



**UNIVERSITAS INDONESIA**

**RANCANG BANGUN SISTEM TELEOPERASI BILATERAL  
2 DOF MENGGUNAKAN DYNAMIXEL AX-12+ DENGAN  
PENGENDALIAN PARAMETER TORSI UNTUK  
MENINGKATKAN SENSASI HAPTIC**

**SKRIPSI**

**FAIZ SALEH BALWEEL  
0706267686**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
OKTOBER 2011**



**UNIVERSITAS INDONESIA**

**RANCANG BANGUN SISTEM TELEOPERASI BILATERAL  
2 DOF MENGGUNAKAN DYNAMIXEL AX-12+ DENGAN  
PENGENDALIAN PARAMETER TORSI UNTUK  
MENINGKATKAN SENSASI HAPTIC**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar  
Sarjana Teknik**

**FAIZ SALEH BALWEEL  
0706267686**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
PEMINATAN KENDALI  
DEPOK  
OKTOBER 2011**

## HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,  
dan semua sumber baik yang dikutip maupun dirujuk  
telah saya nyatakan dengan benar.**

**Nama : Faiz Saleh Balweel**

**NPM : 0706267686**

**Tanda Tangan : **

**Tanggal : 6 Oktober 2011**

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh

Nama : Faiz Saleh Balweel  
NPM : 0706267686  
Program Studi : Teknik Elektro  
Judul Skripsi : Rancang Bangun Sistem Teleoperasi Bilateral 2 DOF Menggunakan Dynamixel AX-12+ Dengan Pengendalian Parameter Torsi Untuk Meningkatkan Sensasi Haptic

**Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia**

Pembimbing : Dr. Abdul Muis, ST., M.Eng (  )

Penguji : Dr. Abdul Halim, M.Eng (  )

Penguji : Dr. Ir. Wahidin Wahab, M.Sc. (  )

Ditetapkan di : Depok

Tanggal : 12 Januari 2012

## KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, proses penulisan skripsi ini dapat terselesaikan. Penulisan skripsi ini dilakukan dalam rangka memenuhi persyaratan dari mata kuliah Skripsi yang terdapat dalam kurikulum program studi Teknik Elektro Universitas Indonesia. Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan skripsi, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

- (1) Dr. Abdul Muis, ST., M.Eng , selaku dosen pembimbing, serta dosen-dosen lainnya, yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
- (2) Orang tua dan keluarga saya yang telah memberikan bantuan berupa dukungan material dan moral;
- (3) Para Sahabat dan teman-teman yang telah banyak membantu saya dalam menyelesaikan penyusunan skripsi ini;

Akhir kata, saya berharap agar Tuhan Yang Maha Esa berkenan membalas segala kebaikan dari semua pihak yang telah membantu. Semoga skripsi ini dapat memberikan manfaat bagi pengembangan ilmu.

Depok, 6 Oktober 2011

Faiz Saleh Balweel

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI  
UNTUK KEPENTINGAN AKADEMIS**

---

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Faiz Saleh Balweel

NPM : 0706267686

Program Studi : Teknik Elektro

Departemen : Teknik Elektro

Fakultas : Teknik

Jenis karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

**RANCANG BANGUN SISTEM TELEOPERASI BILATERAL 2 DOF  
MENGUNAKAN DYNAMIXEL AX-12+ DENGAN PENGENDALIAN  
PARAMETER TORSI UNTUK MENINGKATKAN SENSASI HAPTIC**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis / pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 6 Oktober 2011

Yang menyatakan,



(Faiz Saleh Balweel)

## ABSTRAK

Nama : Faiz Saleh Balweel  
Program Studi : Teknik Elektro  
Judul : RANCANG BANGUN SISTEM TELEOPERASI BILATERAL 2 DOF MENGGUNAKAN DYNAMIXEL AX-12+ DENGAN PENGENDALIAN PARAMETER TORSI UNTUK MENINGKATKAN SENSASI HAPTIC

Sistem teleoperasi bilateral adalah sistem yang dapat melakukan suatu pekerjaan pada jarak jauh dan memungkinkan pengguna merasakan sensasi seakan-akan pengguna tersebut berada pada area yang jauh. Teleoperasi bilateral memungkinkan operator dapat secara langsung mengendalikan sebuah manipulator pada jarak tertentu. Untuk menunjang sistem teleoperasi bilateral maka diperlukan aktuator yang memiliki parameter-parameter yang dapat divariasikan. Dynamixel AX-12+ merupakan serial servo yang memiliki fitur-fitur yang dapat mendukung sistem teleoperasi bilateral seperti pembacaan posisi, pembacaan load, penyesuaian sudut distribusi torsi dan penentuan batas torsi maksimal. Parameter yang paling utama pada dynamixel AX-12+ yang akan diatur nilainya adalah sudut distribusi torsi. Parameter ini berguna untuk memvariasikan taraf keringanan pengguna dalam menggerakkan manipulator. Apabila sudut torsi diatur menjadi landai maka operator dapat dengan mudah menggerakkan manipulator. Sebaliknya, apabila sudut torsi diatur menjadi curam maka operator akan merasa berat dalam menggerakkan manipulator. Pengaturan sudut distribusi torsi ini harus dilakukan secara cermat karena apabila terjadi kesalahan maka dapat merusak gir pada aktuator. Selain itu, kesalahan pengaturan dapat berakibat sistem tidak bekerja secara baik. Perbedaan sistem yang memiliki algoritma penyesuaian torsi dengan sistem yang tidak memiliki algoritma penyesuaian torsi adalah taraf keringanan yang dihasilkan dengan tidak mengesampingkan efek *haptic* yang ada. Apabila sistem tidak menggunakan algoritma penyesuaian torsi secara otomatis maka manipulator dapat tetap dengan ringan digerakkan akan tetapi efek *haptic* yang dihasilkan kecil. Berbeda dengan sistem yang menggunakan algoritma penyesuaian torsi yang dapat menimbulkan efek *haptic* yang baik dengan manipulator yang tetap ringan saat digerakkan. Efek *haptic* ini juga dapat terlihat dari perbedaan posisi antar manipulator pada saat salah satu manipulator menyentuh benda. Perbedaan antar manipulator maksimal pada penelitian ini mencapai  $6.15^\circ$ .

Kata kunci:

Sistem teleoperasi, Sistem bilateral, Haptic, Penyesuaian torsi otomatis, 2-DOF, Dynamixel AX-12+

## ABSTRACT

Name : Faiz Saleh Balweel  
Study Program : Electrical Engineering  
Title : DESIGNING 2-DOF BILATERAL TELEOPERATION SYSTEM USING DYNAMIXEL AX-12+ WITH TORQUE PARAMETER ADJUSTMENT CONTROL TO IMPROVE HAPTIC SENSATION

Bilateral teleoperation system is a system that able to do any work in the remote area and allow user to experience the sensation as if the user is located in that area. This system allow user to control the manipulator directly at a certain distance. To support bilateral teleoperation system, it would require an actuator, which has parameters that can be manually modified. Dynamixel AX-12+ is a serial servo motor which has features that can support bilateral teleoperation system with their parameters such as position reading, load reading, torque angle and torque limit adjustment. The important parameter in dynamixel AX-12+ is to set the value of the torque angle distribution. This parameter is useful to varying the level of user convenience while moving the manipulator. If the torque angle value is set to become more sloping, the user can easily move the manipulator. Conversely, if the torque angle is set to become steeper, the user will feel heavy to move the manipulator. This torque angle arrangement must be done carefully because if there is a mistake in the arrangements, it can damage the internal gear of actuator. In addition, arrangements error can make the whole system troubled. The difference between the systems with torque adjustment algorithms and systems that do not have a torque adjustment algorithm are the level of user convenience and the sense of haptic effects. If the system does not use the torque adjustment algorithm, then the manipulator can still drive easily but there is no haptic effect. In contrast to the system that use torque adjustment algorithm, the manipulator can still drive easily with the sense of haptic effect. This haptic effect is also seen from the difference between manipulator's positions when slave manipulator touches one object. The maximum difference between manipulator in this research reached  $6.15^\circ$ .

Keyword:

Teleoperation system, Bilateral system, Haptic, Torque adjustment, Auto tuning torque, 2-DOF, Dynamixel



## DAFTAR ISI

SKRIPSI .....	i
HALAMAN PERNYATAAN ORISINALITAS .....	ii
HALAMAN PENGESAHAN .....	iii
KATA PENGANTAR .....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS .....	v
ABSTRAK .....	vi
ABSTRACT .....	vii
DAFTAR ISI .....	viii
DAFTAR GAMBAR .....	x
DAFTAR TABEL .....	xiii
DAFTAR ISTILAH .....	xiv
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Tujuan .....	2
1.3 Pembatasan Pembahasan Masalah .....	2
1.4 Metodologi Penelitian.....	2
1.5 Sistematika Penulisan .....	3
<b>BAB 2 SISTEM ROBOTIKA DAN TELEOPERASI BILATERAL.....</b>	<b>4</b>
2.1 Dasar Mengenai Sistem Otomatisasi dan Robotika .....	4
2.2 Dasar Mengenai Sistem Teleoperasi Bilateral .....	4
2.3 Serial Servo sebagai Aktuator Sistem .....	8
2.3.1 Gambaran Umum Dynamixel AX-12+ .....	9
2.3.2 Komunikasi Protokol Dynamixel AX-12+.....	11
2.4 Mikrokontroler ATmega 128 Sebagai Blok Pengendali .....	17
2.4.1 Komunikasi Serial dengan Menggunakan USART .....	19
2.5 <i>Serial Tool</i> Program.....	22
<b>BAB 3 PERANCANGAN SISTEM BILATERAL .....</b>	<b>23</b>
3.1 Perancangan Sistem Teleoperasi Bilateral.....	23
3.1.1 Perancangan Pengendali Sistem Teleoperasi Bilateral .....	24
3.1.2 Perancangan <i>Bandwidth</i> Sistem dengan <i>Low Pass Filter</i> .....	28
3.2 Pengaturan Torsi pada Aktuator Dynamixel AX-12+ .....	30
3.3 Perancangan Perangkat Keras .....	33
3.3.1 Catu Daya dan Penampang .....	35

3.4 Perancangan Perangkat Lunak .....	36
3.4.1 Pengaturan <i>Baud-rate</i> dan waktu cuplik pada Mikrokontroler .....	37
3.4.2 Diagram Alur Program.....	39
3.4.2.1 Pengiriman dan Penerimaan Paket Data .....	39
3.4.2.2 Pembacaan Load dan Pemisahan Nilai Load dan Arah .....	40
3.4.2.3 Auto-Tuning Torque Angle yang akan diterapkan pada sistem....	42
<b>BAB 4 HASIL PENGUJIAN DAN ANALISIS.....</b>	<b>44</b>
4.2 Penentuan Nilai Konstanta $K_p$ dan $K_v$ pada Sistem Bilateral .....	44
4.3 Penentuan Nilai Frekuensi Cut-off pada <i>Low-Pass Filter</i> .....	50
4.4 Penentuan Nilai Parameter <i>Force</i> pada Sistem Bilateral.....	54
4.5 Variasi Nilai Sudut Distribusi Torsi pada Sistem Bilateral .....	59
4.6 <i>Auto-tuning torque angle</i> berdasarkan perbedaan posisi .....	70
<b>KESIMPULAN .....</b>	<b>82</b>
<b>DAFTAR ACUAN .....</b>	<b>83</b>
<b>DAFTAR PUSTAKA.....</b>	<b>84</b>
<b>LAMPIRAN .....</b>	<b>85</b>

## DAFTAR GAMBAR

Gambar 1.1 <i>T-52 Enryu Rescue Dragon tele-rescue robot</i> .....	1
Gambar 2.1 Sistem teleoperasi bilateral .....	5
Gambar 2.2 Tujuan sistem teleoperasi bilateral .....	6
Gambar 2.3 Sistem bilateral 2 kanal .....	7
Gambar 2.4 Sistem Bilateral 4 kanal .....	7
Gambar 2.5 Dynamixel AX-12+ .....	8
Gambar 2.6 Blok diagram pada dynamixel AX-12+.....	9
Gambar 2.7 Konfigurasi pin dynamixel AX-12+.....	10
Gambar 2.8 Konfigurasi koneksi secara <i>daisy-chain</i> .....	11
Gambar 2.9 Mode komunikasi <i>half-duplex</i> dynamixel.....	11
Gambar 2.10 Paket instruksi dan paket status.....	12
Gambar 2.11 Format paket instruksi.....	12
Gambar 2.12 Format paket status .....	13
Gambar 2.13 Grafik pengaturan distribusi torsi pada dynamixel.....	15
Gambar 2.14 Area operasi pergerakan dynamixel AX-12+.....	16
Gambar 2.15 Teleoperasi bilateral dengan blok pengendali .....	17
Gambar 2.16 Modul ATmega 128.....	18
Gambar 2.17 Konfigurasi pin ATmega 128.....	18
Gambar 2.18 Serial tool program .....	22
Gambar 3.1 Blok diagram sistem teleoperasi bilateral 4 kanal .....	24
Gambar 3.2 Arah gaya yang bekerja pada sistem teleoperasi bilateral .....	25
Gambar 3.3 Blok Diagram sistem bilateral dengan pengendali .....	27
Gambar 3.4 Diagram blok fungsi alih low-pass filter pada sistem .....	29
Gambar 3.5 Diagram Bode untuk Low-pass Filter pada sistem.....	30
Gambar 3.6 Parameter yang digunakan pada Distribusi Torsi.....	30
Gambar 3.7 Kemiringan sudut yang dibentuk oleh distribusi torsi .....	31
Gambar 3.8 Variasi yang menghasilkan hasil yang sama pada distribusi torsi ....	32
Gambar 3.9 Variasi sudut yang diperoleh dengan mengubah nilai <i>slope</i> .....	32
Gambar 3.10 Komunikasi sistem bilateral .....	34
Gambar 3.11 Penampang pada sistem bilateral yang telah dibuat .....	35
Gambar 3.12 Diagram alur transfer data antara aktuator dan pengendali .....	39
Gambar 3.13 Diagram alur pembacaan beban pada aktuator.....	41
Gambar 3.14 Algoritma <i>Auto-Tuning Torque Angle</i> .....	43
Gambar 4.1 Grafik posisi master dan slave saat nilai Kp 50 .....	45
Gambar 4.2 Grafik perbedaan posisi master dan slave saat nilai Kp 50.....	45
Gambar 4.3 Grafik posisi master slave saat nilai Kp 190.....	46
Gambar 4.4 Grafik perbedaan posisi master dan slave saat nilai Kp 190.....	47
Gambar 4.5 Grafik posisi master dan slave saat nilai Kp 210 .....	48
Gambar 4.6 Grafik perbedaan posisi master dan slave saat nilai Kp 210.....	48
Gambar 4.7 Grafik posisi master dan slave saat nilai Kp 175 .....	49
Gambar 4.8 Grafik perbedaan posisi master dan slave saat nilai Kp 175.....	50
Gambar 4.9 Respon sistem saat menggunakan frekuensi cut-off sebesar 300 rad/s .....	51
Gambar 4.10 Respon sistem saat menggunakan frekuensi cut-off sebesar 1000 rad/s.....	51

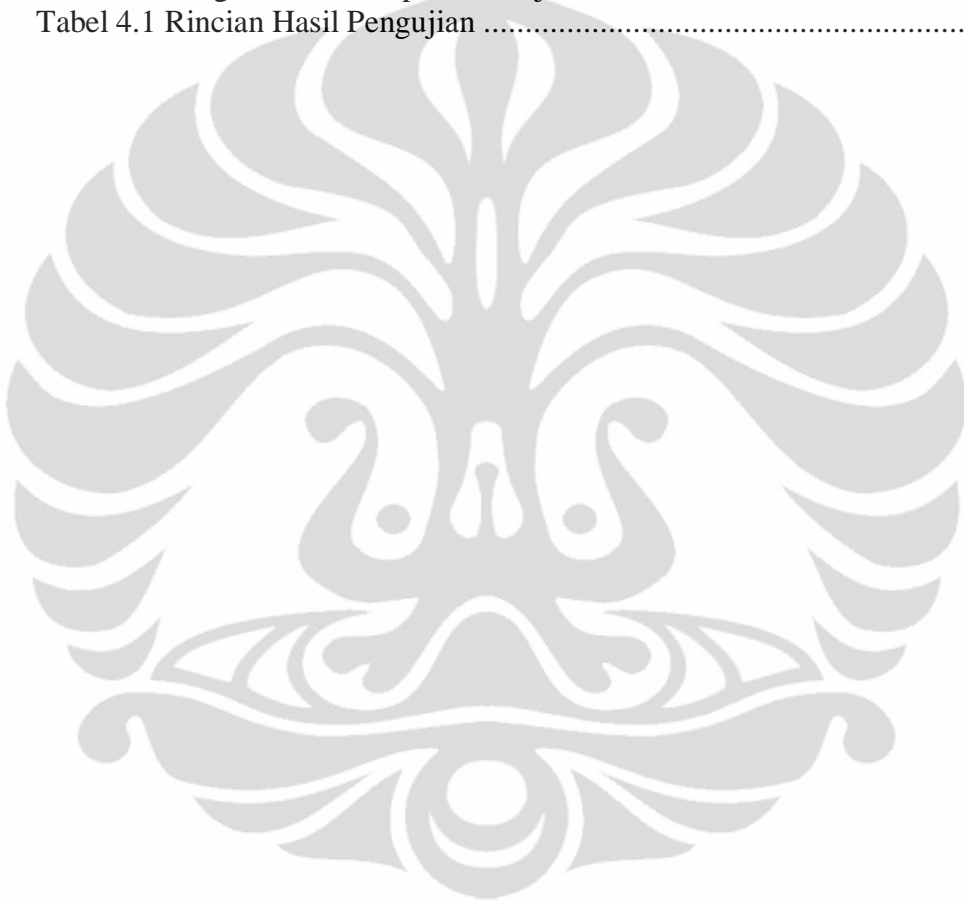
Gambar 4.11 Respon sistem saat menggunakan frekuensi cut-off sebesar 1257 rad/s.....	52
Gambar 4.12 Respon sistem saat menggunakan frekuensi cut-off sebesar 1300 rad/s.....	53
Gambar 4.13 Grafik posisi master dan slave tanpa parameter <i>force</i> .....	54
Gambar 4.14 Grafik perbedaan posisi master dan slave pada sistem.....	55
Gambar 4.15 Grafik posisi master dan slave dengan parameter <i>force</i> .....	55
Gambar 4.16 Grafik perbedaan posisi master dan slave pada sistem.....	56
Gambar 4.17 Grafik pembacaan beban tanpa parameter <i>force</i> .....	56
Gambar 4.18 Grafik pembacaan beban dengan parameter <i>force</i> .....	57
Gambar 4.19 Grafik posisi master dan slave dengan nilai parameter <i>force</i> 6.....	58
Gambar 4.20 Grafik perbedaan posisi master dan slave dengan nilai parameter <i>force</i> 6 .....	58
Gambar 4.21 Grafik posisi master dan slave saat parameter slope bernilai 32.....	59
Gambar 4.22 Grafik perbedaan posisi master dan slave pada sistem .....	60
Gambar 4.23 Grafik posisi master dan slave saat parameter slope bernilai 112 ...	60
Gambar 4.24 Grafik perbedaan posisi master dan slave pada sistem.....	61
Gambar 4.25 Grafik pembacaan beban pada saat pergerakan normal.....	62
Gambar 4.26 Grafik posisi master dan slave saat manipulator slave tertahan .....	62
Gambar 4.27 Grafik perbedaan posisi master dan slave pada sistem.....	63
Gambar 4.28 Grafik posisi master dan slave saat manipulator slave tertahan dengan sudut torsi landai.....	63
Gambar 4.29 Grafik perbedaan posisi master dan slave pada sistem.....	64
Gambar 4.30 Grafik pembacaan beban saat kondisi menyentuh benda dengan nilai slope 32 .....	65
Gambar 4.31 Grafik pembacaan beban saat kondisi menyentuh benda dengan nilai slope 112 .....	66
Gambar 4.32 Grafik pembacaan beban saat kondisi menyentuh benda dengan nilai slope 254 .....	66
Gambar 4.33 Grafik posisi master dan slave saat parameter slope slave dan master bernilai 238.....	67
Gambar 4.34 Grafik perbedaan posisi master dan slave pada sistem.....	68
Gambar 4.35 Grafik posisi master dan slave saat parameter slope master bernilai 238 .....	69
Gambar 4.36 Grafik perbedaan posisi master dan slave saat parameter slope master bernilai 238.....	69
Gambar 4.37 Grafik posisi master dan slave saat kondisi tanpa pengendali <i>auto-tuning</i> .....	70
Gambar 4.38 Grafik perbedaan posisi master dan slave pada sistem.....	71
Gambar 4.39 Grafik pembacaan beban Saat kondisi tanpa pengendali <i>auto-tuning</i> .....	72
Gambar 4.40 Grafik posisi master dan slave saat mode <i>Auto-Tuning</i> dengan nilai batas error 2.92° dengan laju perubahan 10% .....	73
Gambar 4.41 Perbedaan posisi master dan slave pada sistem.....	74
Gambar 4.42 Grafik posisi master dan slave saat mode <i>Auto-Tuning</i> dengan nilai batas error 2.92° dengan laju perubahan 100% .....	74
Gambar 4.43 Perbedaan posisi master dan slave pada sistem.....	75

Gambar 4.44 Grafik posisi master dan slave saat mode <i>Auto-Tuning</i> dengan nilai batas error $1.46^\circ$ dengan laju perubahan 100% .....	76
Gambar 4.45 Perbedaan posisi master dan slave pada sistem.....	76
Gambar 4.46 Penerapan sistem dengan menggunakan 2 DOF .....	77
Gambar 4.47 Penerapan sistem pada sendi pertama.....	78
Gambar 4.48 Penerapan sistem pada sendi kedua .....	78
Gambar 4.49 Perbedaan posisi antar manipulator sistem pada sendi pertama.....	79
Gambar 4.50 Perbedaan posisi antar manipulator sistem pada sendi kedua .....	79



## DAFTAR TABEL

Tabel 2.1 Spesifikasi umum dynamixel AX-12+ .....	10
Tabel 2.2 Tabel byte error pada paket status.....	13
Tabel 2.3 Tabel instruksi yang digunakan pada dynamixel.....	13
Tabel 2.4 Paket data pembacaan beban dynamixel .....	16
Tabel 2.5 Register UCSRA .....	20
Tabel 2.6 Register UCSRB .....	21
Tabel 3.1 Bit pada register TCCR .....	37
Tabel 3.2 Fungsi <i>Prescalar</i> pada kinerja timer .....	37
Tabel 4.1 Rincian Hasil Pengujian .....	80



## DAFTAR ISTILAH

USART	Universal Synchronous Asynchronous Receiver Transmitter
UBRR	16-bit register yang mengatur kecepatan komunikasi USART
UDR	<i>Sharing address I/O</i> yang digunakan untuk mengirim dan menerima data secara serial
UCSRA	Register untuk memonitor status yang dilakukan USART
RXC	Bit pada UCSRA untuk mengindikasi data serial telah diterima
TXC	Bit pada UCSRA untuk mengindikasi data serial telah dikirim
UDRE	Bit pada UCSRA untuk mengindikasi register UDR telah kosong
UCSRB	Register untuk menginisialisasi fungsi pada USART
TXEN	Bit pada UCSRB untuk mengaktifkan pin transmitter
RXEN	Bit pada UCSRB untuk mengaktifkan pin receiver
RX	Receiver
TX	Transmitter

# BAB 1 PENDAHULUAN

## 1.1 Latar Belakang

Saat ini teknologi telah berkembang pesat, termasuk didalamnya teknologi otomatisasi dan robotika. Robot pada periode sekarang telah mampu menggantikan pekerjaan manusia baik yang telah dikerjakan manusia secara individual maupun menggantikan sebuah pekerjaan yang membutuhkan sekelompok orang beserta peralatannya. Salah satu aplikasi dari teknologi ini adalah sistem teleoperasi. Sistem yang pertama kali dikenalkan oleh Ray Goertz ini menggunakan dua buah manipulator yang bergerak secara simultan. Sistem ini memungkinkan pengguna dapat secara langsung mengendalikan sebuah sistem robotika pada jarak tertentu. Aplikasi dari sistem ini banyak digunakan untuk berbagai macam aplikasi salah satunya adalah untuk membantu tim SAR dalam mengeksplorasi lokasi yang terpencil ataupun tidak memungkinkan untuk manusia dapat mengeksplorasi tanpa alat bantuan. Salah satu contohnya adalah T-52 *Enryu-Rescue Dragon* (Tmsuk, Japan) dan *Trauma-Pod* yang pertama kali dikenalkan di Universitas Washington.



Gambar 1.1 T-52 Enryu Rescue Dragon tele-rescue robot

Umumnya pada sistem teleoperasi terdapat dua buah manipulator yaitu manipulator *master* yang digunakan oleh pengguna dan manipulator *slave* yang berhubungan langsung pada lingkungan luar. Sistem ini dapat dikatakan bilateral apabila sistem ini menggunakan informasi masing-masing manipulator yang bergerak dalam dua arah membentuk sebuah sistem *closed loop*. Sehingga antara manipulator *master* dan manipulator *slave* masing-masing mempertukarkan



informasi-informasi secara dua arah. Keuntungan sistem ini adalah apabila manipulator slave tertahan suatu benda atau bersentuhan dengan suatu benda di lingkungan luar, maka informasi tersebut dikirim pada manipulator *master* sehingga pengguna mengetahui bahwa terdapat sesuatu yang menyentuh manipulator *slave*. Sensasi sentuhan tersebut biasa dikenal dengan istilah *haptic*.

Sensasi ini dikirimkan dengan menggunakan pengendali dalam kasus ini adalah ATmega128 kepada manipulator *master*. Begitu juga informasi yang ada pada manipulator *master* yang akan dikirimkan juga kepada manipulator *slave* sehingga membentuk sistem bilateral.

## **1.2 Tujuan**

Tujuan penelitian yang dilakukan pada skripsi ini adalah merancang sebuah sistem teleoperasi bilateral yang dapat meningkatkan efek *haptic* dengan *multi-DOF* 4 kanal yang mencakup pengendali posisi dan torsi (*torque*) menggunakan Dynamixel AX-12+ dengan kemampuan menyesuaikan torsi secara otomatis.

## **1.3 Pembatasan Pembahasan Masalah**

Dalam skripsi ini, permasalahan yang dibahas dibatasi dengan penggunaan serial servo dynamixel AX-12+ dan penggunaan mikrokontroler ATmega128 menjadi sebuah sistem teleoperasi bilateral. Sistem ini menggunakan 4 kanal agar posisi dan gaya dapat bertukar informasi pada masing-masing manipulator. Sistem ini dapat menyesuaikan torsi pada masing-masing sendi pada saat manipulator menyentuh benda sesuai dengan nilai perbedaan posisi yang terdeteksi pada masing-masing sendi.

## **1.4 Metodologi Penelitian**

Penelitian yang didefinisikan pada skripsi ini dilakukan melalui:

1. Pendekatan tinjauan pustaka yaitu dengan melakukan studi literatur dari buku-buku, sumber di internet dan beberapa jurnal
2. Pendekatan diskusi dengan dosen pembimbing skripsi yang berkaitan dengan topik bahasan skripsi

3. Pengambilan data dari pengujian sistem teleoperasi bilateral dengan aktuator motor servo serial dynamixel AX-12+
4. Pengolahan data dan analisis.

### **1.5 Sistematika Penulisan**

Secara umum, skripsi ini dituliskan dalam lima bab. Pada bab yang pertama berjudul pendahuluan dan berisi latar belakang pembahasan skripsi, tujuan pembahasan skripsi, metodologi penulisan skripsi, serta sistematika penulisan skripsi.

Kemudian bab yang kedua berjudul sistem robotika dan teleoperasi bilateral dan secara garis besar berisi teori-teori yang terutama berkaitan dengan judul dari skripsi ini, di antaranya adalah penjelasan sistem teleoperasi bilateral dan penjelasan aktuator yang digunakan yakni dynamixel AX-12+ dan mikrokontroler yang digunakan sebagai pengendali pada sistem ini.

Pada bab ketiga berjudul perancangan sistem bilateral yang akan menjelaskan perancangan sistem yang dibuat pada skripsi ini seperti perancangan pengendali, pengaturan torsi aktuator, perancangan perangkat keras, dan perancangan perangkat lunak termasuk beberapa algoritma yang dibuat untuk menunjang kinerja sistem.

Bab keempat adalah hasil pengujian dan analisis yang berisikan analisis pengujian yang telah dilakukan seperti penentuan nilai parameter pengendali pada sistem, penentuan nilai frekuensi cut-off untuk filter, penentuan nilai parameter gaya dan sudut distribusi torsi dan aplikasi untuk sistem penyesuaian otomatis pada parameter torsi.

Dan pada bab terakhir adalah kesimpulan yang memuat poin-poin penting yang telah dilakukan pada pengujian dan hasil hasil yang telah diuji pada pengujian skripsi ini.

## **BAB 2**

### **SISTEM ROBOTIKA DAN TELEOPERASI BILATERAL**

Sistem teleoperasi bilateral adalah salah satu aplikasi turunan dari sistem robotika. Pada sistem robotika dikenal beberapa komponen seperti pengendali (*controller*) dan aktuator. Pada sistem teleoperasi juga memiliki beberapa komponen seperti sistem robotika yaitu pengendali dan dua buah aktuator yang biasa disebut manipulator. Pada bab ini akan dibahas tentang pengenalan sistem robotika secara umum, pengenalan sistem bilateral secara umum, aktuator berupa dynamixel AX-12+, pengendali yang digunakan berbasis ATmega128 dan serial tool program sebagai program untuk mengakuisisi data.

#### **2.1 Dasar Mengenai Sistem Otomatisasi dan Robotika**

Sistem otomatisasi dan robotika adalah sebuah sistem yang menggabungkan bidang mekanika, elektronika, dan pengendali yang berupa sistem berbasis komputer (PLC atau mikrokontroler). Sistem otomatisasi ini sangatlah penting digunakan pada aplikasi-aplikasi industri karena sistem ini dapat bekerja sendiri (*unsupervised*) sehingga memiliki efektifitas yang tinggi. Sistem ini memerlukan sistem kendali, baik menggunakan sistem mekanika, elektronika maupun komputer. Saat ini, mayoritas sistem pengendali yang digunakan berbasis komputer. Seperti pada industri-industri yang menggunakan PLC sebagai sistem pengendali. Lain hal, dengan mayoritas pemakaian non-industri (skala kecil) lebih banyak menggunakan mikrokontroler sebagai otak dari sistem yang telah dibuat.

#### **2.2 Dasar Mengenai Sistem Teleoperasi Bilateral**

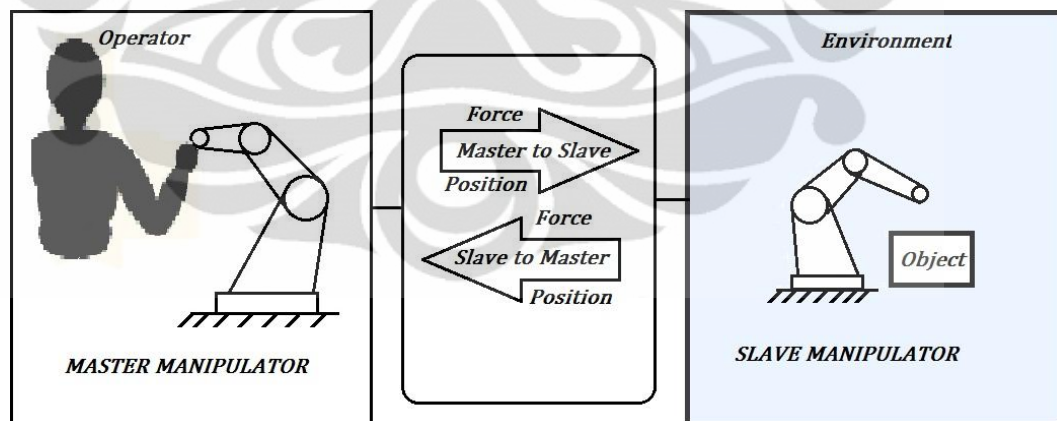
Pada saat sistem robotik dengan manipulator diaplikasikan pada sesuatu pekerjaan yang tidak terstruktur, maka pengendalian akan terbagi menjadi dua yaitu menggunakan fungsi otomatis dan menggunakan fungsi manual. Walaupun pengendalian otomatis yang canggih dapat diterapkan, teleoperasi (pengendalian manual) untuk sebuah manipulator harus disediakan sebagai sebuah cadangan atau pelengkap [1]. Teleoperasi, yang salah satu manipulatornya digerakkan oleh

operator, memiliki fungsi yang sangat baik untuk merasakan umpan balik yang diberikan oleh manipulator lain yang berada pada suatu tempat.

*Telerobotic* adalah bagian dari bidang robotik yang mengkhususkan ilmu dengan pengendalian robot jarak jauh. Pada umumnya telerobotik memiliki dua fungsi, *teleoperation* dan *telepresence* [2]. *Teleoperation* berarti melakukan sesuatu pekerjaan dari jarak jauh sementara *telepresence* memungkinkan pengguna merasakan seakan-akan pengguna tersebut berada pada area yang jauh tersebut.

Sistem teleoperasi telah lama digunakan dengan berbagai aplikasi. Biasanya sistem teleoperasi digunakan pada eksplorasi tempat yang ekstrim atau kegiatan yang dilakukan dalam kondisi ekstrim. Teleoperasi digunakan untuk eksplorasi laut dalam [3], bedah [4], pengaturan pembuangan limbah berbahaya [5], *tele-rescue* [6], dan banyak aplikasi lainnya.

Sistem teleoperasi dapat didefinisikan sebagai sistem pengendali bilateral yang memiliki komponen-komponen penting yaitu pengguna, master, slave, pengendali, dan lingkungan [7]. Pengguna akan menggunakan manipulator master untuk menggerakkan manipulator slave yang berada di suatu tempat untuk merasakan lingkungan yang ada disekitar manipulator slave.



Gambar 2.1 Sistem teleoperasi bilateral

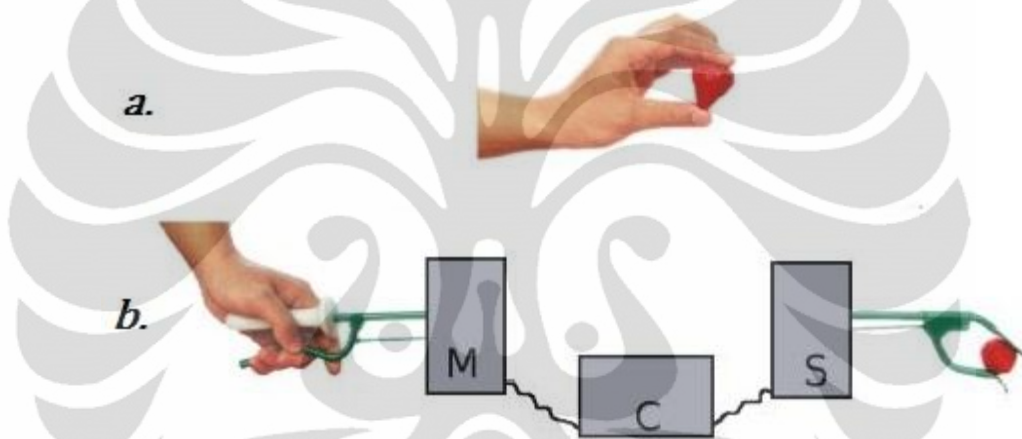
Untuk merasakan lingkungan pada manipulator slave, maka manipulator tersebut harus berfungsi layaknya tangan operator itu sendiri. Pada beberapa literatur tangan manusia dapat terbagi dalam 2 fungsi sensor: *cutaneous* dan *kinesthetic*[7]. *Cutaneous* ditemukan pada permukaan kulit tangan manusia. Dengan kemampuan ini manusia dapat mendeteksi getaran, jenis permukaan dan

bentuk permukaan. *Kinesthetic* ditemukan pada sendi-sendi dan otot manusia. Dengan kemampuan ini manusia dapat mendeteksi berbagai informasi tentang pergerakan sendi, dan skala yang lebih besar seperti bentuk benda dan kekerasan material.

Lebar pita frekuensi pada masing-masing fungsi sensor manusia tersebut berbeda-beda dapat dilihat sebagai berikut [7]

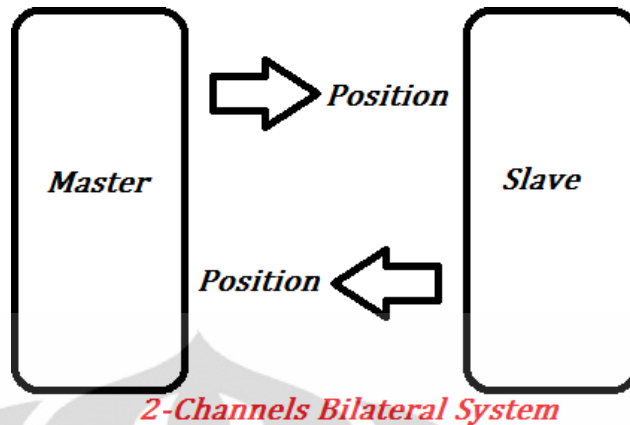
- Lebar pita frekuensi *Kinesthetic*: 20Hz
- Lebar pita frekuensi *Cutaneous*: 320 Hz

Sistem teleoperasi bilateral memiliki tujuan seperti pada gambar ilustrasi berikut.



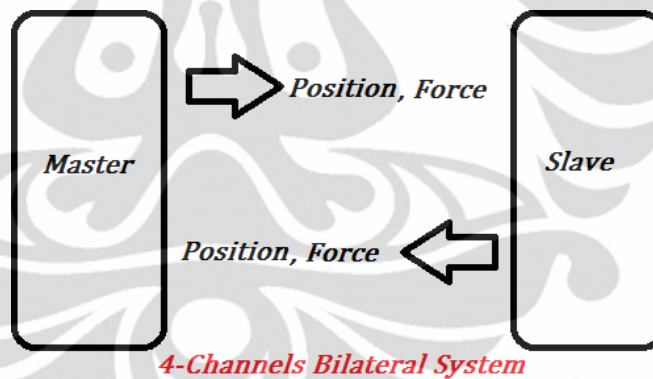
Gambar 2.2 Tujuan sistem teleoperasi bilateral

Sistem bilateral seperti gambar bagian 'B' diatas memiliki master manipulator dan slave manipulator yang saling terhubung, sehingga pengguna yang menggunakan manipulator master dapat merasakan sensasi yang dialami manipulator slave begitu pula sebaliknya. Diharapkan walaupun menggunakan manipulator master dan slave, sensasi gaya yang diakibatkan dari menyentuh sesuatu atau mengangkat sesuatu sama seperti hubungan langsung dengan menggunakan tangan seperti yang dapat dilihat pada gambar bagian 'A'.



Gambar 2.3 Sistem bilateral 2 kanal

Terdapat dua jenis sistem teleoperasi bilateral yaitu sistem bilateral 2 kanal (*2-channels*) dan sistem bilateral 4 kanal (*4-channels*). Pada sistem bilateral 2 kanal, masing-masing manipulator master dan slave hanya mempertukarkan informasi posisi. Sedangkan pada sistem bilateral 4 kanal, masing-masing manipulator mempertukarkan informasi posisi dan gaya yang dialami oleh masing-masing manipulator.



Gambar 2.4 Sistem Bilateral 4 kanal

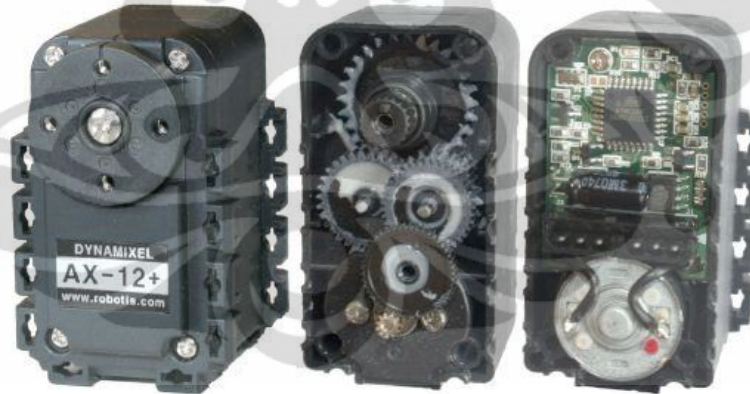
Untuk sistem bilateral yang dibuat dalam penelitian ini adalah sistem bilateral dengan 4 kanal. Sistem ini lebih baik dibandingkan dengan 2 kanal karena sistem 4 kanal ini memiliki nilai beban yang akan ditukarkan informasinya. Untuk mengakomodir kebutuhan sistem dengan 4 kanal maka aktuator yang dibutuhkan adalah sebuah aktuator yang mampu membaca nilai gaya yang terjadi pada sistem dengan kata lain akan dipergunakan aktuator berupa serial servo.



### 2.3 Serial Servo sebagai Aktuator Sistem

Aktuator yang digunakan untuk sebuah sistem teleoperasi bilateral dapat berbagai macam. Salah satunya adalah dengan menggunakan serial servo. Serial servo berbeda dengan motor servo biasa. Motor servo biasa dapat digerakkan dengan memberikan sinyal PWM pada pengendali motor servo tersebut, sedangkan dengan menggunakan serial servo pengguna dapat menggerakkan motor servo ini dengan menggunakan komunikasi serial.

Pada motor servo biasa, parameter untuk pengendali bersifat statis yang berarti motor servo akan menerima sinyal untuk mencapai suatu target posisi dan motor servo akan menghitung torsi yang dibutuhkan untuk mencapai posisi tersebut. Kecepatan motor atau batas maksimum torsi servo tidak dapat diatur. Selain itu beban tidak dapat diidentifikasi langsung oleh motor servo sehingga pengguna kesulitan [8]. Hal ini tidak ditemukan pada serial servo yang dapat mengubah langsung parameter-parameter pengendali yang terdapat didalamnya seperti, torsi maksimum, kecepatan, membaca posisi, membaca suhu, dan lain sebagainya.



Gambar 2.5 Dynamixel AX-12+

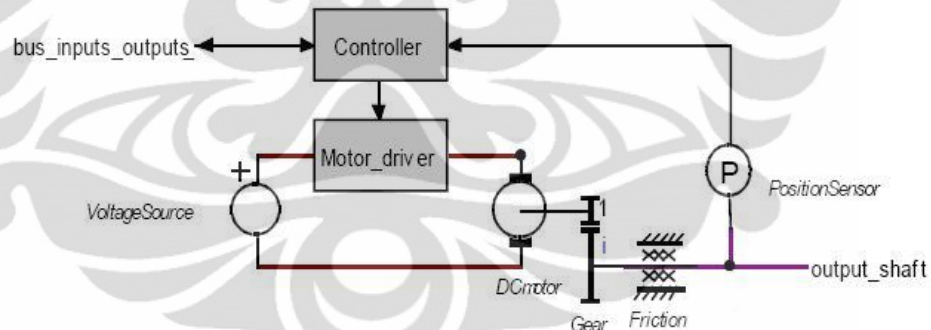
Dynamixel adalah sebuah serial servo yang cukup *powerful* yang biasa dijadikan aktuator dari sebuah robot. Berbagai macam tipe dynamixel yang ada bergantung dari segi kecepatan motor bergerak, torsi yang dihasilkan, dan jenis gir yang ada didalam motor tersebut.

Dynamixel AX-12+ dapat dijadikan aktuator dari sistem bilateral karena selain parameter-parameter yang dapat disesuaikan dengan kebutuhan sistem, aktuator ini juga dapat membaca berbagai macam informasi-informasi yang ada

pada dynamixel. Hal yang terpenting adalah aktuator ini dapat membaca posisi dan gaya yang terjadi pada saat *real-time* sehingga sebuah sistem bilateral 4 kanal dapat dengan mudah direalisasikan. Selain itu serial servo ini termasuk motor yang terjangkau dari segi biaya pada kelasnya.

### 2.3.1 Gambaran Umum Dynamixel AX-12+

Dynamixel AX-12+ adalah salah satu dari banyak tipe dynamixel serial servo. Dynamixel ini paling umum digunakan dalam pembentukan robot-robot terutama dibidang robot *humanoid*. Serial servo ini memiliki gir yang terbuat dari plastik yang kuat. Terdapat sebuah pengendali berupa ATmega8 sehingga dengan bantuan fitur-fitur yang ada pada ATmega8, dynamixel ini mampu mengukur posisi, kecepatan, temperatur, tegangan dan lain lain. Dynamixel ini mendapat setpoint melalui komunikasi serial dan diproses untuk menggerakkan motor DC yang terdapat didalam dynamixel. Output yang dihasilkan berupa posisi yang dibaca dengan sensor posisi untuk mengukur posisi sekarang sebagai umpan balik pengendali.



Gambar 2.6 Blok diagram pada dynamixel AX-12+

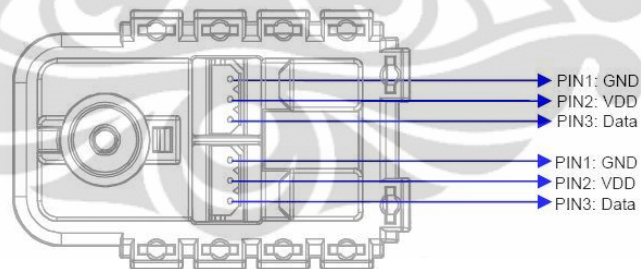
Selain itu dynamixel AX-12+ ini memiliki *precision control* yang memiliki resolusi 1024 step, *compliance driving* sehingga sudut dari penyesuaian torsi dapat diatur, Umpan balik berupa umpan balik posisi, kecepatan, dan torsi beban. Dynamixel ini juga memiliki sistem alarm yang dapat ditentukan dari parameter yang diinginkan pengguna seperti suhu internal, tegangan, dan torsi. Dynamixel AX-12+ menggunakan *distributed control* yang berarti posisi, kecepatan, dan torsi dapat diatur dengan sebuah paket data [9]. Berikut adalah spesifikasi utama dari Dynamixel AX-12+.



Tabel 2.1 Spesifikasi umum dynamixel AX-12+

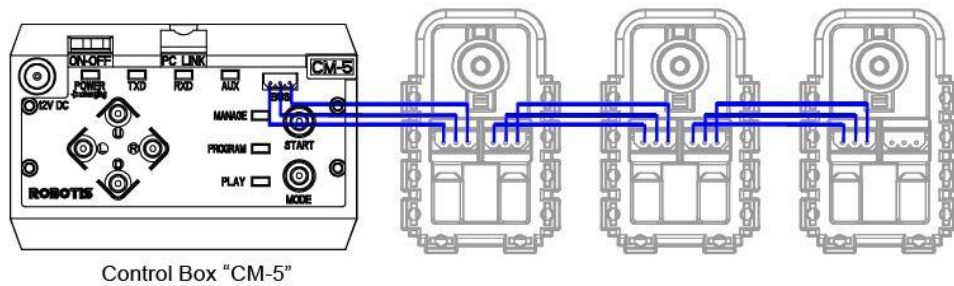
Parameter	Nilai
Limit Tegangan	7-14 Volt
Arus Maksimum	900mA
Torsi	12-16.5 kgf.cm
Kecepatan	0.269 sec/60° ~ 0.196 sec/60°
Resolusi	0.29°/step (1024 step)
Sudut Operasi	300°, <i>Endless Turn</i>
Temperatur Operasi	-5° C ~ +85° C
Sinyal Pengendali	<i>Digital Packet</i>
Tipe Protokol	<i>Half Duplex Asynchronous Serial Communication (8 bit, 1 stop, No Parity)</i>
BaudRate	7343bps – 1Mbps
Level Sinyal	TTL
ID	254 ID

Terdapat dua buah port konektor pada dynamixel yang memungkinkan penggunaan koneksi *daisy-chain* antar dynamixel dan pengendali sehingga dapat mengurangi penggunaan port pada pengendali (mikrokontroler).

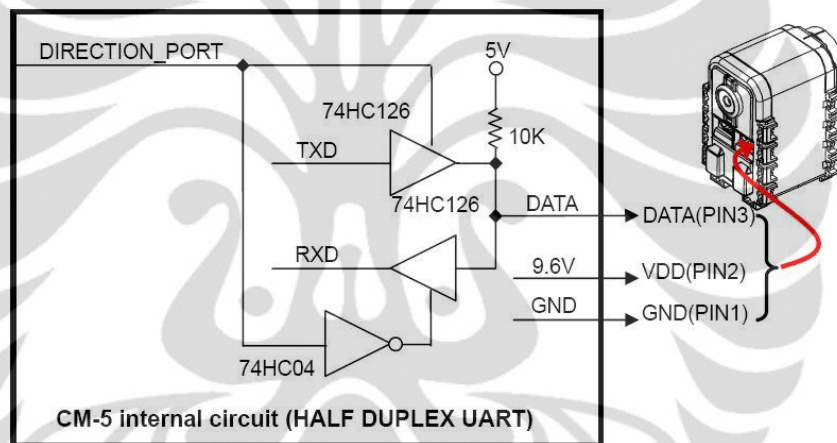


Gambar 2.7 Konfigurasi pin dynamixel AX-12+

Dengan menggunakan kedua port pada dynamixel maka pengkabelan dapat dilakukan seperti gambar berikut.

Gambar 2.8 Konfigurasi koneksi secara *daisy-chain*

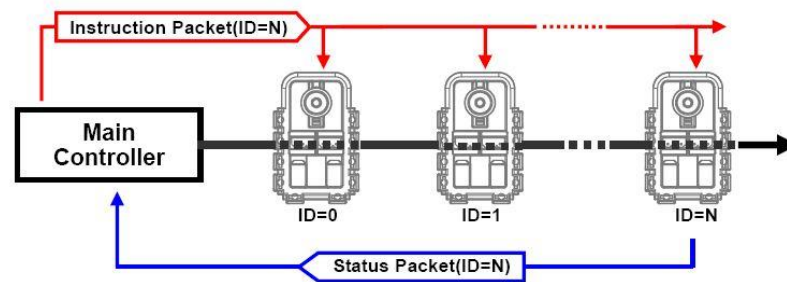
Komunikasi yang menghubungkan pengendali dengan dynamixel adalah jenis komunikasi UART (*Universal Asynchronous Receiver/Transmitter*). UART yang digunakan berupa *half-duplex* sehingga pengiriman paket dan penerimaan paket dilakukan secara bergantian.

Gambar 2.9 Mode komunikasi *half-duplex* dynamixel

Pada gambar 2.9, dapat dilihat bahwa DIRECTION\_PORT diubah menjadi nilai logika '1' apabila ingin mengirim paket data dari pengendali ke dynamixel dan diubah menjadi nilai logika '0' apabila ingin menerima paket data dari dynamixel menuju ke pengendali.

### 2.3.2 Komunikasi Protokol Dynamixel AX-12+

Terdapat dua jenis paket data yang digunakan dynamixel untuk berkomunikasi dengan pengendali yaitu paket instruksi dan paket status. Paket instruksi adalah paket yang dikirimkan oleh pengendali untuk menginstruksikan dynamixel melakukan sebuah pekerjaan. Sementara paket status adalah paket yang dikirimkan dynamixel untuk pengendali yang berupa informasi status pekerjaan yang telah dilakukan dynamixel.



Gambar 2.10 Paket instruksi dan paket status

Pengendali akan mengirimkan sebuah paket instruksi yang akan diterima oleh dynamixel, dengan paket tersebut dynamixel akan melakukan perintah dari paket instruksi tersebut. Dengan jeda waktu yang ditentukan sebelumnya, dynamixel akan mengirimkan status paket yang berisikan informasi status dan jenis-jenis kesalahan yang terjadi pada dynamixel tersebut. Paket status tersebut akan diterima pengendali untuk lebih lanjut dianalisis informasinya.

Struktur paket instruksi yang dikirim pengendali ke dynamixel mengikuti format berikut



Gambar 2.11 Format paket instruksi

Paket data ini menggunakan bilangan heksadesimal untuk setiap byte. Berikut adalah penjelasan setiap byte dari paket instruksi

- **0xFF 0xFF** Paket ini diawali oleh sebuah header yang berisikan dua buah bilangan heksadesimal yang menyatakan awal dari paket yang dikirimkan.
- **ID Byte** ID diisi dengan nilai ID pada dynamixel tersebut dengan kisaran 0 – 254 atau 0x00 – 0xFD dalam bilangan heksadesimal. Terdapat sebuah *Broadcast ID* (0xFE) yang berarti paket ini ditujukan untuk semua dynamixel yang terhubung dengan nomor ID berapapun.
- **LENGTH** Menunjukkan nilai panjang paket yang bernilai ‘Jumlah Parameter (N) + 2’.
- **INSTRUCTION** Jenis instruksi yang diperintahkan kepada dynamixel. Dapat dilihat pada tabel 2.3.
- **PARAMETER1..N** Digunakan sebagai tambahan untuk melengkapi instruksi.
- **CHECK SUM** Digunakan sebagai sebuah tanda selesainya paket instruksi. Nilai dari check sum ini dapat dihitung dengan persamaan berikut.

$Check\ Sum = \sim(ID+Length+Instruction+Parameter\ 1+\dots+Parameter\ N)$  (2.1)  
Sementara struktur paket status yang dikirim dynamixel ke pengendali mengikuti format berikut



Gambar 2.12 Format paket status

Format paket status hampir sama dengan paket instruksi yang diawali dengan sebuah header dan diakhiri dengan sebuah checksum. Perbedaan yang dapat dilihat adalah terdapat byte error pada paket ini. Byte error berfungsi untuk mengirimkan informasi apabila terdapat error dari dynamixel. Byte error ini dapat dilihat artinya pada setiap bit dari tabel 2.2 berikut.

Tabel 2.2 Tabel byte error pada paket status

Bit	Nilai	Keterangan
Bit 7	0	-
Bit 6	Instruction Error	Bernilai 1 jika instruksi yang diterima tidak terdefinisi
Bit 5	Overload Error	Bernilai 1 jika nilai beban melebihi torsi maksimum
Bit 4	Checksum Error	Bernilai 1 jika nilai check sum pada paket instruksi salah
Bit 3	Range Error	Bernilai 1 jika jumlah parameter paket instruksi melebihi kisaran yang terdefinisi (Length)
Bit 2	Overheating Error	Bernilai 1 jika suhu dynamixel melebihi temperatur operasi yang ditentukan
Bit 1	Angle Limit Error	Bernilai 1 jika dynamixel diperintahkan bergerak melewati batas yang ditentukan
Bit 0	Input Voltage Error	Bernilai 1 jika tegangan input pada dynamixel melewati batas teganganm operasi yang ditentukan

Instruksi yang tersedia pada dynamixel AX-12+ dapat dilihat pada tabel berikut.

Tabel 2.3 Tabel instruksi yang digunakan pada dynamixel

Instruksi	Fungsi	Nilai	Jumlah Parameter
PING	Digunakan untuk mengecek apakah dynamixel tersebut ada	0x01	0

READ DATA	Membaca nilai yang ada pada register tertentu	0x02	2
WRITE DATA	Menulis nilai yang ada pada register tertentu	0x03	2~
REG WRITE	Seperti WRITE DATA, akan tetapi perintah ini akan bersifat menunggu sampai instruksi ACTION diberikan	0x04	2~
ACTION	Memicu perintah yang dilakukan oleh instruksi REG WRITE	0x05	0
RESET	Reset menjadi <i>factory default</i>	0x06	0
SYNC WRITE	Untuk menuliskan perintah untuk banyak aktuator dalam waktu yang sama	0x07	4~

Dynamixel AX-12+ dapat dioperasikan dengan membaca nilai register ataupun menulis perintah pada register-register tertentu. Register tersebut berada dalam RAM dan EEPROM. Berikut ini adalah penjelasan untuk beberapa alamat register yang sering digunakan dalam mengendalikan dynamixel.

Address 0x03      **ID**. Alamat ini digunakan untuk merubah ID yang bersifat unik pada masing-masing dynamixel.

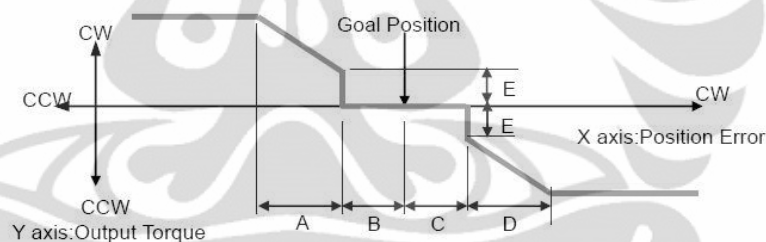
Address 0x04      **Baudrate**. Alamat ini digunakan untuk merubah kecepatan komunikasi dynamixel. Alamat ini diisi dengan hasil perhitungan berdasarkan persamaan berikut.

$$Speed(bps) = 2000000 / (Address\ 0x04 + 1) \quad (2.2)$$

Address 0x0E,0x0F,0x22,0x23

**Maximum Torque.** Alamat ini digunakan untuk mengatur nilai torsi maksimum yang digunakan pada dynamixel. Terdapat dua lokasi untuk mengatur nilai ini yaitu pada EEPROM (alamat 0x0E,0x0F) dan pada RAM (alamat 0x22,0x23). Pada saat dynamixel pertama kali hidup maka nilai yang ada pada EEPROM akan disalin ke lokasi alamat RAM sehingga dynamixel hanya membaca nilai yang ada pada RAM.

Address 0x1A~0x1D **Compliance Margin dan Slope.** Pada alamat ini kita dapat mengatur pembagian distribusi torsi pada dynamixel dan mengatur pembagian error. Dengan adanya pengaturan ini, output yang dihasilkan pada dynamixel dapat bergerak lebih mulus tanpa merusak gir dalam dynamixel. Grafik dari pengaturan ini dapat dilihat pada gambar berikut.



Gambar 2.13 Grafik pengaturan distribusi torsi pada dynamixel

Dimana

A: Kemiringan torsi untuk berlawanan arah jarum jam

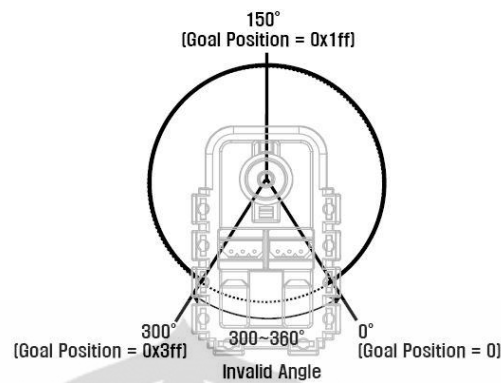
B: Error yang terjadi untuk berlawanan arah jarum jam

C: Error yang terjadi untuk searah jarum jam

D: Kemiringan torsi untuk searah jarum jam

E: Torsi minimum pada saat mencapai posisi yang diinginkan

Address 0x1E,0x1F **Goal Position.** Alamat ini digunakan untuk memerintahkan dynamixel untuk bergerak menuju posisi angular yang diinginkan. Posisi yang memungkinkan untuk dynamixel bergerak dapat dilihat pada gambar berikut.



Gambar 2.14 Area operasi pergerakan dynamixel AX-12+

Address 0x24,0x25 **Present Position**. Digunakan untuk membaca posisi angular terkini dynamixel.

Address 0x28,0x29 **Present Load**. Digunakan untuk membaca besar load pada dynamixel dan arahnya. Pembagian nilai beban dan arah dapat dilihat pada tabel berikut.

Tabel 2.4 Paket data pembacaan beban dynamixel

BIT	15~11	10	9	8	7	6	5	4	3	2	1	0
Value	0	Load Direction	Load Value									

Contoh penggunaan paket instruksi dan paket status pada dynamixel AX-12+ adalah sebagai berikut

- Mengubah nilai torsi maksimum yang ada pada register EEPROM menjadi 50% dari nilai maksimal yang mungkin dicapai dynamixel dengan ID 0

Instruksi **WRITE DATA**

Address **0x0E**

Data **0xFF,0x01** (Karena nilai 100% adalah 0x3FF, maka nilai 50% adalah 0x1FF)

[Paket Instruksi]: FF FF 00 05 03 0E FF 01 E9

[Paket Status]: FF FF 00 02 00 FD

Dalam hal ini tidak terdapat error.

- Membaca nomor model dan versi *firmware* pada dynamixel dengan ID 1



Instruksi **READ DATA**

Address **0x00**

Length **0x03**

[Paket Instruksi]: FF FF 01 04 02 00 03 F5

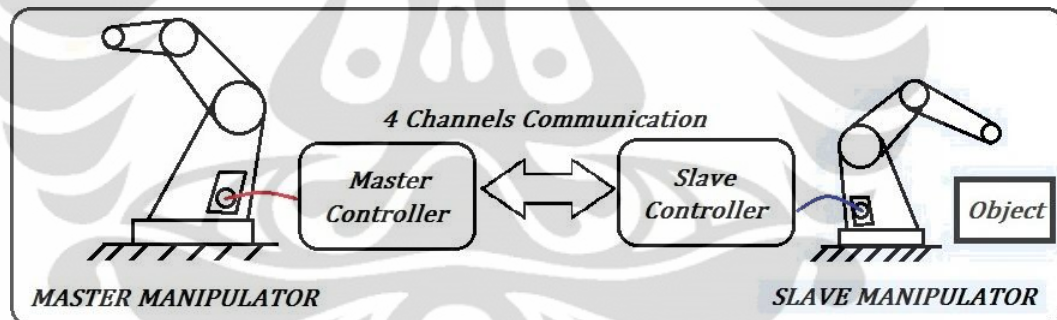
[Paket Status]: FF FF 01 05 00 74 00 08 7D

Dalam kasus ini nomor model adalah 0x74 (116) dan versi firmware 0x08, tidak terdapat error.

Untuk nilai nilai register secara keseluruhan dapat dilihat pada tabel yang ada pada bagian lampiran.

#### 2.4 Mikrokontroler ATmega 128 Sebagai Blok Pengendali

Sistem teleoperasi bilateral memerlukan blok pengendali sebagai pusat komputasi dan komunikasi untuk seluruh sistem baik untuk manipulator master maupun manipulator slave. Ilustrasi sebuah sistem teleoperasi beserta blok pengendalinya dapat dilihat pada gambar berikut.

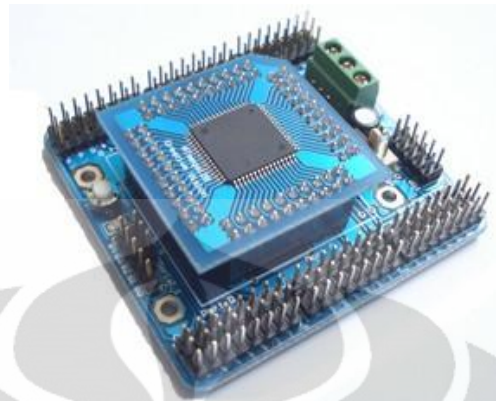


Gambar 2.15 Teleoperasi bilateral dengan blok pengendali

Pengendali dapat berupa sistem berbasis komputer seperti PC, PLC, atau mikrokontroler. Mikrokontroler merupakan sebuah mikroprosesor yang telah dilengkapi dengan perangkat antarmuka dengan peralatan luar yang terintegrasi dalam satu chip. Mikrokontroler AVR memiliki arsitektur RISC (*Reduced Instruction Set Computing*) 8-bit sehingga sebagian besar instruksi dieksekusi dalam satu siklus instruksi *clock*. Apabila mikrokontroler AVR menggunakan pewaktuan sebesar 16MHz maka AVR ini dapat mengeksekusi instruksi mencapai 16 juta instruksi dalam satu detik atau biasa disebut 16 MIPS (*Millions of instructions per second*). Hal ini sangat berbeda dengan mikrokontroler keluarga MCS51 yang masih memiliki arsitektur CISC (*Complex Instruction Set*

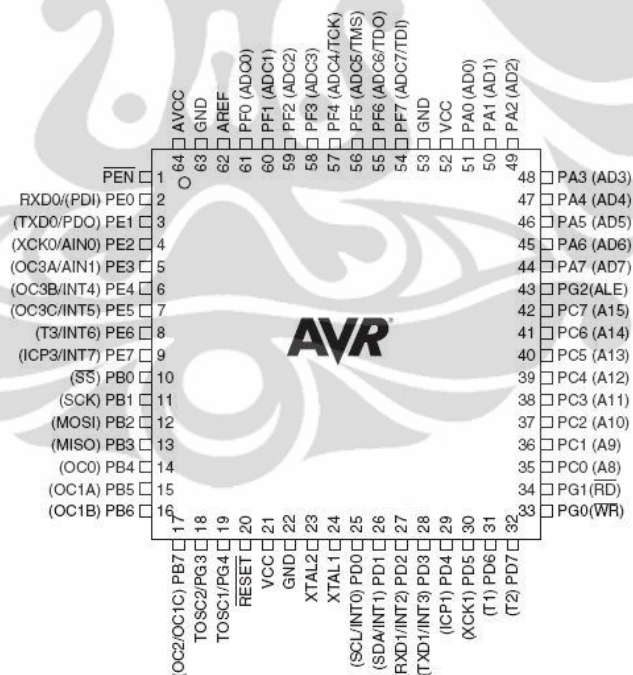


Computing) yang membutuhkan 12 siklus pewaktuan untuk mengeksekusi instruksi.



Gambar 2.16 Modul ATmega 128

Terdapat beberapa kelas mikrokontroler AVR yaitu, ATtiny, AT90Sxx, ATmega, dan AT86RFxx. Pada skripsi ini pengujian dilakukan dengan menggunakan ATmega 128, dengan konfigurasi pin sebagai berikut.



Gambar 2.17 Konfigurasi pin ATmega 128

Sistem berbasis ATmega ini dapat direalisasikan untuk menunjang kinerja dari sistem bilateral yang akan dibuat. Oleh karena aktuator yang digunakan adalah serial servo, yang diperintahkan dengan menggunakan sebuah paket data serial seperti yang telah dijelaskan sebelumnya, maka komunikasi menjadi hal

yang terpenting dalam sistem ini. ATmega 128 memiliki 2 buah USART sehingga dapat mengakomodir komunikasi ke pada aktuator dynamixel dan dapat berkomunikasi dengan pengendali ATmega lainnya untuk menginformasikan parameter antar manipulator.

#### 2.4.1 Komunikasi Serial dengan Menggunakan USART

Komunikasi serial adalah sebuah proses pengiriman sejumlah data dalam bit dengan koneksi *single wire*. Masing-masing bit yang dikirimkan akan dipisahkan dengan waktu sehingga peranti yang akan menerima data tersebut dapat mengartikan level logika yang dikirimkan pada masing-masing bit.

USART (*Universal Synchronous Asynchronous Receiver Transmitter*) digunakan untuk berkomunikasi antara mikrokontroler dengan peranti lain. Peranti lain ini dapat berupa mikrokontroler, PC, ataupun alat yang memiliki koneksi komunikasi serial. Terdapat dua mode untuk komunikasi ini yaitu sinkronus dan asinkronus. Pada komunikasi sinkronus diperlukan sinyal pewaktuan untuk mensinkronkan kecepatan data, sementara pada komunikasi asinkronus tidak diperlukan sinyal pewaktuan. Komunikasi yang digunakan pada skripsi ini adalah mode asinkronus. Mode komunikasi serial asinkronus menggunakan *start bit* dan *stop bit* pada awal dan akhir data agar penerima mengetahui waktu yang dibutuhkan oleh masing-masing bit [10].

USART memiliki 3 buah register yaitu *control and status register*, *baud rate register*, dan *data register*. Pada register kontrol dan status terdapat 3 buah register 8-bit, UCSRA, UCSRB, dan UCSRC. Sedangkan untuk register baud rate terdapat 16-bit register yaitu UBRRH yang berisi byte tertinggi dari satu word dan UBRRL yang berisi byte terendah dari satu word. Terdapat register yang menyimpan data yang akan ditulis ataupun dibaca bernama UDR. UDR adalah 2 register yang memiliki satu alamat I/O (*sharing address*). Masing-masing UDR ini berfungsi sebagai register tulis (*write-only register*) yang berisi byte yang akan ditransmisikan dan register baca (*read-only register*) yang berisi byte yang telah diterima. Maka apabila program membaca UDR, berarti program membaca data yang telah diterima oleh UDR secara serial. Sedangkan apabila program menulis register UDR berarti program ini menulis data untuk dikirimkan secara serial.

Register UCSRA menggambarkan status yang sedang dilakukan oleh USART tersebut. Berikut adalah arti dari masing-masing bit dalam register UCSRA

Tabel 2.5 Register UCSRA

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM

Bit	Deskripsi
RXC	Set untuk mengindikasikan data serial telah diterima
TXC	Set untuk mengindikasikan data serial telah dikirim
UDRE	Set untuk mengindikasikan bahwa register pengiriman UDR telah kosong
FE	Set untuk mengindikasikan bila terjadi <i>framing error</i>
DOR	Set untuk mengindikasikan bila terjadi <i>overrun error</i>
PE	Set untuk mengindikasikan bila terjadi <i>parity error</i>
U2X	Set untuk mengubah kecepatan transmisi menjadi dua kali lipat
MPCM	Set untuk mengaktifkan mode komunikasi <i>multi-processor</i>

Jika sebuah byte telah berhasil dikirimkan dari sebuah piranti menuju USART pada mikrokontroler, pada saat bit terakhir dalam byte tersebut diterima, bit RXC pada register UCSRA akan diset menjadi logika 1 yang mengartikan bahwa byte telah selesai diterima. Program menggunakan bit RXC untuk mengetahui kapan waktu yang tepat untuk membaca data pada register UDR yang telah diterima oleh USART. Data tersebut harus langsung dibaca atau dimasukkan ke dalam suatu tempat penyimpanan (*buffer*) agar tidak tertimpa oleh data yang akan diterima selanjutnya. Sedangkan UDRE digunakan untuk mengindikasikan bahwa register pengiriman pada UDR telah kosong dan memungkinkan untuk diisi byte selanjutnya untuk dikirim.

Register UCSRB digunakan untuk menginisialisasi fungsi-fungsi pada USART. Berikut arti masing-masing bit pada register UCSRB

Tabel 2.6 Register UCSRB

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8

Bit	Deskripsi
RXCIE	Mask bit untuk <i>Receive interrupt enable</i> . Set logika 1 untuk mengaktifkan interupsi
TXCIE	Mask bit untuk <i>Transmit interrupt enable</i> . Set logika 1 untuk mengaktifkan interupsi
UDRIE	Mask bit untuk <i>Data Register Empty interrupt enable</i> . Set logika 1 untuk mengaktifkan interupsi
RXEN	Set logika 1 untuk mengaktifkan penerima serial ( <i>Serial Receiver</i> )
TXEN	Set logika 1 untuk mengaktifkan pengiriman serial ( <i>Serial Transmitter</i> )
UCSZ2	-
RXB8	Bit ke -9 yang diterima pada mode penerimaan 9-bit
TXB8	Bit ke -9 yang dikirim pada mode pengiriman 9-bit

3-bit awal adalah mask bit yang berhubungan dengan interupsi pada USART. *RXCIE* adalah bit mask interupsi yang terjadi saat byte serial diterima. *TXCIE* adalah bit mask interupsi yang terjadi saat USART menyelesaikan pengiriman byte. *UDRIE* adalah mask bit interupsi yang terjadi saat register UDR siap untuk mengirim byte selanjutnya.

Register UBRR mengatur kecepatan komunikasi dalam USART. Register 16-bit ini terbagi dua yaitu UBRR<sub>L</sub> untuk 8-bit pada byte rendah, dan UBRR<sub>H</sub> untuk 8-bit pada byte tinggi. Nilai yang diisikan pada register ini mengikuti persamaan berikut.

$$UBRR = \frac{\text{System Clock}}{16 * \text{Baudrate}} - 1 \quad (2.3)$$

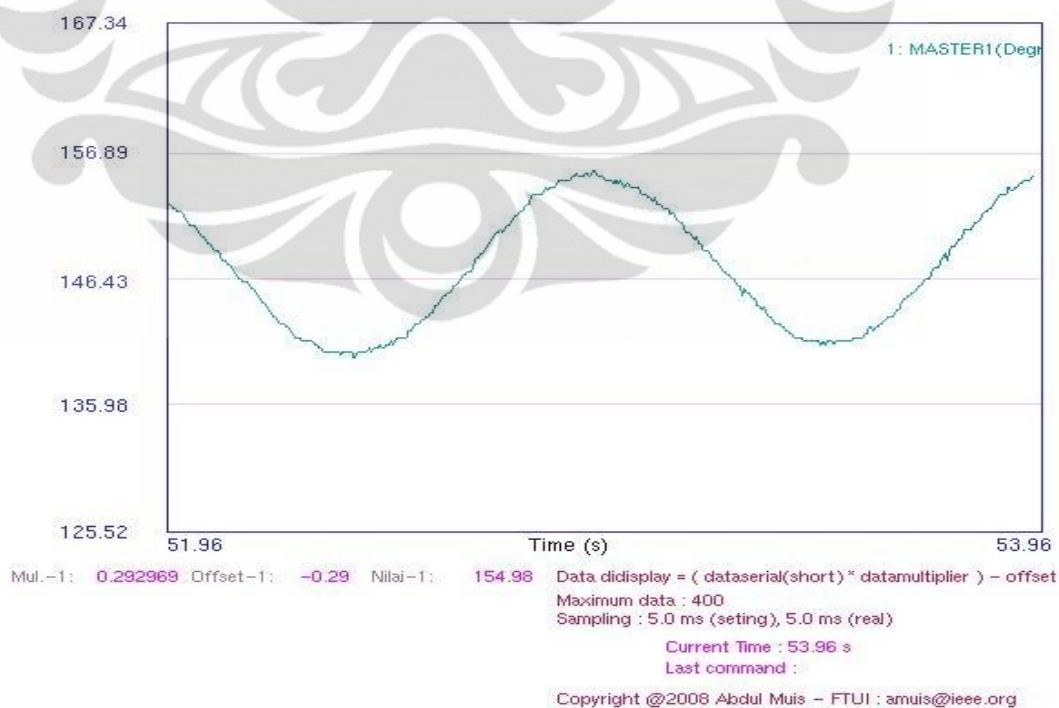
Mengenai komunikasi antara pengendali dan aktuator akan dijelaskan pada bab selanjutnya pada bagian perancangan.

## 2.5 Serial Tool Program

Untuk menerima dan memonitor data yang dikirim dan diterima oleh masing-masing mikrokontroler diperlukan sebuah program untuk membaca data dan akan ditampilkan pada PC. Program yang digunakan adalah *Serial Tool Program*<sup>1</sup>. Dengan menggunakan program ini data yang dikirim dapat direkam dan disimpan oleh komputer dan data tersebut dapat langsung ditampilkan dalam bentuk grafik secara *real-time*. Program ini memiliki komponen antara lain

- cygwin1.dll : *standard library*
- glut32.dll : *openGL library*
- graph232.exe : file eksekusi untuk display grafik berbasis openGL
- serial\_test.exe : file eksekusi untuk membaca port serial
- serial.ini : mengatur konfigurasi untuk komunikasi serial
- dataset.ini : mengatur konfigurasi data yang akan ditampilkan
- result.dat : hasil pembacaan data

Untuk pembacaan data pada sistem yang telah dibuat menggunakan baudrate sebesar 38400 bps dengan waktu cuplik sebesar 5 ms. Berikut adalah tampilan Serial Tool Program.



Gambar 2.18 Serial tool program

<sup>1</sup> Dr. Abdul Muis, ST.,M.Eng

## **BAB 3**

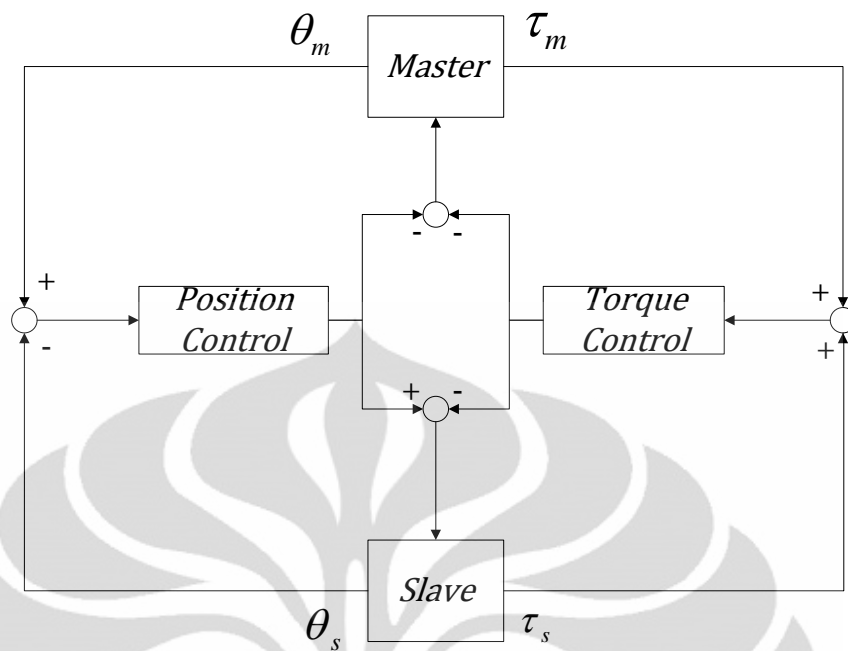
### **PERANCANGAN SISTEM BILATERAL**

Perancangan ini menjadi hal utama dalam penelitian sistem bilateral yang akan dibuat. Beberapa parameter yang ada pada bab perancangan ini akan masuk dalam tahap pengujian pada bab selanjutnya.

Pada bab ini akan dibahas antara lain sistem bilateral yang akan digunakan secara lebih mendalam, perancangan dan penggunaan filter, distribusi torsi yang akan diterapkan pada sistem, perancangan perangkat keras, dan perancangan perangkat lunak. Perancangan perangkat keras yang dibahas meliputi sistem pengkabelan pada pengujian dan beberapa komponen-komponen dasar yang dilakukan untuk menunjang pengujian. Sementara itu, perancangan perangkat lunak yang dilakukan berupa pengaturan baudrate dan pewaktuan yang menghasilkan waktu cuplik, algoritma komunikasi antara pengendali dengan aktuator, algoritma tentang pembacaan beban pada aktuator, dan algoritma *auto-tuning* yang akan diterapkan pada sistem bilateral.

#### **3.1 Perancangan Sistem Teleoperasi Bilateral**

Sistem teleoperasi bilateral yang dibuat pada skripsi ini menggunakan sistem bilateral 4 kanal dengan menggunakan dua buah pengendali yaitu pengendali posisi (*Position Control*) dan pengendali torsi (*Torque Control*). Hasil dari masing-masing pengendali tersebut nantinya akan dinormalisasi sehingga dapat menghasilkan satu jenis keluaran (output). Berikut adalah gambar blok diagram awal pada sistem bilateral.



Gambar 3.1 Blok diagram sistem teleoperasi bilateral 4 kanal

Pada gambar 3.1, terlihat bahwa terdapat dua buah pengendali yaitu pengendali posisi dan torsi. Dari gambar tersebut dapat dilihat sistem secara umum yang dapat dirumuskan kedalam persamaan berikut.

$$\theta_m^{Res} = \theta Controller. (\theta_s^{Ref} - \theta_m^{Ref}) - \tau Controller. (\tau_m^{Ref} + \tau_s^{Ref}) \quad (3.1)$$

$$\theta_s^{Res} = \theta Controller. (\theta_m^{Ref} - \theta_s^{Ref}) - \tau Controller. (\tau_m^{Ref} + \tau_s^{Ref}) \quad (3.2)$$

Penjelasan berupa pengendali dan persamaan-persamaan pendukungnya akan dijelaskan lebih rinci pada subbab berikutnya.

### 3.1.1 Perancangan Pengendali Sistem Teleoperasi Bilateral

Untuk mendapatkan persamaan masing-masing pengendali pada sistem bilateral, maka sistem ini dapat dianggap bekerja pada kondisi ideal yang linear. Sistem bilateral 4 kanal yang linear dapat direpresentasikan menjadi sebuah sistem 2-port yang memiliki matriks sebagai berikut [11].

$$\begin{bmatrix} \tau_m \\ \theta_m \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} \theta_s \\ \tau_s \end{bmatrix} \quad (3.3)$$

Untuk sistem yang ideal, parameter matriks yang digunakan adalah

$$\begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (3.4)$$

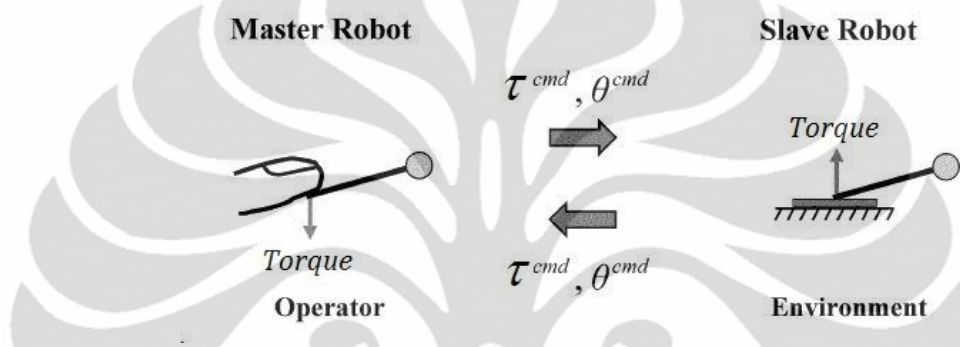
Sehingga persamaan menjadi



$$\tau_m = -\tau_s \quad (3.5)$$

$$\theta_m = \theta_s \quad (3.6)$$

Dengan persamaan ini maka posisi master akan selalu sama dengan slave begitu pula sebaliknya. Sedangkan torsi yang terjadi pada slave, akan sama dengan torsi yang terjadi pada master hanya saja berlawanan arahnya. Gaya tersebut mengikuti kaidah hukum aksi-reaksi yang sesuai dengan hukum Newton yang ketiga yaitu “gaya aksi dan reaksi dari dua buah benda memiliki besar yang sama, dengan arah terbalik, dan segaris”.



Gambar 3.2 Arah gaya yang bekerja pada sistem teleoperasi bilateral

Masing-masing torsi dan posisi yang dipergunakan untuk sistem bilateral memiliki pengendali masing-masing untuk torsi dan posisi yaitu pengendali *common mode* untuk torsi ( $\tau_c$ ) dan *differential mode* untuk posisi ( $\theta_d$ ) [12].

$$\tau_c = \tau_m + \tau_s \quad (3.7)$$

$$\theta_d = \theta_m - \theta_s \quad (3.8)$$

Dari persamaan (3.7) dan (3.8), untuk mendapat hubungan antara pengendali posisi dan pengendali torsi maka persamaan tersebut diubah menjadi dimensi percepatan mengingat masing-masing komponen memiliki hubungan dengan percepatan yaitu

$$\tau = I\alpha \quad (3.9)$$

$$\frac{d^2\theta}{dt^2} = \alpha \quad (3.10)$$

Sehingga pengendali untuk torsi dapat ditulis sebagai persamaan

$$\dot{\theta}_c^{ref} = \frac{K(\ddot{\theta}_c - \tau_c)}{I} \quad (3.11)$$



saat  $\ddot{\theta}_c \rightarrow 0$

$$\ddot{\theta}_c^{ref} = -\frac{K(\tau_c)}{I} \quad (3.12)$$

Berdasarkan persamaan (3.7), maka didapat persamaan berikut

$$\ddot{\theta}_c^{ref} = -\frac{K(\tau_m + \tau_s)}{I} \quad (3.13)$$

$$\ddot{\theta}_c^{ref} = -K_f(\tau_m + \tau_s) \quad (3.14)$$

Sedangkan pengendali untuk posisi dapat ditulis sebagai persamaan

$$\ddot{\theta}_d = \frac{d^2\theta_d}{dt^2} \quad (3.15)$$

Pada persamaan (3.6), dijabarkan bahwa nilai ideal posisi master dan slave adalah sama sehingga dapat dituliskan sesuai dengan persamaan berikut

$$\frac{\theta_m}{\theta_s} = 1 \quad (3.16)$$

Sehingga pada dimensi percepatan, perbandingan nilai percepatan master dan nilai percepatan slave yang ideal adalah sesuai dengan persamaan berikut

$$\frac{\theta_m}{\theta_s} = \frac{as^2 + bs + c}{as^2 + bs + c} \quad (3.17)$$

Persamaan diatas dapat dijabarkan sebagai berikut

$$a\ddot{\theta}_m + b\dot{\theta}_m + c\theta_m = a\ddot{\theta}_s + b\dot{\theta}_s + c\theta_s \quad (3.18)$$

$$a(\ddot{\theta}_m - \ddot{\theta}_s) = b(\dot{\theta}_s - \dot{\theta}_m) + c(\theta_s - \theta_m) \quad (3.19)$$

Berdasarkan persamaan (3.8) dan (3.15) didapat perumusan berikut

$$\dot{\theta}_d = -(K_p + K_v s)(\theta_m - \theta_s) \quad (3.20)$$

$$\ddot{\theta}_d = -K_d(\dot{\theta}_m - \dot{\theta}_s) \quad (3.21)$$

Sehingga dari persamaan (3.14) dan (3.21) dapat dituliskan menjadi

$$\begin{bmatrix} \ddot{\theta}_c \\ \ddot{\theta}_d \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_m \\ \ddot{\theta}_s \end{bmatrix} \quad (3.22)$$

Untuk mendapatkan masing-masing komponen master dan slave maka persamaan diatas dapat diubah menjadi persamaan sebagai berikut

$$\begin{bmatrix} \ddot{\theta}_m \\ \ddot{\theta}_s \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_c \\ \ddot{\theta}_d \end{bmatrix} \quad (3.23)$$

Dari persamaan (3.23) dan substitusi oleh persamaan (3.21) dan (3.14) maka didapat persamaan berikut

$$\ddot{\theta}_m = \frac{1}{2}K_d(\theta_s - \theta_m) - \frac{1}{2}K_f(\tau_m + \tau_s) \quad (3.24)$$

$$\ddot{\theta}_s = \frac{1}{2}K_d(\theta_m - \theta_s) - \frac{1}{2}K_f(\tau_m + \tau_s) \quad (3.25)$$

Sesuai dengan persamaan (3.17), pada salah satu bagian manipulator master atau slave, persamaan orde dua tersebut dapat didekati dengan persamaan berikut.

$$as^2 + bs + c = s^2 + 2\xi\omega_n s + \omega_n^2 \quad (3.26)$$

Sehingga didapat pendekatan nilai parameter sesuai persamaan berikut.

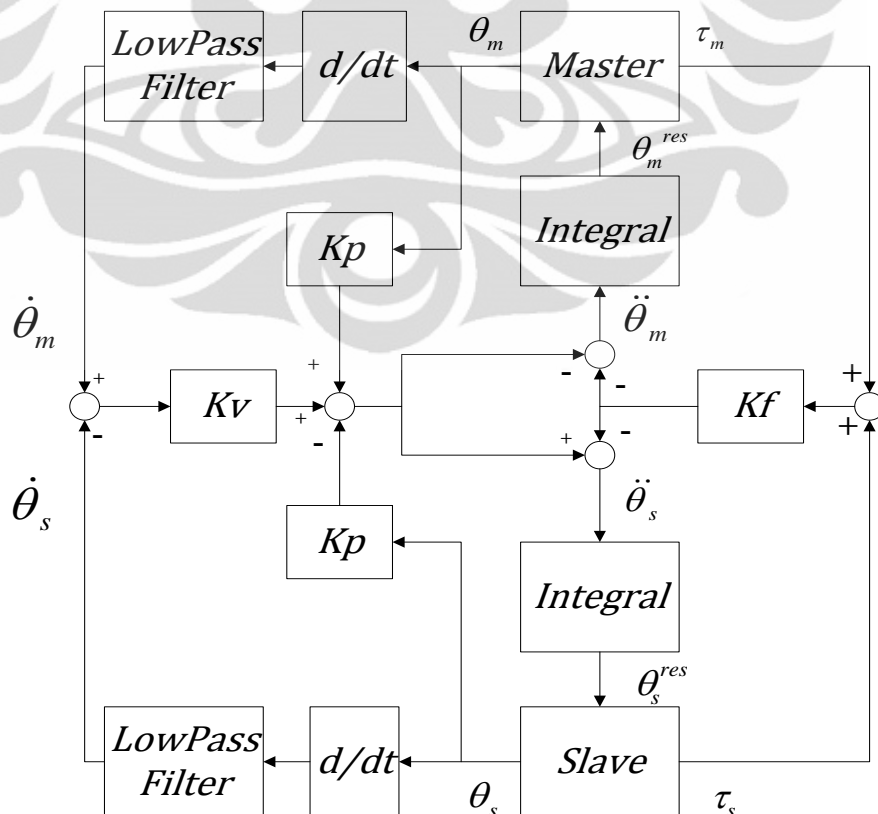
$$Kp = \omega_n^2 \quad (3.27)$$

$$Kv = 2\xi\omega_n \quad (3.28)$$

Dari persamaan (3.27) dan (3.28) maka parameter Kp nantinya akan disesuaikan dengan waktu cuplik sistem dan parameter Kv yang optimal akan mengikuti parameter Kp sesuai dengan persamaan berikut

$$Kv = 2 \cdot (0.7) \cdot \sqrt{Kp} \quad (3.29)$$

Apabila gambar 3.1 kita gambarkan dengan membuka pengendali posisi dan pengendali torsi maka akan dihasilkan blok diagram sebagai berikut.



Gambar 3.3 Blok Diagram sistem bilateral dengan pengendali

### 3.1.2 Perancangan *Bandwidth* Sistem dengan *Low Pass Filter*

Seperti yang telah dibahas pada subbab sebelumnya, bahwa sistem bilateral ini bekerja pada dimensi percepatan dikarenakan sistem ini menggunakan nilai torsi yang berhubungan secara linear pada percepatan. Oleh karena pada dynamixel pembacaan data yang dapat dilakukan adalah pembacaan data posisi dan beban, maka data posisi yang telah dibaca dynamixel, sebagai nilai referensi, akan diturunkan fungsinya. Penurunan langsung dengan menggunakan waktu cuplik ini sangat rentan terhadap gangguan (*noise*). Agar nilai turunan dari posisi ini dapat digunakan maka diperlukan sebuah *Low-Pass Filter* yang berfungsi untuk meredam frekuensi tinggi yang berperan sebagai noise. Filter yang digunakan adalah filter berorde satu yang memiliki fungsi alih sebagai berikut

$$\frac{Y(s)}{U(s)} = \frac{1}{\tau s + 1} \quad (3.30)$$

Dimana  $\tau$  adalah konstanta waktu yang berhubungan langsung dengan frekuensi seperti pada persamaan berikut.

$$f_c = \frac{1}{2\pi\tau} \quad (3.31)$$

Dimana  $f_c$  adalah *cut-off frequency* dengan satuan Hertz yang berfungsi sebagai batas frekuensi kritis yang dapat dilewati pada filter. Dari persamaan diatas dapat disimpulkan bahwa

$$\tau = \frac{1}{\omega_c} \quad (3.32)$$

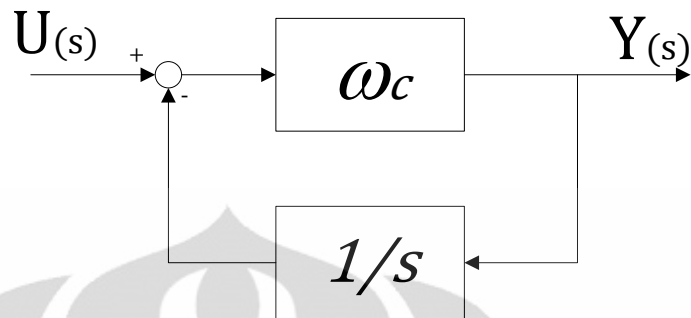
Dimana  $\omega_c$  adalah *angular frequency* dengan satuan rad/s. Seperti yang telah dijelaskan sebelumnya bahwa filter ini digunakan untuk nilai input hasil dari proses differensial dari posisi, sehingga fungsi alih dapat dirumuskan sebagai berikut

$$\frac{Y(s)}{sU(s)} = \frac{\omega_c}{s + \omega_c} \quad (3.33)$$

$$\frac{Y(s)}{U(s)} = \frac{s\omega_c}{s + \omega_c} \quad (3.34)$$

$$\frac{Y(s)}{U(s)} = \frac{\omega_c}{1 + \frac{\omega_c}{s}} \quad (3.35)$$

Sehingga dari persamaan (3.35) dapat dibentuk sebuah diagram blok sebagai berikut.



Gambar 3.4 Diagram blok fungsi alih low-pass filter pada sistem

Persamaan (3.29) dapat dijabarkan sebagai berikut

$$\frac{Y(s)}{U(s)} = \frac{\omega_c}{1 + \frac{\omega_c}{s}} \quad (3.36)$$

$$U(s) \cdot \omega_c = Y(s) \left(1 + \frac{\omega_c}{s}\right) \quad (3.37)$$

$$U(s) \cdot \omega_c = Y(s) + \omega_c \cdot \frac{Y(s)}{s} \quad (3.38)$$

$$Y(s) = \omega_c \cdot \left(U(s) - \frac{Y(s)}{s}\right) \quad (3.39)$$

$$Y(t) = \omega_c \cdot \left(U(t) - \int Y(t) \cdot dt\right) \quad (3.40)$$

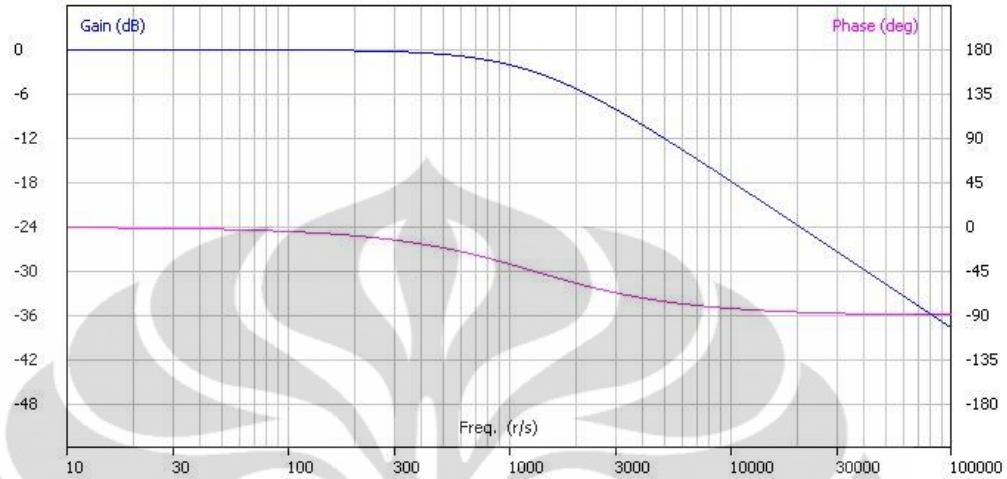
Sistem yang dibuat adalah sistem dengan waktu cuplik sebesar 5 ms. Dari waktu cuplik ini dapat dihitung bahwa perkiraan frekuensi sistem dengan persamaan berikut.

$$f = \frac{1}{T} \quad (3.41)$$

Dengan persamaan (3.41) diatas didapat bahwa perkiraan frekuensi sistem bekerja adalah 200 Hz. Agar hasil proses penurunan tetap optimal dan dapat bekerja pada waktu cuplik 5 ms atau setara dengan 200 Hz maka frekuensi cut-off harus bernilai minimal 200 Hz atau setara dengan 1256.63 rad/s. Pada sistem ini digunakan angular cut-off sebesar 1300 rad/s atau setara dengan 206.9 Hz, sehingga fungsi alih dapat dituliskan dengan persamaan berikut.

$$\frac{Y(s)}{sU(s)} = \frac{1300}{s + 1300} \quad (3.42)$$

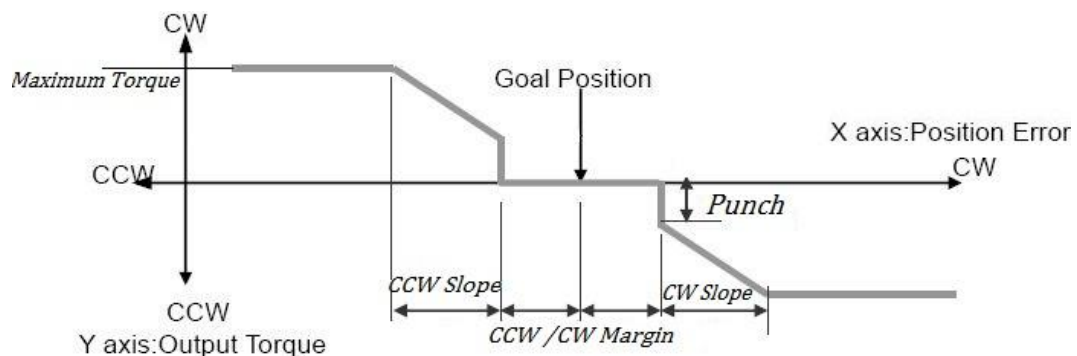
Nilai ini didapat dari metode pencarian nilai frekuensi cut-off yang dijelaskan pada bab selanjutnya. Dengan fungsi alih tersebut, maka sistem dapat dicitrakan dengan menggunakan *bode plot* seperti grafik berikut.



Gambar 3.5 Diagram Bode untuk Low-pass Filter pada sistem

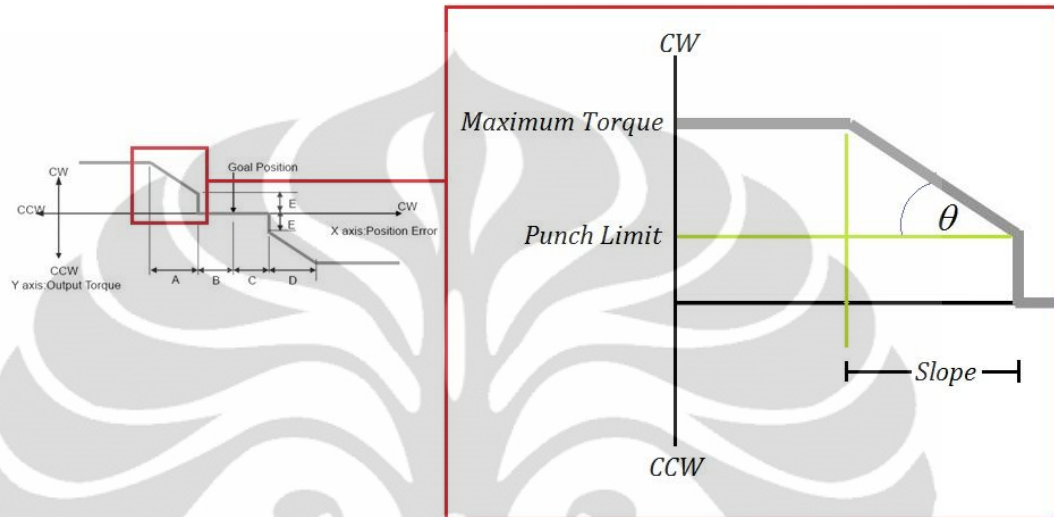
### 3.2 Pengaturan Torsi pada Aktuator Dynamixel AX-12+

Dynamixel memiliki fungsi tersendiri agar pengguna dapat mengatur distribusi torsi pada motor secara manual. Seperti pada gambar 3.6, terdapat beberapa parameter yang dapat diatur secara manual terhadap motor yaitu, besarnya nilai penurunan torsi dari nilai maksimum torsi menuju pada posisi yang diinginkan (*Slope*), besarnya nilai maksimum torsi (*Maximum Torque Limit*), besarnya nilai error yang diperbolehkan (*Margin*), dan besarnya nilai minimal ambang batas torsi pada posisi yang diinginkan (*Punch*). Masing-masing parameter ini dapat dibedakan nilainya untuk setiap arah yaitu arah berlawanan jarum jam dan searah jarum jam.



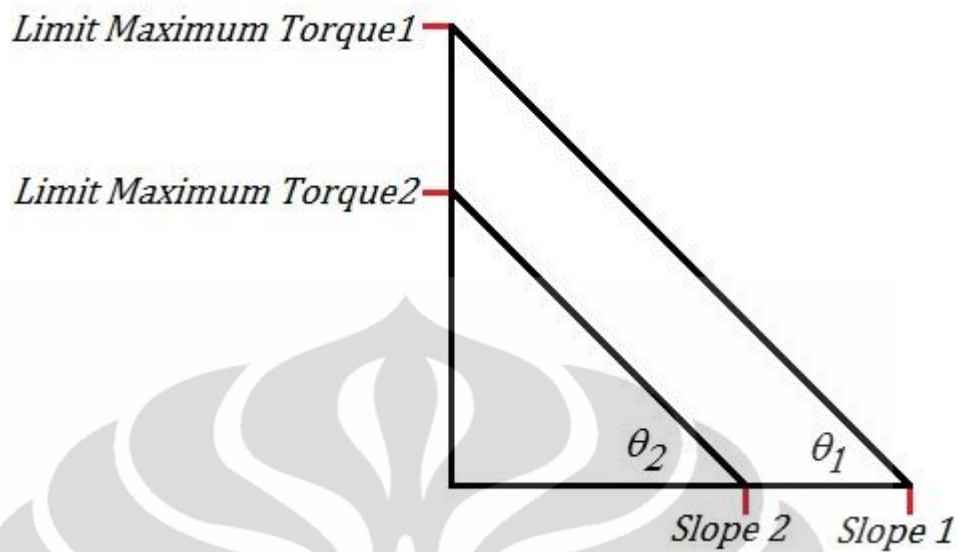
Gambar 3.6 Parameter yang digunakan pada Distribusi Torsi

Untuk sistem bilateral, distribusi torsi ini berlaku dalam sudut kemiringan torsi dikarenakan pergerakan posisi manipulator tidak akan pernah menyentuh nilai maksimum torsi karena pergerakan posisi pada manipulator dilakukan dalam skala posisi angular yang kecil. Oleh karena itu kemiringan distribusi torsi lebih sering digunakan.



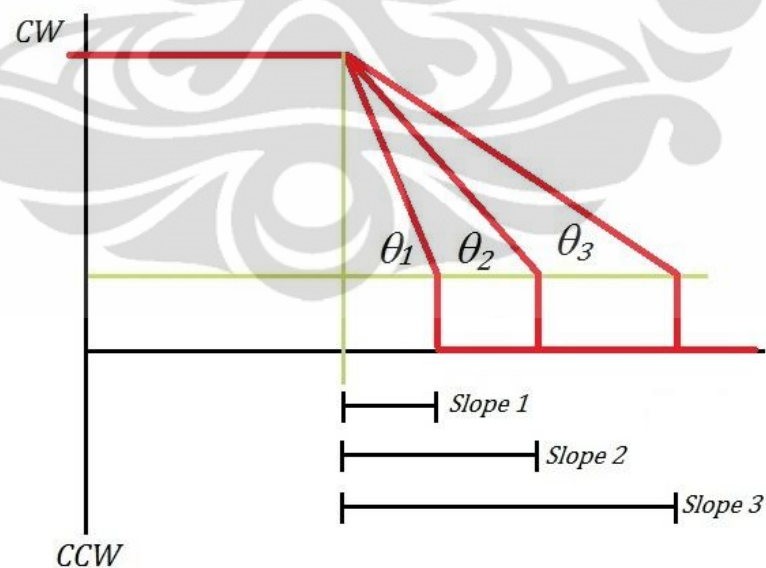
Gambar 3.7 Kemiringan sudut yang dibentuk oleh distribusi torsi

Secara logika apabila kita dapat mengatur kemiringan distribusi pada motor maka kita dapat mengatur kekuatan torsi motor tersebut di sekitar posisi yang diinginkan. Semakin landai sudut torsi maka akan semakin mudah untuk menggerakkan manipulator, sebaliknya semakin curam sudut torsi maka akan semakin sulit (berat) untuk menggerakkan manipulator. Hal ini terjadi karena pada saat sudut distribusi landai maka selang beberapa derajat dari posisi yang diinginkan memiliki nilai torsi yang lebih kecil dibandingkan dengan pada saat sudut distribusi torsi yang curam. Dari parameter-parameter yang dapat diatur, terdapat dua parameter dominan yang berpengaruh langsung dengan sudut distribusi torsi yaitu limit maksimum torsi dan besar nilai slope. Diantara kedua parameter tersebut salah satu parameter harus berlaku sebagai konstanta dan yang lain sebagai variabel. Hal ini dikarenakan apabila kedua parameter tersebut dapat diatur dan bersifat variabel maka akan terdapat beberapa variasi yang menghasilkan sudut yang sama pada distribusi torsi.



Gambar 3.8 Variasi yang menghasilkan hasil yang sama pada distribusi torsi

Seperti pada gambar 3.8, dapat dilihat besar nilai  $\theta_1$  sama dengan nilai pada  $\theta_2$ . Sudut distribusi torsi akan menghasilkan hasil yang sama dengan variasi perbandingan nilai yang sama. Oleh karena itu pada riset ini nilai yang dijadikan variabel adalah nilai slope dan nilai yang dijadikan konstanta adalah nilai maksimum torsi sehingga sudut-sudut distribusi torsi dapat bervariasi.



Gambar 3.9 Variasi sudut yang diperoleh dengan mengubah nilai slope

Nilai-nilai yang diperoleh dari parameter ini adalah nilai pada bilangan desimal. Nilai ini telah diubah kedalam bentuk digital yang semulanya adalah

nilai analog. Nilai ini memiliki arti sebagai resolusi posisi angular sesuai dengan rumusan berikut.

$$Resolusi = \frac{Sudut\ Maksimum}{1024} \quad (3.43)$$

Dengan rumusan diatas maka resolusi pada dynamixel AX-12+ yang memiliki kisaran jangkauan  $300^\circ$  adalah sebesar  $0.29296875^\circ$ . Maka resolusi setiap pertambahan nilai 1 ke dalam pembacaan berarti penambahan sebesar  $0.29298675^\circ$ .

Nilai yang digunakan sebagai nilai limit maksimum torsi adalah sebesar 50% yaitu setara dengan nilai 511. Sedangkan nilai yang dapat dihasilkan oleh variasi besar slope adalah mulai dari 1-254. Akan tetapi, nilai slope yang terlalu kecil yang dapat menghasilkan sudut yang terlalu curam dapat merusak gir dalam motor karena sudut tersebut tidak dapat mengkompensasi besarnya torsi pada posisi yang diinginkan. Oleh karena itu slope terkecil yang digunakan adalah berdasarkan nilai inisial pada dynamixel yaitu sebesar 32. Persamaan untuk mendapatkan sudut kemiringan distribusi torsi dapat dibentuk seperti persamaan berikut.

$$\theta = \tan^{-1} \frac{(Maksimum\ Torsi - Punch)}{Slope} \quad (3.44)$$

Dari batasan-batasan parameter dan persamaan diatas, maka sudut maksimum dan sudut minimum yang dapat dibentuk oleh variasi slope adalah sebagai berikut.

$$\theta_{MAX} = \tan^{-1} \frac{(511 - 32)}{32} = 86.17^\circ$$

$$\theta_{MIN} = \tan^{-1} \frac{(511 - 32)}{254} = 62.06^\circ$$

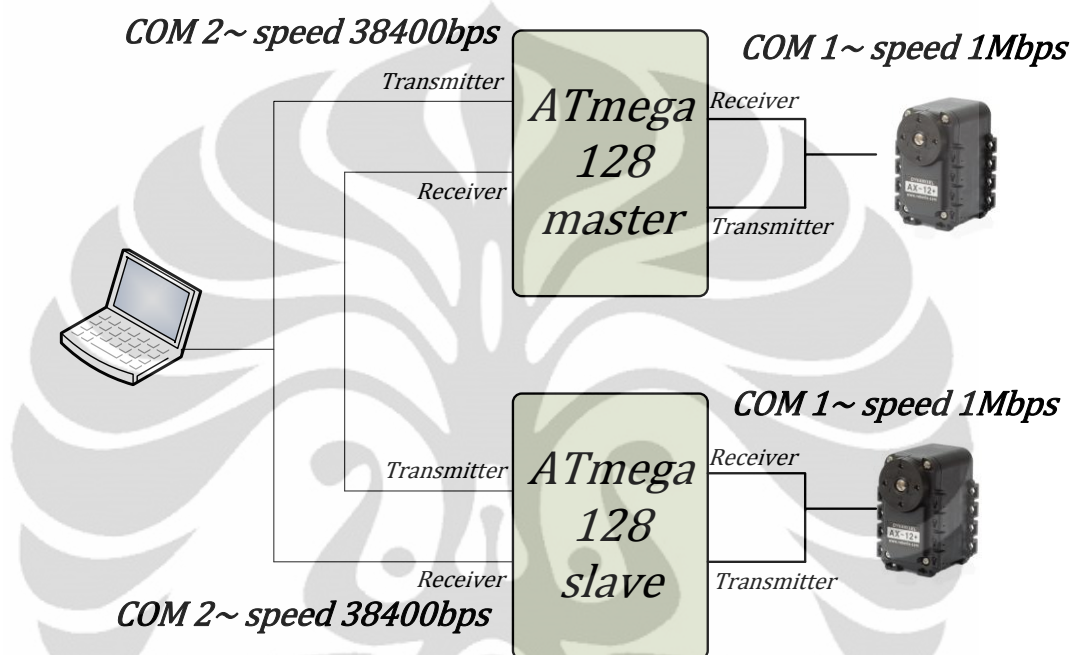
Dari hasil diatas maka dapat disimpulkan sudut kemiringan torsi dapat divariasikan antara  $62.06^\circ - 86.17^\circ$ .

### 3.3 Perancangan Perangkat Keras

Perangkat keras yang digunakan untuk menjadi sebuah sistem bilateral berupa dua buah pengendali berbasis ATmega128. Untuk mengakomodir kebutuhan manipulator master dan slave, motor servo serial Dynamixel AX-12+



dirangkai berikut penampangnya agar dapat membentuk lebih dari satu derajat kebebasan. Catu daya yang digunakan diadaptasi dari catu daya komputer agar dapat mengakomodir kebutuhan motor dalam bekerja. Selain itu untuk mengakuisisi data dibutuhkan PC yang memiliki koneksi USB sehingga data dari mikrokontroler dapat dilihat menggunakan PC. Berikut gambaran dari rangkaian yang digunakan.



Gambar 3.10 Komunikasi sistem bilateral

Perbedaan tipe komunikasi antara dynamixel yang menggunakan half-duplex dan ATmega128 yang menggunakan full-duplex menyebabkan diperlukannya perubahan sistem komunikasi. Pada lembar manual dynamixel terdapat rangkaian pengubah dari full-duplex menjadi half-duplex yang dapat dilihat pada gambar 2.9.

Pada rangkaian sebenarnya, untuk menjadikan mode komunikasi full-duplex menjadi half-duplex, dilakukan dengan cara menggabung pin transmitter dan receiver pada mikrokontroler secara langsung seperti pada gambar 3.10. Oleh karena itu, tidak dibutuhkan lagi pin yang membedakan proses pengiriman dan penerimaan. Untuk membedakannya dilakukan dengan pengaturan pada dalam mikrokontroler itu sendiri yang akan dibahas pada subbab selanjutnya.

Sementara itu untuk komunikasi antar pengendali dalam hal ini mikrokontroler dilakukan dengan mode komunikasi full-duplex sehingga pin TX

pada mikrokontroler master akan disambung pada pin RX pada mikrokontroler slave dan begitu pula sebaliknya. Untuk meminimalisir terjadinya kesalahan, maka port pada mikrokontroler yang berfungsi sebagai USART baik COM-1 maupun COM-2 di pull-up secara internal untuk menghilangkan *floating value*.

### 3.3.1 Catu Daya dan Penampang

Untuk memenuhi kebutuhan sumber tenaga listrik pada perangkat yang digunakan maka digunakan catu daya berupa catu daya yang biasa digunakan pada komputer. Hal ini dipertimbangkan karena catu daya tersebut memiliki keluaran arus yang besar sehingga dapat mengakomodir lonjakan arus yang dibutuhkan oleh peralatan terutama motor servo.

Sementara itu, selain menggunakan penampang asli yang diperoleh dari dynamixel itu sendiri, dibuat dua buah penampang lagi yang digunakan untuk menghubungkan antara sendi terdiri dari dua jenis yaitu penampang pertama dengan panjang 35 mm dan penampang kedua dengan panjang 135 mm. Penampang untuk tempat duduk dynamixel juga digunakan agar dynamixel dapat bekerja dengan lebih stabil. Penampang-penampang ini dapat dilihat pada gambar berikut.



Gambar 3.11 Penampang pada sistem bilateral yang telah dibuat

### 3.4 Perancangan Perangkat Lunak

Perancangan perangkat lunak yang paling penting adalah pemisahan data yang diterima dan dikirim karena perbedaan mode komunikasi serial antara dynamixel dan mikrokontroler ATmega128. Dengan cara menggabungkan langsung pin TX dan RX ada ATmega128 maka harus dibuat sebuah mekanisme untuk memisahkan proses pengiriman dan penerimaan data. Seperti yang telah dibahas sebelumnya, pemisahan ini perlu dilakukan mengingat terdapat dua buah register UDR yang berfungsi membaca dan menulis yang memiliki satu alamat I/O (*Sharing address*). Apabila tidak dipisahkan maka akan terjadi kesalahan proses penulisan isi register (pengiriman) dan pembacaan isi register (penerimaan). Oleh karena itu dibutuhkan pengaturan pada 8-bit register kontrol dan status yang ada pada mode komunikasi USART.

Pada saat pertama kali data akan dikirimkan maka nilai dari register UCSRB akan diset 1 pada bit ke-3 yaitu bit TXEN dan menonaktifkan pin lainnya sehingga hanya bit TXEN yang berfungsi sebagai aktifasi pin TX sajalah yang aktif. Setelah bit TXEN diaktifkan maka pengiriman dapat berlangsung. Setelah pengiriman selesai, pengaturan harus mengaktifkan pin RX dan menonaktifkan pin TX. Agar tidak terjadi kesalahan maka indikasi bahwa proses pengiriman telah selesai harus diketahui. Untuk itu digunakan register status dan kontrol UCSRA yang akan mengecek bit UDRE yang berfungsi sebagai indikator bahwa register pengiriman UDR telah kosong. Proses ini akan menunggu sampai bit UDRE menjadi satu yang berarti register pengiriman UDR telah kosong setelah itu pin RX akan diaktifkan dengan cara mengaktifkan bit RXEN pada register UCSRB.

Penggunaan komunikasi antara dynamixel dengan mikrokontroler tidak menggunakan tempat penyimpanan sementara (*buffer*) karena penggunaan tempat penyimpanan sementara akan memperlambat kinerja komunikasi serial antara dynamixel dan mikrokontroler. Oleh karena itu, setiap data yang diterima oleh mikrokontroler dari dynamixel harus tepat. Untuk mengakomodir kebutuhan itu maka setiap setelah menerima 1 byte data dari dynamixel maka sebuah register UCSRA pada bit ke-7 yaitu bit RXC harus diperhatikan karena bit ini akan

bernilai 1 pada saat data telah diterima. Untuk diagram alur lebih lengkapnya akan dibahas pada subbab selanjutnya.

### 3.4.1 Pengaturan *Baud-rate* dan waktu cuplik pada Mikrokontroler

Timer/Counter adalah fungsi pada mikrokontroler yang paling sering digunakan untuk mengukur nilai yang berhubungan dengan waktu. Untuk ATmega128 terdapat 2 buah 8-bit Timer (Timer-0 dan Timer-2) dan 2 buah 16-bit Timer (Timer-1 dan Timer-3). Timer ini, terutama untuk timer 16-bit, memiliki register tersendiri yaitu *Timer Counter Control Register* (TCCR<sub>x</sub>). Untuk dapat menentukan kecepatan timer dapat diadaptasi dari nilai *system clock* ataupun masukan lain dari pin eksternal. Pada register TCCR terdapat 8-bit dengan 3-bit yang paling kecil berfungsi menentukan kecepatan timer yang akan digunakan seperti tabel berikut

Tabel 3.1 Bit pada register TCCR

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					CS12	CS11	CS10

Untuk menentukan berapa kecepatan timer tersebut 3 bit paling akhir (CS12,CS11,CS10) dapat diisikan data seperti yang terdapat pada tabel berikut ini

Tabel 3.2 Fungsi *Prescaler* pada kinerja timer

CS12	CS11	CS10	Fungsi Clock
0	0	0	Timer 1 Tidak berfungsi
0	0	1	Kecepatan = System Clock
0	1	0	Kecepatan = System Clock/8
0	1	1	Kecepatan = System Clock/64
1	0	0	Kecepatan = System Clock/256
1	0	1	Kecepatan = System Clock/1024
1	1	0	Pin Eksternal T1, Hitung pada saat <i>a falling edge</i>
1	1	1	Pin Eksternal T1, Hitung pada saat <i>a rising edge</i>

Mode yang digunakan untuk menentukan waktu cuplik adalah mode *Clear Timer on Compare* (CTC) dengan *Interrupt on Compare A match* (OCRnA)

dengan kecepatan 2 MHz ( System Clock/8). Dengan mode ini nilai counter/timer akan menjadi 0 kembali (*clear*) pada saat nilai counter/timer pada register TCNTn sesuai dengan nilai OCRnA. Nilai OCRnA akan disesuaikan nilai puncaknya sesuai dengan waktu cuplik yang diinginkan.

Pada saat nilai yang dicocokkan sesuai maka program yang ada pada fungsi interupsi akan dijalankan. Program yang ada pada fungsi ini adalah mengaktifkan flag sebagai penanda waktu cuplik dimulai. Pada fungsi program utama, program akan menunggu sampai flag bernilai satu. Setelah itu program jalan dan flag di reset kembali dan program kembali menunggu sampai flag bernilai 1 dan begitu seterusnya.

Untuk komunikasi antara mikrokontroler dengan mikrokontroler memiliki baudrate yang berbeda dengan komunikasi antara mikrokontroler dengan dynamixel. Penentuan nilai baudrate ini dapat dirumuskan seperti pada persamaan (2.3). Baudrate yang digunakan untuk komunikasi ATmega dengan dynamixel digunakan 1Mbps sedangkan komunikasi antara mikrokontroler dengan mikrokontroler digunakan 38400 bps.

Antara ATmega dengan dynamixel dibutuhkan baudrate maksimal dikarenakan banyaknya byte perintah maupun status yang dikirimkan dalam 1 paket data. Untuk 1 paket data normal berkisar 8-10 byte yang dikirimkan. Sedangkan untuk pengiriman paket data yang memiliki banyak parameter pengiriman dapat mencapai 12-16 byte dalam sekali pengiriman untuk satu perintah. Oleh karena itu agar dalam satu kali waktu cuplik, data yang dikirimkan dapat terakomodir maka baudrate maksimal yang digunakan Antara ATmega dan dynamixel.

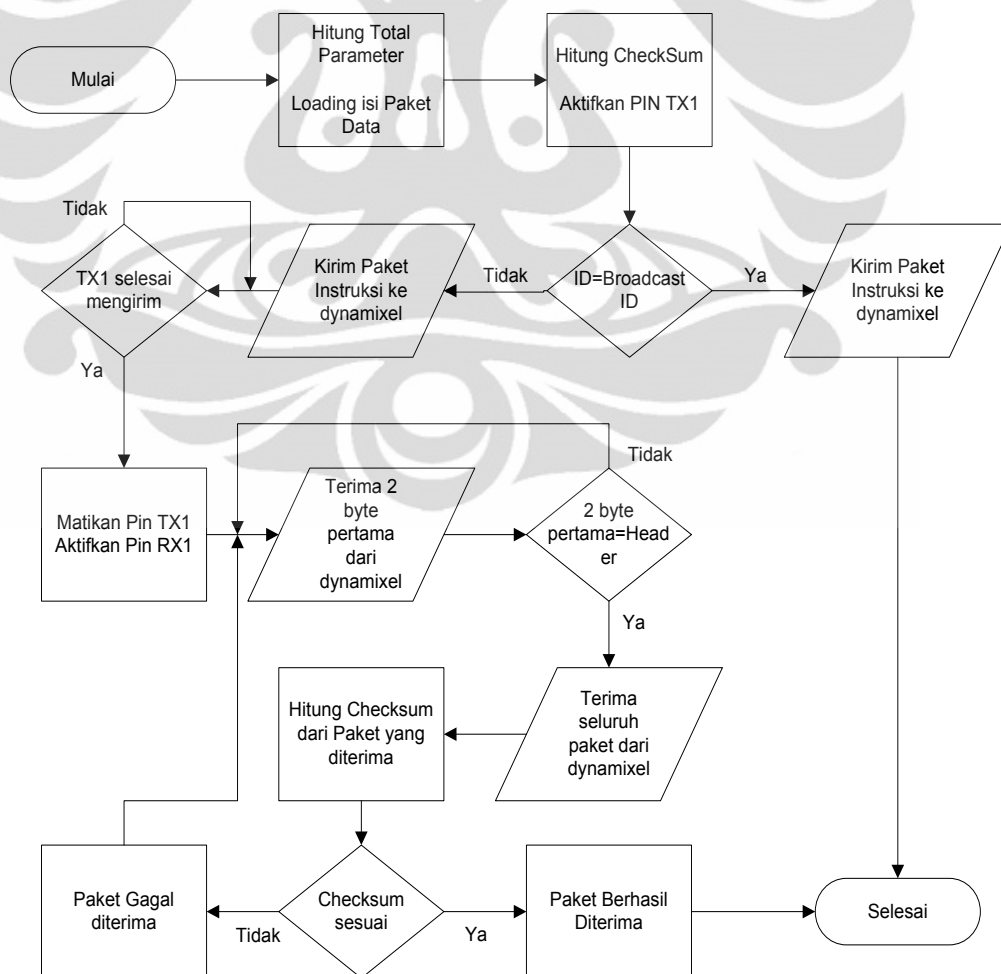
Untuk hubungan antar mikrokontroller digunakan baudrate sebesar 38400 bps dikarenakan tidak terlalu banyak data yang dikirimkan. Data yang dikirimkan biasanya berupa data posisi 2-byte dan data force 2-byte untuk masing-masing sendi. Sehingga jumlahnya masih dapat terakomodir. Selain itu pemilihan baudrate 38400 bps agar jalur komunikasi ini tetap dapat dimonitoring oleh PC yang kecepatannya tidak sampai dengan 1Mbps.

### 3.4.2 Diagram Alur Program

Pada subbab ini akan menjelaskan diagram alur dan program yang diaplikasikan pada pengendali berupa mikrokontroler ATmega128.

#### 3.4.2.1 Pengiriman dan Penerimaan Paket Data

Pengiriman dan penerimaan paket data merupakan hal yang paling krusial dalam komunikasi antara pengendali dengan aktuatornya. Informasi-informasi yang dikirimkan maupun yang diterima harus tepat secara nilai dan waktu. Informasi-informasi ini berupa status aktuator, perintah untuk aktuator dan pembacaan kondisi aktuator yang akan diproses lebih lanjut oleh pengendali dalam hal ini berupa ATmega128. Pengiriman dan penerimaan data seperti yang telah dibahas sebelumnya menggunakan mode half-duplex sehingga antara pengiriman data dan penerimaan data dilakukan secara bergantian. Berikut adalah diagram alur dari pengiriman dan penerimaan data.



Gambar 3.12 Diagram alur transfer data antara aktuator dan pengendali

Sesuai diagram alur pada gambar 3.12, pertama kali yang harus dilakukan adalah mengisi paket data yang sesuai untuk dikirimkan. Setelah itu pin TX1 baru diaktifkan. Setelah paket siap untuk dikirim, ID dynamixel terlebih dahulu dicek apakah ID tersebut adalah ID yang unik atau termasuk dalam ID yang umum yang dapat mengendalikan semua dynamixel (*Broadcast ID*). Hal ini diperlukan karena apabila ID berupa sebuah ID broadcast maka paket status ,paket yang dikirimkan oleh dynamixel kepada pengendali, tidak akan dikirimkan. Apabila hal ini tidak dipisahkan maka pengendali yang merupakan ATmega 128 yang menggunakan USART tanpa buffer akan terus menunggu paket sampai paket diterima. Hal ini akan menghasilkan program yang salah. Dapat dilihat apabila paket dikirimkan kepada ID broadcast maka setelah pengiriman paket intruksi fungsi ini selesai tanpa menunggu paket status yang datang dari dynamixel.

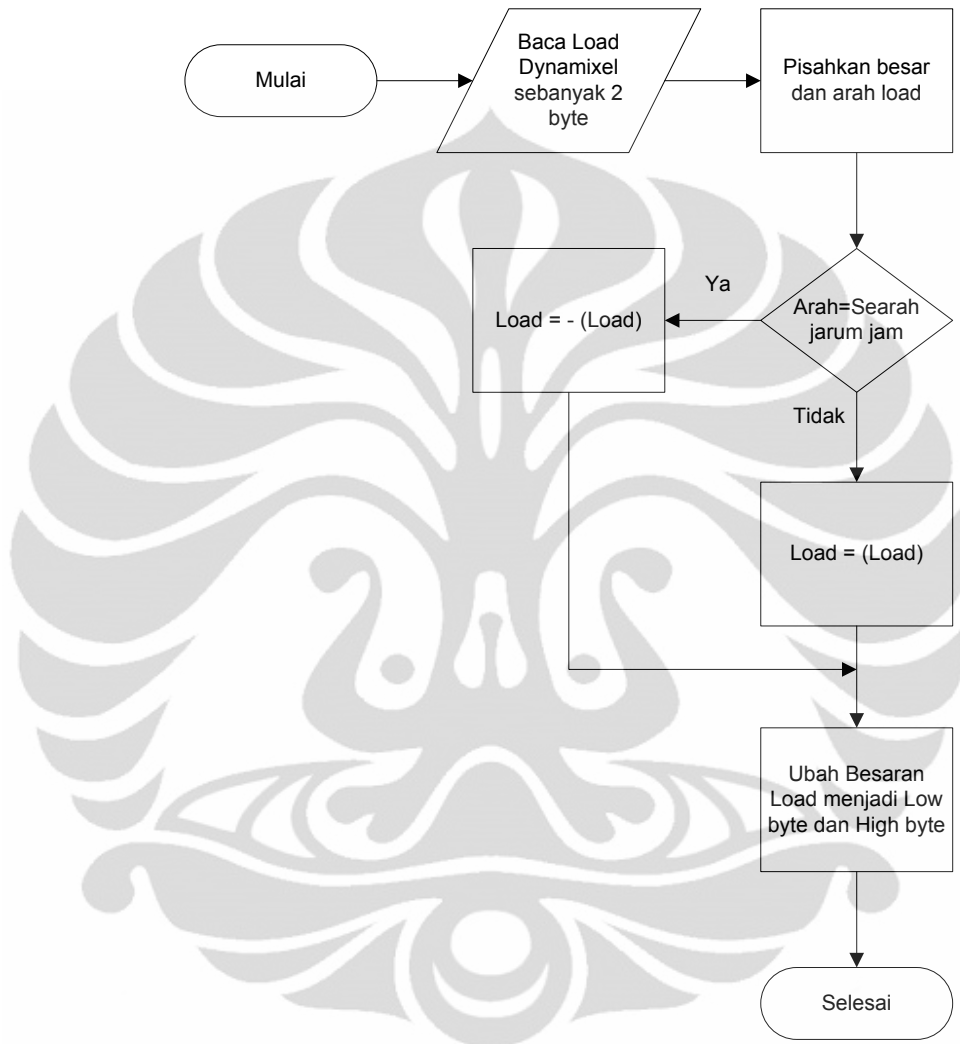
Apabila paket instruksi dikirimkan kepada ID selain ID broadcast maka setelah proses pengiriman paket maka fungsi ini akan menunggu sampai paket data pada byte terakhir benar-benar telah terkirim dengan mengecek register pengirim yang telah kosong. Apabila kondisi ini dipenuhi maka pin TX akan dimatikan dan pin RX akan diaktifkan dan siap untuk menerima paket status dari dynamixel. Untuk memperkecil kesalahan data yang diterima, fungsi ini menyaring 2 byte pertama yang seharusnya bernilai sama dengan header yang telah ditentukan. Fungsi ini akan terus mengambil data sampai 2 byte awal data adalah byte header. Setelah 2 byte benar adalah header dari dynamixel maka sisa paket selanjutnya akan diterima dengan jumlah byte yang telah diketahui sebelumnya.

Setelah itu sebagai media konfirmasi terakhir, byte stop dari paket status akan dihitung secara manual dan dicocokkan dengan byte terakhir sebagai upaya memperkecil kesalahan penerimaan paket. Setelah semua dipenuhi maka informasi dari paket status tersebut siap untuk digunakan lebih lanjut.

#### **3.4.2.2 Pembacaan Load dan Pemisahan Nilai Load dan Arah**

Proses pengiriman paket instruksi dan penerimaan paket status seperti diatas, diaplikasikan untuk segala bentuk pengiriman paket antara pengendali dan aktuator yang digunakan untuk membaca posisi, memerintahkan pergerakan ke suatu posisi, membaca load, mengatur nilai torsi dan lain sebagainya. Untuk

pembacaan load agak berbeda dikarenakan dalam satu paket terdapat nilai dan arah load. Nilai dan arah ini harus dipisahkan agar tidak membingungkan dan mempermudah dalam pengolahan data. Diagram alur untuk pembacaan beban dapat dilihat sebagai gambar berikut.



Gambar 3.13 Diagram alur pembacaan beban pada aktuator

Pemisahan ini dilakukan juga untuk membedakan nilai agar mudah dideteksi dalam grafik. Setelah dipisahkan arah beban searah jarum jam dan berlawanan arah jarum jam maka untuk searah jarum jam nilainya akan bernilai minus dan apabila berlawanan arah jarum jam akan bernilai positif.



### 3.4.2.3 Auto-Tuning Torque Angle yang akan diterapkan pada sistem

Untuk akhir dari pengendalian sistem bilateral 4 kanal ini, digunakan sebuah algoritma yang akan membaca kondisi sistem dan akan selalu mengubah nilai sudut distribusi torsi pada manipulator tersebut seperti yang telah dijelaskan pada subbab 3.2. Fungsi dari algoritma ini adalah mengurangi nilai perbedaan posisi antar manipulator sehingga sistem dapat bekerja secara optimal.

Algoritma ini bekerja pada saat perbedaan posisi yang telah ditentukan dan akan mengubah nilai slope yang berakibat berubahnya nilai sudut distribusi torsi.

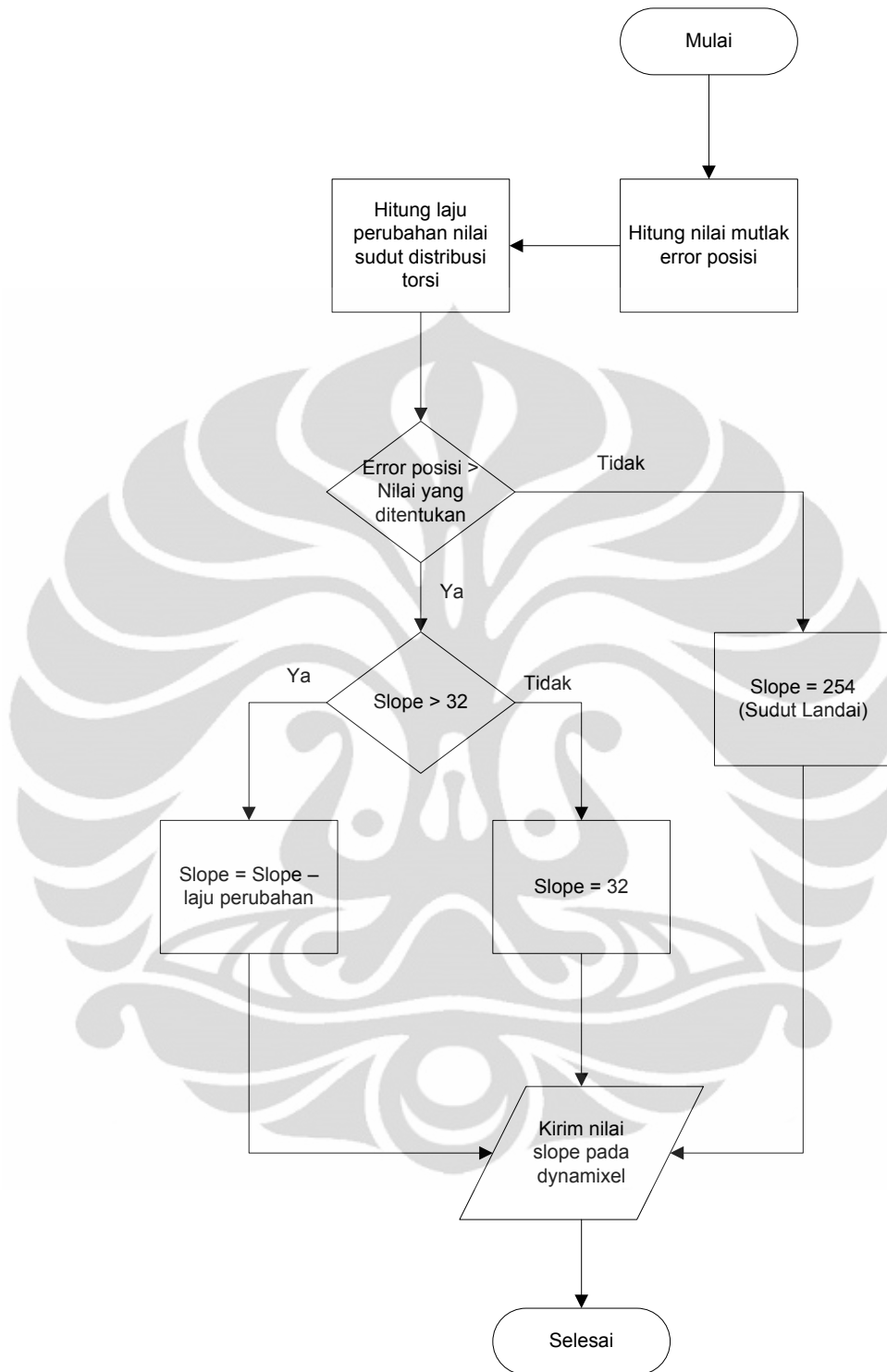
Variasi nilai slope ,seperti yang dijelaskan pada subbab 3.2 sebelumnya tentang pengaturan sudut torsi pada aktuator dynamixel AX-12+, mengakibatkan berubahnya sudut distribusi torsi sehingga sudut torsi menjadi landai atau curam. Sudut torsi yang curam mengakibatkan manipulator lebih sulit digerakkan sedangkan sudut torsi yang landai mengakibatkan manipulator lebih mudah untuk digerakkan sehingga operator dapat dengan ringan menggerakkan manipulator pada sistem teleoperasi bilateral.

Dapat dilihat pada gambar 3.14, apabila perbedaan posisi antar manipulator (error) lebih besar dari nilai yang ditentukan maka sudut distribusi torsi akan bertambah curam setiap waktu cuplik dengan nilai maksimal kecuraman sebesar  $86.17^\circ$  yang setara dengan nilai slope sebesar 32. Sedangkan apabila perbedaan posisi antar manipulator (error) lebih kecil dari nilai yang ditentukan maka sudut distribusi kembali landai dengan sudut  $62.06^\circ$  atau sebanding dengan nilai slope sebesar 254.

Laju perubahan nilai sudut distribusi torsi dihitung dengan persamaan berikut

$$\text{Laju Perubahan} = \frac{\text{Persentase laju perubahan}}{100} \times 48 \quad (3.45)$$

Nilai 48 adalah nilai perubahan maksimal slope untuk setiap waktu cuplik, sehingga apabila persentase laju perubahan diset menjadi 100% maka slope akan berkurang sebesar 48 setiap waktu cuplik. Sedangkan apabila persentase laju perubahan diset menjadi 10% maka slope akan berkurang sebesar 4 setiap waktu cuplik.



Gambar 3.14 Algoritma *Auto-Tuning Torque Angle*

Rincian langkah-langkah prosedur pemakaiannya akan diterangkan pada bab selanjutnya.

## **BAB 4**

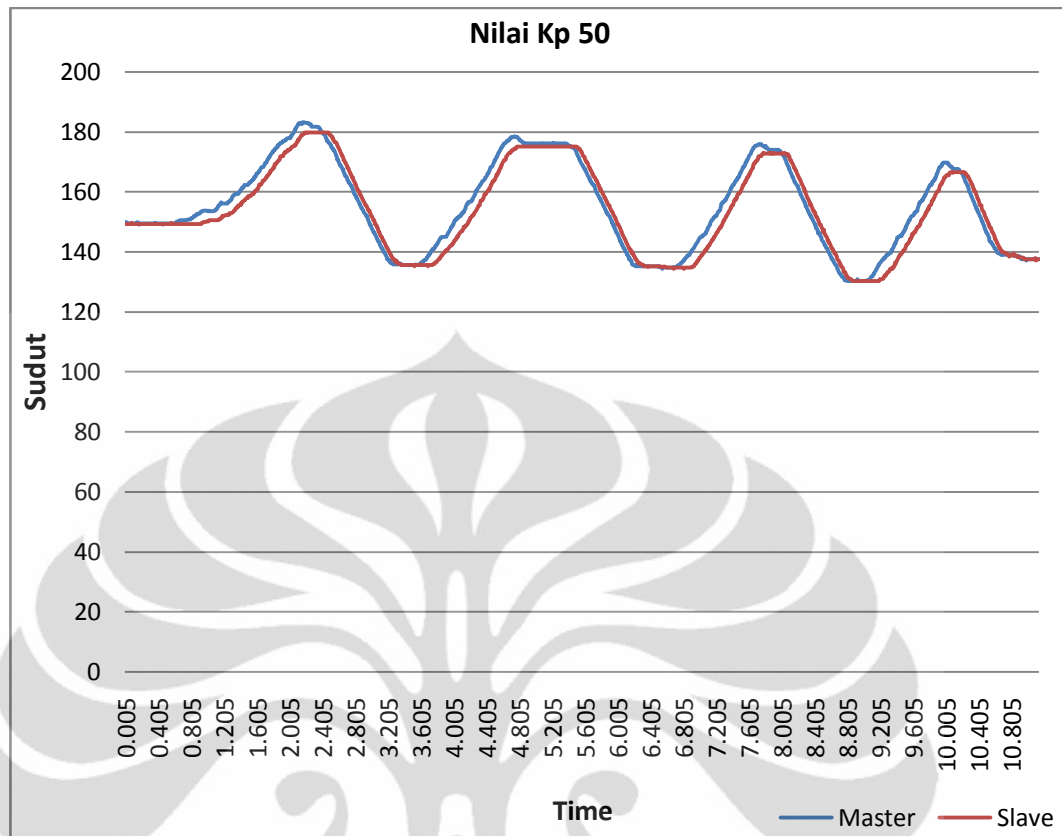
### **HASIL PENGUJIAN DAN ANALISIS**

Untuk memvalidasi rancangan sistem teleoperasi bilateral yang telah dibuat, pada bab ini akan dilakukan pengujian dari rancangan yang telah dibuat. Beberapa parameter yang telah ditentukan pada perancangan sebelumnya juga akan diujicobakan agar mendapat hasil yang optimal. Pengujian yang akan dilakukan adalah penentuan parameter konstanta pengendali, frekuensi cut off pada filter, penentuan konstanta gaya, variasi nilai sudut distribusi torsi aktuator, dan implementasi algoritma penyesuaian torsi secara otomatis untuk sistem yang memiliki 2 derajat kebebasan.

#### **4.2 Penentuan Nilai Konstanta $K_p$ dan $K_v$ pada Sistem Bilateral**

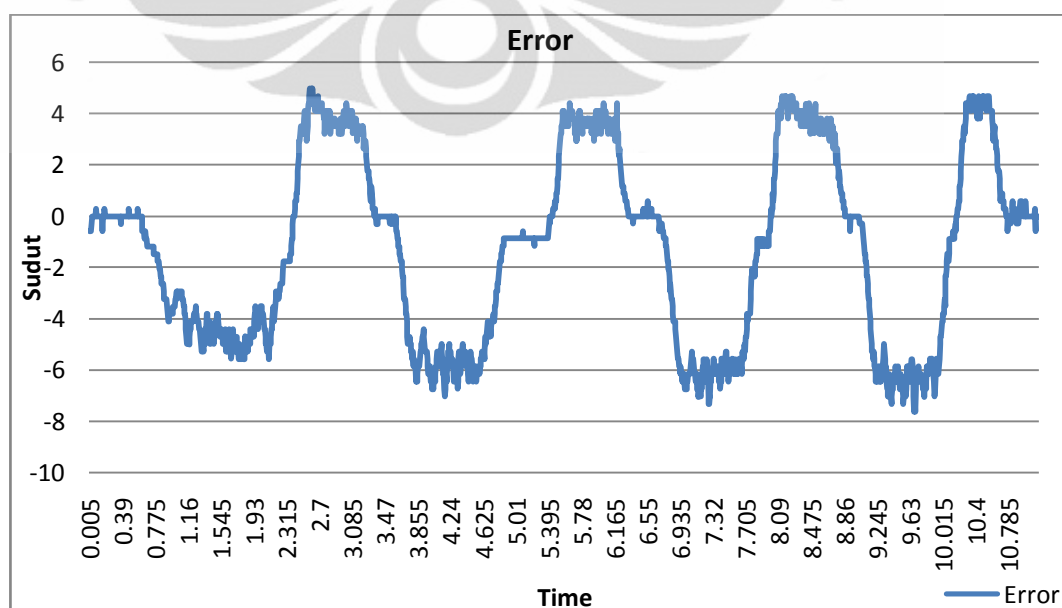
Nilai konstanta  $K_p$  yang ideal adalah mendekati 1. Sehingga apabila sebuah sistem bekerja dengan hasil yang menggunakan operasi integral dan sistem bekerja pada waktu cuplik tertentu maka nilai  $K_p$  yang sesuai adalah nilai pembuat satu. Hal ini dikarenakan hasil akhir pada sistem yang dikalikan oleh waktu cuplik sesuai dengan operasi integral. Oleh karena itu, dengan sistem yang sama dengan penjelasan diatas dan dengan waktu cuplik sebesar 5 ms, maka nilai  $K_p$  yang sesuai adalah sebesar 200.

Pengujian-pengujian yang akan dilakukan adalah pengujian pencarian nilai  $K_p$  dan  $K_v$  yang sesuai dengan kriteria. Berikut ini akan ditampilkan hasil pengujian apabila nilai  $K_p$  yang digunakan berbeda jauh dari nilai ideal optimalnya yaitu 50.



Gambar 4.1 Grafik posisi master dan slave saat nilai Kp 50

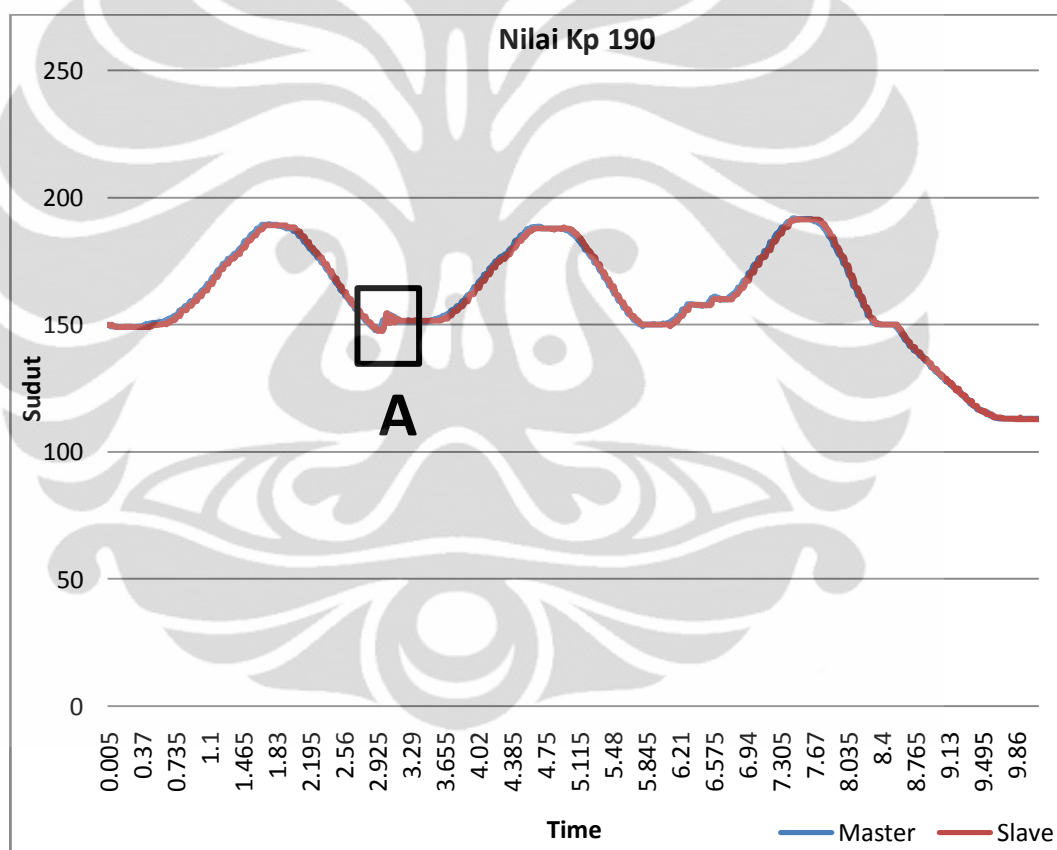
Dapat dilihat apabila nilai Kp yang digunakan untuk sistem terlalu kecil, maka manipulator slave pada sistem akan tetap dapat mengikuti manipulator master dengan respon yang lambat sehingga memiliki nilai perbedaan posisi yang cukup signifikan.



Gambar 4.2 Grafik perbedaan posisi master dan slave saat nilai Kp 50

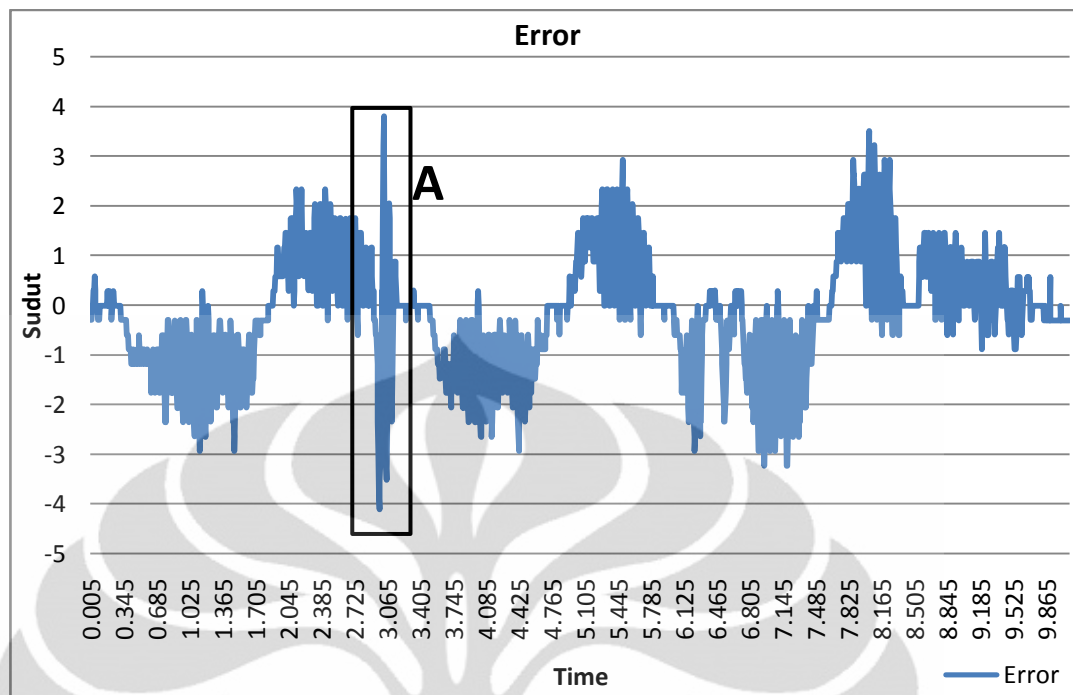
Dapat dilihat bahwa nilai maksimum error yang terjadi oleh perbedaan posisi antara master dan slave mencapai  $4.98^\circ$  dengan rata rata nilai  $2.62^\circ$  pada arah putar searah jarum jam dan mencapai  $7.61^\circ$  dengan rata-rata nilai sebesar  $3.6^\circ$  pada arah berlawanan jarum jam. Hal ini disebabkan nilai konstanta  $K_p$  yang terlalu kecil dibandingkan dengan waktu cuplik sistem. Walaupun demikian sistem masih dapat dikatakan stabil walaupun dengan respon yang lambat.

Untuk mengurangi nilai kesalahan tersebut maka nilai  $K_p$  akan dinaikkan mendekati nilai yang ideal yaitu 200. Berikut adalah hasil pengujian selanjutnya dengan nilai  $K_p$  sebesar 190.



Gambar 4.3 Grafik posisi master slave saat nilai  $K_p$  190

Dapat dilihat bahwa pergerakan manipulator master dan slave menjadi lebih baik dibandingkan pengujian sebelumnya. Akan tetapi terdapat sedikit osilasi pada pergerakannya yang dapat dilihat pada kotak berlabel 'A' pada gambar 4.3 diatas.

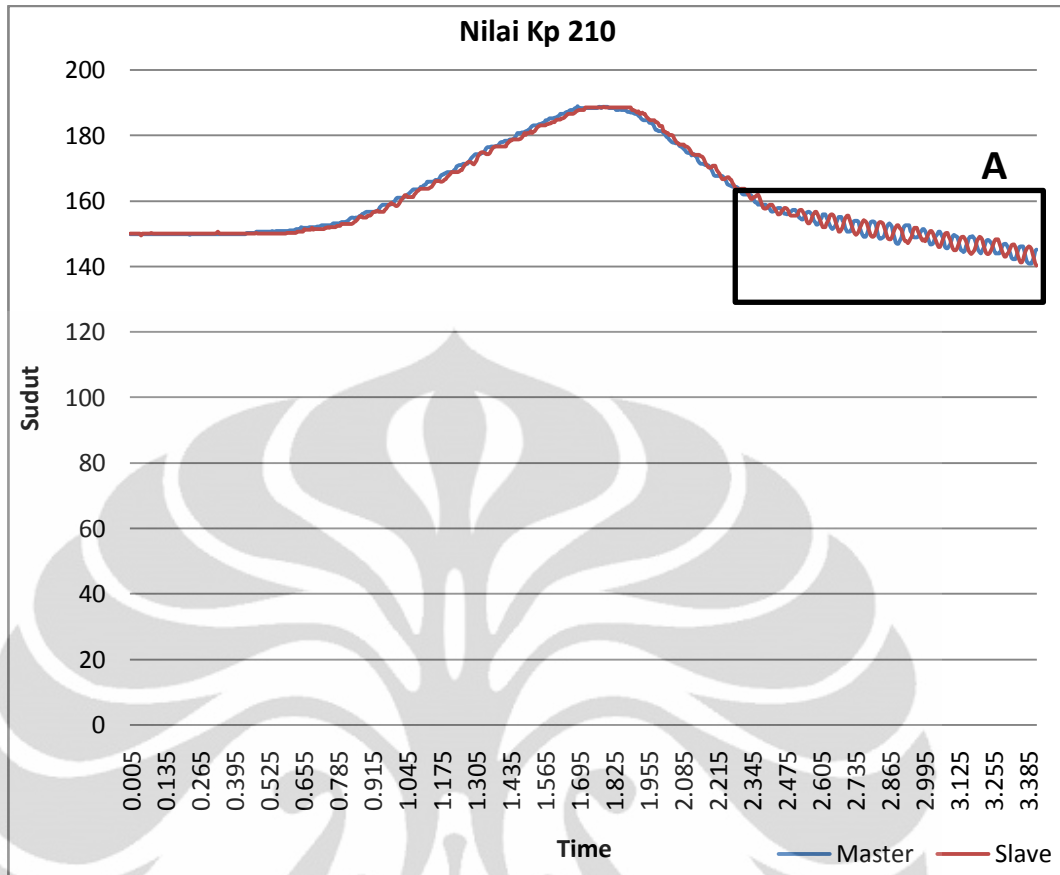


Gambar 4.4 Grafik perbedaan posisi master dan slave saat nilai Kp 190

Dari grafik kesalahan diatas, terlihat bahwa dengan menggunakan nilai Kp yang mendekati nilai waktu cuplik maka kesalahan akan diminimalisir. Dapat dilihat juga bahwa terjadi sedikit osilasi pada bagian kotak berlabel 'A' diatas. Hal ini disebabkan oleh nilai Kp yang masih sedikit terlalu besar. Nilai kesalahan tertinggi apabila sistem ini terjadi tanpa osilasi adalah sebesar  $3.52^\circ$  dengan nilai rata-rata  $0.89^\circ$  pada bagian searah jarum jam dan  $3.22^\circ$  dengan nilai rata-rata  $1.21^\circ$  pada arah berlawanan jarum jam. Akan tetapi karena terdapat osilasi yang menyebabkan nilai kesalahan maksimum naik menjadi  $3.81^\circ$  pada searah jarum jam dan  $4.1^\circ$  pada searah jarum jam.

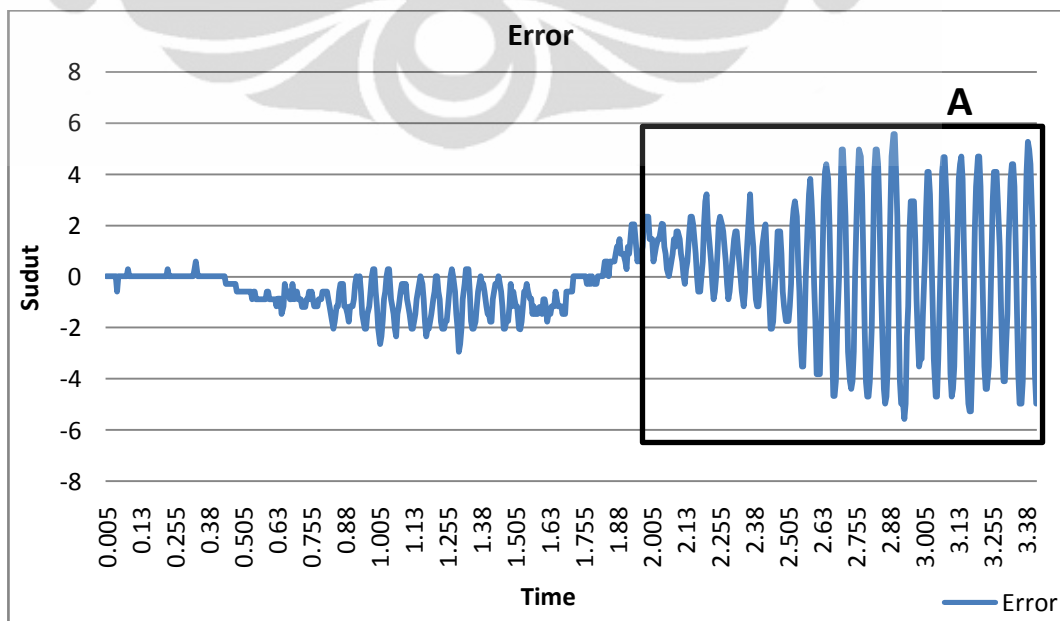
Dapat dilihat bahwa apabila nilai Kp terlalu kecil digunakan maka sistem tetap stabil akan tetapi dengan nilai perbedaan yang cukup besar dengan kata lain repon sistem lambat. Sementara apabila nilai Kp yang terlalu besar akan menyebabkan sistem tidak stabil dan berakibat sistem yang berosilasi.

Sebagai contoh dapat dilihat pada grafik berikut yang menggunakan nilai Kp sebesar 210 yang lebih besar dari nilai yang ideal berdasarkan waktu cuplik.



Gambar 4.5 Grafik posisi master dan slave saat nilai Kp 210

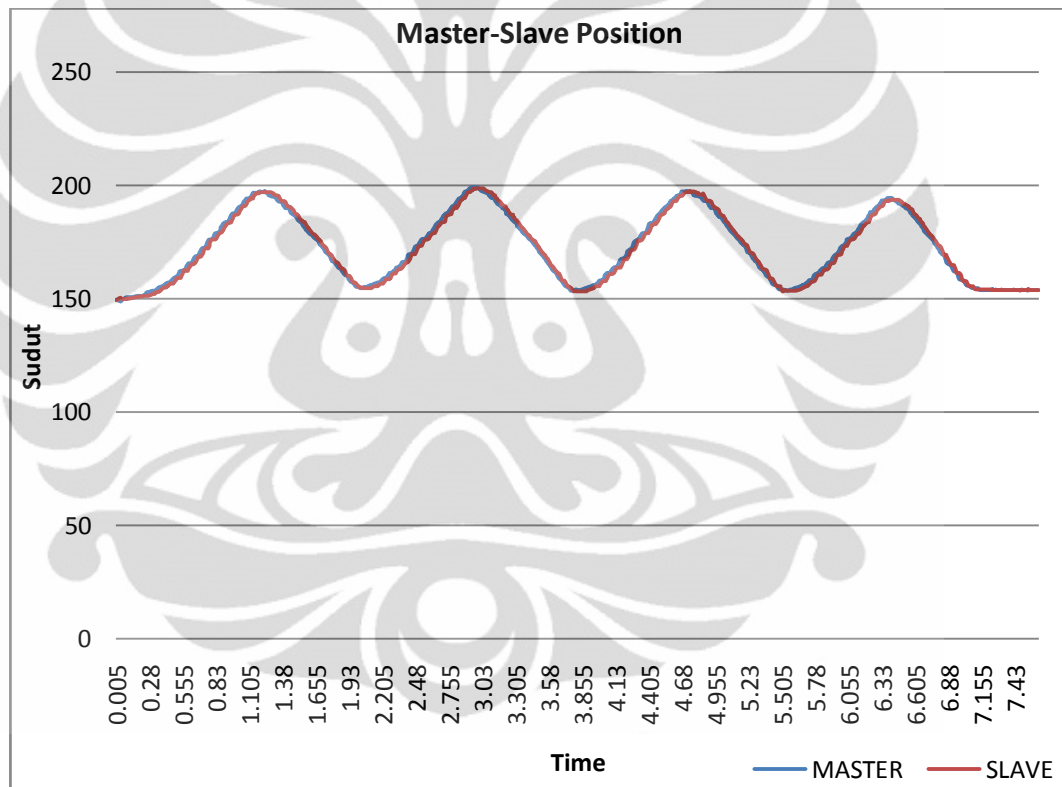
Dapat dilihat bahwa nilai yang terlalu besar mengakibatkan sistem yang apabila terkena gangguan maka akan beresilasi sehingga nilai kesalahannya pun semakin tinggi.



Gambar 4.6 Grafik perbedaan posisi master dan slave saat nilai Kp 210

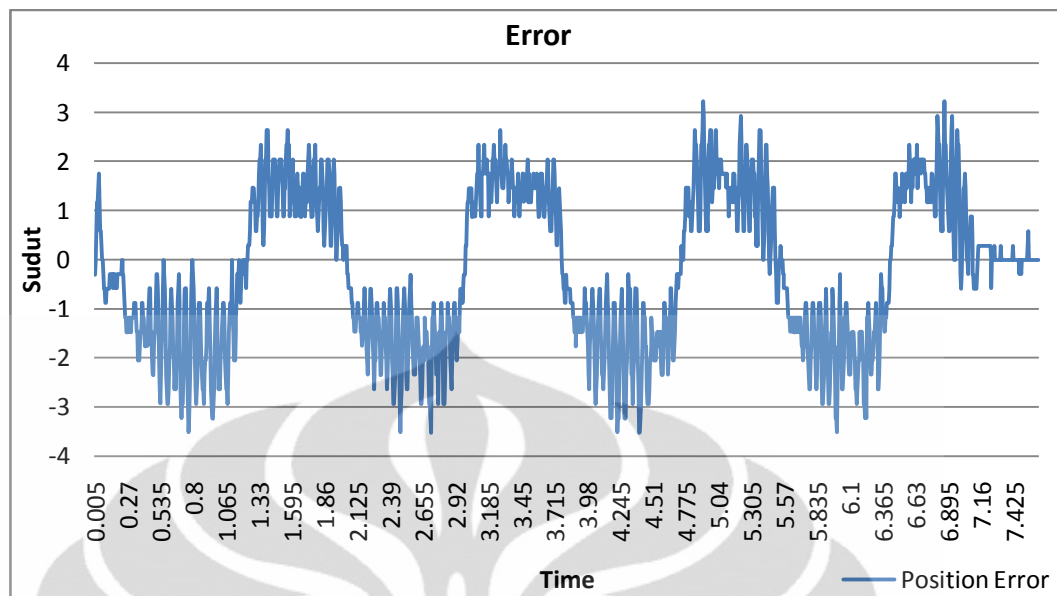
Terlihat pada gambar 4.6 grafik perbedaan posisi manipulator, pada daerah yang berada diluar kotak berlabel 'A' nilai perbedaan posisi maksimal adalah  $2.93^\circ$  dengan rata-rata nilai  $0.88^\circ$ . nilai ini terlihat lebih baik dari sebelumnya, akan tetapi pada akhir pergerakan sistem berosilasi seperti yang terlihat pada label 'A' pada gambar 4.5 dan 4.6.

Oleh karena itu nilai  $K_p$  yang digunakan adalah sebesar 175 dengan harapan sistem akan tetap stabil dan respon yang lambat akan dibantu dengan nilai  $K_v$  yang optimal sesuai dengan persamaan (3.29). Sesuai persamaan (3.29) maka didapat nilai  $K_v$  ideal sebesar 18.52, sehingga didapat pencitraan grafik sebagai berikut.



Gambar 4.7 Grafik posisi master dan slave saat nilai  $K_p$  175





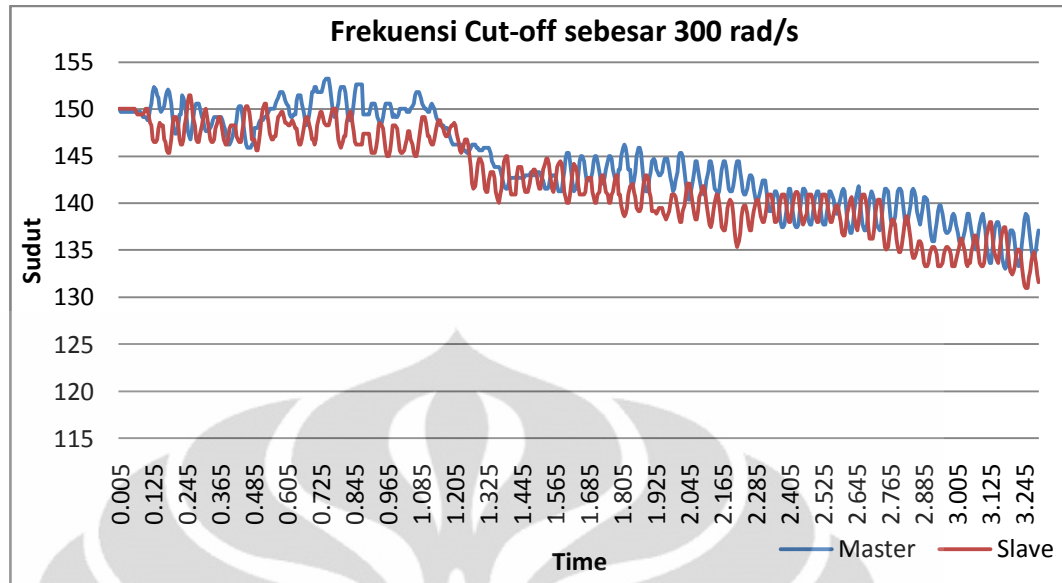
Gambar 4.8 Grafik perbedaan posisi master dan slave saat nilai Kp 175

Dengan nilai parameter yang sesuai maka didapat kesalahan maksimum perbedaan manipulator master dan slave sebesar  $3.23^\circ$  dengan nilai rata rata  $1.3^\circ$  pada arah searah jarum jam dan  $3.52^\circ$  dengan nilai rata-rata  $1.5^\circ$  pada arah berlawanan jarum jam. Nilai kesalahan ini hampir sama dengan nilai pada saat Kp bernilai 190, hanya saja tidak terjadi osilasi pada sistem ini.

#### 4.3 Penentuan Nilai Frekuensi Cut-off pada *Low-Pass Filter*

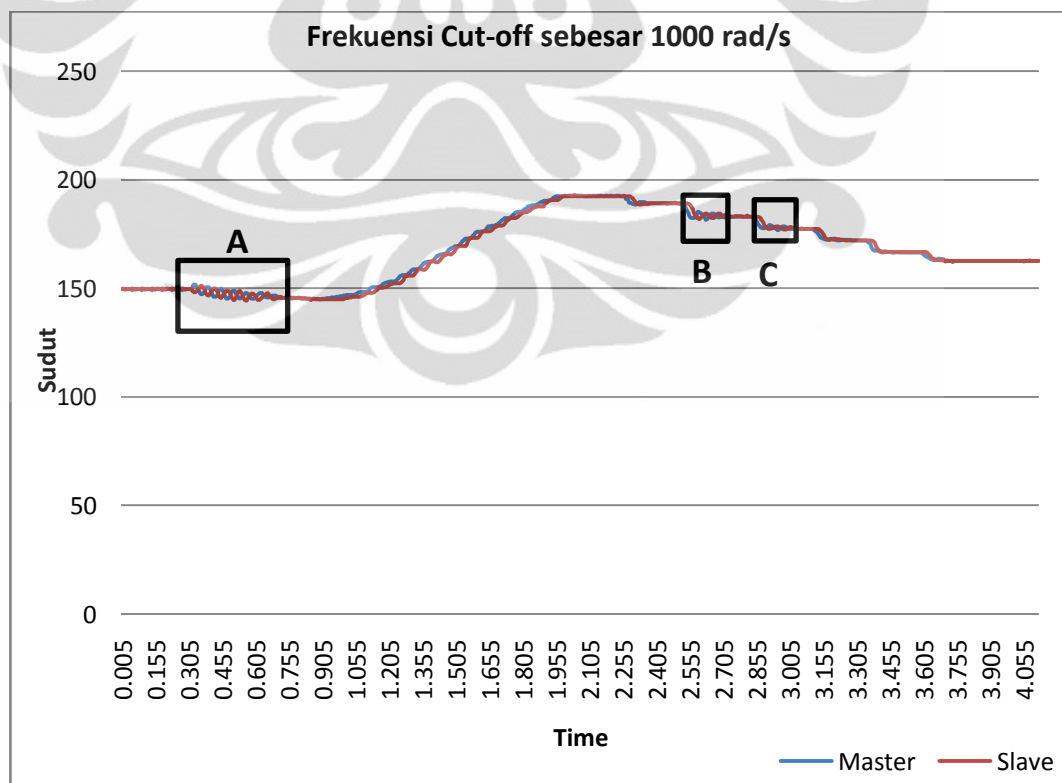
Karena sistem ini menggunakan nilai Kv maka diperlukan sebuah lowpass filter untuk meredam gangguan (noise) hasil dari penurunan posisi terhadap waktu cuplik seperti yang telah dijelaskan sebelumnya. Frekuensi yang ideal sesuai dengan perhitungan adalah sebesar 200 Hz atau sebesar 1256.6 rad/s. Pada pengujian ini, frekuensi cut off akan divariasikan dari nilai terkecilnya dan di naikkan perlahan untuk mendapatkan hasil terbaik.

Lowpass filter ini akan memotong sinyal yang masuk yang memiliki frekuensi lebih besar dibandingkan dengan frekuensi cut-off yang ditentukan dan memperbolehkan sinyal yang memiliki frekuensi dibawah frekuensi cut-off untuk lewat. Apabila frekuensi cut-off lebih kecil dari frekuensi pada sistem sebenarnya maka sinyal yang telah diproses pada sistem tidak akan dapat lewat secara keseluruhan sehingga menyebabkan sistem yang tidak stabil.



Gambar 4.9 Respon sistem saat menggunakan frekuensi cut-off sebesar 300 rad/s

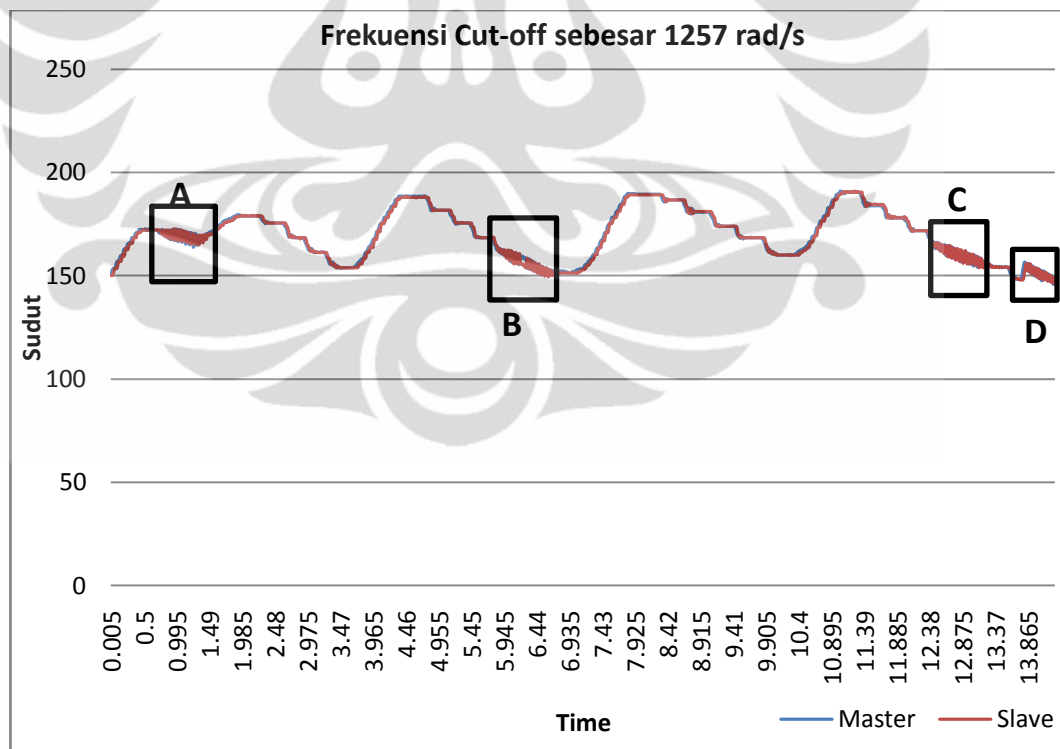
Dapat dilihat pada grafik gambar 4.9 memiliki nilai frekuensi cut-off yang jauh dibawah nilai frekuensi sistem. Oleh karena hal tersebut, maka pengujian ini akan diuji mulai dari 1000 rad/s sampai menemukan kondisi ideal. Untuk kondisi dengan frekuensi cut off sebesar 1000 rad/s dapat dilihat pada gambar berikut.



Gambar 4.10 Respon sistem saat menggunakan frekuensi cut-off sebesar 1000 rad/s

Pengujian dengan pergerakan manipulator master secara perlahan tentu akan menghasilkan hasil yang baik bahkan dengan nilai kesalahan yang kecil. Oleh karena itu, pengujian untuk menentukan frekuensi cut-off dari filter dilakukan dengan cara menggerakkan master secara perlahan dan memberi gerakan secara tiba-tiba pada master dengan harapan akan terlihat respon sistem pada saat manipulator digerakkan secara tiba-tiba atau mendadak.

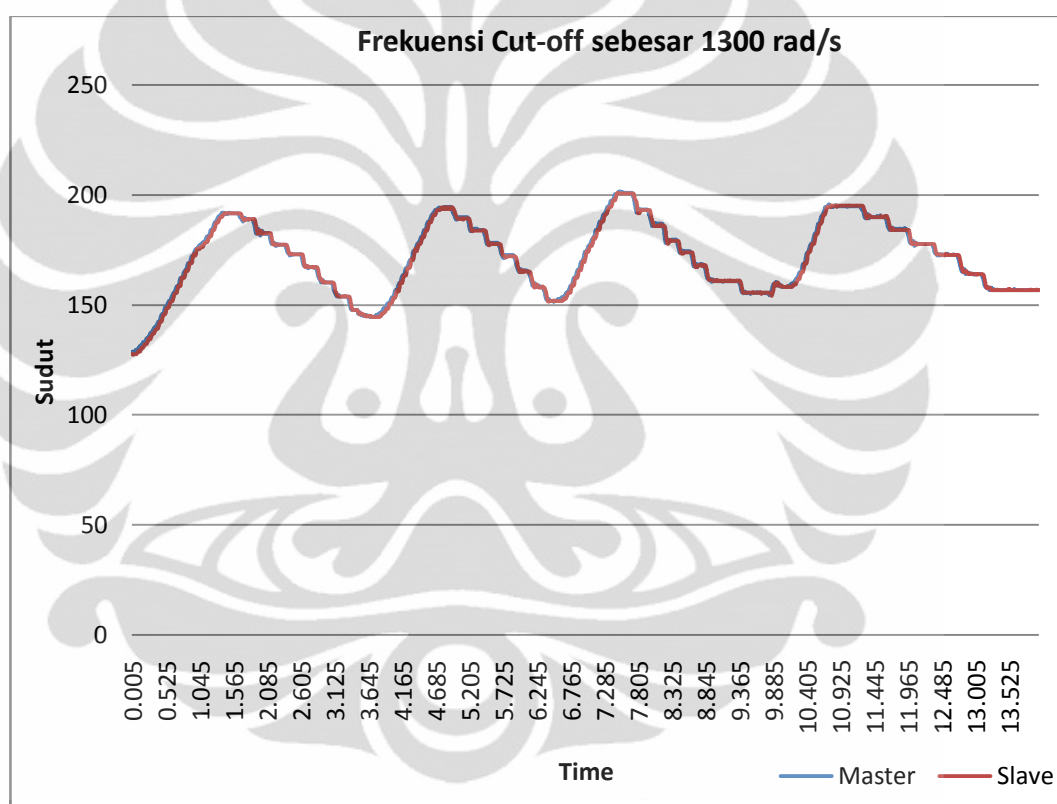
Pada grafik 4.10 diatas terdapat beberapa titik osilasi yang terjadi dengan filter yang memiliki frekuensi cut-off sebesar 1000 rad/s atau sebesar 159.6 Hz. Hal ini ditunjukkan dengan kotak hitam yang menandakan titik-titik osilasi yang terjadi pada sistem. Pada label 'A' pada gambar 4.10, terlihat bahwa pada pergerakan normal saja respon dapat berosilasi. Hal ini juga terlihat pada label 'B' dan 'C' yang tidak dapat mengikuti nilai referensi yang diberikan secara tiba-tiba sehingga pada awal pergerakan, sistem sempat berosilasi walaupun pada akhirnya menuju stabil. Hal ini terjadi dikarenakan nilai frekuensi yang masih terlalu kecil dibandingkan dengan frekuensi sistem.



Gambar 4.11 Respon sistem saat menggunakan frekuensi cut-off sebesar 1257 rad/s

Pada pengujian kedua, digunakan frekuensi cut-off sebesar 1257 rad/s atau sebesar 200.05 Hz. Secara teoritis frekuensi ini adalah frekuensi yang ideal dengan perhitungan yang mendekati 200 Hz.

Dari beberapa pengujian dengan pergerakan mendadak dapat terlihat sistem dapat mengikuti nilai referensi yang diberikan seperti respon yang terjadi sebelum area berlabel 'B' tetapi pada akhir sistem, pada label 'B', 'C' dan 'D', respon mulai berosilasi. Sebenarnya respon ini cukup bagus digunakan apabila pergerakan manipulator tidak dilakukan secara tiba-tiba. Untuk pergerakan secara tiba-tiba respon tidak dapat mengikuti nilai referensi yang diberikan.



Gambar 4.12 Respon sistem saat menggunakan frekuensi cut-off sebesar 1300 rad/s

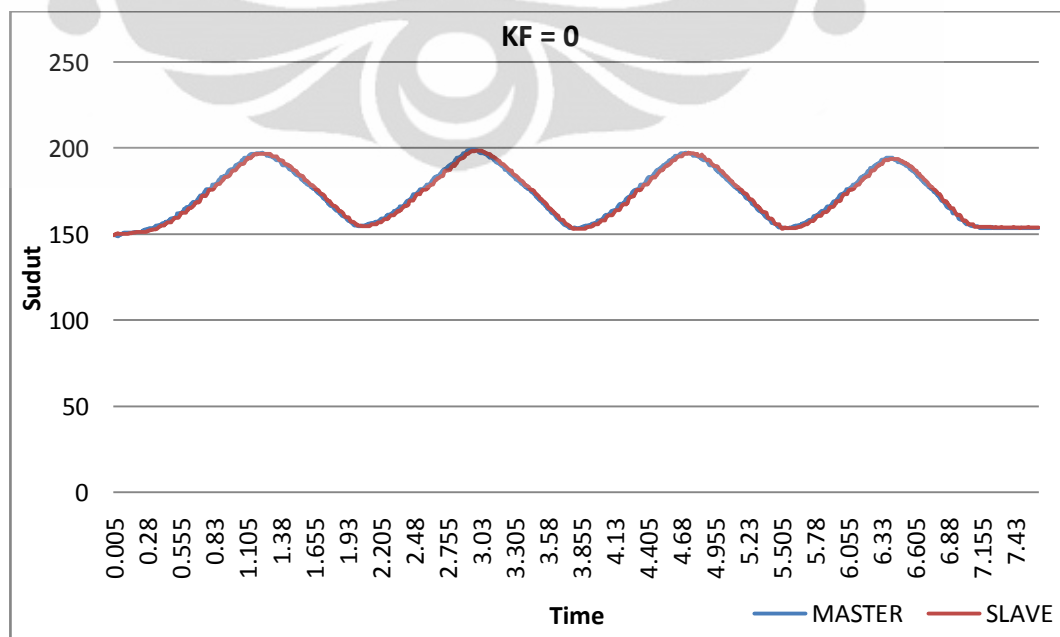
Oleh karena itu, nilai frekuensi cut-off dinaikkan kembali sebesar 1300 rad/s atau setara dengan 206.9 Hz. Dapat dilihat bahwa sistem dapat mengikuti nilai referensi secara keseluruhan dalam kondisi gerakan perlahan maupun gerakan tiba-tiba. Sehingga nilai ini dijadikan nilai acuan frekuensi cut-off untuk pengujian-pengujian selanjutnya. Apabila nilai ini tetap dinaikkan, maka gangguan-gangguan (noise) akan ikut masuk pada sistem yang juga dapat menyebabkan sistem berosilasi.

#### 4.4 Penentuan Nilai Parameter *Force* pada Sistem Bilateral

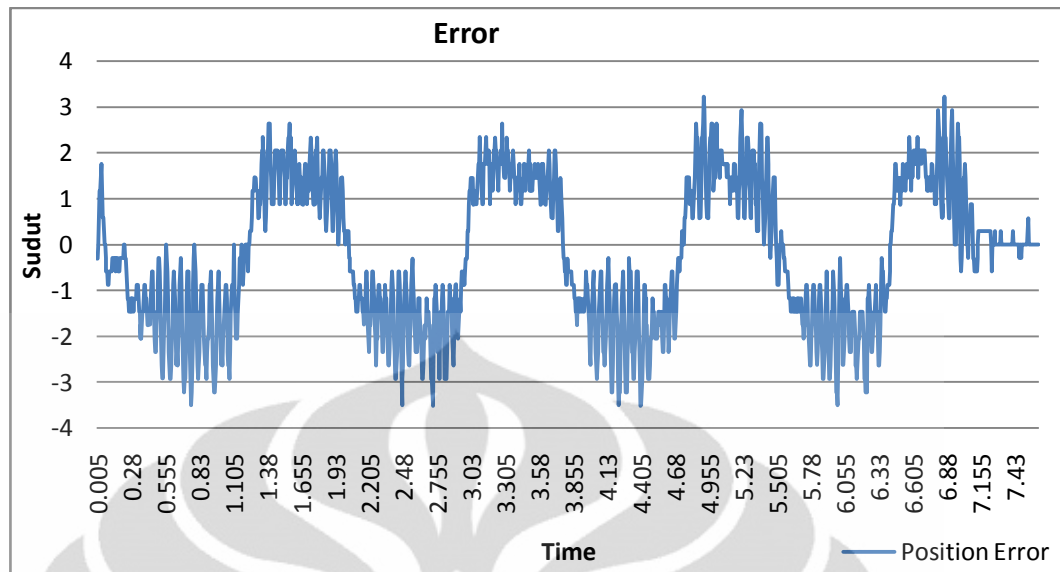
Pengujian-pengujian yang telah dilakukan diatas masih dalam sebatas sistem bilateral 2 kanal yaitu hanya mempertukarkan informasi posisi masing-masing manipulator. Untuk mengubah sistem menjadi sistem bilateral 4 kanal maka masing-masing manipulator mempertukarkan juga informasi gaya masing-masing.

Informasi gaya ini harus diintegrasikan agar dimensi gaya, yang setara dengan dimensi percepatan, dapat mengeluarkan nilai yang setara dengan dimensi posisi. Proses integral yang dibuat pada sistem telah dijelaskan pada bab sebelumnya melalui sistem pada persamaan (3.14), (3.24) dan (3.25).

Nilai parameter *force* yang ditukarkan pada masing-masing manipulator berfungsi untuk mempermudah gerakan manipulator sehingga manipulator bergerak tidak hanya berdasarkan perbedaan posisi, akan tetapi juga terhadap perbedaan tekanan. Apabila sistem hanya menggunakan perbedaan posisi maka manipulator slave akan bergerak apabila manipulator master telah bergerak sehingga menimbulkan perbedaan posisi. Jika menggunakan sistem bilateral 4 kanal, maka sebagai contoh apabila manipulator master diberikan suatu tekanan oleh operator tanpa ada perbedaan posisi yang signifikan maka manipulator slave akan tetap bergerak sesuai dengan tekanan yang diberikan. Secara tidak langsung maka manipulator akan lebih mudah digerakkan oleh operator. Berikut adalah grafik sistem tanpa adanya nilai parameter gaya.

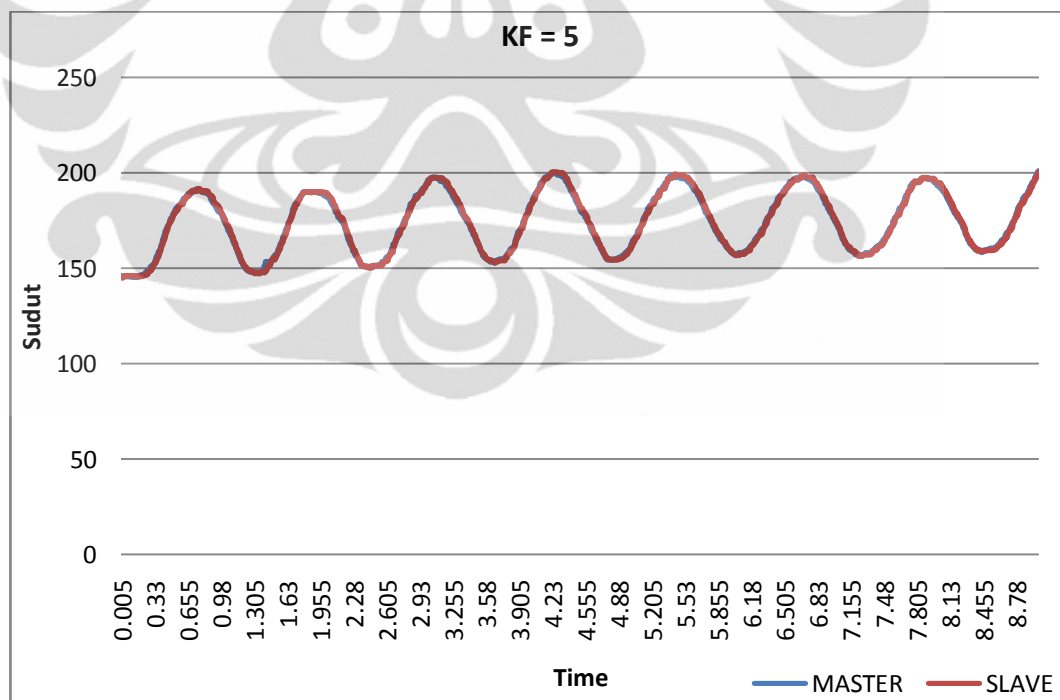


Gambar 4.13 Grafik posisi master dan slave tanpa parameter *force*



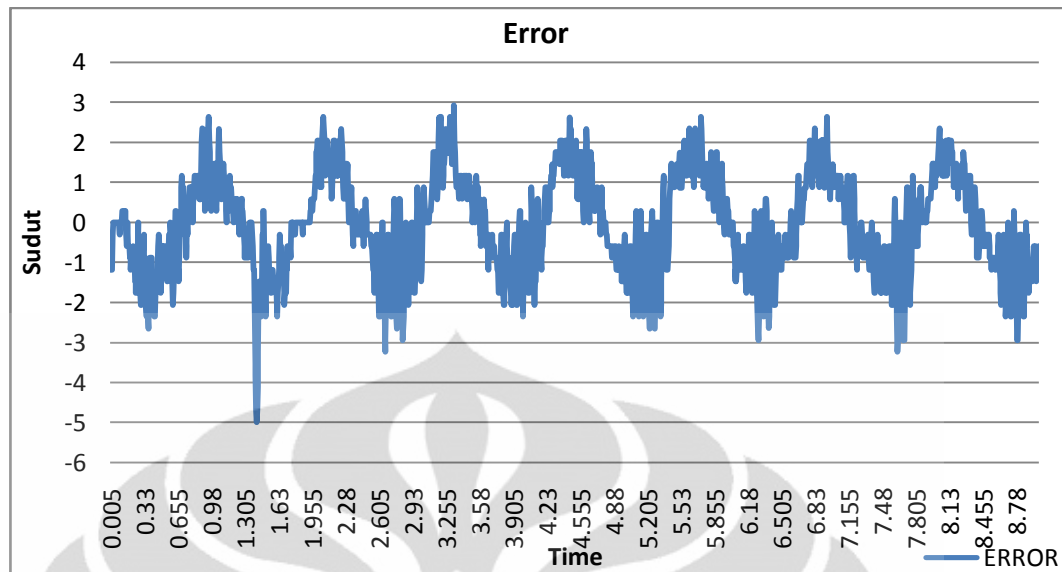
Gambar 4.14 Grafik perbedaan posisi master dan slave pada sistem

Dapat dilihat pada grafik 4.14 tanpa adanya faktor gaya yang dikirimkan, sistem berjalan normal dengan perbedaan posisi (error) bernilai seperti pada pengujian-pengujian sebelumnya. Berikut grafik dengan adanya faktor *force* pada sistem.



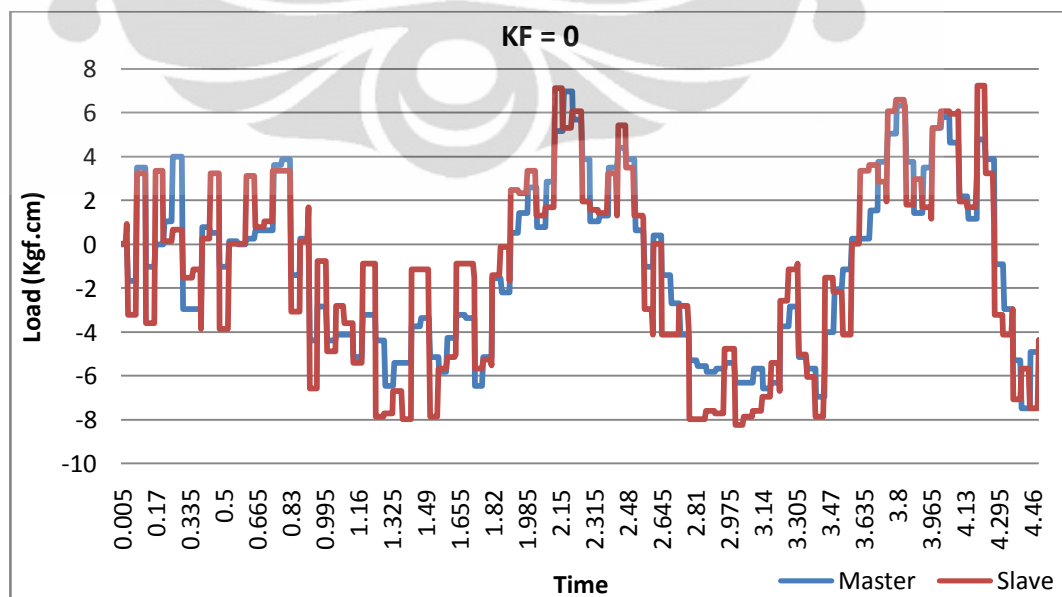
Gambar 4.15 Grafik posisi master dan slave dengan parameter *force*





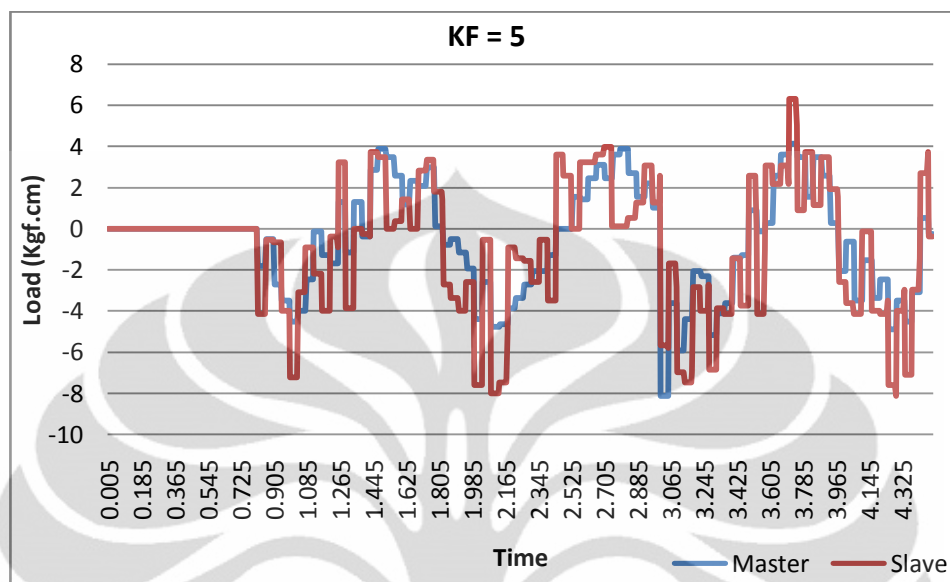
Gambar 4.16 Grafik perbedaan posisi master dan slave pada sistem

Pada saat sistem menggunakan nilai gaya yang ditukarkan antar manipulator, sistem menjadi lebih mudah untuk digerakkan. Dari gambar 4.16, terlihat bahwa grafik kesalahan yang terjadi berkurang taraf osilasi pada bagian puncaknya dibandingkan grafik kesalahan sebelum menggunakan parameter gaya pada gambar 4.14. Hal ini disebabkan oleh pergerakan manipulator yang lebih mudah digerakkan dan lebih mulus saat digerakkan mengurangi taraf osilasi perbedaan posisi antar dua buah manipulator. Untuk dapat melihat lebih jelas maka ditampilkan grafik pembacaan beban berikut ini.



Gambar 4.17 Grafik pembacaan beban tanpa parameter *force*

Terlihat pada grafik 4.17, bahwa pada saat sistem bekerja dan manipulator master digerakkan maka nilai beban puncak rata-rata yang terukur adalah berkisar antara nilai 6 Kgf.cm.

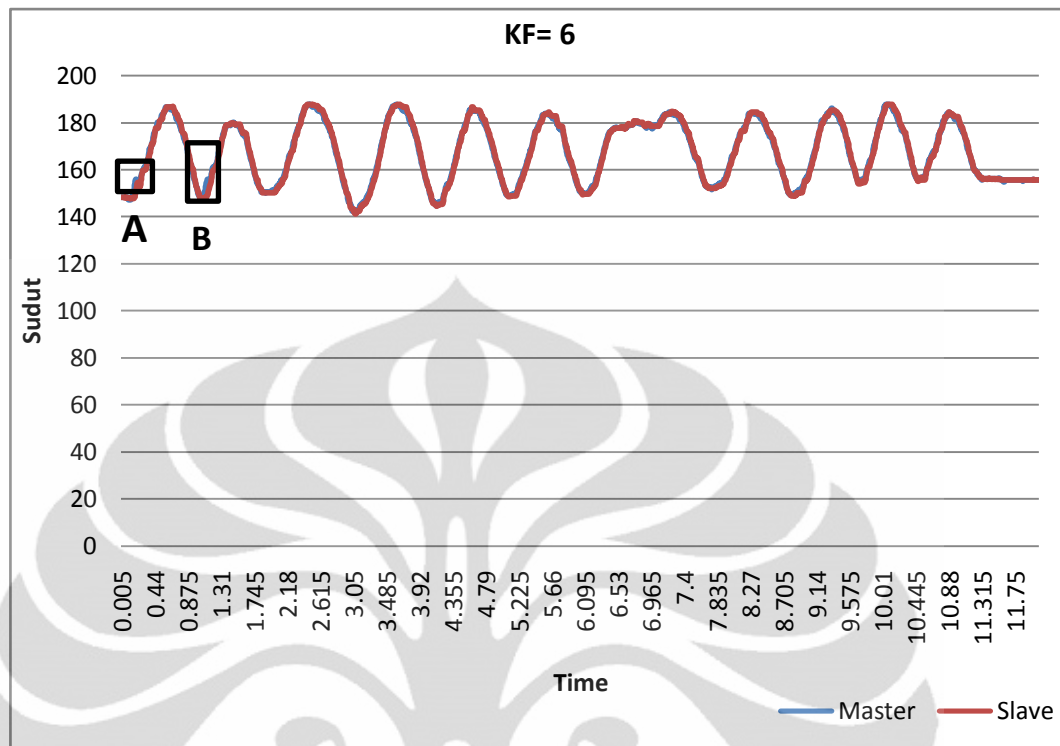


Gambar 4.18 Grafik pembacaan beban dengan parameter *force*

Sementara pada saat sistem menggunakan nilai parameter gaya, maka nilai beban puncak rata-rata pada sistem adalah berkisar antara nilai 4 Kgf.cm. Dari kedua gambar diatas dapat dilihat bahwa pada saat sistem tidak menggunakan nilai gaya maka nilai beban lebih besar dibandingkan dengan pada saat sistem menggunakan nilai gaya. Hal ini menyimpulkan bahwa usaha yang digunakan untuk menggerakkan manipulator tanpa parameter gaya akan lebih besar dibandingkan dengan usaha untuk menggerakkan manipulator dengan parameter gaya.

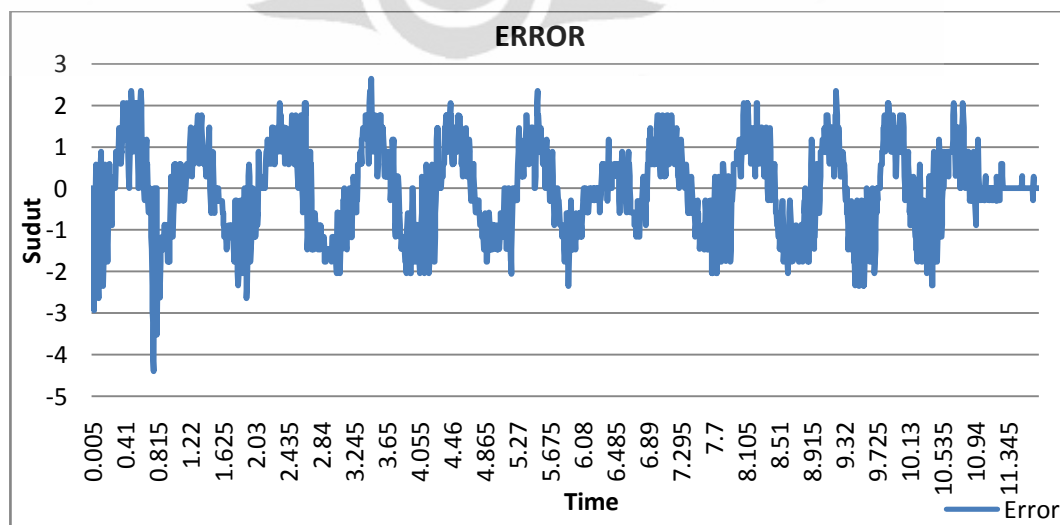
Parameter gaya pada sistem memiliki kisaran konstanta tersendiri. Nilai konstanta ini tidak dapat terlalu besar dikarenakan informasi gaya, yang setara dengan dimensi percepatan, terdapat banyak sekali gangguan (noise) yang terjadi bila nilainya terlalu besar. Dari beberapa pengujian yang dilakukan, maka kisaran nilai konstanta gaya adalah sebesar 0 – 5. Sebagai contoh dapat dilihat gambar berikut dengan nilai konstanta gaya sebesar 6.





Gambar 4.19 Grafik posisi master dan slave dengan nilai parameter *force* 6

Dapat dilihat sistem masih tergolong stabil akan tetapi terdapat beberapa kondisi loncatan-loncatan kedua manipulator sehingga sistem diperkirakan rawan sekali terjadi osilasi. Dapat dilihat pada kotak berlabel 'A' dan 'B' pada gambar diatas, sistem sangat berpotensi untuk mengalami osilasi apabila terdapat gangguan (noise) pada pembacaan nilai beban dari motor. Berikut grafik perbedaan posisi (kesalahan) pada sistem diatas.

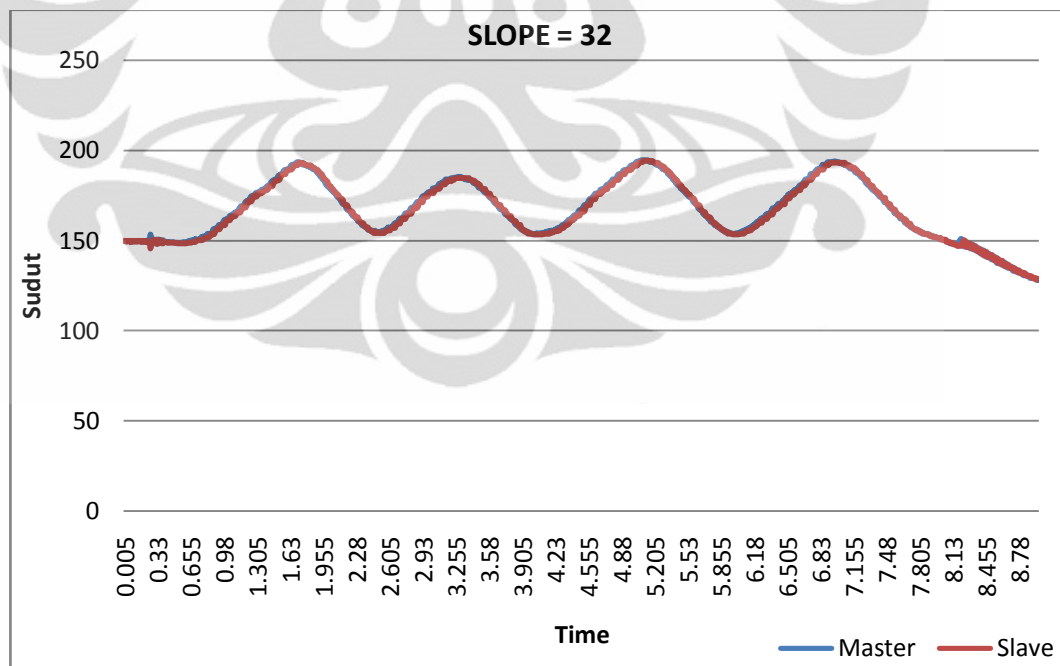


Gambar 4.20 Grafik perbedaan posisi master dan slave dengan nilai parameter *force* 6

#### 4.5 Variasi Nilai Sudut Distribusi Torsi pada Sistem Bilateral

Selain dengan memvariasikan nilai konstanta gaya pada bagian motor, untuk menjadikan manipulator ringan agar semakin mudah digerakkan maka sudut distribusi torsi juga berperan penting pada sistem. Semakin landai sudut distribusi torsi, dengan kata lain semakin besar nilai slope, maka akan semakin ringan usaha operator untuk menggerakkan manipulator. Sebaliknya semakin curam sudut distribusi torsi maka akan semakin berat usaha yang dilakukan operator untuk menggerakkan manipulator. Oleh karena itu memvariasikan sudut distribusi torsi akan sama dengan memvariasikan nilai slope pada motor. Nilai inisial slope untuk dynamixel AX-12+ adalah sebesar 32 atau setara dengan sudut  $86.17^\circ$ . Sudut ini adalah sudut maksimal untuk variasi yang dilakukan sehingga tentu saja apabila nilai ini digunakan, maka diperlukan usaha yang lebih berat untuk menggerakkan manipulator.

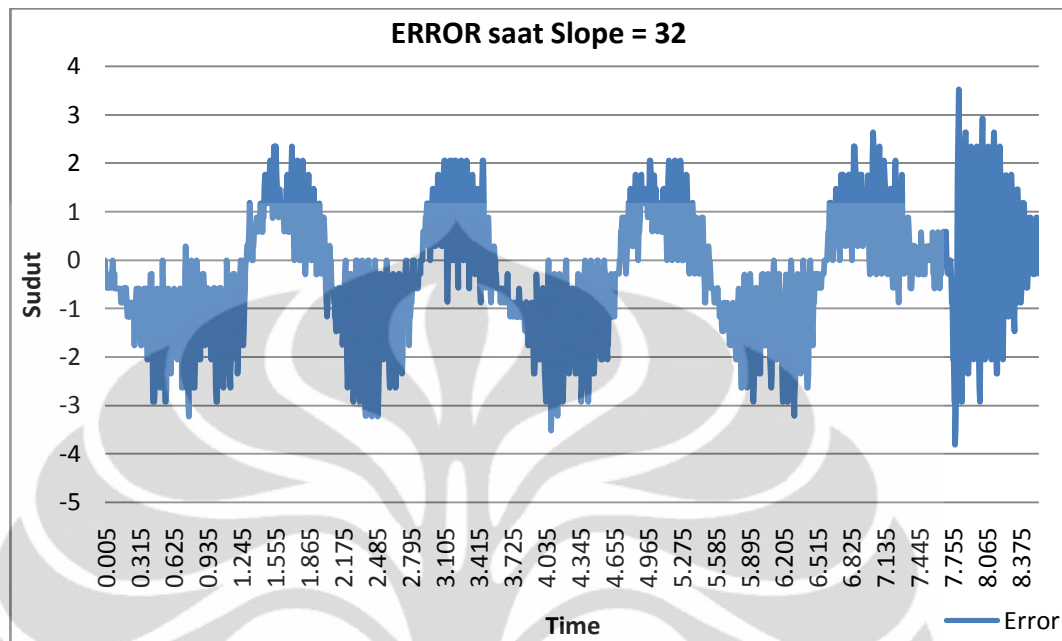
Walaupun usaha yang dilakukan operator untuk menggerakkan manipulator dengan sudut distribusi torsi yang landai lebih ringan, akan tetapi tingkat kesalahannya pun makin bertambah.



Gambar 4.21 Grafik posisi master dan slave saat parameter slope bernilai 32

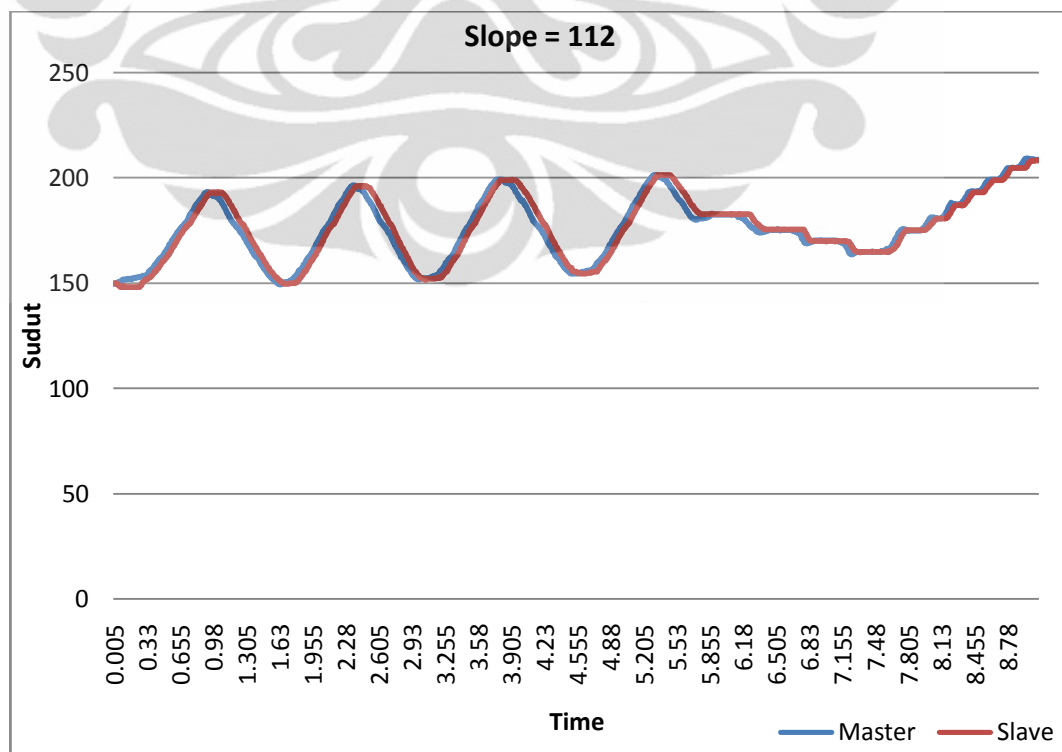
Untuk nilai sudut distribusi torsi yang curam, nilai posisi manipulator slave akan dengan cepat mengikuti manipulator master sehingga error yang

ditimbulkan relatif lebih kecil walaupun usaha yang dilakukan operator untuk menggerakkan manipulator lebih besar.



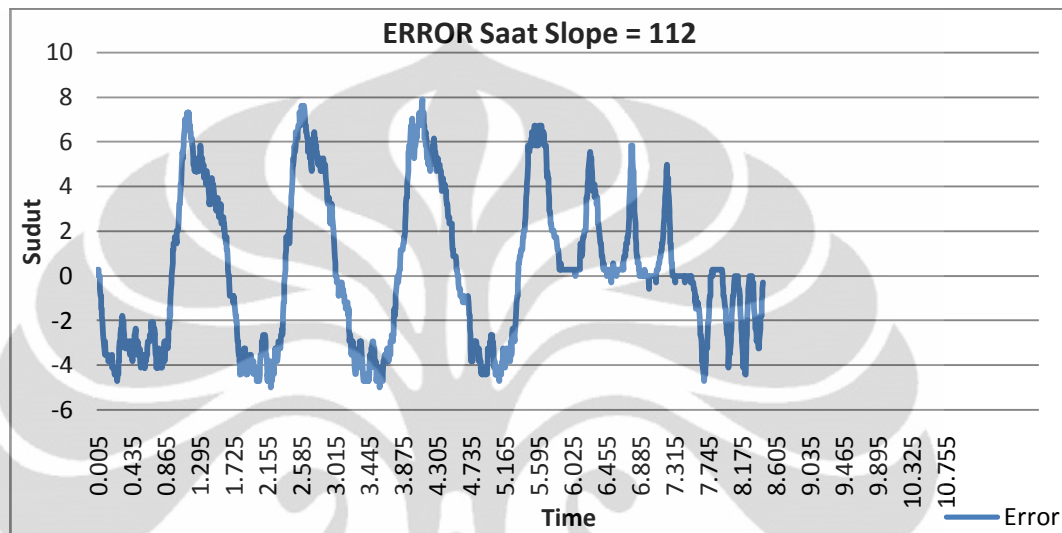
Gambar 4.22 Grafik perbedaan posisi master dan slave pada sistem

Dapat dilihat pada gambar 4.22 bahwa error maksimal yang terjadi adalah  $3.51^\circ$ . Semakin berkurangnya nilai taraf kemiringan sudut distribusi torsi akan menambah nilai error yang diperoleh dari perbedaan posisi master dengan slave.



Gambar 4.23 Grafik posisi master dan slave saat parameter slope bernilai 112

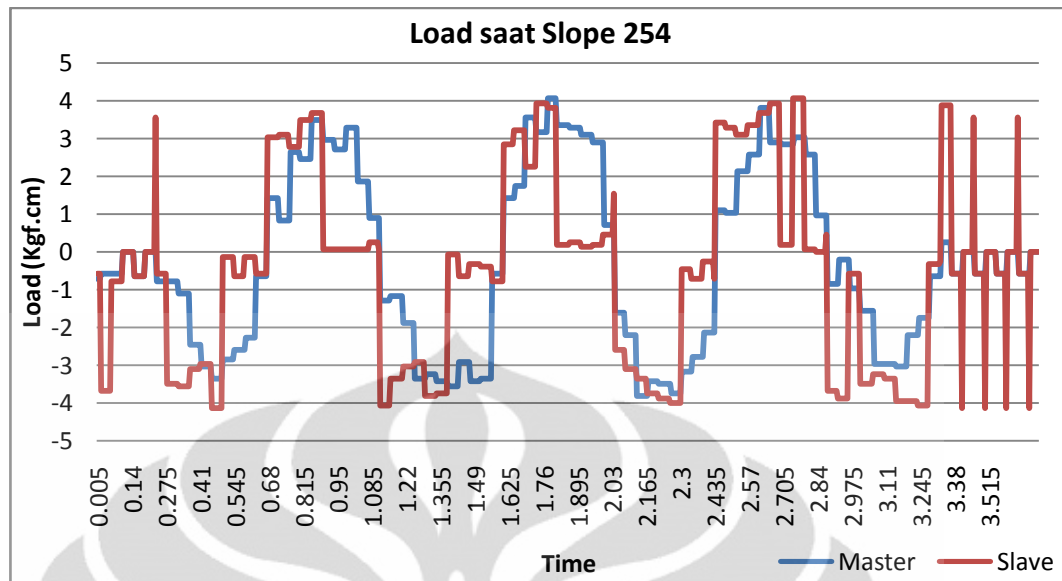
Dapat dilihat pada gambar 4.23 bahwa semakin besar nilai slope atau dengan kata lain semakin landai sudut distribusi torsi maka perbedaan posisi akan bertambah sehingga manipulator slave tidak dengan cepat mengikuti posisi master. Walaupun demikian usaha yang dilakukan oleh operator untuk menggerakkan manipulator relatif lebih mudah.



Gambar 4.24 Grafik perbedaan posisi master dan slave pada sistem

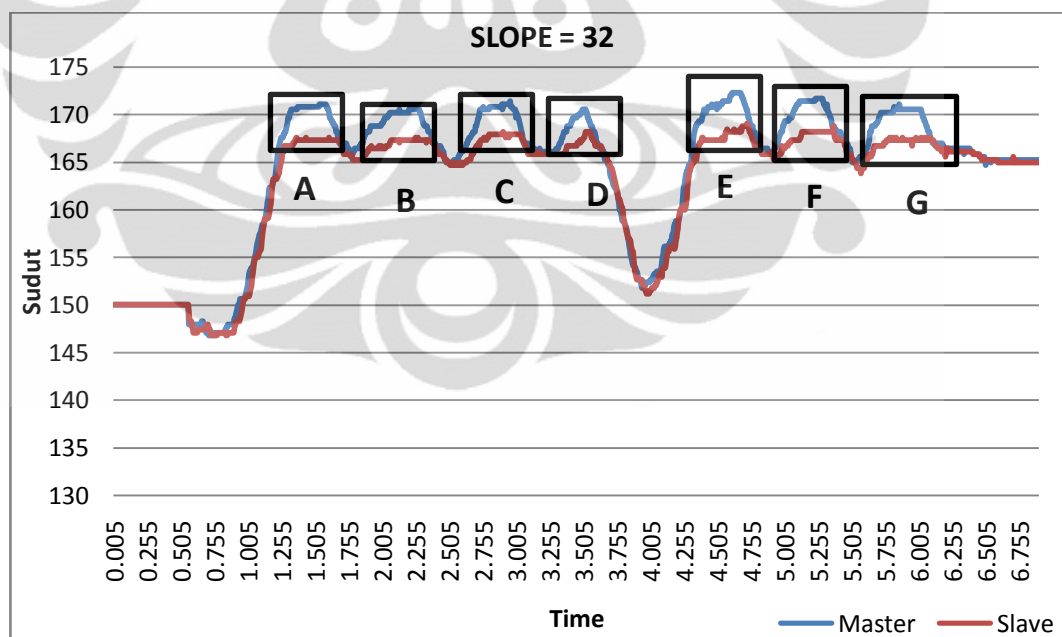
Pada grafik gambar 4.24, error yang terjadi lebih besar dibandingkan dengan nilai sudut yang curam. Dari hasil perhitungan didapat nilai error terbesar bernilai  $7.62^\circ$ .

Pada saat sistem berjalan dengan normal tanpa ada gangguan sentuhan sensasi pada manipulator slave maka nilai beban yang terbaca didominasi dengan nilai master karena nilai beban master memiliki gaya langsung yaitu gaya dorong oleh operator. Sedangkan nilai beban pada manipulator slave terbaca lebih kecil karena tidak mendapat gaya dorongan melainkan hanya mendapat perintah untuk bergerak menuju posisi yang diinginkan berdasarkan informasi perhitungan dari manipulator master. Berikut grafik pembacaan beban pada kondisi pergerakan normal.

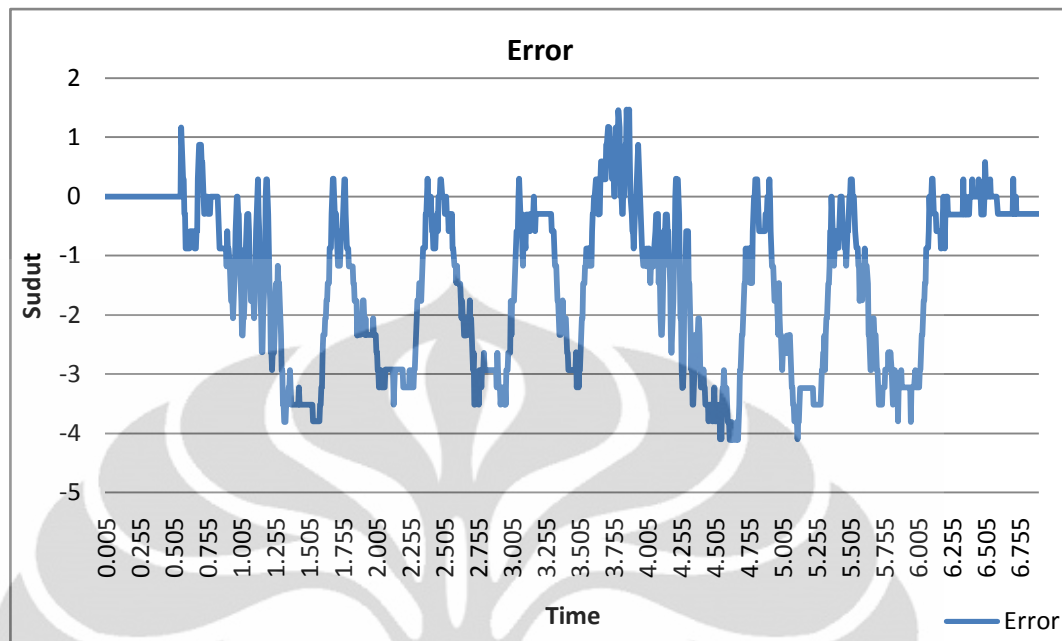


Gambar 4.25 Grafik pembacaan beban pada saat pergerakan normal

Sementara pada saat manipulator slave mendapat sensasi sentuhan pada lingkungan sekitarnya maka nilai perbedaan posisi dan beban akan bertambah. Berikut grafik posisi dan perbedaan posisi yang didapat pada setiap kisaran sudut distribusi torsi.

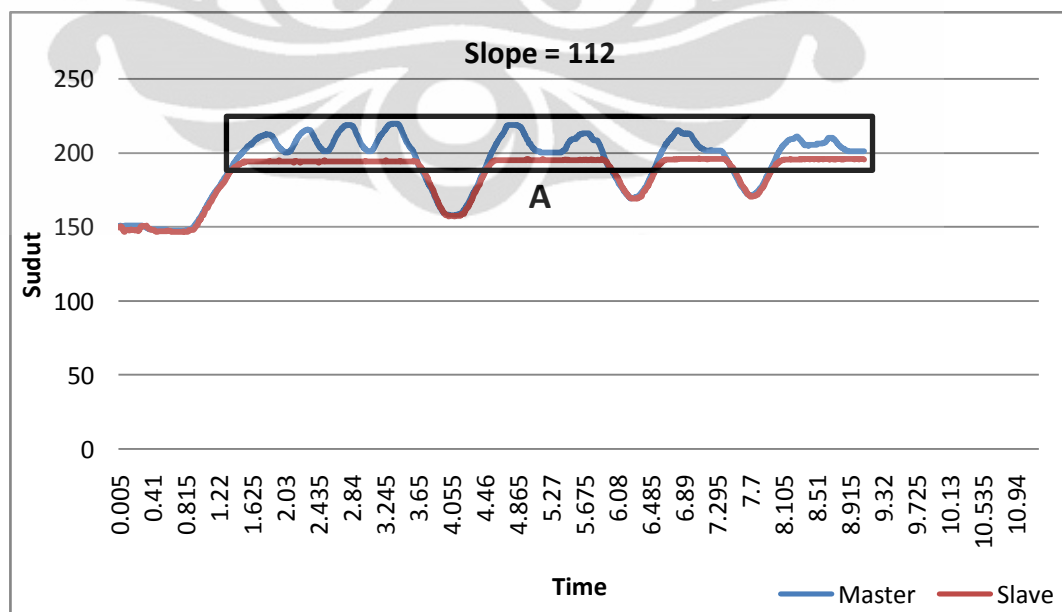


Gambar 4.26 Grafik posisi master dan slave saat manipulator slave tertahan



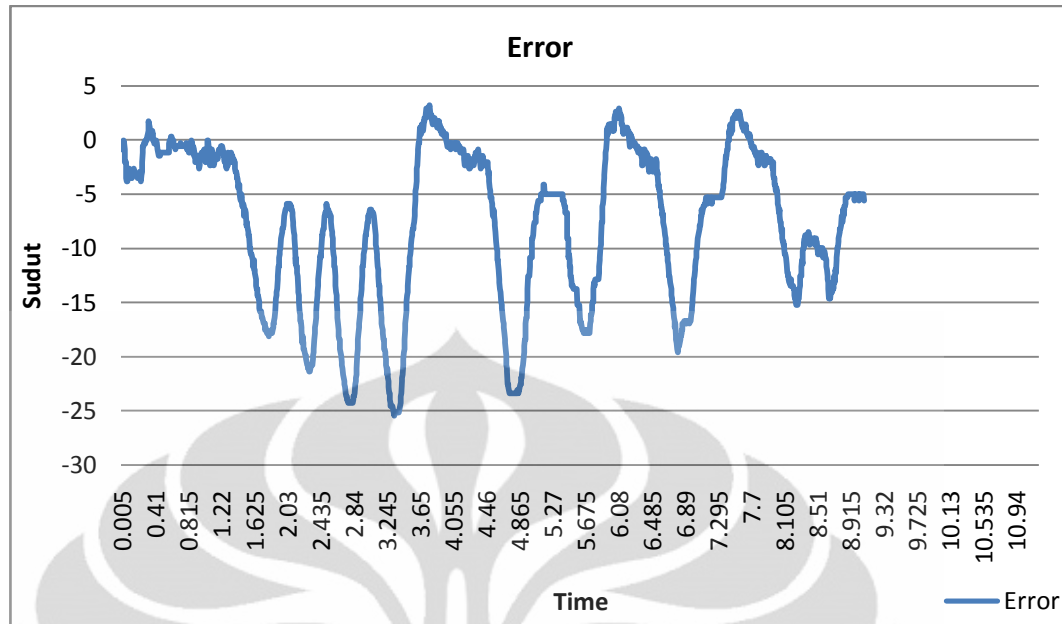
Gambar 4.27 Grafik perbedaan posisi master dan slave pada sistem

Dapat dilihat pada gambar 4.26 dan 4.27, bahwa pada saat besar nilai slope kecil atau dapat dikatakan nilai sudut distribusi torsi semakin curam maka nilai perbedaan manipulator master dan slave tidak terlalu besar. Berdasarkan pengolahan data nilai maksimum pada perbedaan posisi pada saat menyentuh sesuatu adalah sebesar  $4.1^\circ$ .



Gambar 4.28 Grafik posisi master dan slave saat manipulator slave tertahan dengan sudut torsi landai

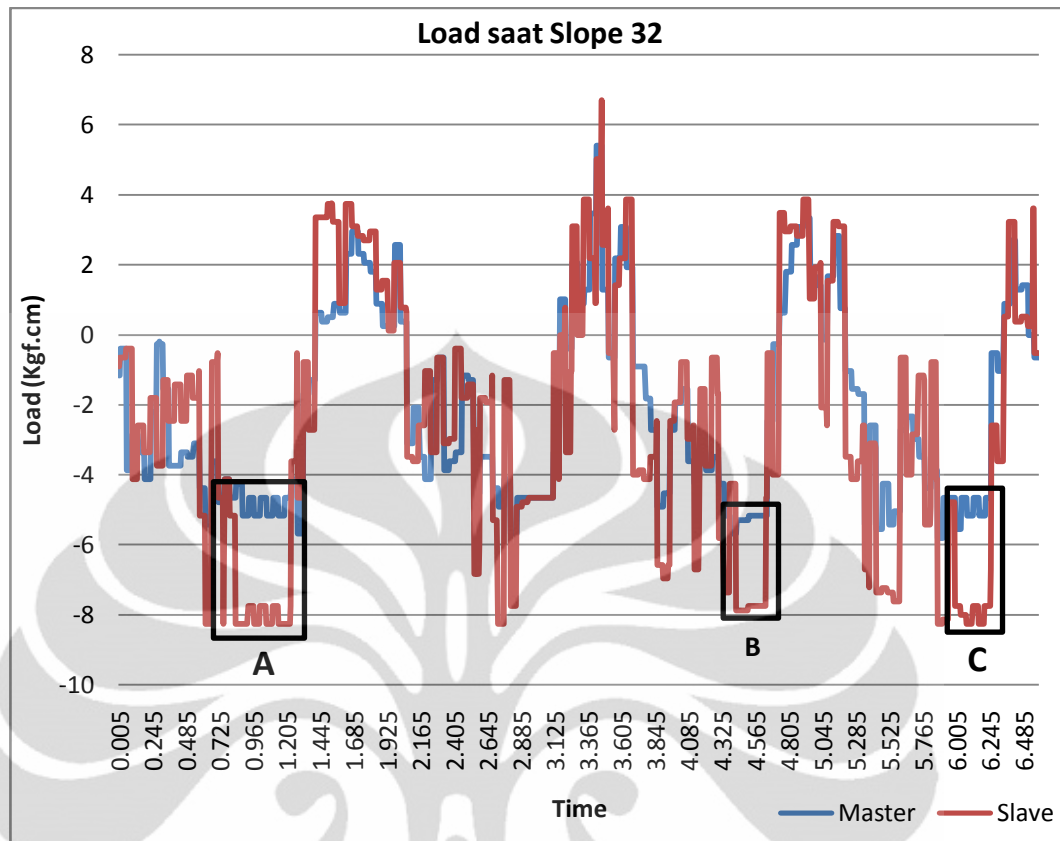




Gambar 4.29 Grafik perbedaan posisi master dan slave pada sistem

Perbedaan yang mencolok terjadi pada saat sudut distribusi torsi landai. Pada gambar 4.24 terlihat bahwa perbedaan sudut maksimum pada saat kondisi normal tidak menyentuh benda adalah  $7.62^\circ$  sedangkan pada sudut distribusi torsi yang sama dan kondisi menyentuh benda perbedaan posisi maksimal mencapai  $25.2^\circ$ . Nilai ini sangat besar walaupun operator dapat dengan mudah menggerakkan manipulator karena sudut distribusi torsi yang landai akan tetapi, perbedaan posisi yang dihasilkan saat menyentuh benda sangatlah besar. Hal ini dapat mengurangi efek sensasi *haptic* pada sistem.

Kondisi menyentuh benda ini juga dapat dilihat pada nilai beban yang terbaca pada motor. Berikut adalah grafik pembacaan beban pada motor pada variasi sudut distribusi torsi (slope).



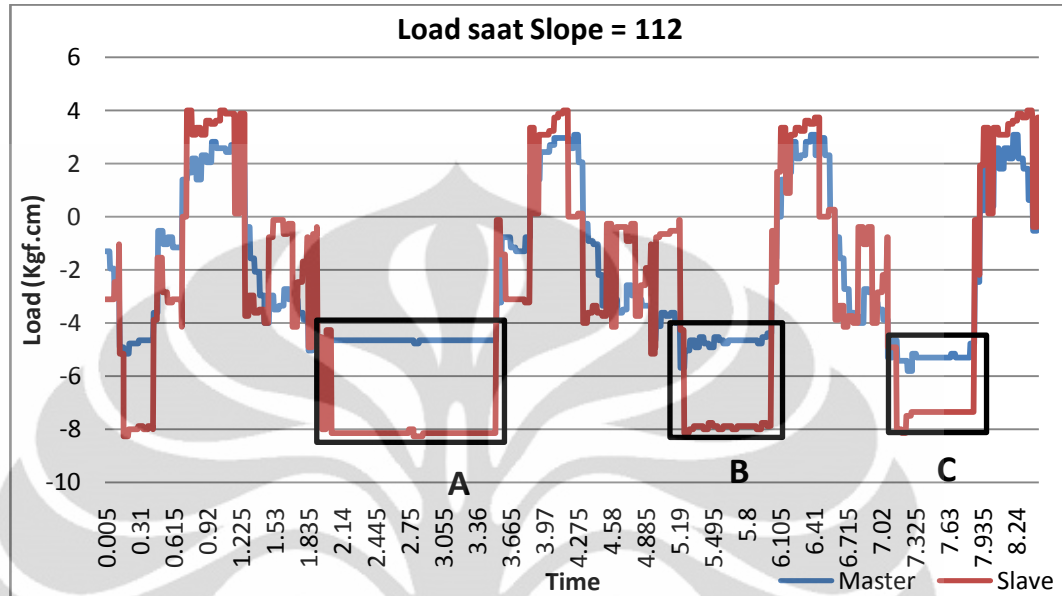
Gambar 4.30 Grafik pembacaan beban saat kondisi menyentuh benda dengan nilai slope 32

Dapat dilihat pada gambar 4.30 diatas, bagian yang ditandai dengan kotak berlabel 'A', 'B' dan 'C' adalah nilai beban yang terbaca pada saat kondisi manipulator slave menyentuh sesuatu. Pada kondisi sudut distribusi torsi yang curam seperti ini, apabila manipulator slave menyentuh suatu benda, maka manipulator master akan merasakan sensasinya dan nilai perbedaan posisi antara master dan slave tidak akan bernilai sangat besar karena sistem memiliki sudut yang curam untuk distribusi torsinya seperti pada gambar 4.27 sebelumnya.

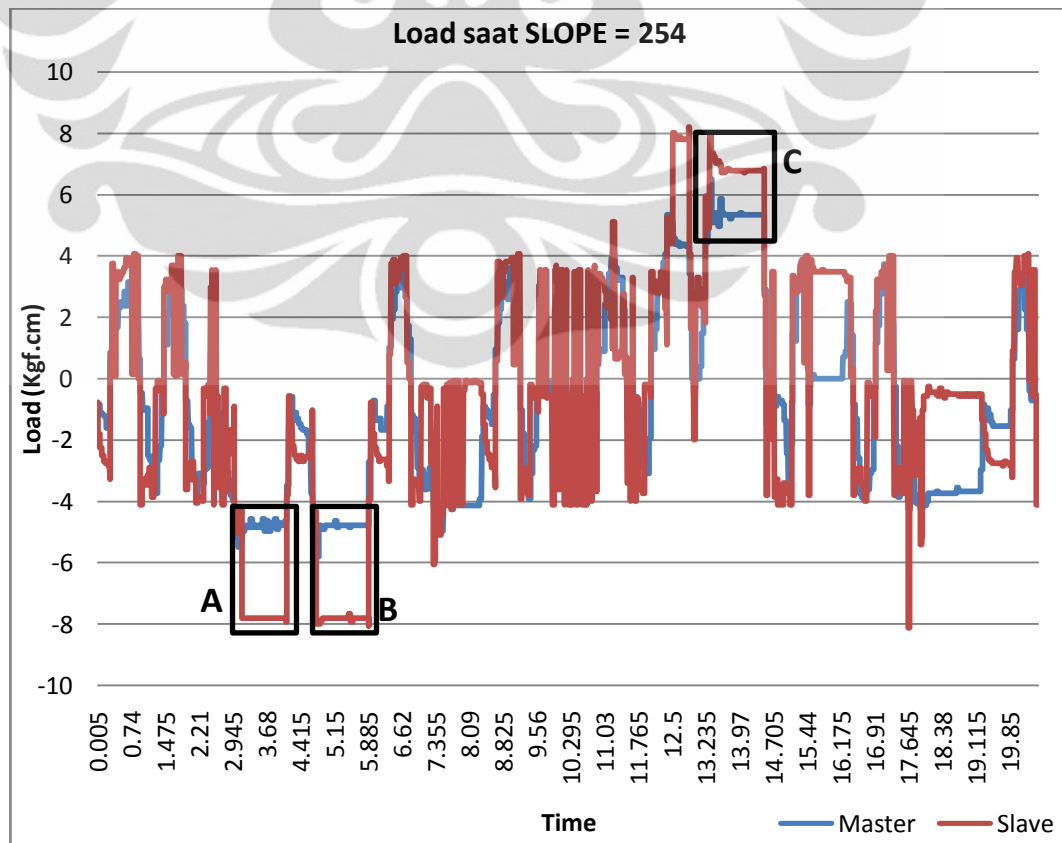
Kondisi gambar berlabel 'A' dan 'C' adalah kondisi pada saat menyentuh benda dan manipulator master tetap mempertahankan posisinya melawan gaya yang diberikan oleh operator sehingga menimbulkan pembacaan beban yang bergerigi. Dapat dilihat pada saat menyentuh benda, nilai beban yang terbaca pada manipulator slave adalah sebesar 8.25 Kgf.cm. Hal ini disebabkan karena pada aktuator dynamixel AX-12+ telah ditentukan bahwa nilai torsi maksimal yang digunakan adalah sebesar 50 % dari nilai maksimal (16.5 Kgf.cm) sehingga pada saat menyentuh sesuatu dan saat manipulator tetap ditekan secara paksa maka nilai beban tidak akan lebih dari 8.25 Kgf.cm. Apabila tetap dipaksakan bergerak



maka kemungkinan besar aktuator akan rusak karena melebihi batas torsi maksimalnya. Sedangkan nilai beban pada manipulator master saat menyentuh terbaca pada data dengan rata rata nilai 4.6 Kgf.cm.



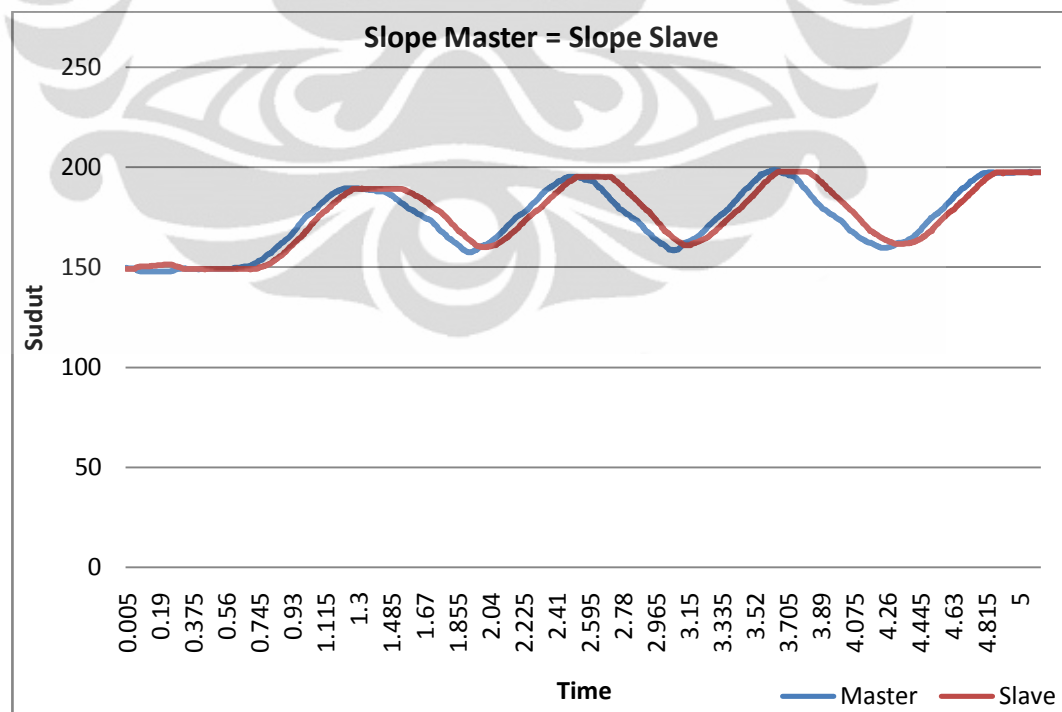
Gambar 4.31 Grafik pembacaan beban saat kondisi menyentuh benda dengan nilai slope 112



Gambar 4.32 Grafik pembacaan beban saat kondisi menyentuh benda dengan nilai slope 254

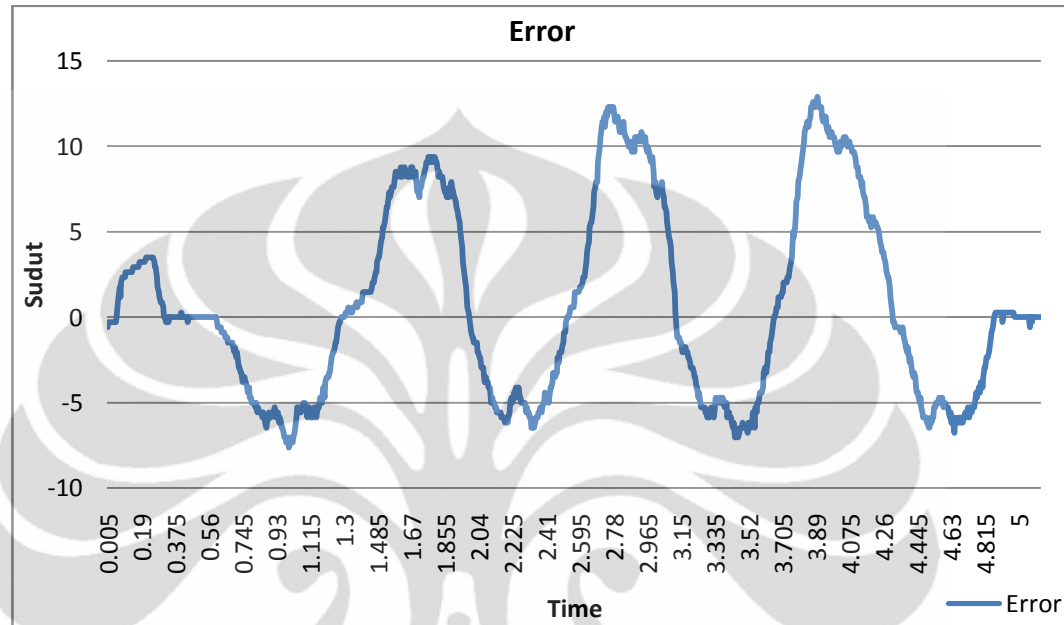
Dari kedua gambar diatas yang berlabel 'A', 'B' dan 'C' dapat terlihat bahwa perbedaan nilai sudut distribusi torsi tidak mempengaruhi perbedaan nilai beban yang terbaca secara signifikan walaupun sangat mempengaruhi perbedaan posisi seperti pada contoh gambar 4.27 dan 4.29. Dapat dilihat pada nilai sudut distribusi torsi sesuai dengan gambar 4.31 dan 4.32, pada saat kondisi menyentuh benda nilai beban pada manipulator master mencapai rata-rata 4.83 Kgf.cm dan untuk manipulator slave mencapai rata-rata 8 Kgf.cm. Hal ini tidak berbeda dengan gambar sebelumnya yang memiliki sudut distribusi torsi yang lebih curam.

Pengujian yang telah dilakukan diatas, diterapkan dengan mengatur sudut di kedua ruas manipulator baik bagian slave maupun master. Untuk meminimalisir kesalahan berupa perbedaan posisi antar manipulator yang terjadi, dengan menjaga sistem agar tetap dapat menggerakkan manipulator dengan ringan, maka secara logika manipulator yang diperlukan untuk digerakkan operator secara ringan adalah hanya manipulator master. Sehingga pengujian selanjutnya adalah perbedaan tingkat kesalahan dengan memvariasikan nilai slope masing-masing manipulator.



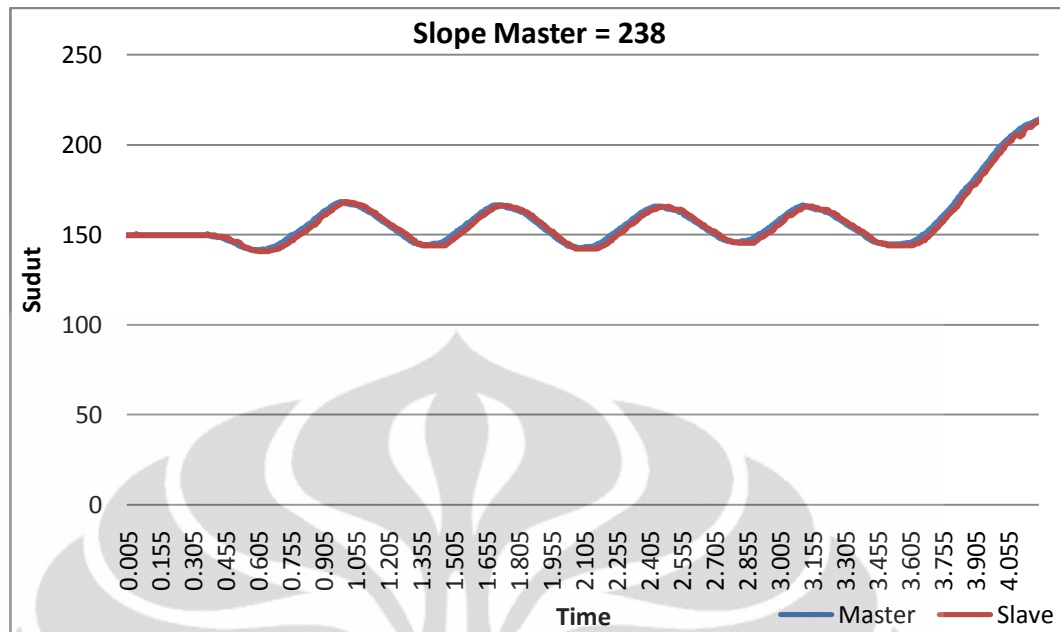
Gambar 4.33 Grafik posisi master dan slave saat parameter slope slave dan master bernilai 238

Apabila slope manipulator master diatur sama dengan nilai slope manipulator slave yang memiliki sudut distribusi torsi yang landai, maka kesalahan yang terjadi relatif besar dan respon relatif lambat dalam mengikuti nilai referensinya seperti yang terlihat pada gambar 4.33 diatas.



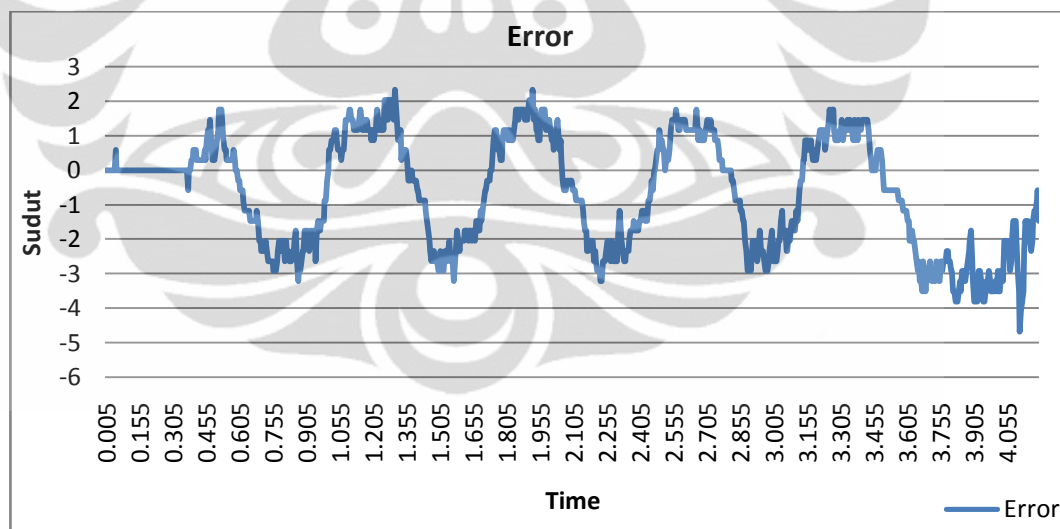
Gambar 4.34 Grafik perbedaan posisi master dan slave pada sistem

Nilai slope yang diberikan pada manipulator master dan manipulator slave adalah sebesar 238. Dari gambar 4.34 diatas terlihat bahwa error yang dihasilkan sangat besar dengan nilai maksimal error sebesar  $12.5^{\circ}$ . Untuk mengurangi nilai error ini maka pengujian selanjutnya adalah nilai slope pada manipulator master yang diberikan dengan nilai 238, sedangkan nilai slope untuk manipulator slave dibiarkan tetap dengan nilai inisialnya yaitu sebesar 32.



Gambar 4.35 Grafik posisi master dan slave saat parameter slope master bernilai 238

Dapat dilihat bahwa dengan taraf keringanan dalam menggerakkan manipulator yang sama dengan pengujian sebelumnya, respon tetap baik dan dapat dengan cepat mengikuti nilai referensinya.



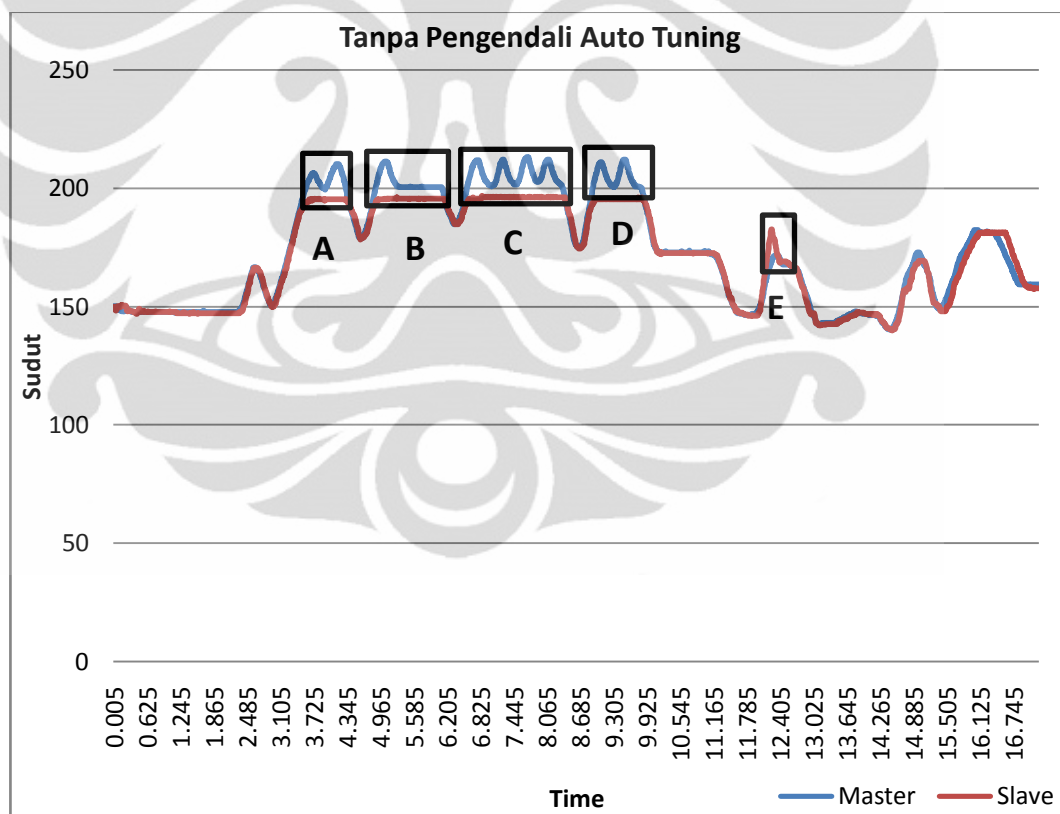
Gambar 4.36 Grafik perbedaan posisi master dan slave saat parameter slope master bernilai 238

Dapat dilihat pada gambar 4.36 bahwa perbedaan posisi antar manipulator yang terjadi jauh lebih kecil dibandingkan dengan pengujian sebelumnya. Kesalahan yang terjadi memiliki nilai maksimal  $2.3^\circ$  pada saat bergerak searah jarum jam dan  $4.6^\circ$  pada saat bergerak berlawanan arah jarum jam. Maka untuk pengujian-pengujian selanjutnya, nilai sudut distribusi torsi yang akan

divariasikan merupakan nilai dari manipulator master saja sebab manipulator master inilah yang akan langsung berhubungan dengan operator.

#### 4.6 *Auto-tuning torque angle* berdasarkan perbedaan posisi

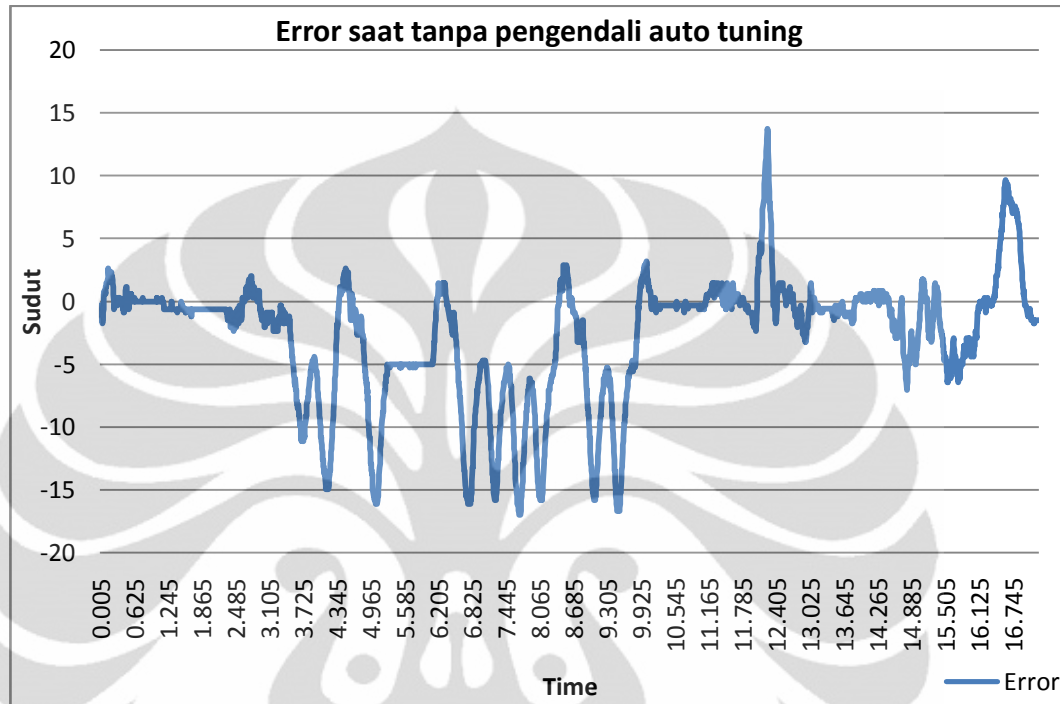
Setelah mendapatkan sistem yang ideal seperti pada akhir sub bab sebelumnya, maka pada sistem ini diterapkan variasi sudut distribusi torsi pada manipulator master sedangkan sudut distribusi torsi untuk manipulator slave tidak divariasikan melainkan mengikuti nilai inisial dari dynamixel itu sendiri. Hasil dari pengujian ini didapat perbedaan posisi masing-masing manipulator yang relatif lebih kecil dengan taraf keringanan operator untuk menggerakkan manipulator master lebih baik, sehingga operator dapat mudah menggerakkan manipulator master. Akan tetapi, apabila sistem bekerja pada kondisi manipulator slave menyentuh benda pada lingkungan, maka nilai perbedaan posisi akan bertambah.



Gambar 4.37 Grafik posisi master dan slave saat kondisi tanpa pengendali *auto-tuning*

Dapat dilihat pada gambar diatas, kondisi sistem yang sama dengan yang ada pada gambar 4.35 pada sub bab sebelumnya menghasilkan perbedaan posisi yang signifikan pada saat manipulator slave menyentuh benda seperti yang

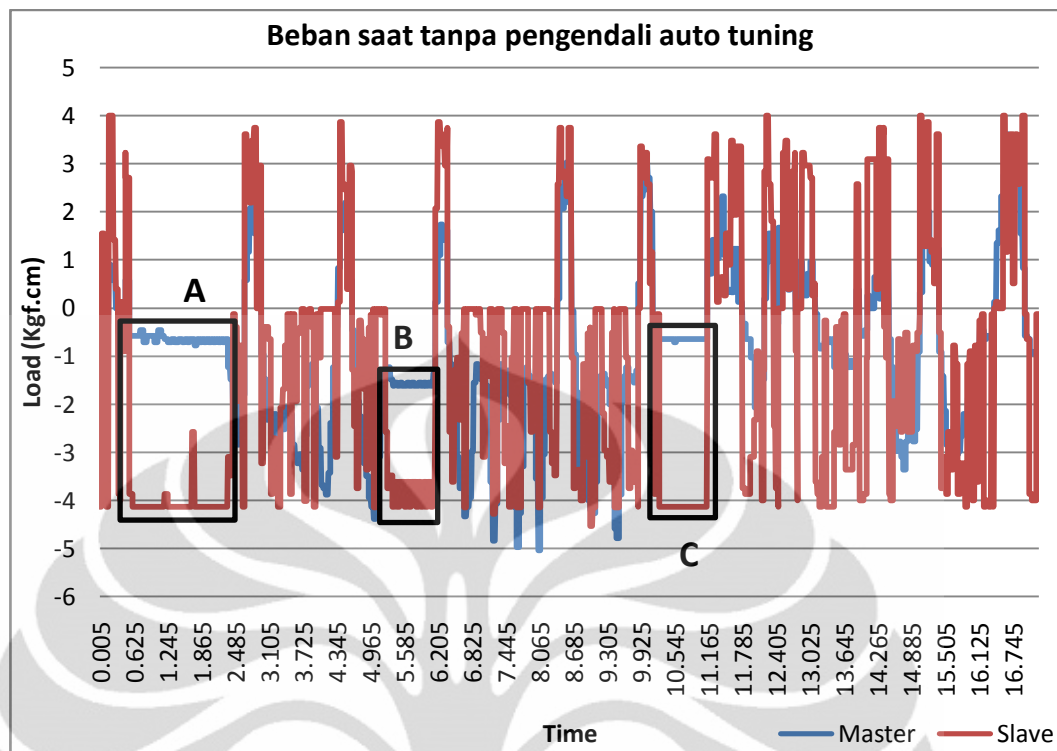
ditandai kotak berlabel 'A' sampai 'E' pada grafik diatas. Pengujian ini dilakukan dengan menggerakkan manipulator master dan pada posisi yang telah ditentukan dibuat pengganjal yang terletak didepan manipulator slave sehingga manipulator slave tertahan.



Gambar 4.38 Grafik perbedaan posisi master dan slave pada sistem

Terlihat bahwa perbedaan posisi antar manipulator mencapai nilai maksimal sebesar  $16.7^\circ$ . Nilai ini dapat dilihat juga sebagai perbandingan dengan gambar 4.29, dimana pada gambar tersebut perbedaan posisi antar manipulator mencapai nilai  $25.1^\circ$ . Nilai pada gambar 4.38 yang bernilai  $16.7^\circ$ , merupakan setengah dari nilai sebelumnya dikarenakan pada kondisi ini manipulator yang diatur sudut distribusinya hanya manipulator master sedangkan pada subbab sebelumnya sudut distribusi torsi yang diatur adalah mencakup untuk kedua manipulator.





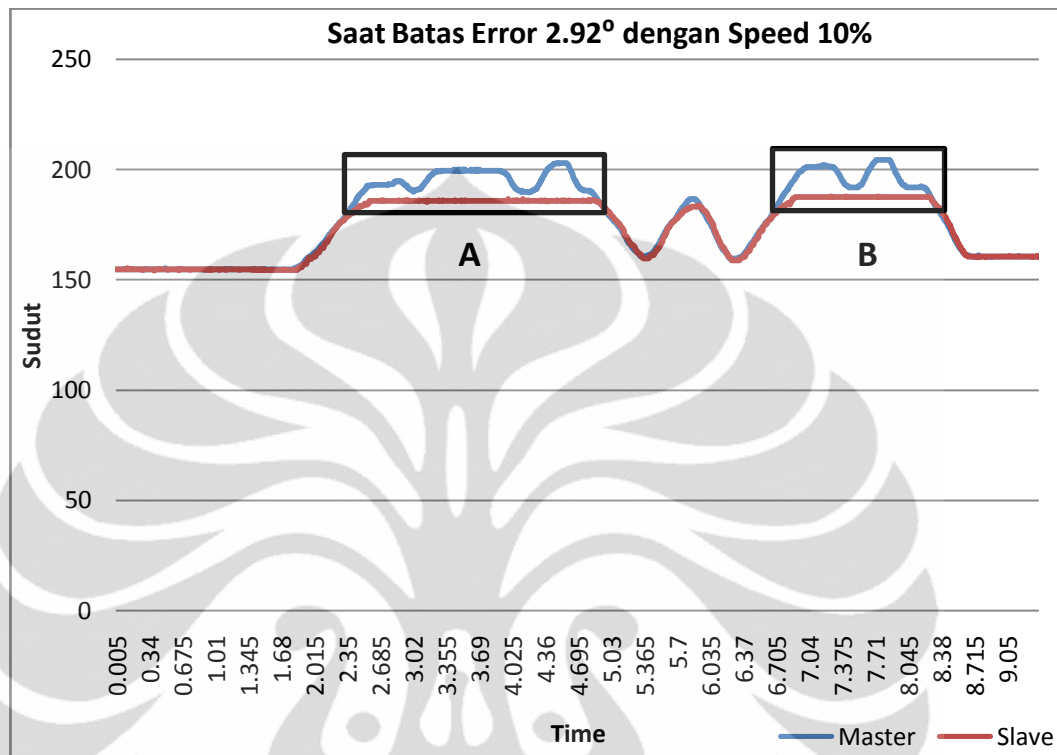
Gambar 4.39 Grafik pembacaan beban Saat kondisi tanpa pengendali *auto-tuning*

Pada grafik pembacaan beban pada pengujian yang sama, terlihat bahwa pada bagian yang diberi tanda kotak diatas, nilai beban master lebih kecil dibandingkan dengan nilai beban slave yang terlihat lebih besar. Hal ini disebabkan karena pengaturan sudut distribusi torsi pada manipulator master bernilai lebih landai atau memiliki nilai slope yang lebih besar dibandingkan dengan manipulator slave yang diatur agar sudut distribusi torsinya lebih curam. Hal ini juga yang menyebabkan apabila manipulator slave tersangkut atau menyentuh benda maka manipulator master masih dapat digerakkan sampai kepada batas maksimal  $16.7^\circ$ .

Untuk mensiasati kondisi ini maka diperlukan suatu algoritma yang apabila manipulator slave menyentuh benda maka sudut distribusi torsi pada manipulator master menjadi lebih besar. Parameter yang dapat digunakan adalah perbedaan posisi antar manipulator. Untuk parameter beban yang terbaca tidak dapat digunakan karena terlalu banyak gangguan pada hasil pembacaannya.

Seperti yang telah dijelaskan secara umum pada bab sebelumnya, pada gambar 3.14, perbedaan posisi antar manipulator akan dicari harga mutlaknya pada setiap waktu cuplik dan apabila perbedaan posisi tersebut lebih besar dari

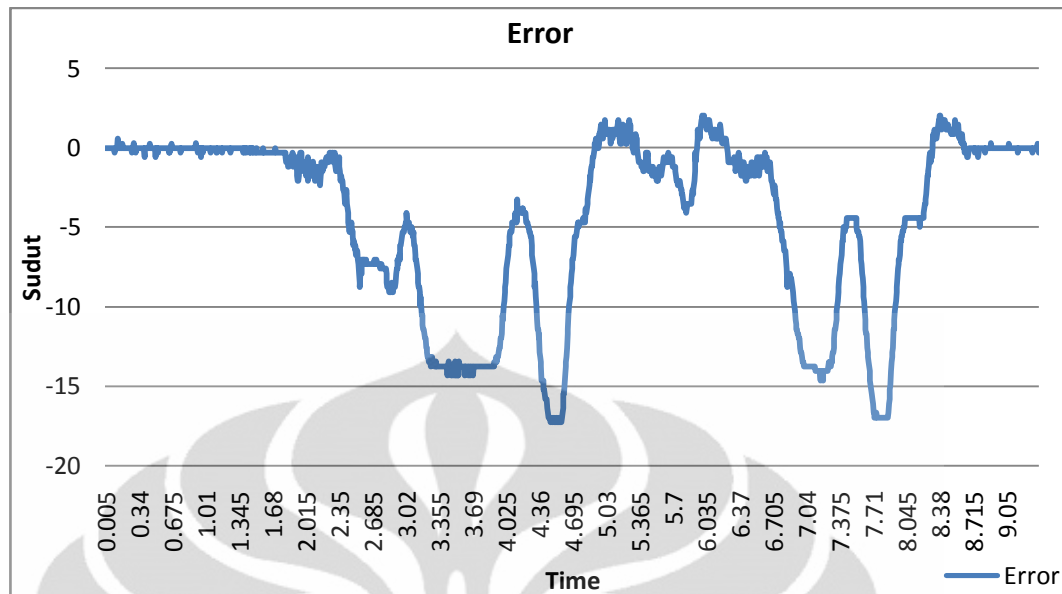
nilai yang telah ditentukan sebagai batas error, maka nilai slope akan dikurangi oleh sebuah nilai yang telah diformulakan sebagai laju perubahan sudut distribusi torsi.



Gambar 4.40 Grafik posisi master dan slave saat mode *Auto-Tuning* dengan nilai batas error  $2.92^\circ$  dengan laju perubahan 10%

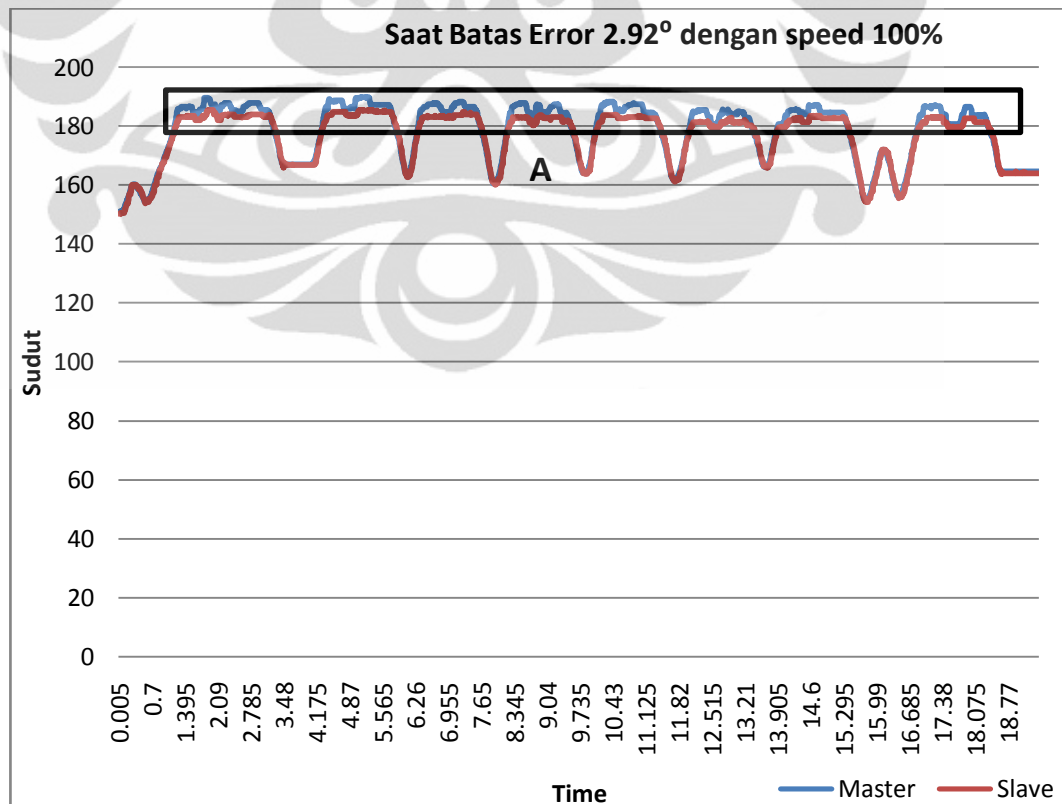
Pada pengujian pertama, sistem diterapkan dengan parameter batas error sebesar  $2.92^\circ$  dan laju perubahan sudut distribusi torsi diatur menjadi 10%. Laju perubahan sudut distribusi torsi ini memiliki kisaran 100% yaitu pada saat nilai slope akan berkurang sebanyak 48 atau setara  $14.06^\circ$  pada setiap waktu cupliknya, sehingga pada kecepatan 10% maka nilai slope akan dikurangi oleh nilai 4 pada setiap waktu cupliknya.



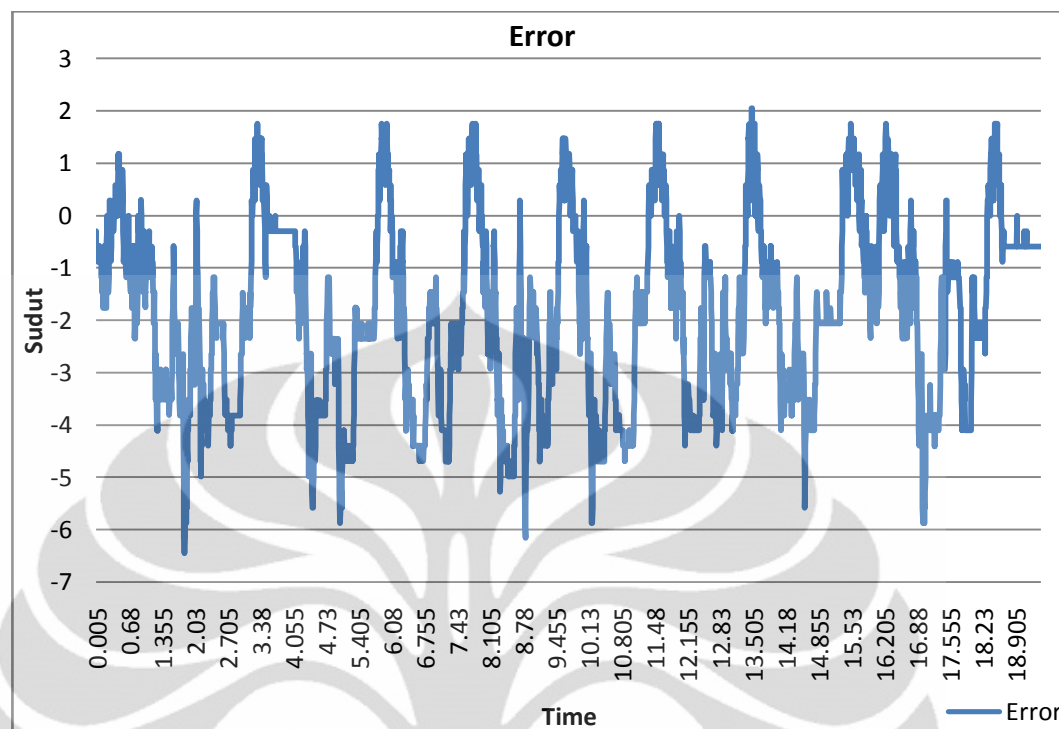


Gambar 4.41 Perbedaan posisi master dan slave pada sistem

Dapat dilihat pada laju kecepatan perubahan nilai sudut yang terlalu kecil mengakibatkan perbedaan posisi antar manipulator menjadi sama seperti sistem tanpa menggunakan algoritma *auto-tuning* ini. Berikut grafik posisi manipulator dengan laju perubahan nilai sudut distribusi torsi sebesar 100%.

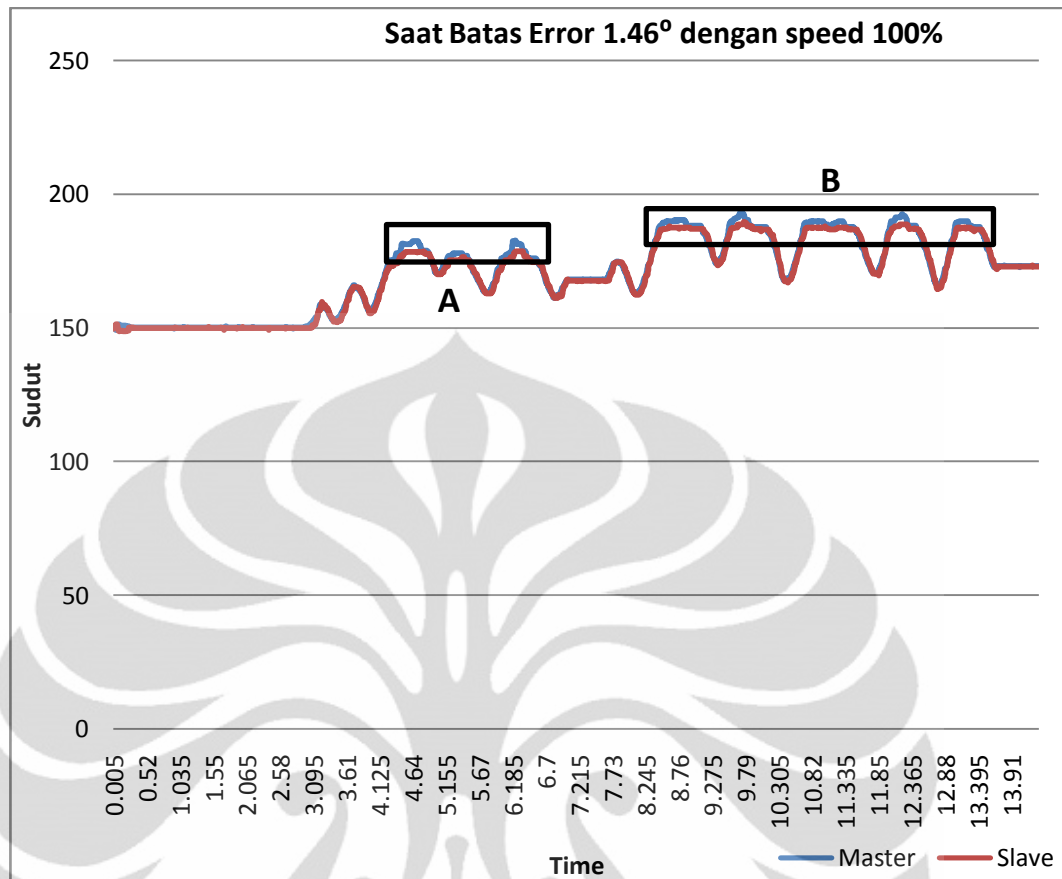


Gambar 4.42 Grafik posisi master dan slave saat mode *Auto-Tuning* dengan nilai batas error  $2.92^\circ$  dengan laju perubahan 100%

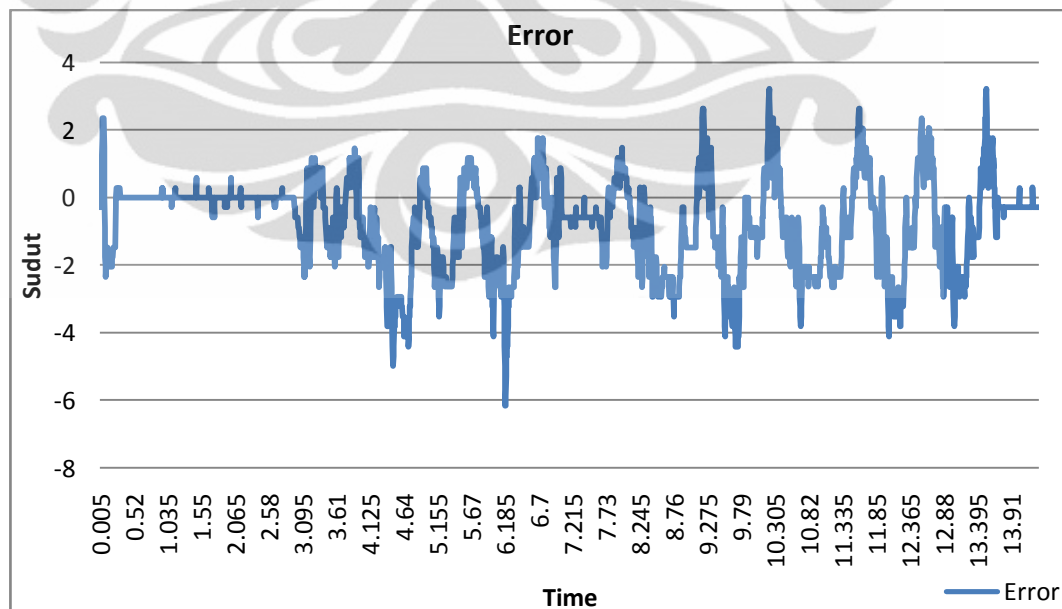


Gambar 4.43 Perbedaan posisi master dan slave pada sistem

Dapat dilihat bahwa nilai maksimum perbedaan posisi antar manipulator pada saat manipulator slave menyentuh benda adalah  $6.44^\circ$  dengan rata-rata perbedaan sebanyak  $4^\circ$ . Hal ini menunjukkan adanya peningkatan kinerja sistem dengan mengurangi nilai perbedaan posisi antar manipulator pada saat sistem mengalami sensasi dari luar (*disturbance*). Pada kondisi ini manipulator tetap dapat digerakkan oleh operator secara mudah namun pada saat manipulator slave menyentuh suatu benda maka manipulator master yang digerakkan oleh operator akan memiliki sensasi seperti menahan karena nilai sudut distribusi torsinya yang meningkat. Oleh karena itu pada sistem ini terjadi peningkatan dari sebelumnya yang hanya mengutamakan keringanan operator dalam menggerakkan manipulator dengan mengesampingkan efek *haptic* yang ada menjadi sistem yang dapat tetap menjaga keringanan dalam menggerakkan manipulator dan tetap menimbulkan efek *haptic*.



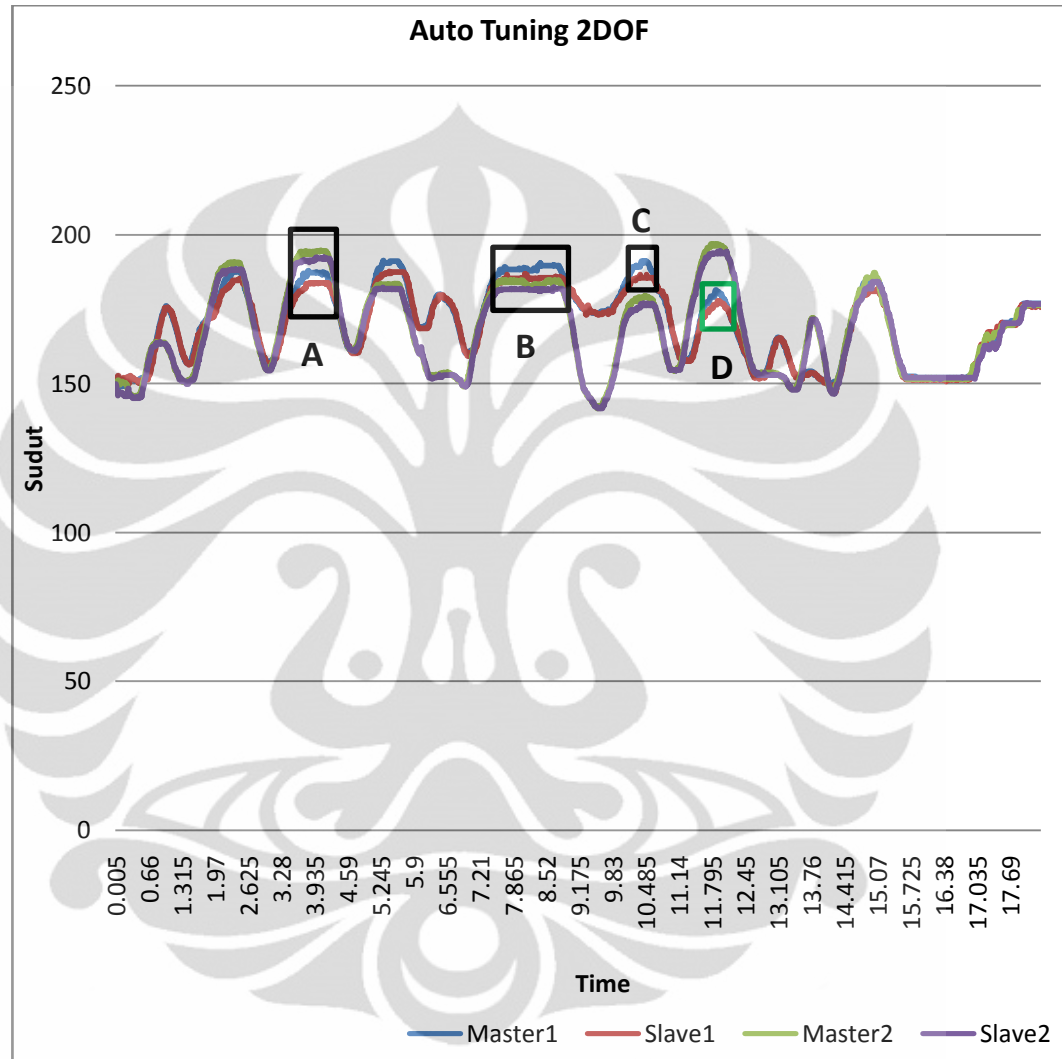
Gambar 4.44 Grafik posisi master dan slave saat mode *Auto-Tuning* dengan nilai batas error  $1.46^\circ$  dengan laju perubahan 100%



Gambar 4.45 Perbedaan posisi master dan slave pada sistem

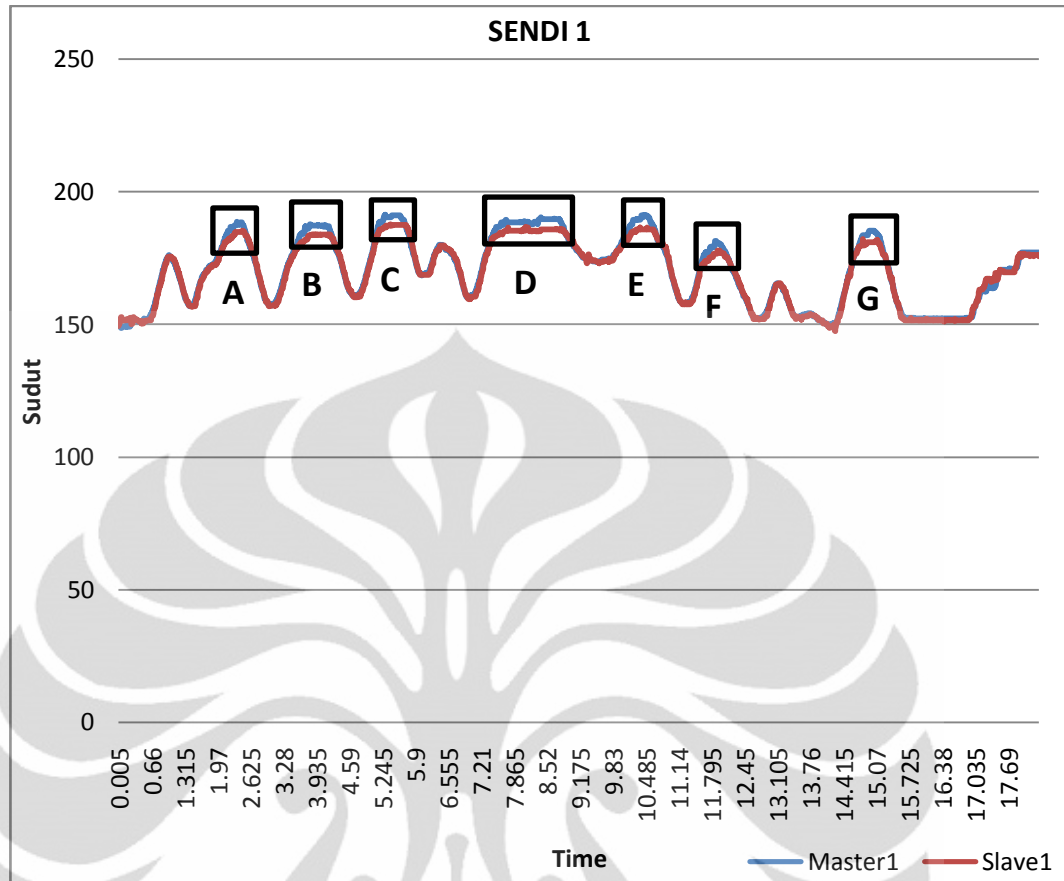
Pada grafik diatas, dilihat bahwa nilai maksimum perbedaan posisi antar manipulator pada saat manipulator slave menyentuh benda adalah  $6.15^\circ$  dengan

rata-rata perbedaan sebanyak  $2.94^\circ$ . Nilai ini mendekati nilai rata-rata perbedaan posisi pada saat sistem bekerja ideal dengan normal tanpa manipulator slave menyentuh benda. Berikut adalah grafik penerapan sistem dengan menggunakan 2-DOF.

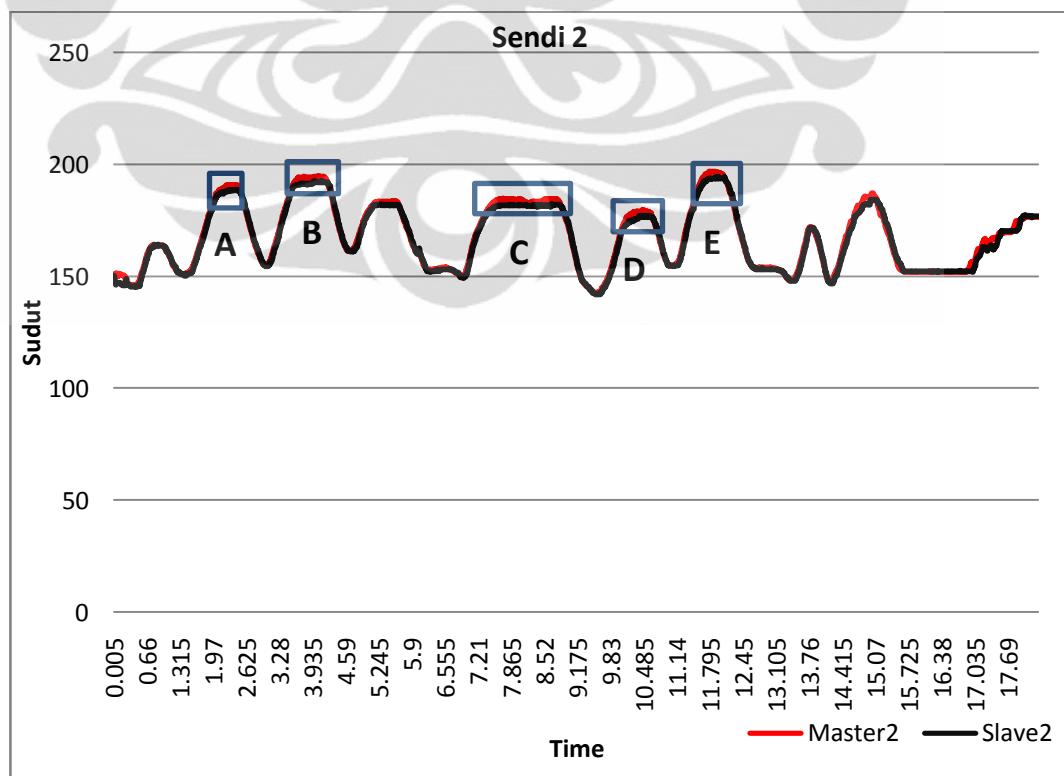


Gambar 4.46 Penerapan sistem dengan menggunakan 2 DOF

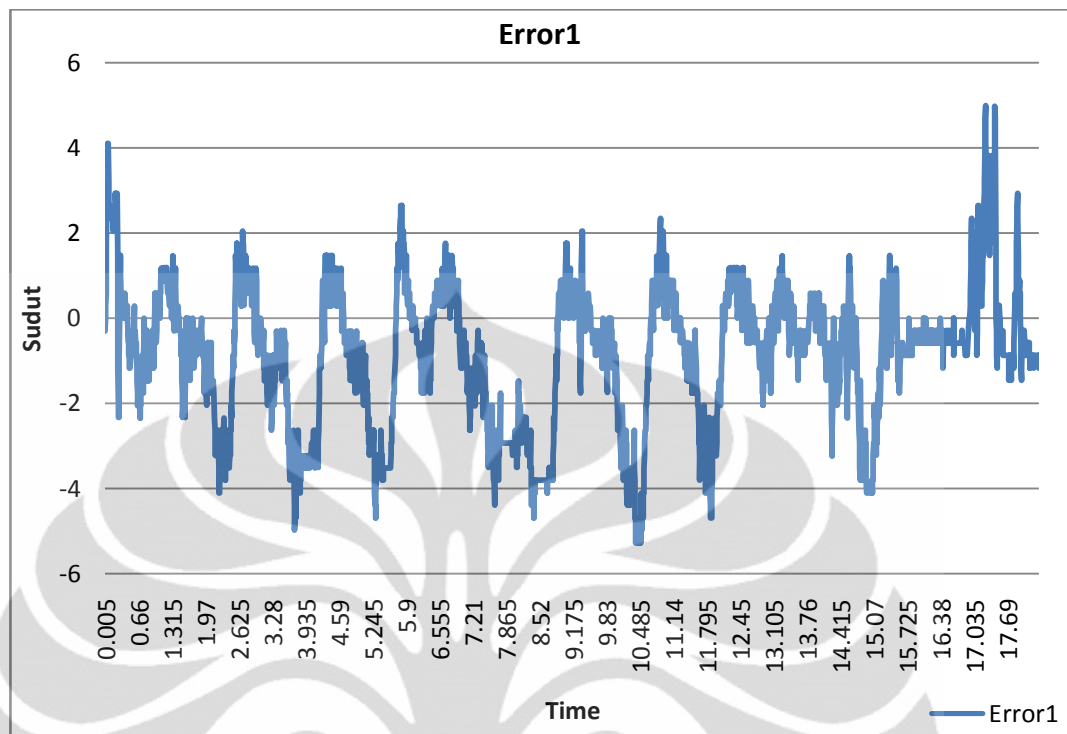
Penerapan terhadap 2-DOF berlaku sama seperti untuk sistem auto-tuning pada pembahasan sebelumnya hanya saja untuk membedakan sensasi usaha yang dilakukan untuk menggerakkan kedua sendi pada manipulator master oleh operator maka sendi kedua diberikan nilai konstanta *force* ( $K_f$ ) yang lebih besar dari sendi pertama. Hal ini disebabkan oleh beban yang ada pada sendi pertama sedikit lebih besar dibandingkan dengan sendi kedua karena secara normal sendi pertama juga menerima beban berat motor sendi kedua.



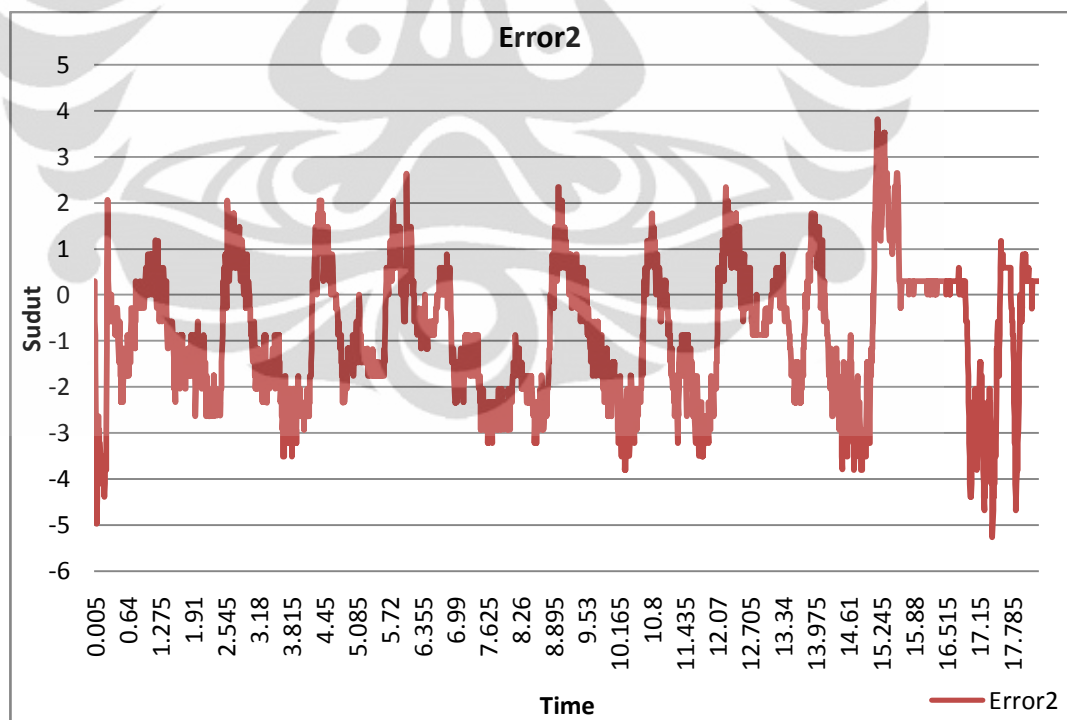
Gambar 4.47 Penerapan sistem pada sendi pertama



Gambar 4.48 Penerapan sistem pada sendi kedua



Gambar 4.49 Perbedaan posisi antar manipulator sistem pada sendi pertama



Gambar 4.50 Perbedaan posisi antar manipulator sistem pada sendi kedua

Terlihat jelas bahwa pada kedua sendi nilai maksimal perbedaan sudut antar manipulator pada kondisi menyentuh sesuatu berkisar diantara rata-rata  $3.5^\circ$ .

Hal ini jauh lebih baik dibandingkan sistem tanpa algoritma auto-tuning untuk mengatur sudut distribusi torsi.

Berikut disertakan tabel rincian hasil dari beberapa pengujian yang telah dilakukan sebelumnya beserta parameter yang didapat agar sistem bekerja secara optimal.

Tabel 4.1 Rincian Hasil Pengujian

Pengujian	Hasil
Penentuan Parameter Kp dan Kv	<ul style="list-style-type: none"> <li>• Pengujian dilakukan dengan pergerakan normal (<i>free motion</i>)</li> <li>• Nilai Optimal Kp sebesar 175 dan Kv sebesar 18.52</li> </ul>
Penentuan Frekuensi Cut-off pada filter	<ul style="list-style-type: none"> <li>• Pengujian dilakukan dengan pergerakan normal (<i>free motion</i>) dan pergerakan sentakan (secara tiba-tiba)</li> <li>• Nilai optimal frekuensi cut-off sebesar 1300 rad/s atau 206.9 Hz</li> </ul>
Penentuan Parameter Kf	<ul style="list-style-type: none"> <li>• Pengujian dilakukan dengan pergerakan normal (<i>free motion</i>)</li> <li>• Nilai optimal untuk Kf pada sendi pertama sebesar 3.5 dan pada sendi kedua sebesar 4.5</li> </ul>
Variasi sudut distribusi torsi	<ul style="list-style-type: none"> <li>• Pengujian dilakukan dengan pergerakan normal (<i>free motion</i>) dan pergerakan saat manipulator tertahan</li> <li>• Nilai optimal parameter slope manipulator slave sebesar 32 dan manipulator master sebesar 254.</li> <li>• Perbedaan posisi maksimum 2.3° CW dan 4.6° CCW</li> <li>• Perbedaan posisi rata rata 1.2° CW dan 1.6° CCW</li> </ul>
<i>Auto-Tuning Torque</i>	<ul style="list-style-type: none"> <li>• Pengujian dilakukan dengan pergerakan normal (<i>free motion</i>) dan</li> </ul>

	<p>pergerakan saat manipulator tertahan</p> <ul style="list-style-type: none"> <li>• Nilai optimal untuk nilai error yang ditentukan adalah 1.46 dengan laju perubahan 100%</li> <li>• Perbedaan posisi maksimum saat pergerakan normal <math>3.22^\circ</math> dan saat manipulator tertahan sebesar <math>6.15^\circ</math></li> <li>• Perbedaan posisi saat pergerakan normal rata-rata <math>0.64^\circ</math> dan saat manipulator tertahan rata-rata <math>1.6^\circ</math></li> </ul>
Aplikasi pada sistem dengan 2 DOF	<ul style="list-style-type: none"> <li>• Pengujian dilakukan dengan pergerakan normal (<i>free motion</i>) dan pergerakan saat manipulator tertahan</li> <li>• Nilai optimal untuk sistem adalah nilai-nilai yang telah disebutkan diatas</li> </ul>



## KESIMPULAN

Dari keseluruhan hasil pengujian yang telah dilakukan dalam skripsi ini, dapat disimpulkan beberapa poin penting yaitu:

1. Sistem teleoperasi bilateral yang digunakan dengan kemampuannya untuk menyesuaikan torsi secara otomatis yang berguna untuk meningkatkan sensasi haptic menggunakan Dynamixel AX-12+ berbasis ATmega128 dapat direalisasikan.
2. Penggunaan umpan balik *force* (Bilateral 4 kanal) dapat meningkatkan efek haptic untuk operator dibandingkan dengan sistem tanpa umpan balik *force* (Bilateral 2 kanal).
3. Dengan adanya pengaturan sudut distribusi torsi pada manipulator master, maka operator semakin mudah untuk menggerakkan manipulator master dan sistem tetap dapat dengan cepat mengikuti nilai referensi yang diberikan.
4. Pada pergerakan free motion tanpa ada hambatan manipulator dari masing-masing sendi dapat mengikuti dengan rata-rata perbedaan posisi sebesar  $0.64^\circ$  dengan nilai maksimal sebesar  $3.22^\circ$ .
5. Dengan adanya algoritma penyesuaian torsi motor maka pada saat manipulator slave tertahan, perbedaan posisi yang dihasilkan relatif lebih kecil dibandingkan dengan saat tidak menggunakan algoritma penyesuaian torsi sehingga efek *haptic* dapat ditimbulkan.
6. Pada pergerakan saat manipulator slave tertahan didapat rata-rata perbedaan posisi antar manipulator sebesar  $1.6^\circ$  dengan nilai maksimal sebesar  $6.15^\circ$

## DAFTAR ACUAN

- [1] Hannaford, B. (1989). A Design Framework for Teleoperators with Kinesthetic Feedback. *IEEE Transactions on Robotics and Automation* , 5, 426-433.
- [2] Ching, H. (2006, December). Internet Based Bilateral Teleoperation.
- [3] Anderson, R. J., & Spong, M. W. (1989). Bilateral Control of Teleoperators with Time Delay. *IEEE Transactions on Automatic Control* , 34, 494-501.
- [4] Flemmer, H., Eriksson, B., & Wikander, J. (1999). Control Design and Stability Analysis of A Surgical Teleoperator. *Mechatronics* , 9, 843-866.
- [5] Daniel, R., & McAree, R. (1998). Fundamental limits of performance for force reflecting teleoperation. *International Journal of Robotics Research* , 17, 811-830
- [6] Hirche, S., Stanczyk, B., & Buss, M. (2003). Transparent Exploration of Remote Environment by Internet Telepresence. *International Conference on Humanoid Robots* .
- [7] *Haptic control for dummies: An Introduction and analysis*. (2006). Eindhoven: Control System Technology Group.
- [8] Tira-Thompson, E. (2008). Digital Servo Calibration and Modelling.
- [9] Mensink, A. *Characterization and Modelling of Dynamixel servo*. University of twente.
- [10] Barnett, R. H., Cox, S., & O'Cull, a. L. (2007). *Embedded C Programming and The Atmel AVR 2nd Edition*. Thomson Delmar Learning.
- [11] Muis, A., & Ohnishi, K. (2006, December). Teleoperation with soft sensation based on Transformer and Gyration Theorems. *IEEE International Conference on Industrial Technology* , 346-351.
- [12] Katsura, S., Matsumoto, Y., & Ohnishi, K. (2005). Realization of "Law of Action and Reaction" by Multilateral Control. *IEEE Transactions on Industrial electronics* , 52, 1196-1205.

## DAFTAR PUSTAKA

- Barnett, R. H., Cox, S., & O'Cull, a. L. (2007). *Embedded C Programming and The Atmel AVR 2nd Edition*. Thomson Delmar Learning.
- Burns, R. S. (2001). *Advanced Control Engineering*. Oxford: Butterworth-Heinemann.
- Ching, H. (2006, December). Internet Based Bilateral Teleoperation.
- Eady, F. (2009, April). Unwinding The AX-12+ Communication Protocol. *SERVO Magazine*, pp.30-37.
- Hannaford, B. (1989). A Design Framework for Teleoperators with Kinesthetic Feedback. *IEEE Transactions on Robotics and Automation* , 5, 426-433.
- Haptic control for dummies: An Introduction and analysis*. (2006). Eindhoven: Control System Technology Group.
- Houpis, C. H., & Lamont, G. B. (1992). *Digital Control Systems : Theory, Hardware, Software*. Singapore: McGraw-Hill.
- Katsura, S., Matsumoto, Y., & Ohnishi, K. (2005). Realization of "Law of Action and Reaction" by Multilateral Control. *IEEE Transactions on Industrial electronics* , 52, 1196-1205.
- Mensink, A. *Characterization and Modelling of Dynamixel servo*. University of twente.
- Muis, A., & Ohnishi, K. (2006, December). Teleoperation with soft sensation based on Transformer and Gyrator Theorems. *IEEE International Conference on Industrial Technology* , 346-351.
- Ogata, K. (2010). *Modern Control Engineering 5th Edition*. Upper Saddle River: Prentice Hall.
- Paynter, R. T. (1989). *Introductory Electronic Devices and Circuits*. Englewood Cliffs, New Jersey: Prentice-Hall.
- Sandin, P. E. (2003). *Robot Mechanisms and Mechanical Devices*. Chicago: McGraw-Hill.
- Tira-Thompson, E. (2008). Digital Servo Calibration and Modelling.

## LAMPIRAN

Address	Item	Access	Initial Value	Area
0(0x00)	Model Number (L)	RD	12(0x0C)	EEPROM
1(0x01)	Model Number (H)	RD	0(0x00)	EEPROM
2(0x02)	Version of Firmware	RD	?	EEPROM
3(0x03)	ID	RD,WR	1(0x01)	EEPROM
4(0x04)	Baud Rate	RD,WR	1(0x01)	EEPROM
5(0x05)	Return Delay Time	RD,WR	250(0xFA)	EEPROM
6(0x06)	CW Angle Limit(L)	RD,WR	0(0x00)	EEPROM
7(0x07)	CW Angle Limit(H)	RD,WR	0(0x00)	EEPROM
8(0x08)	CCW Angle Limit(L)	RD,WR	255(0xFF)	EEPROM
9(0x09)	CCW Angle Limit(H)	RD,WR	3(0x03)	EEPROM
10(0x0A)	(Reserved)	-	0(0x00)	EEPROM
11(0x0B)	the Highest Limit Temperature	RD,WR	85(0x55)	EEPROM
12(0x0C)	the Lowest Limit Voltage	RD,WR	60(0x3C)	EEPROM
13(0x0D)	the Highest Limit Volage	RD,WR	190(0xBE)	EEPROM
14(0x0E)	Max Torque(L)	RD,WR	255(0xFF)	EEPROM
15(0x0F)	Max Torque(H)	RD,WR	3(0x03)	EEPROM
16(0x10)	Status Return Level	RD,WR	2(0x02)	EEPROM
17(0x11)	Alarm LED	RD,WR	4(0x04)	EEPROM
18(0x12)	Alarm Shutdown	RD,WR	4(0x04)	EEPROM
19(0x13)	(Reserved)	RD,WR	0(0x00)	EEPROM
20(0x14)	Down Calibration(L)	RD	?	EEPROM
21(0x15)	Down Calibration(H)	RD	?	EEPROM
22(0x16)	Up Calibration(L)	RD	?	EEPROM
23(0x17)	Up Calibration(H)	RD	?	EEPROM
24(0x18)	Torque Enable	RD,WR	0(0x00)	RAM
25(0x19)	LED	RD,WR	0(0x00)	RAM
26(0x1A)	CW Compliance Margin	RD,WR	0(0x00)	RAM
27(0x1B)	CCW Compliamce Margin	RD,WR	0(0x00)	RAM
28(0x1C)	CW Compliance Slope	RD,WR	32(0x20)	RAM
29(0x1D)	CCW Compliance Slope	RD,WR	32(0x20)	RAM
30(0x1E)	Goal Position(L)	RD,WR	[Addr36]value	RAM
31(0x1F)	Goal Position(H)	RD,WR	[Addr37]value	RAM
32(0x20)	Moving Speed(L)	RD,WR	0(0x00)	RAM
33(0x21)	Moving Speed(H)	RD,WR	0(0x00)	RAM
34(0x22)	Torque Limit(L)	RD,WR	[Addr14]value	RAM
35(0x23)	Torque Limit(H)	RD,WR	[Addr15]value	RAM
36(0x24)	Present Position(L)	RD	?	RAM

37(0x25)	Present Position(H)	RD	?	RAM
38(0x26)	Present Speed(L)	RD	?	RAM
39(0x27)	Present Speed(H)	RD	?	RAM
40(0x28)	Present Load(L)	RD	?	RAM
41(0x29)	Present Load(H)	RD	?	RAM
42(0x2A)	Present Voltage	RD	?	RAM
43(0x2B)	Present Temperature	RD	?	RAM
44(0x2C)	Registered Instruction	RD,WR	0(0x00)	RAM
45(0x2D)	(Reserved)	-	0(0x00)	RAM
46(0x2E)	Moving	RD	0(0x00)	RAM
47(0x2F)	Lock	RD,WR	0(0x00)	RAM
48(0x30)	Punch(L)	RD,WR	32(0x20)	RAM
49(0x31)	Punch(H)	RD,WR	0(0x00)	RAM

