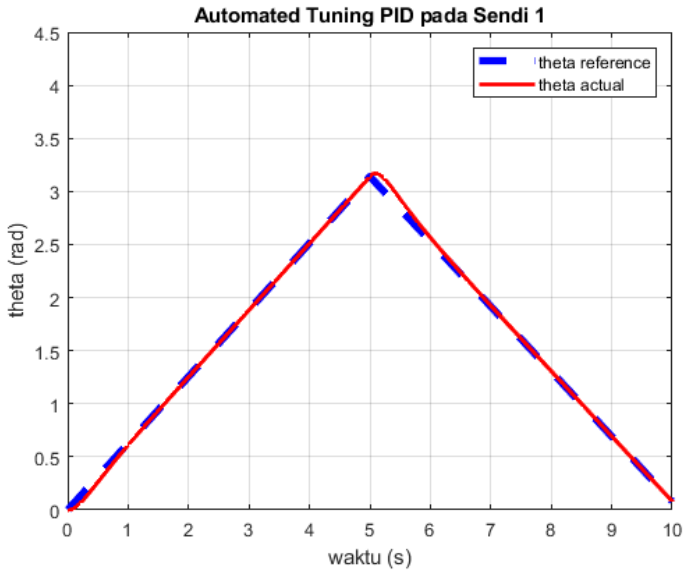
## 4.4 Analisa Trajectory Bebas

### 4.4.1 Kontrol PID menggunakan Advance Tuning PID

Advance PID tuning pada blok diagram adalah dengan menekan tombol tune. Kemudian program MATLAB akan mengidentifikasi plant yang dibuat dan memberikan parameter-parameter PID yang disesuaikan dengan plant. Advance PID tuning mampu menyesuaikan parameter gain sistem secara cepat untuk mendapatkan desain yang robust dengan respon waktu yang bagus, sehingga didapatkan nilai  $K_p$ ,  $K_i$ , dan  $K_d$  secara optimal. Hasil yang didapatkan dalam advance tuning PID yaitu

Tabel 4.5: Tabel Tuning PID pada Trajectory Bebas dengan *Automated Tuning*

	Nilai $K_p$	Nilai $K_i$	Nilai $K_d$
Sendi 1	376.2043	695.7069	-3.5435
Sendi 2	1227.9766	7283.8264	5.2512
Sendi 3	646.6129	2076.7024	-1.0778

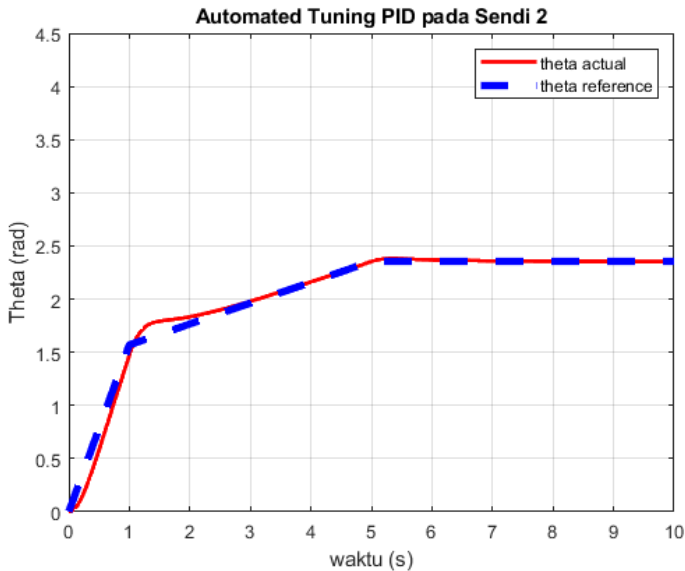


Gambar 4.54: Grafik Sendi 1 menggunakan *Automated Tuning* PID pada *Trajectory* Bebas

Hasil simulasi *Automated tuning* PID pada sendi 1 ditunjukkan pada gambar 4.54. Pada grafik tersebut menunjukkan sudut yang ditempuh oleh sendi 1. Sudut yang diatur adalah naik pada detik ke-5 sebesar 3,14 radian, kemudian turun 0 radian pada detik ke-10. Grafik warna biru menunjukkan referensi sudut yang harus dicapai oleh sendi. Grafik warna merah menunjukkan sudut aktual yang digerakkan oleh sendi 1.

Gerakan sudut pada sendi 1 dipengaruhi oleh kontrol PID. Semakin baik respon yang diberikan PID maka semakin cepat dan presisi sendi 1 mengikuti sudut referensi yang diinginkan. Pada simulasi *trajectory* bebas, sendi bergerak namun pergerakan ini sangat kecil. Hal ini mempengaruhi tingkat kepresisian gerak dari sendi 1. Kejadian ini disebabkan besarnya torsi yang diterima oleh motor sehingga motor tidak bisa menahan dari besarnya torsi yang

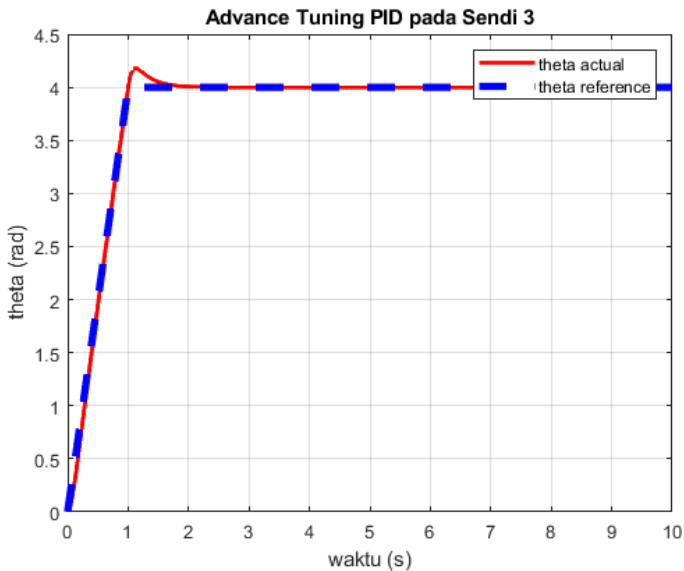
diterima. Dari nilai  $K_p$ ,  $K_i$ , dan  $K_d$  tersebut, didapatkan nilai *settling time* = 1.27 detik, *rise time* = 0.151 detik, persen *overshoot* = 12.1%.



Gambar 4.55 Grafik Sendi 2 menggunakan *Automated Tuning* PID pada *Trajectory Bebas*

Hasil simulasi *Automated tuning* PID pada sendi 2 ditunjukkan pada gambar 4.55. Pada grafik tersebut menunjukkan sudut yang ditempuh oleh sendi 2. Sudut yang diatur adalah 1.6 radian selama 1 detik, kemudian sudut dinaikkan senilai 2,45 radian sampai detik ke-5. Pada detik ke-5 sampai detik ke-10, sudut diatur konstan pada nilai 2.45 radian. Grafik warna biru menunjukkan referensi sudut yang harus dicapai oleh sendi. Grafik warna merah menunjukkan sudut aktual yang digerakkan oleh sendi 2.

Gerakan sudut pada sendi 2 dipengaruhi oleh kontrol PID. Semakin baik respon yang diberikan PID maka semakin cepat dan presisi sendi 2 mengikuti sudut referensi yang diinginkan. Dari nilai  $K_p$ ,  $K_i$ , dan  $K_d$  tersebut, didapatkan nilai *settling time* = 0.0484 detik, *rise time* = 0.402 detik, persen *overshoot* = 10%.

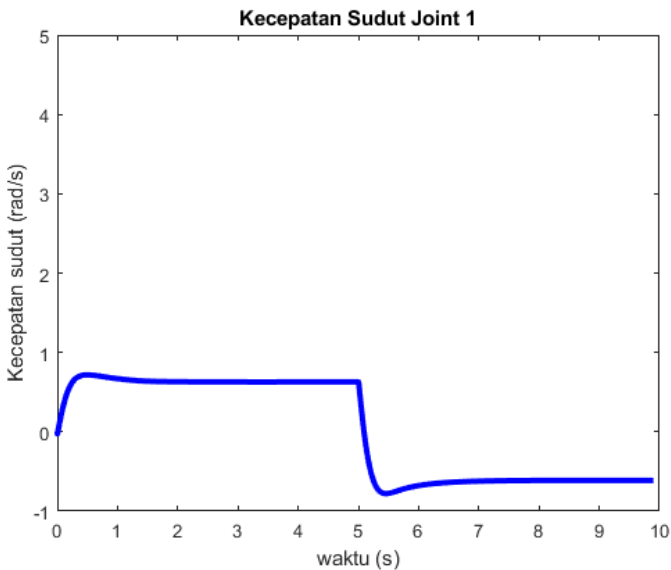


Gambar 4.56 Grafik Sendi 3 menggunakan Advance Tuning PID pada *Trajectory Bebas*

Hasil simulasi *Automated tuning* PID pada sendi 3 ditunjukkan pada gambar 4.56. Pada grafik tersebut menunjukkan sudut yang ditempuh oleh sendi 3 Sudut yang diatur naik adalah 4 radian selama 1 detik. Pada detik ke-1 sampai detik ke-10, sudut diatur konstan pada nilai 4 radian. Grafik warna biru menunjukkan referensi sudut yang harus dicapai oleh sendi. Grafik warna merah menunjukkan sudut aktual yang digerakkan oleh sendi 3.

Gerakan sudut pada sendi 3 di-pengaruhi oleh kontrol PID. Semakin baik respon yang diberikan PID maka semakin cepat dan presisi sendi 2 mengikuti sudut referensi yang diinginkan. Dari nilai  $K_p$ ,  $K_i$ , dan  $K_d$  tersebut, didapatkan nilai *settling time* = 0.743 detik, *rise time* = 0.088 detik, persen *overshoot* = 11.4%.

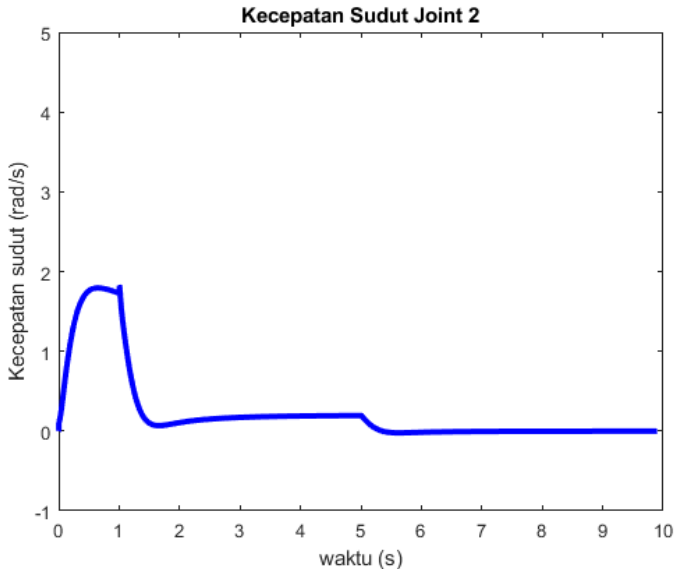
#### 4.4.1.1 Kecepatan Sudut pada tiap Sendi



Gambar 4.57 Kecepatan sudut sendi 1 Menggunakan *Automated Tuning* PID pada *Trajectory* Bebas

Hasil simulasi kecepatan sudut *Automated tuning* PID pada sendi 1 ditunjukkan pada gambar 4.57. Pada grafik tersebut menunjukkan kecepatan sudut yang ditempuh oleh sendi 1. Hasil simulasi pada kecepatan sudut didapatkan dari turunan sudut aktual sendi 1 terhadap waktu. Pada detik ke-0, kecepatan sudut naik mencapai 0,8 rad/s. Pada detik ke-1 hingga detik ke-5 stabil pada

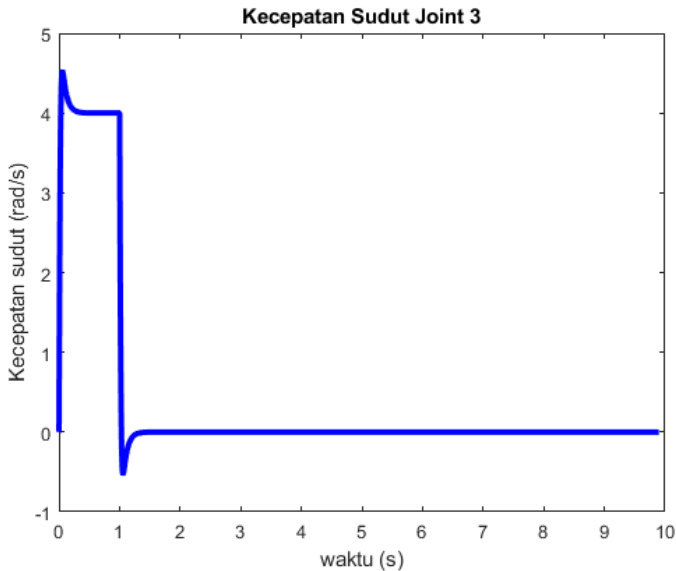
0,7 rad/s, kemudian mengalami penurunan hingga -0,8 rad/s. Pada detik ke-6 hingga detik ke-10 stabil pada -0,7 rad/s.



Gambar 4.58 Kecepatan sudut sendi 2 Menggunakan *Automated Tuning* PID pada *Trajectory* Bebas

Hasil simulasi kecepatan sudut *Automated tuning* PID pada sendi 2 ditunjukkan pada gambar 4.58. Pada grafik tersebut menunjukkan kecepatan sudut yang ditempuh oleh sendi 2. Hasil simulasi pada kecepatan sudut didapatkan dari turunan sudut aktual sendi 2 terhadap waktu. Pada detik ke-0 hingga detik ke-1, mengalami kenaikan hingga mencapai 0,9 rad/s. Pada detik ke-2 hingga detik ke-5, kecepatan sudut naik dari 0 rad/s menjadi 0,2 rad/s. Pada detik ke-5 hingga detik ke-10, kecepatan sudut stabil pada 0 rad/s.

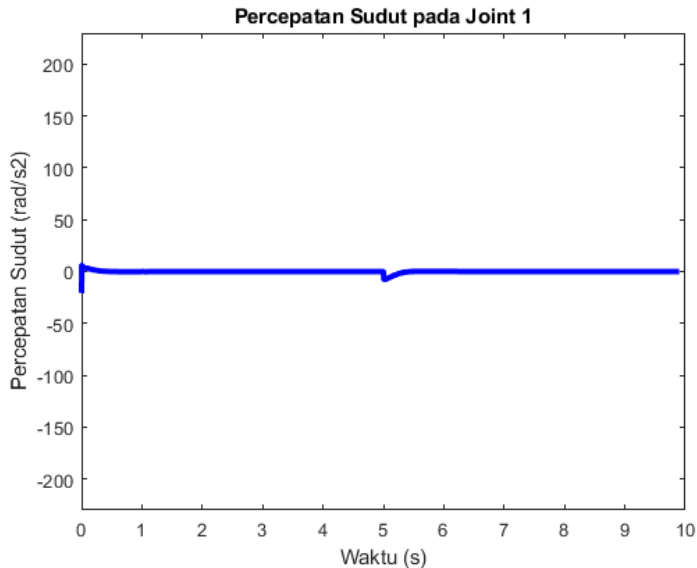




Gambar 4.59 Kecepatan sudut sendi 3 Menggunakan *Automated Tuning* PID pada *Trajectory* Bebas

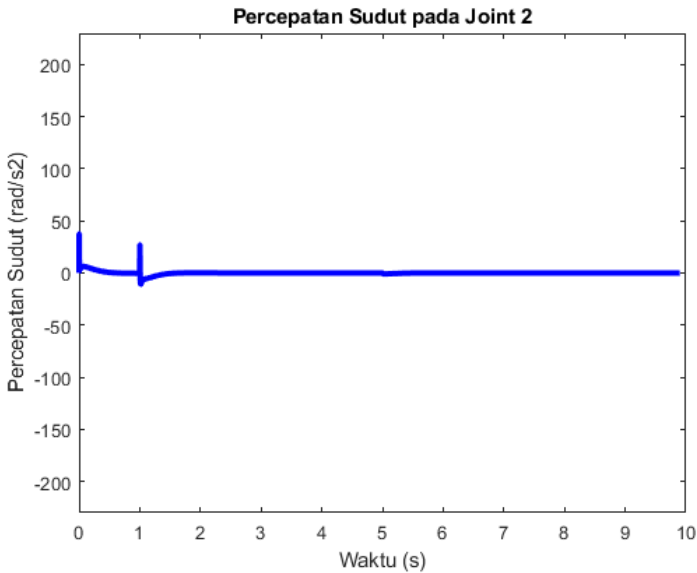
Hasil simulasi kecepatan sudut *Automated tuning* PID pada sendi 3 ditunjukkan pada gambar 4.59. Pada grafik tersebut menunjukkan kecepatan sudut yang ditempuh oleh sendi 3. Hasil simulasi pada kecepatan sudut didapatkan dari turunan sudut aktual sendi 3 terhadap waktu. Pada detik ke-0 mengalami kenaikan mencapai 4,5 rad/s. Pada detik ke-1 turun dari 4 rad/s hingga -0,5 rad/s. Kecepatan sudut yang stabil dari detik ke-2 hingga detik ke-10.

#### 4.4.1.2 Percepatan Sudut pada tiap Sendi



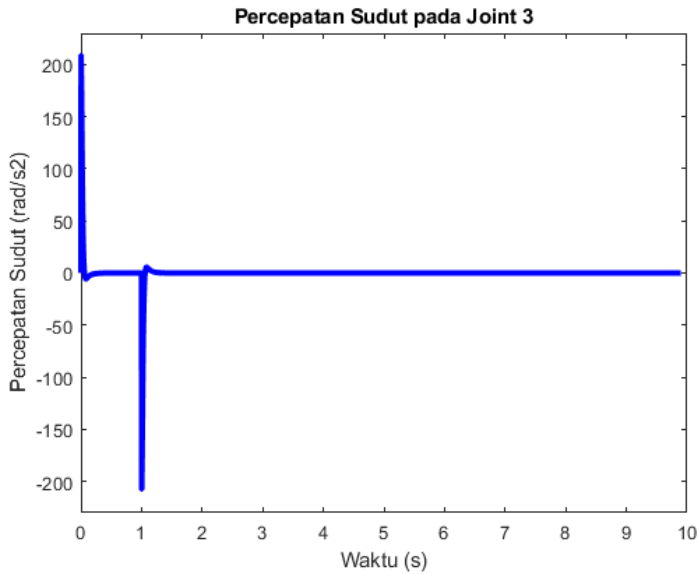
Gambar 4.60 Percepatan Sudut Sendi 1 menggunakan *Automated Tuning* PID pada *Trajectory* Bebas

Hasil simulasi percepatan sudut *Automated tuning* PID pada sendi 1 ditunjukkan pada gambar 4.60. Pada grafik tersebut menunjukkan percepatan sudut yang ditempuh oleh sendi 1. Hasil simulasi pada percepatan sudut didapatkan dari turunan kecepatan sudut sendi 1 terhadap waktu. Pada detik ke-0, mengalami penurunan hingga  $-20 \text{ rad/s}^2$ , kemudian mengalami kenaikan hingga mencapai  $3 \text{ rad/s}^2$  lalu turun lagi hingga mencapai  $0 \text{ rad/s}^2$ . Pada detik ke-0 hingga detik ke-5 stabil pada  $0 \text{ rad/s}^2$ .



Gambar 4.61 Percepatan Sudut Sendi 2 menggunakan *Automated Tuning* PID pada Trajectory Bebas

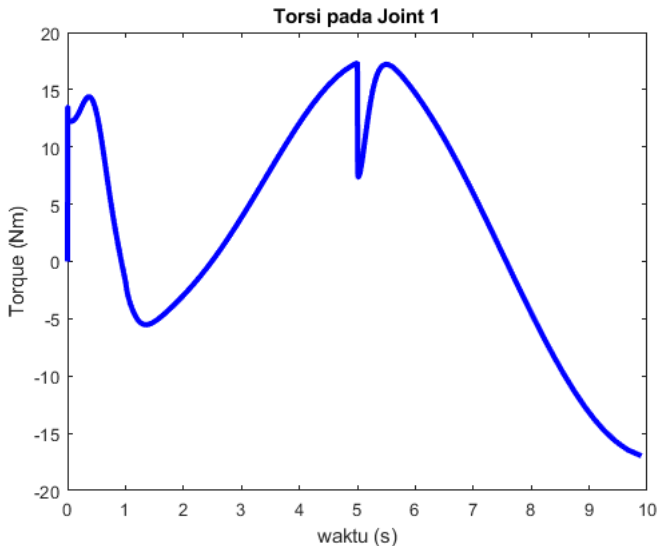
Hasil simulasi percepatan sudut *Automated tuning* PID pada sendi 2 ditunjukkan pada gambar 4.61. Pada grafik tersebut menunjukkan percepatan sudut yang ditempuh oleh sendi 2. Hasil simulasi pada percepatan sudut didapatkan dari turunan kecepatan sudut sendi 2 terhadap waktu. Pada detik ke-0, mengalami kenaikan hingga  $37 \text{ rad/s}^2$ , kemudian mengalami penurunan hingga mencapai  $20 \text{ rad/s}^2$ . Pada detik ke-1 mengalami penurunan hingga  $-10 \text{ rad/s}^2$ , kemudian kembali ke  $0 \text{ rad/s}^2$  dan stabil hingga detik ke-10.



Gambar 4.62 Percepatan Sudut Sendi 3 menggunakan *Automated Tuning* PID pada *Trajectory* Bebas

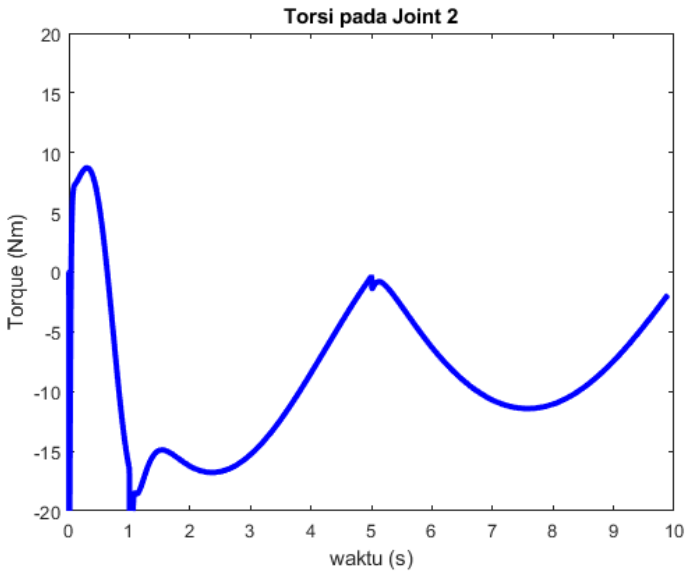
Hasil simulasi percepatan sudut *Automated tuning* PID pada sendi 3 ditunjukkan pada gambar 4.62. Pada grafik tersebut menunjukkan percepatan sudut yang ditempuh oleh sendi 3. Hasil simulasi pada percepatan sudut didapatkan dari turunan kecepatan sudut sendi 3 terhadap waktu. Pada detik ke-0, mengalami kenaikan hingga  $200 \text{ rad/s}^2$ , kemudian mengalami penurunan hingga mencapai  $0 \text{ rad/s}^2$ . Pada detik ke-1 mengalami penurunan hingga  $-200 \text{ rad/s}^2$ , kemudian kembali ke  $0 \text{ rad/s}^2$ . Pada detik ke-2 hingga detik ke-10 stabil pada  $0 \text{ rad/s}^2$ .

#### 4.4.1.3 Torsi pada tiap Sendi



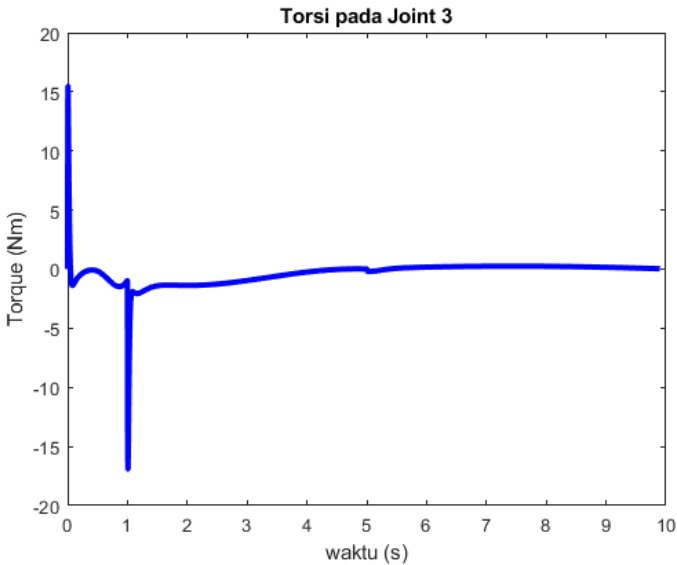
Gambar 4.63 Torsi Sendi 1 menggunakan *Automated Tuning* PID pada *Trajectory* Bebas

Hasil simulasi torsi *Automated tuning* PID pada sendi 1 ditunjukkan pada gambar 4.63. Pada grafik tersebut menunjukkan torsi yang ditempuh oleh sendi 1. Hasil simulasi besarnya torsi dipengaruhi sudut, kecepatan sudut, dan percepatan sudut pada sendi 1. Pada detik ke-0, mengalami kenaikan hingga 12 Nm. Pada detik ke-0,3 naik menjadi 14 Nm, kemudian turun pada detik ke-1 menjadi -5,5 Nm. Pada detik ke-5 mengalami kenaikan menjadi 17 Nm, kemudian turun menjadi 7,4 Nm. Saat menuju detik ke-6 mengalami kenaikan menjadi 17 Nm. Pada detik ke-10 mengalami penurunan hingga mencapai -16 Nm.



Gambar 4.64 Torsi Sendi 2 menggunakan *Automated Tuning* PID pada *Trajectory* Bebas

Hasil simulasi torsi *Automated tuning* PID pada sendi 2 ditunjukkan pada gambar 4.64. Pada grafik tersebut menunjukkan torsi yang ditempuh oleh sendi 2. Hasil simulasi besarnya torsi dipengaruhi sudut, kecepatan sudut, dan percepatan sudut pada sendi 2. Pada detik ke-0 mengalami penurunan hingga -20 Nm, kemudian mengalami kenaikan mencapai 8,5 Nm. Saat menuju detik ke-1, mengalami penurunan hingga mencapai -20 Nm, kemudian kembali naik menjadi -15 Nm. Pada detik ke-2,5 kembali mengalami penurunan mencapai -16 Nm, kemudian naik hingga -0,8 Nm. Pada detik ke-7 mengalami penurunan hingga -11 Nm, kemudian mengalami kenaikan pada detik ke-10 mencapai -1 Nm.



Gambar 4.65 Torsi Sendi 3 menggunakan *Automated Tuning* PID pada *Trajectory* Bebas

Hasil simulasi torsi *Automated tuning* PID pada sendi 3 ditunjukkan pada gambar 4.65. Pada grafik tersebut menunjukkan torsi yang ditempuh oleh sendi 3. Hasil simulasi besarnya torsi dipengaruhi sudut, kecepatan sudut, dan percepatan sudut pada sendi 3. Pada detik ke-0, mengalami kenaikan hingga -15 Nm. Saat detik ke-1, mengalami penurunan hingga mencapai -15 Nm, kemudian kembali ke 0 Nm. Pada detik ke-2 hingga -10 Nm, torsi berada pada nilai -0,2 Nm.

#### 4.4.2 Kontrol PID Menggunakan Metode Ziegler-Nichols

Simulasi pergerakan lengan robot dengan kontrol PID dilakukan dengan metode ziegler-nichol. Metode ini digunakan untuk mendapatkan parameter-parameter seperti  $K_p$ ,  $K_i$ , dan  $K_d$ . Parameter tersebut selanjutnya dilakukan tuning pada diagram blok PID. Diagram blok PID kemudian disambungkan dengan motor DC dan lengan robot. Berikut parameter-parameter yang didapatkan saat simulasi menggunakan MATLAB.

Tabel 4.6 : Tabel Tuning PID Metode Ziegler-Nichols Lingkaran

	Nilai $K_p$	Nilai $K_i$	Nilai $K_d$
Sendi 1	140.991	28795.2754	0.17258
Sendi 2	1014.2902	220727.3244	1.1652
Sendi 3	190.5715	37687.3572	0.24091

Pada saat dilakukan simulasi, tuning dengan metode ziegler-nichols yang dimasukkan ke dalam blok diagram PID tidak bisa menstabilkan sistem. Akibatnya lengan robot bergerak osilasi dan tidak beraturan.



## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Kesimpulan dari penelitian ini sebagai berikut

1. Pada penelitian ini dianalisa dinamika lengan robot yang memiliki 3 derajat kebebasan. Simulasi dinamis menggunakan Simscape Multibody. Blok diagram lengan robot digabungkan dengan blok diagram PID agar model lengan robot tersebut dapat dikontrol sehingga dapat berjalan sesuai *trajectory* yang direncanakan.
2. Untuk membuat Kontrol PID menggunakan *automated tuning*, terlebih dahulu *plant* harus linier agar algoritma pada *automated tuning* berjalan dan *plant* dapat dikontrol menggunakan kontrol PID. Hasil dari *automated tuning* berupa nilai Kp, Ki, dan Kd kemudian dimasukkan ke blok diagram PID, selanjutnya PID akan mengontrol pergerakan dari lengan robot. Lengan robot digerakkan sesuai *trajectory* setengah lingkaran, lingkaran, dan bebas. Hasil dari pergerakan lengan robot menggunakan *automated tuning* dibandingkan Ziegler-Nichols mempunyai persen *overshoot* yang lebih kecil.
3. Untuk membuat Kontrol PID menggunakan metode Ziegler-Nichols terlebih dahulu mencari *transfer function* dari tiap-tiap motor. Dari *transfer function* tersebut, dicari nilai Kp, Ki, dan Kd menggunakan aturan-aturan yang telah ditetapkan oleh Ziegler-Nichols. Hasil dari Metode Ziegler Nichols berupa nilai Kp, Ki, dan Kd kemudian dimasukkan ke blok diagram PID, selanjutnya PID akan mengontrol pergerakan dari lengan robot. Lengan robot digerakkan sesuai *trajectory* setengah lingkaran, lingkaran, dan bebas. Hasil dari pergerakan lengan robot menggunakan Ziegler-Nichols

dibandingkan *automated tuning* mempunyai *rise time* dan *settling time* yang lebih cepat.

4. Analisis dinamika dari pergerakan lengan robot bergantung pada sudut, kecepatan sudut, percepatan sudut, dan torsi dari masing-masing sendi. Hasil dari simulasi menunjukkan bahwa nilai dari kecepatan sudut diturunkan dari sudut terhadap waktu. Nilai dari percepatan sudut diturunkan dari kecepatan sudut terhadap waktu. Besarnya torsi dipengaruhi oleh besarnya sudut, kecepatan sudut, dan percepatan sudut pada masing-masing sendi.

## 5.2 **Saran**

Saran dari penelitian ini adalah

1. Dibutuhkan sistem kontrol yang baik agar kesalahan yang dilakukan oleh pengendali sekecil mungkin dan tingkat kepresisian yang tinggi.
2. Diperlukan pemodelan motor yang sesuai dengan spesifikasi yang digunakan.

## DAFTAR PUSTAKA

- Agbaraji, E., Inyiyama, H., Okezie, C., 2017. *Dynamics Modeling of 3 DOF Articulated Manipulator Based on Independent Joint Scheme*. Physcal Sains International Journal.
- Bilal, dkk. 2018. *Design and Control of 6 DOF Robotic Manipulator*. University of Engineering and Technology Lahore.
- CTMS, Mei. 2019. DC Motor Position: Simulink Model. <http://ctms.engin.umich.edu/CTMS/index.php?example=MotorPosition&section=SimulinkModeling>.
- Kabir, Usman dkk. 2019. *Performance Analysis Of Pid, Pd And Fuzzy Controllers For Position Control Of 3-Dof Robot Manipulator*. Zaria Journal of Electrical Engineering Technology
- Kharidge A., Jianbiao D., Zhang Y. 2016. *Performance Study of PID and Fuzzy Controllers for Position Control of 6 DOF arm Manipulator with Various Defuzzification Strategies*. MATEC Web of Conferences.
- Lafmejani, HS., Zarabadipour, H. 2014. *Modeling, Simulation, Position Control of 3 DOF Articulated Manipulator*. Indonesian Journal of Electrical Engineering and Informatics.
- Manjare S., Thomas, M. 2016. *Modeling of Multi-DOF Robotic Manipulator using sim-Mechanics Software*. Indian Journal of Science and Technology.
- Nguyen VT., Dumitriu, Stroe. 2017. *Controlling the Motion of a Planar 3-DOF Manipulator Using PID Controllers*. INCAS BULLETIN.
- Niku, B Saeed. 2001. *Introduction to Robotics Analysis, Systems, Application*. New Jersey: Prentice Hall, Inc.
- Nise, N. S., 2011. *Control Systems Engineering, sixth Edition*. John Wiley & Sons, Inc.

- Nurhakim, Rahmad Hardiansah. 2010. *Sistem Kendali Gerak Continous Path Tracking dengan menggunakan Cubic Trajectory Planning Pada Robot Manipulator 4 DOF*. Universitas Indonesia.
- Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik edisi kedua*. Jakarta: Penerbit Erlangga.
- Qahtani HM., Mohammed, AA. Sunar, 2016. *Dynamics and Control of a Robotic Arm Having Four Link*. Arab J Sci Eng
- Singh TP., Suresh P, Chandan S., 2017. *Forward and Inverse Kinematics Analysis of Robotic Manipulator*. International Research Journal of Engineering and Technology.
- Tsai, Lung-wen. 1999. *Robot Analysis: The Mechanics of Serial and Parallel Manipulator*. New York: Jhon Wiley & Sons, Inc.

## LAMPIRAN

### Matlab

```
1 function [x, y, z] = fcn(t) 1 function [x, y, z] = fcn(t)
2                               2
3 - T = 10;                    3 - T = 10;
4 - f = 1/T;                    4 - f = 1/T;
5                               5
6 - x=0.65*cos(pi*f*t);        6 - x=0.65*cos(2*pi*f*t);
7 - y=0.65*sin(pi*f*t);        7 - y=0.65*sin(2*pi*f*t);
8 - z=0.5;                      8 - z=0.5;
9                               9
```

Gambar A.1 : Coding Untuk Menggambar *Trajectory*

```
1 function [theta1, theta2, theta3] = fcn(x, y, z)
2 % DH parameter
3 a1=0.105; d1=0.173; alp1=90;
4 a2=0.7; d2=0; alp2=0;
5 a3=0.31; d3=0; alp3=0;
6
7 %% posisi end effector
8 Pos = [x, y, z]; %dari input
9 Tend = trvec2tform(Pos);
10 % Tend = [ux vx wx qx; %qx adalah posisi ee terhadap base
11 % uy vy wy qy; %qy adalah posisi ee terhadap base
12 % uz vz wz qz; %qz adalah posisi ee terhadap base
13 % 0 0 0 1];
14 %% mencari theta
15 %theta 1
16 theta1 = atan2(Tend(2,4),Tend(1,4)); %dalam radian
17 %theta 2
18 r2 = Tend(3,4)-d1;
19 r1 = sqrt(Tend(1,4)^2+Tend(2,4)^2)-a1;
20 r3 = sqrt(r2^2+r1^2);
21 phi2 = atan2(r2,r1);
22 phi1 = acos((a2^2+r3^2-a3^2)/(2*a2*r3));
23 theta2 = phi2-phi1; %dalam radian
24 %theta 3
25 phi3 = acos((a2^2+a3^2-r3^2)/(2*a2*a3));
26 theta3 = deg2rad(180)-phi3; %dalam radian
```

Gambar A.2 : Coding Untuk Membuat *Inverse Kinematics*

```

1 - J = 0.5;
2 - b = 0.1;
3 - Kt = 0.5;
4 - Ke = 0.5;
5 - R = 10.5;
6 - L = 0.4;
7 - s = tf('s');
8 - P_motor = Kt / ((L * J * s^2) + (L * b + R * J) * s) + (R * b + Kt * Ke)

```

---

```

Command Window
>> transferfunction

P_motor =

          0.5
-----
0.2 s^2 + 5.29 s + 1.3

Continuous-time transfer function.

```

Gambar A.3 : *Transfer Function* pada motor 1

```

1 - J = 0.01;
2 - b = 0.1;
3 - Kt = 0.1;
4 - Ke = 0.1;
5 - R = 3.5;
6 - L = 0.05;
7 - s = tf('s');
8 - P_motor = Kt / ((L * J * s^2) + ((L * b + R * J) * s) + (R * b + Kt * Ke))

```

---

```

Command Window
>> transferfunction

P_motor =

          0.1
-----
0.0005 s^2 + 0.04 s + 0.36

Continuous-time transfer function.

```

Gambar A.4 : *Transfer Function* pada motor 2

```
1 - J = 0.003;
2 - b = 0.01;
3 - Kt = 0.01;
4 - Ke = 0.01;
5 - R = 2.5;
6 - L = 0.001;
7 - s = tf('s');
8 - P_motor = Kt / ((L * J * s^2) + (L * b + R * J) * s + (R * b + Kt * Ke))

Command Window
>> transferfunction

P_motor =

          0.01
-----
3e-06 s^2 + 0.00751 s + 0.0251

Continuous-time transfer function.
```

Gambar A.5 : *Transfer Function* pada motor 3

```

1 %Tuning PID metode ziegler-nichols
2 %close all; clear all; clc;
3 s = tf('s');
4 sys = 1/(8.4*10^-6*s^2+2.52*s+4.282);
5
6 %obtain step response of system
7 [y,t] = step(sys);
8 plot(t,y,'linewidth',2);grid on; xlabel('Time(s)');ylabel('Amplitude');
9 title('open loop response');
10
11 %obtain inflection point and draw tangent
12 yp = diff(y);
13 ypp = diff(yp,2);
14 %find the root using FZERO
15 t_infl = fzero(@(T) interp1(t(2:end-1),ypp,T,'linear','extrap'),0);
16 y_infl = interp1(t,y,t_infl,'linear');
17 hold on;
18 plot(t_infl,y_infl,'ro');
19
20 %Draw tangent at inflection point
21 h = mean(diff(t));
22 dy = gradient(y,h);
23 [~,idx] = max(dy);
24 b = [t([idx-1,idx+1]) ones(2,1)] \ y([idx-1,idx+1]);
25 tv = [-b(2)/b(1); (1-b(2))/b(1)];
26 f = [tv ones(2,1)] * b;
27
28 plot(tv,f,'r','linewidth',1.5)
29 ylim([0 max(y)]);
30
31 %%finding T and L
32 L = tv(1);
33 T = tv(2) - tv(1);
34
35 %PID parameters
36 a = L/T;
37 Kp = 1.2/a;
38 Ti = 2*L;
39 Td = L/2;
40 Ki = Kp/Ti;
41 Kd = Kp*Td;
42 disp('PID parameters')
43 disp(['Kp = ' num2str(Kp), ' Ti = ' num2str(Ti), ' Td = ' num2str(Td), ' Ki = ' num2str(Ki), ' Kd = ' num2str(Kd)])
44
45 %cont = Kp(1 + 1/(s*Ti) * s*Td);
46 cont = Kp + Kp/(s*Ti) + Kp*Td*s;
47 cl_sys = feedback(cont*sys,1);
48 t = [0:0.01:3];
49 [yc,tc] = step(cl_sys,1);
50 figure;
51 plot(tc,yc,'linewidth',2);xlabel('waktu (s)');ylabel('amplitudo');
52 title('Ziegler-Nichol Closed Loop Response');
53 grid on;

```

Gambar A.6 Metode Ziegler-Nichols pada Motor 1



```

1 %Tuning PID metode ziegler-nichols
2 %close all; clear all; clc;
3 s = tf('s');
4 sys = 1/(6*10^-6*s^2+1.8*s+1.282);
5
6 %obtain step response of system
7 [y,t] = step(sys);
8 plot(t,y,'linewidth',2);grid on; xlabel('Time (s)');ylabel('Amplitude');
9 title('open loop response');
10
11 %obtain inflection point and draw tangent
12 yp = diff(y);
13 ypp = diff(yp,2);
14 %find the root using FZERO
15 t_infl = fzero(@(T) interp1(t(2:end-1),ypp,T,'linear','extrap'),0);
16 y_infl = interp1(t,y,t_infl,'linear');
17 hold on;
18 plot(t_infl,y_infl,'ro');
19
20 %Draw tangent at inflection point
21 h = mean(diff(t));
22 dy = gradient(y,h);
23 [~,idx] = max(dy);
24 b = [t([idx-1,idx+1]) ones(2,1)] \ y([idx-1,idx+1]);
25 tv = [-b(2)/b(1) (1-b(2))/b(1)];
26 f = [tv ones(2,1)] * b;
27
28 plot(tv,f,'r','linewidth',1.5);
29 ylim([0 max(y)]);
30
31 %%finding T and L
32 L = tv(1);
33 T = tv(2) - tv(1);
34
35 %PID parameters
36 a = L/T;
37 Kp = 1.2/a;
38 Ti = 2*L;
39 Td = L/2;
40 Ki = Kp/Ti;
41 Kd = Kp*Td;
42 disp('PID parameters')
43 disp(['Kp = ' num2str(Kp), ' Ti = ' num2str(Ti), ' Td = ' num2str(Td), ' Ki = ' num2str(Ki), ' Rd = ' num2str(Kd)])
44
45 %cont = Kp(1 + 1/(s*Ti) + s*Td);
46 cont = Kp + Kp/(s*Ti) + Kp*Td*s;
47 cl_sys = feedback(cont*sys,1);
48 t = [0:0.01:3];
49 [yc,tc] = step(cl_sys,1);
50 figure;
51 plot(tc,yc,'linewidth',2);xlabel('waktu (s)');ylabel('amplitudo');
52 title('Ziegler-Nichol Closed Loop Response');
53 grid on;

```

Gambar A.7 Metode Ziegler-Nichols pada Motor 2

```

1 %Tuning PID metode ziegler-nichols
2 %close all; clear all; clc;
3 s = tf('s');
4 sys = 1/(1.2*10^-6*s^2+0.36*s+4.282);
5
6 %obtain step response of system
7 [y,t] = step(sys);
8 plot(t,y,'linewidth',2);grid on; xlabel('Time(s)');ylabel('Amplitude');
9 title('open loop response');
10
11 %obtain inflection point and draw tangent
12 yp = diff(y);
13 ypp = diff(yp);
14 %find the root using FZERO
15 t_infl = fzero(@(T) interp1(t(2:end-1),ypp,T,'linear','extrap'),0);
16 y_infl = interp1(t,y,t_infl,'linear');
17 hold on;
18 plot(t_infl,y_infl,'ro');
19
20 %Draw tangent at inflection point
21 h = mean(diff(t));
22 dy = gradient(y,h);
23 [~,idx] = max(dy);
24 b = [t([idx-1,idx+1]) ones(2,1)] \ y([idx-1,idx+1]);
25 tv = [-b(2)/b(1); (1-b(2))/b(1)];
26 f = [tv ones(2,1)] * b;
27
28 plot(tv,f,'g','linewidth',1.5)
29 ylim([0 max(y)]);
30
31 %%finding T and L
32 L = tv(1);
33 T = tv(2) - tv(1);
34
35 %PID parameters
36 a = L/T;
37 Kp = 1.2/a;
38 Ti = 2*L;
39 Td = L/2;
40 Ki = Kp/Ti;
41 Kd = Kp*Td;
42 disp('PID parameters')
43 disp(['Kp = ' num2str(Kp), ' Ti = ' num2str(Ti), ' Td = ' num2str(Td), ' Ki = ' num2str(Ki), ' Kd = ' num2str(Kd)])
44
45 %cont = Kp(1 + 1/(s*Ti) * s*Td);
46 cont = Kp + Kp/(s*Ti) + Kp*Td*s;
47 cl_sys = feedback(cont*sys,1);
48 t = [0:0.01:3];
49 [yc,tc] = step(cl_sys,1);
50 figure;
51 plot(tc,yc,'linewidth',2);xlabel('waktu (s)');ylabel('amplitudo');
52 title('Ziegler-Nichol Closed Loop Response');
53 grid on;

```

Gambar A.8 Metode Ziegler-Nichols pada Motor 3