



UNIVERSITAS INDONESIA

**RANCANG BANGUN ROBOT SAR BERKAKI EMPAT TIGA
SENDI DENGAN TEKNIK TELEOPERATION
MENGUNAKAN EMPAT LENGAN DUA SENDI**

SKRIPSI

IKHWANNUL KHOLIS

0806455276

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI 2012**



UNIVERSITAS INDONESIA

**RANCANG BANGUN ROBOT SAR BERKAKI EMPAT TIGA
SENDI DENGAN TEKNIK TELEOPERATION
MENGUNAKAN EMPAT LENGAN DUA SENDI**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

IKHWANNUL KHOLIS

0806455276

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI 2012**

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Ikhwannul Kholis

NPM : 0806455276

Tanda Tangan : 

Tanggal : 13 Juni 2012

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Ikhwannul Kholis
NPM : 0806455276
Program Studi : Teknik Elektro
Judul Seminar : Rancang Bangun Robot SAR Berkaki Empat Tiga Sendi dengan Teknik Teleoperation Menggunakan Empat Lengan Dua Sendi

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

Dewan Penguji

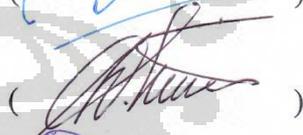
Pembimbing :

Dr. Ir. Abdul Muis, ST., M. Eng

()

Penguji :

Ir. Wahidin Wahab, M. Sc., Ph.D

()

Prof. Dr.Eng. Drs.Benyamin Kusumoputro, M.Eng.

()

Ir. Aries Subiantoro, M.SEE.

()

Ditetapkan di : Depok

Hari/Tanggal : 10 Juli 2012

UCAPAN TERIMA KASIH

Puji Syukur saya panjatkan kepada Allah swt. karena atas nikmat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penyusunan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Departemen Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, baik dari masa perkuliahan sampai pada penyusunan skripsi ini 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 yang telah menyediakan waktu, tenaga, dan pikiran dalam mengarahkan saya dalam penyusunan skripsi ini;
- (2) Orang tua dan seluruh keluarga saya yang telah memberikan dukungan moril maupun materiil.; dan
- (3) Sahabat-sahabat, yaitu Elkom 2008, Tim Robotika UI, serta senior Teknik Elektro, dan orang-orang terdekat, terutama teman-teman sebimbingan, yaitu Wisnu, JP, Imam, Anto, Haris, Alan, dan khususnya Tsalatsatun Andini, yang telah banyak membantu saya dalam menyelesaikan skripsi ini.

Saya berharap Allah Swt. berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 13 Juni 2012



Ikhwanul Kholis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS
AKHIR UNTUK KEPENTINGAN AKADEMIS**

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

Nama : Ikhwannul Kholis
NPM : 0806455276
Program Studi : Teknik Elektro
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis Karya : Skripsi

demikian 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 ROBOT SAR BERKAKI EMPAT TIGA SENDI
DENGAN TEKNIK TELEOPERATION MENGGUNAKAN EMPAT
LENGAN DUA SENDI**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/format-kan, mengelolanya dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya tanpa perlu meminta ijin dari 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 : 13 Juni 2012

Yang menyatakan



Ikhwannul Kholis

ABSTRAK

Nama : Ikhwannul Kholis
Program Studi : Teknik Elektro
Judul Skripsi : Rancang Bangun Robot SAR Berkaki Empat Tiga Sendi dengan Teknik Teleoperation Menggunakan Empat Lengan Dua Sendi

Sistem *bilateral teleoperation* menggunakan dua jenis informasi yaitu *teleoperation* dan *telepresence* yang bergerak dua arah membentuk *closed loop*. Sistem ini memungkinkan operator dapat secara langsung mengendalikan sebuah manipulator pada jarak tertentu dengan merasakan *feedback* dari robot. Dalam skripsi ini dibahas mengenai rancang bangun robot SAR berkaki empat tiga sendi dengan teknik teleoperation menggunakan empat lengan dua sendi dengan menggunakan serial servo Kondo dan prosesor Vortex86DX. Konsep *Master-Slave Transformation* digunakan untuk mengakomodasi bilateral teleoperation. Sistem ini menggunakan *forward kinematics* untuk menentukan koordinat posisi pada sisi *master* dan *slave transformation* untuk menentukan respon *slave*. Evaluasi kinerja sistem dilakukan dengan memperhatikan respon *slave* terhadap perubahan posisi *master*. Berdasarkan pengujian yang telah dilakukan, system ini masih memiliki *time delay* yang cukup besar akibat sistem mekanik yang kurang baik.

Kata kunci: *Bilateral teleoperation, Forward kinematics, Master-Slave Transformation.*

ABSTRACT

Name : Ikhwannul Kholis
Program : Electrical Engineering
Title : SAR Robot Development with Four 3-DOF Legs by Using
Teleoperation Through Four Two-DOF Link Manipulator.

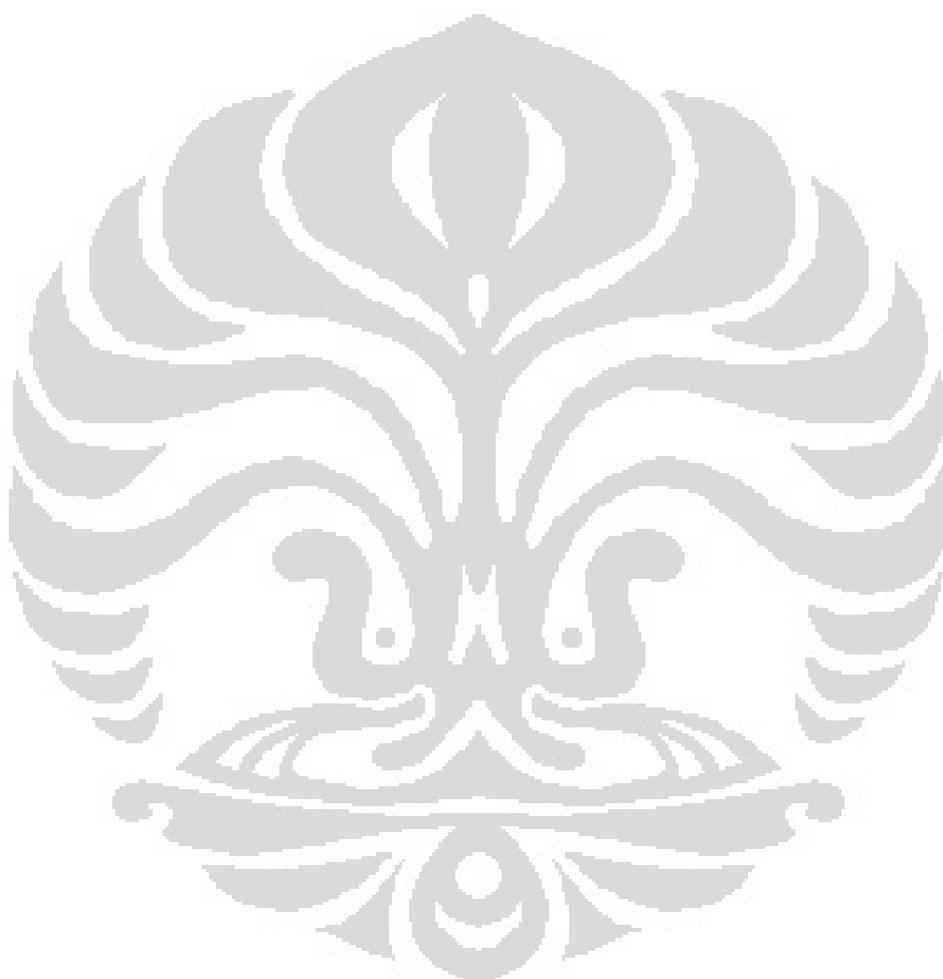
Bilateral teleoperation system use two kind of information which is teleoperation and telepresence exchanged forming a closed loop system. This system allow user to control the manipulator directly at a certain distance. This thesis explains about Design of SAR Robot with a 2 DOF - 3 DOF Bilateral Teleoperation using serial servo Kondo and Vortex86DX processor. Master-Slave Transformation is used to accommodate bilateral teleoperation. This system use forward kinematics to define coordinat position at master and inverse kinematics to define the angle of servo at *slave*. The system evaluated by monitoring slave responses against master at position change. At evaluation of this system, it still has the big time delay relatively because of mechanical system lack.

Keyword : Bilateral teleoperation, Forward kinematics, Master-Slave Transformation.

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS	ii
PENGESAHAN	iii
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR GAMBAR.....	x
DAFTAR TABEL	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan Penulisan.....	3
1.4 Pembatasan Masalah	3
1.5 Metode Penelitian	3
1.6 Sistematika Penulisan	3
BAB 2 LANDASAN TEORI	5
2.1 Prinsip Bilateral Teleoperation.....	5
2.2 Forward Kinematics untuk Robot Manipulator	6
2.3 Development board	13
2.4 Motor Serial Servo Kondo KRS 2552 HV.....	18
2.5 Komunikasi Data dengan Socket Programming	24
BAB 3 PERANCANGAN SISTEM	27
3.1 Konsep Bilateral Teleoperation	27
3.2 Perangkat Keras	28
3.2.1 Roboard	29
3.2.2 Servo Kondo.....	31
3.2.3 Wireless USB dan Access Point.....	32
3.2.4 Komponen Pendukung	33
3.3 Perangkat Lunak.....	36
3.3.1 Fungsi UDP sebagai komunikasi antar Roboard.....	37
3.3.2 Fungsi Komunikasi serial Servo kondo.....	40
3.3.3 Algoritma.....	41
3.4 Sistem Terintegrasi.....	46
BAB 4 PENGUJIAN DAN ANALISIS.....	48
4.1 Pengujian Bilateral Teleoperation 1	48
4.1.1 Percobaan Bilateral Teleoperation 1	48
4.1.2 Analisis Percobaan 1	53
4.2 Pengujian Bilateral Teleoperation 2	54
4.2.1 Pengujian Sistem	54
4.2.2 Analisis Percobaan 2	56
4.3 Pengujian Bilateral Teleoperation 3	57

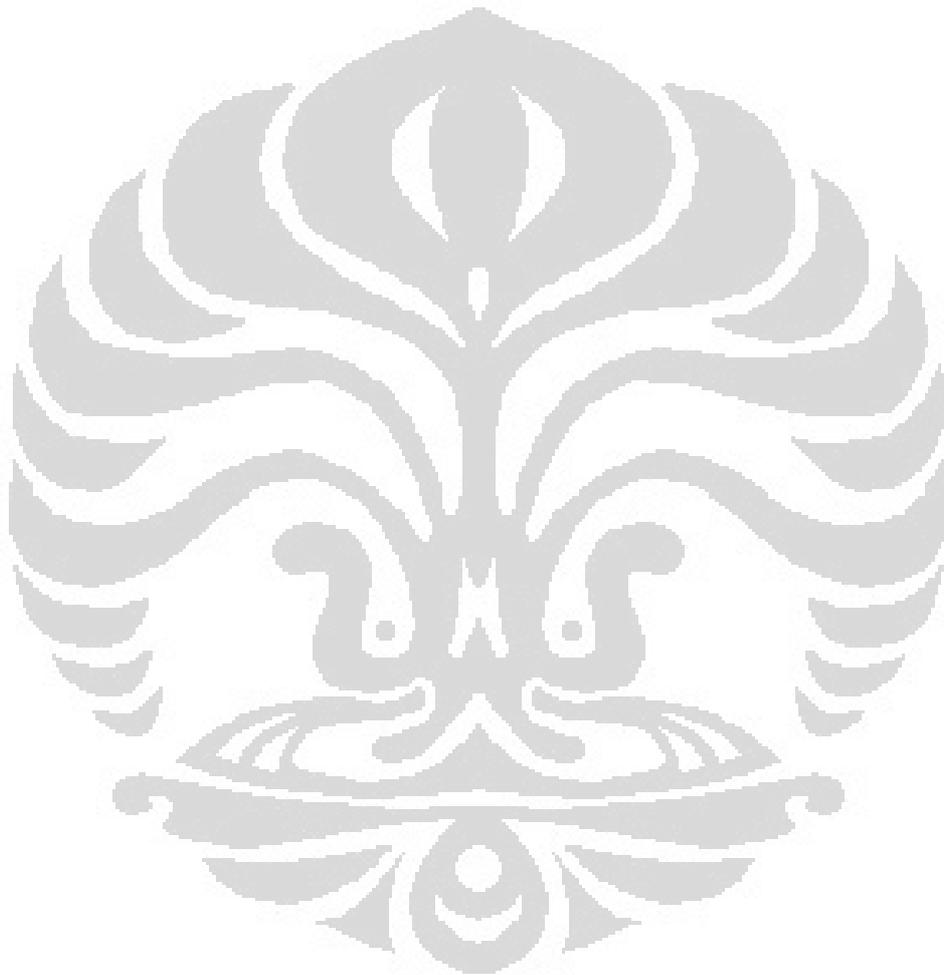
4.3.1 Pengujian Sistem	57
4.3.2 Analisis Percobaan 2	58
BAB 5 KESIMPULAN	59
DAFTAR ACUAN.....	60
DAFTAR REFERENSI.....	61
LAMPIRAN.....	62



DAFTAR GAMBAR

Gambar 1.1. Bilateral Teleoperation (Telesurgery)	2
Gambar 2.1. Blok Diagram Bilateral Teleoperation	6
Gambar 2.2. Forward Kinematics pada Koordinat Cartesian 2 Dimensi.....	7
Gambar 2.3. Dua Solusi pada Inverse Kinematics.....	8
Gambar 2.4. Konfigurasi Singular	10
Gambar 2.5. Forward Kinematics pada Koordinat Cartesian 3 Dimensi.....	11
Gambar 2.6. Dua Solusi pada Inverse Kinematics.....	12
Gambar 2.7. RB-100	13
Gambar 2.8. Vortex86DX Single On Chip (SoC).....	14
Gambar 2.9. Perbandingan RB-100 dengan CPU Board Lain.....	14
Gambar 2.10. Tampilan Lubuntu 10.04 LTS yang Dijalankan Oleh Roboard Pada Monitor.....	18
Gambar 2.11. Sudut Operasi Servo Kondo KRS 2552 HV	19
Gambar 2.12. Servo Kondo KRS 2552 HV	19
Gambar 2.13. Bentuk Komunikasi UDP.....	26
Gambar 3.1. Blok Diagram Sistem Bilateral yang Dirancang	27
Gambar 3.2. Blok Diagram Secara umum sistem Bilateral Teleoperation pada skripsi ini.....	29
Gambar 3.3. RB-100	30
Gambar 3.4. RB-110	30
Gambar 3.5. COM 3 (TTL, Full Duplex UART).....	31
Gambar 3.6. Kabel Servo Kondo pada Roboard.....	32
Gambar 3.7. Konfigurasi Servo Kondo, (a) Robot SAR, (b) Sisi Operator.....	32
Gambar 3.8. Wireless Router dan Wireless USB	33
Gambar 3.9. Kabel Serial TTL Half Duplex UART.....	33
Gambar 3.10. Gambar Sistem Mekanik Bilateral Teleoperation.....	34
Gambar 3.11. Desain Mekanik di AutoCad 2007	34
Gambar 3.12. Desain Mekanik (a) per bagian bracket servo, (b) per bagian kaki robot, (c) Robot SAR, dan (d) Robot Joystik Operator.....	36
Gambar 3.13. Flow Chart Sistem Bilateral teleoperation	42
Gambar 3.14. ID Servo pada Sistem (a) Sisi Operator (b) Sisi Robot.....	47
Gambar 4.1. Perubahan Posisi 1 Kaki pada Sisi Operator (a) Posisi Awal, dan (b) Posisi Akhir	49
Gambar 4.2. Perubahan Posisi Kaki Robot pada Pengujian Bilateral Teleoperation 1 (a) Posisi Awal, (b) Posisi Kedua, (c) Posisi Ketiga, (d) Posisi Keempat, dan (e) Posisi Akhir.....	50
Gambar 4.3. Grafik Pengujian Bilateral Teleoperation 1 (Sampling 1)	51

Gambar 4.4. Grafik Pengujian Bilateral Teleoperation 1 (Sampling 2)	51
Gambar 4.5. Grafik Pengujian Bilateral Teleoperation 1 (Sampling 3)	52
Gambar 4.6. Grafik Pengujian Bilateral Teleoperation 1 (Sampling 4)	52
Gambar 4.7. Grafik Pengujian Bilateral Teleoperation 1 (Sampling 5)	53
Gambar 4.8. Grafik Hasil Pengujian Bilateral Teleoperation 2 Sampling 1.....	55
Gambar 4.9. Grafik Hasil Pengujian Bilateral Teleoperation 2 Sampling 2.....	55
Gambar 4.10. Grafik Pengujian Bilateral Teleoperation 2 sampling 3.....	56
Gambar 4.11. Grafik Hasil Pengujian Bilateral Teleoperation 3	58



DAFTAR TABEL

Table 2.1. Spesifikasi RB-100 dan RB-110	15
Table 2.2. Paket Instruksi Serial Servo Kondo KRS 2552 HV	20
Table 2.3. Command pada Protokol ICS 3.0	20
Table 2.4. Jenis SubCommand pada Kondo KRS 2552 HV	21
Table 2.5. Paket Status pada Protokol ICS 3.0	21
Table 2.6. Pengaturan Posisi Servo Kondo KRS 2552 HV	22
Table 2.7. Format POS_H dan POS_L	22
Table 2.8. Paket Instruksi Sudut 90° pada Servo ID 1	22
Table 2.9. Paket Status Pengiriman Instruksi Sudut 90° pada ID 1	23
Table 2.10. Instruksi Freerun Mode pada Servo ID 1	23
Table 2.11. Packet Status Servo Kondo KRS 2552 HV	23
Table 2.12. Pembacaan Speed pada Servo ID 1	24
Table 2.13. Paket Status Pembacaan Speed pada Servo	24
Table 2.14. Penulisan Speed pada Servo ID 1 dengan Data Speed Sebesar 2	24
Table 3.1. Transmisi Data Antar Roboard	39
Table 3.2. Pengiriman Data <i>Master Slave</i>	43
Table 3.3. Konfigurasi Data Dx dan Dy Pada Setiap Data yang Dikirim	43
Table 3.4. Gerakan Robot	45

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi robotika saat ini sangat pesat. Hal ini ditandai dengan banyaknya teknologi robotika yang ditemukan dengan tujuan memudahkan pekerjaan manusia. Pada dasarnya, robot dibuat untuk memudahkan pekerjaan manusia yang bersifat *dumb* (bodoh), *dirty* (kotor), *dull* (membosankan), dan *dangerous* (berbahaya). Dengan demikian, tujuan diciptakannya robot memang untuk kebaikan kehidupan manusia.

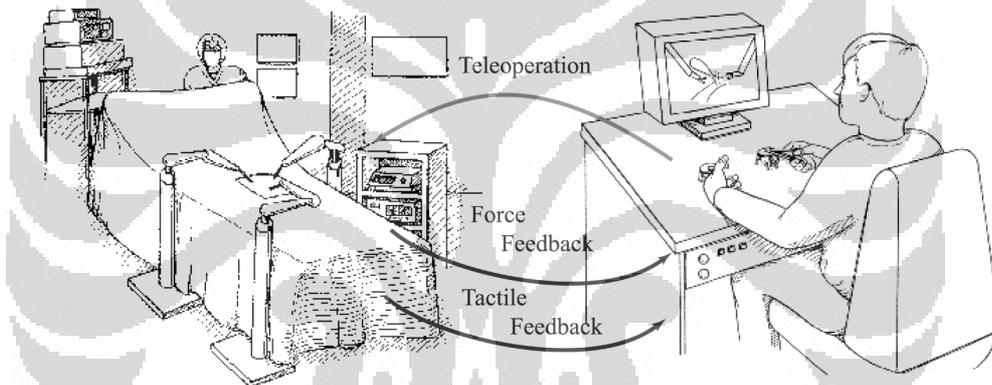
Terdapat beberapa bidang pada teknologi robotika, salah satunya adalah *Telerobotics*. *Telerobotics* merupakan salah satu bidang pada teknologi robotika yang membahas tentang pengendalian robot dari jarak jauh. *Telerobotics* memiliki dua informasi dalam penggunaannya, yaitu *Teleoperation*, yaitu kemampuan operator untuk memanipulasi objek dari jarak jauh, dan *Telepresence*, yaitu umpan balik dari lingkungan robot untuk dikirimkan kepada operator dalam berbagai bentuk, seperti visual, audio, atau umpan balik tekanan (*haptic*). *Teleoperation* membutuhkan remote control untuk mengendalikan berbagai mesin, misalnya robot manipulator, *remotely operated vehicle* (ROV), *unmanned Aerial Vehicle* (UAV), mesin industri, dan lain-lain. Selain itu, teleoperation dapat pula melibatkan pengendalian suatu alat dari jarak jauh dengan skala yang berbeda antara user dan lingkungan robot.

Sistem teleoperation yang tidak memiliki *force feedback* biasanya disebut sebagai *open-loop system* karena sistem hanya memiliki satu jalur aliran informasi. Sistem ini disebut juga dengan *unilateral teleoperation*. Sedangkan, sistem teleoperation yang menggunakan *force feedback* biasanya disebut sebagai *closed-loop System* karena sistem memiliki 2 jalur aliran informasi. Sistem ini biasanya disebut juga dengan *bilateral teleoperation*.

Sistem yang memiliki *teleoperation* dengan *telepresence* berupa *haptic* termasuk sebagai *Bilateral Teleoperation System*. Hal ini karena informasi

mengalir melalui dua jalur antara operator dan robot sehingga membentuk suatu *closed-loop system*. *Haptic* berasal dari bahasa Yunani, *haphe*, yang berarti sensasi sentuhan. *Teleoperation* dengan *telepresence* berupa *haptic* memungkinkan adanya interaksi dengan user melalui force feedback pada joystick. Operator, yang disebut *master*, dapat merasakan *force feedback* yang diterima dari robot, yang disebut *slave*.

Teleoperation telah banyak dilakukan di berbagai bidang, misalnya pertambangan, operasi penjinakan bom, proses pembuangan sampah nuklir, dan lain-lain. Salah satu contoh penggunaan *bilateral teleoperation* adalah *telesurgery*, yaitu pembedahan jarak jauh. Dokter dapat melakukan pembedahan secara jarak jauh, misalnya melalui internet, dengan menggunakan kamera dan sensasi sentuhan pada joystick. Berikut ini adalah gambar telesurgery.



Gambar 1.1. Bilateral Teleoperation (Telesurgery)

Sumber : <http://www.emeraldinsight.com/>

Pada skripsi ini, kata *teleoperation* dimaksudkan untuk mendeskripsikan *bilateral teleoperation*. Perancangan sistem teleoperation pada skripsi ini terkait dengan proyek pembuatan robot *search and rescue* untuk kondisi pasca gempa. *Teleoperation* ini dimaksudkan untuk mengetahui kondisi robot di lingkungan pasca gempa sehingga operator dapat mengambil tindakan selanjutnya.

1.2 Perumusan Masalah

Perancangan sistem *teleoperation* pada skripsi ini merupakan pengembangan dari sistem-sistem *teleoperation* yang telah dirancang sebelumnya di lingkungan Universitas Indonesia. Permasalahan yang dibahas pada skripsi ini adalah bagaimana merancang suatu sistem *teleoperation* multi dof dengan

menggunakan serial servo dengan memetakan sistem 2 DOF di sisi *master* ke sistem 3 DOF di sisi *slave*.

1.3 Tujuan Penulisan

Tujuan penulisan skripsi ini adalah merancang dan mengoperasikan suatu sistem multi DOF *bilateral teleoperation* pada robot suatu SAR dengan menggunakan Master Empat Lengan Dua Sendi menggunakan mekanisme transformasi 2 DOF ke 3 DOF.

1.4 Pembatasan Masalah

Pembahasan dalam skripsi ini meliputi penggunaan board RB 100 dan RB 110 sebagai *development board*, pengendalian servo Kondo KRS 2552 HV, integrasi *Development Board* dengan serial servo, penggunaan komunikasi data, dan pemetaan sistem 2 DOF di *master* ke sistem 3 DOF di *slave*. Skripsi ini tidak membahas penambahan pengendali eksternal.

1.5 Metode Penelitian

Metode penelitian yang digunakan pada penyusunan skripsi ini adalah sebagai berikut.

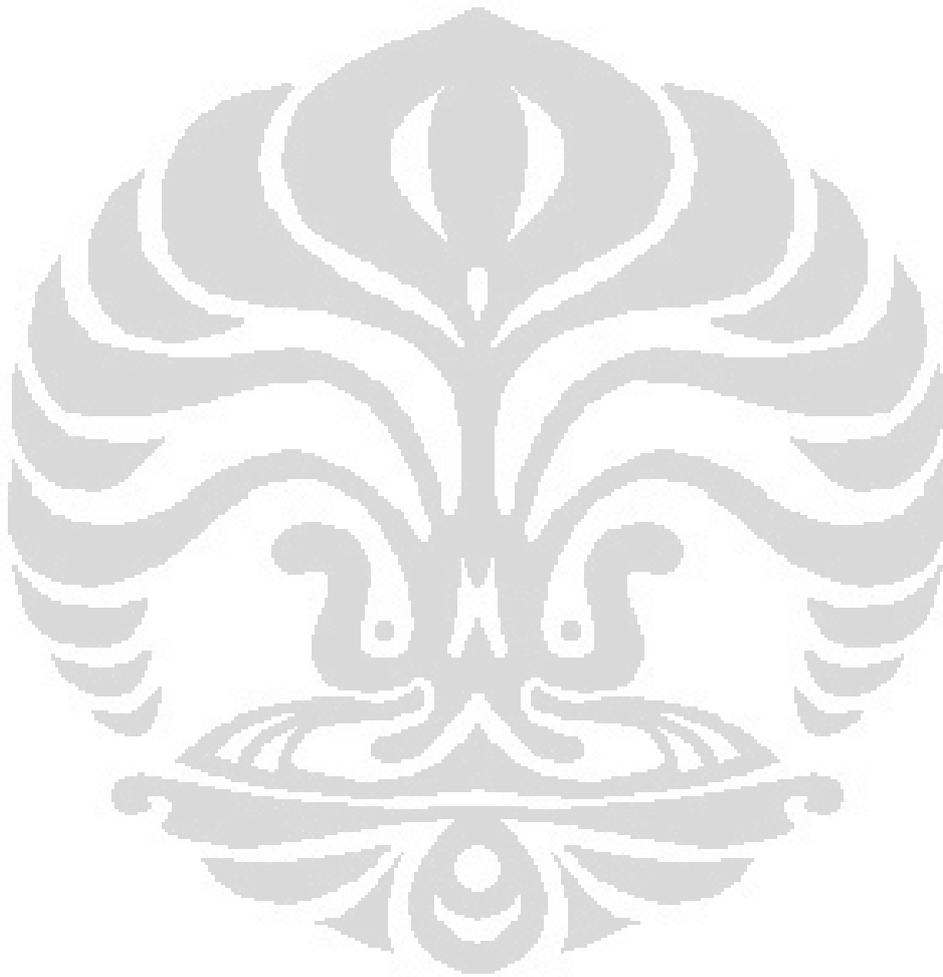
1. Studi Literatur mengenai sistem teleoperation yang pernah dikembangkan di lingkungan Universitas Indonesia.
2. Diskusi dengan pembimbing skripsi
3. Perakitan perangkat keras dan perancangan perangkat lunak
4. Pengujian sistem terintegrasi

1.6 Sistematika Penulisan

Skripsi ini ditulis dalam 5 bab, yaitu

1. BAB I Pendahuluan, berisi latar belakang, perumusan masalah, tujuan penulisan, pembatasan masalah, metode penelitian, dan sistematika penulisan.
2. BAB II Landasan Teori, yaitu berisi tentang landasan ilmu mengenai prinsip *bilateral teleoperation*, perangkat keras sistem *teleoperation* yang dirancang, dan pemetaan sistem 2 DOF ke sistem 3 DOF.
3. BAB III Perancangan, yaitu berisi tentang metode perancangan, langkah perancangan sistem hingga konfigurasi yang digunakan.

4. BAB IV Pengujian dan Analisa, berisi tentang analisa kinerja sistem dan pembahasan mengenai evaluasi kinerja sistem.
5. BAB V Penutup, berisi kesimpulan skripsi ini, dan saran.



BAB 2

LANDASAN TEORI

Pada bab ini, dijelaskan landasan ilmu dan teori yang digunakan untuk mengoperasikan sistem *bilateral teleoperation* 3 DOF dengan 2 DOF. Landasan teori yang dibahas adalah Prinsip *Bilateral Teleoperation*, *Forward Kinematics* pada Robot Manipulator, *Development Board* yang digunakan termasuk Sistem Operasinya, Servo kondo, dan Fungsi Komunikasi antara *master* dan *slave*.

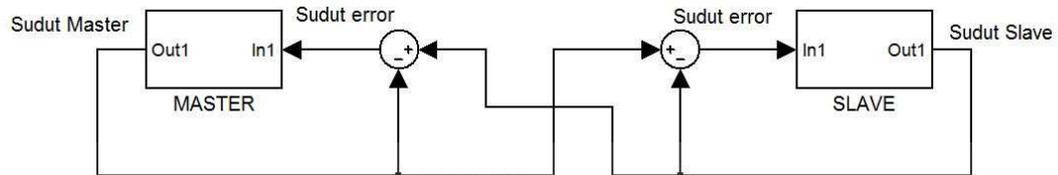
2.1 Prinsip Bilateral Teleoperation

Sistem Bilateral Teleoperation merupakan sistem teleoperation yang memiliki *force feedback* sehingga memungkinkan operator dalam merasakan kondisi yang terdapat di robot. Saat operator menggerakkan manipulator *master*, manipulator *slave* akan bergerak sesuai dengan gerakan manipulator *master*. Selain itu, manipulator *slave* akan mengirimkan *force feedback* ke sisi *master* sehingga operator dapat merasakan sensasi sentuhan yang diterima oleh manipulator *slave*. Operator dapat merasakan objek yang disentuh robot seolah-olah berinteraksi dengan objek secara langsung.

Pada pengendalian bilateral teleoperation di skripsi ini, *master* yang memiliki sistem 2 DOF dengan 4 kaki akan memberikan perintah kepada *slave* yang memiliki sistem 3 DOF dengan 4 kaki berupa posisi koordinat virtual yang akan dijalankan oleh manipulator *slave*. Selain itu, *slave* akan memberikan *force feedback* kepada *master* ketika membentur suatu objek sehingga *master* dapat merasakan kondisi yang dirasakan oleh *slave*. Dengan data berupa posisi koordinat virtual, dilakukan komputasi dalam menentukan sudut dari masing-masing servo pada tubuh robot yang berfungsi sebagai *joint* (sudut gerakan robot).

Berdasarkan data yang dikirim dari *master*, *slave* bergerak mengikuti gerakan *master*. Pergerakan *slave* memungkinkan adanya sudut *error* yang terjadi antara *slave* dengan *master*. Sudut *error* tersebut dikirimkan dari *slave* ke *master* untuk dilakukan komputasi selanjutnya. Ketika sudut *error* lebih besar daripada toleransi sudut *error* yang telah ditentukan, *master* akan bereaksi dengan memberikan torsi lebih besar sehingga operator dapat merasakan sensasi *haptic* yang diterima *slave*.

Manipulator *master* dan *slave* terkoneksi melalui suatu jaringan untuk saling mengirimkan sudut *error*. Jaringan yang digunakan merupakan UDP (User Datagram Protocol). Berikut adalah blok diagram ilustrasi bilateral teleoperation yang dirancang.



Gambar 2.1. Blok Diagram Bilateral Teleoperation

2.2 Forward Kinematics untuk Robot Manipulator

Robot manipulator merupakan robot yang menggunakan lengan dalam pergerakannya. Robot manipulator dirancang dengan menggunakan *links* (panjang lengan) yang dihubungkan dengan *joints* (aktuator pada robot) ke dalam suatu proses rantai kinematika. *Joints* dapat berupa *revolute*, yaitu aktuator yang bergerak secara rotasi, dan *prismatic*, yaitu aktuator yang bergerak secara linear.

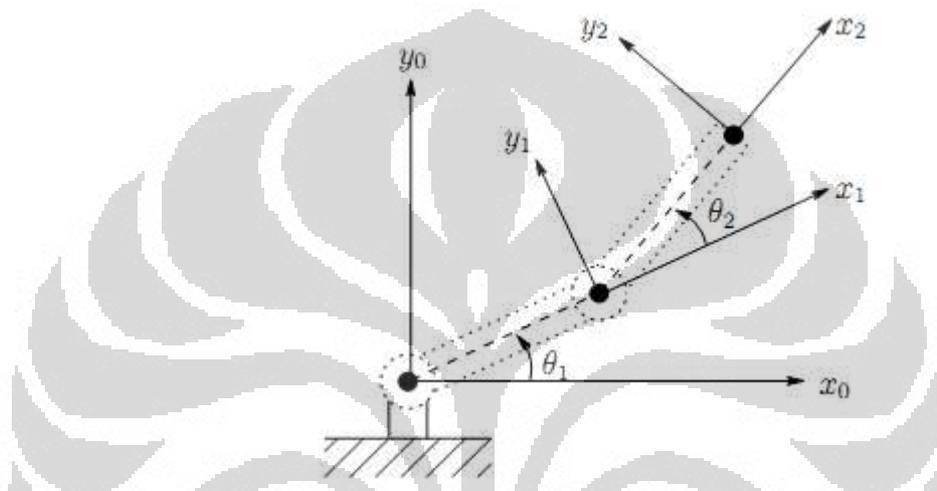
Banyaknya joint yang digunakan menentukan banyaknya Derajat kebebasan (*Degree of Freedom/DOF*) dari manipulator. Banyaknya DOF yang digunakan menentukan banyaknya pergerakan manipulator. Manipulator biasanya memiliki 6 DOF, yaitu 3 untuk Posisi dan 3 untuk Orientasi manipulator. Kesulitan dalam mengendalikan manipulator semakin bertambah dengan semakin banyaknya links pada manipulator.

Pengendalian robot manipulator memerlukan suatu proses pemetaan *end-effector*, yaitu ujung manipulator, terhadap *origin*, yaitu titik awal manipulator. Pemetaan tersebut dapat dilakukan dengan menggunakan *Forward Kinematics*. *Forward Kinematics* merupakan suatu fungsi untuk memetakan posisi terhadap sudut suatu manipulator. *Forward Kinematics* dapat digunakan untuk menentukan koordinat sistem terhadap semua objek terhadap manipulator. Koordinat ini biasanya dibentuk dalam koordinat cartesian.

Penggunaan *Forward Kinematics* sudah familiar di kalangan akademisi dan peneliti dalam perhitungan koordinat system dan benda-benda di sekitar system. Pada skripsi ini, *Forward Kinematics* yang digunakan meliputi Forward

Kinematics di bidang 2 Dimensi dan Forward Kinematics di bidang 3 Dimensi. Forward Kinematics di bidang 2 Dimensi mengakomodasi kebutuhan akan keberadaan posisi *end-effector* di sisi master (operator) dengan 2 DOF. Sedangkan, Forward Kinematics di bidang 3 Dimensi mengakomodasi kebutuhan akan keberadaan posisi *end-effector* di sisi slave (robot) dengan 3 DOF system.

2.2.1. Forward Kinematics untuk Robot Manipulator pada bidang 2 Dimensi



Gambar 2.2. Forward Kinematics pada Koordinat Cartesian 2 Dimensi.

Sumber: Spong, Mark, et all. *Robot Dynamics and Control*.

Dari gambar 2.4, terlihat bahwa lengan robot dapat diketahui posisi *end-effector* terhadap origin pada bidang 2 Dimensi. Gambar tersebut memenuhi persamaan *Forward Kinematics* sebagai berikut.

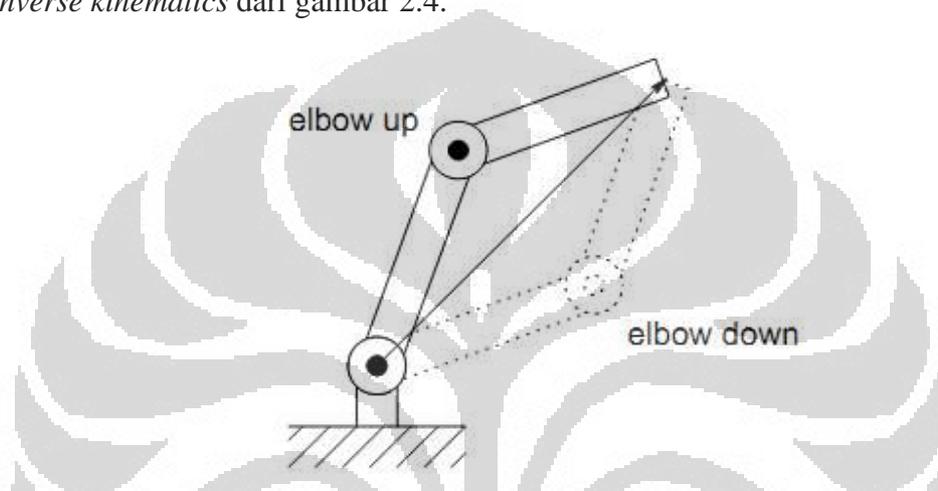
$$\begin{aligned} x &= l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ y &= l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \end{aligned} \quad [2.1]$$

Persamaan tersebut memiliki l_1 sebagai panjang lengan 1 dan l_2 sebagai panjang lengan 2. Dengan titik (x_2, y_2) sebagai titik akhir *end-effector* dan (x_0, y_0) sebagai titik origin, Persamaan tersebut dapat dibentuk menjadi suatu matriks. Berikut ini adalah matriks Forward Kinematics untuk sistem pada gambar 2.4.

$$\begin{bmatrix} x_2 - x_0 & y_2 - y_0 \\ x_2 \cdot y_0 & y_2 \cdot x_0 \end{bmatrix} = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix} \quad [2.2]$$

Forward kinematics hanya dapat menentukan posisi *end-effector*. Permasalahan yang dihadapi adalah bagaimana menggerakkan *joints* manipulator sehingga menghasilkan posisi yang diinginkan. Permasalahan ini disebut sebagai

Forward Kinematics Problem. Permasalahan ini dapat diatasi dengan menggunakan *inverse kinematics*. Inverse kinematics merupakan suatu fungsi yang memetakan sudut manipulator terhadap posisi. Karena forward kinematics merupakan fungsi non-linear, Inverse kinematics tidak mudah untuk diterapkan. Hal ini karena ada banyak solusi dalam menentukan sudut dari posisi yang diinginkan. Dari gambar 2.4, terdapat dua solusi untuk *inverse kinematics* yang dapat disebut juga sebagai *elbow-up* dan *elbow-down*. Berikut ini adalah ilustrasi *inverse kinematics* dari gambar 2.4.



Gambar 2.3. Dua Solusi pada Inverse Kinematics

Sumber: Spong, Mark, et all. *Robot Dynamics and Control*.

Untuk mendapatkan persamaan *inverse kinematics*, diperlukan acuan terhadap gerakan manipulator yang diinginkan. Untuk mendapatkan posisi seperti yang terlihat pada Gambar 2.4, dilakukan *inverse* cosinus untuk mendapatkan θ_2 . Berikut adalah persamaan *inverse kinematics* dari Gambar 2.4.

$$\cos(\theta_2) = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2} = D \quad [2.3]$$

$$\theta_2 = \cos^{-1}(D)$$

Dari persamaan tersebut, dapat diperoleh θ_2 . Namun, cara yang lebih baik untuk menentukan θ_2 adalah dengan memperhatikan $\cos(\theta_2)$ dan $\sin(\theta_2)$ sehingga menghasilkan $\tan(\theta_2)$. Dari fungsi tangen, diperoleh θ_2 . Berikut adalah persamaan tersebut.

$$\begin{aligned}\sin(\theta_2) &= \pm\sqrt{1-D^2} \\ \theta_2 &= \tan^{-1} \frac{\pm\sqrt{1-D^2}}{D}\end{aligned}\quad [2.4]$$

Setelah mendapatkan θ_2 , dapat ditentukan θ_1 sebagai berikut.

$$\theta_1 = \tan^{-1}(y/x) - \tan^{-1}\left(\frac{l_2 \sin\theta_2}{l_1 + l_2 \cos\theta_2}\right)\quad [2.5]$$

Karena adanya *multiple solution* pada penggunaan *inverse kinematics*, *inverse kinematics* sulit diterapkan pada manipulator yang bergerak secara kontinyu. Pergerakan manipulator yang kontinyu menyebabkan adanya kecepatan pergerakan yang disebut sebagai *velocity kinematics*. *Velocity kinematics* memungkinkan adanya perubahan posisi sudut dari manipulator sehingga posisi dari *end-effector* dapat ditentukan. Persamaan ini dapat ditentukan dengan pendekatan persamaan *differential*. Berdasarkan Gambar 2.4, dapat diperoleh *velocity kinematics* dengan pendekatan persamaan *differential partial* sebagai berikut.

$$\begin{aligned}\dot{x} &= -(l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2))\dot{\theta}_1 - l_2 \sin(\theta_1 + \theta_2)\dot{\theta}_2 \\ \dot{y} &= (l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2))\dot{\theta}_1 + l_2 \cos(\theta_1 + \theta_2)\dot{\theta}_2\end{aligned}\quad [2.6]$$

Persamaan tersebut dapat dijadikan dalam bentuk matrik dengan $\dot{x} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$

dan $\dot{\theta} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$ sebagai berikut.

$$\begin{aligned}\dot{x} &= \begin{bmatrix} -(l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \dot{\theta} \\ \dot{x} &= J\dot{\theta}\end{aligned}\quad [2.7]$$

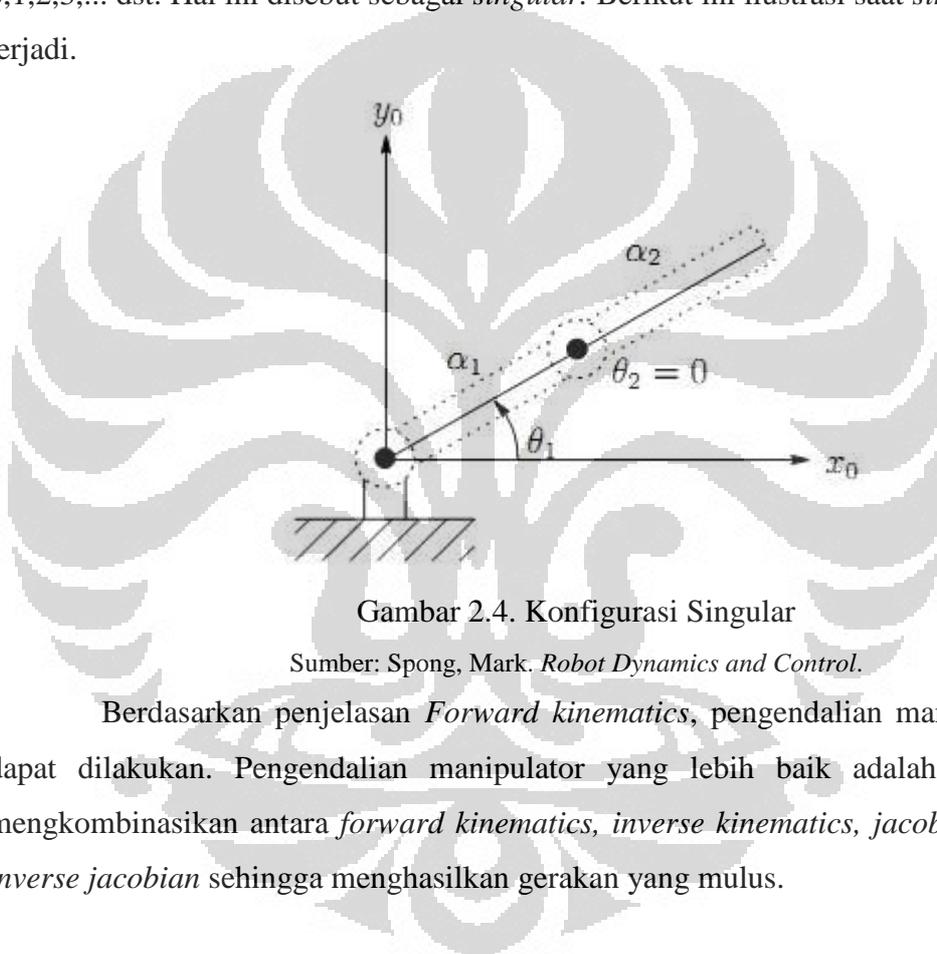
Pada persamaan matrik di atas, notasi J merupakan persamaan matrik Jacobian. Matrik Jacobian merupakan suatu matrik yang memetakan perubahan perubahan kecil posisi terhadap perubahan kecil sudut. Matrik Jacobian dapat di-*inverse* untuk dapat menentukan perubahan kecil sudut terhadap perubahan kecil posisi yang disebut *inverse Jacobian*. Persamaan *inverse Jacobian* diberikan sebagai berikut

$$\dot{\theta} = J^{-1} \dot{x}$$

$$\dot{\theta} = \frac{1}{l_1 l_2 \sin(\theta_2)} \begin{bmatrix} l_2 \cos(\theta_1 + \theta_2) & l_2 \sin(\theta_1 + \theta_2) \\ -l_1 \cos(\theta_1) - l_2 \cos(\theta_1 + \theta_2) & -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) \end{bmatrix} \dot{x}$$

[2.8]

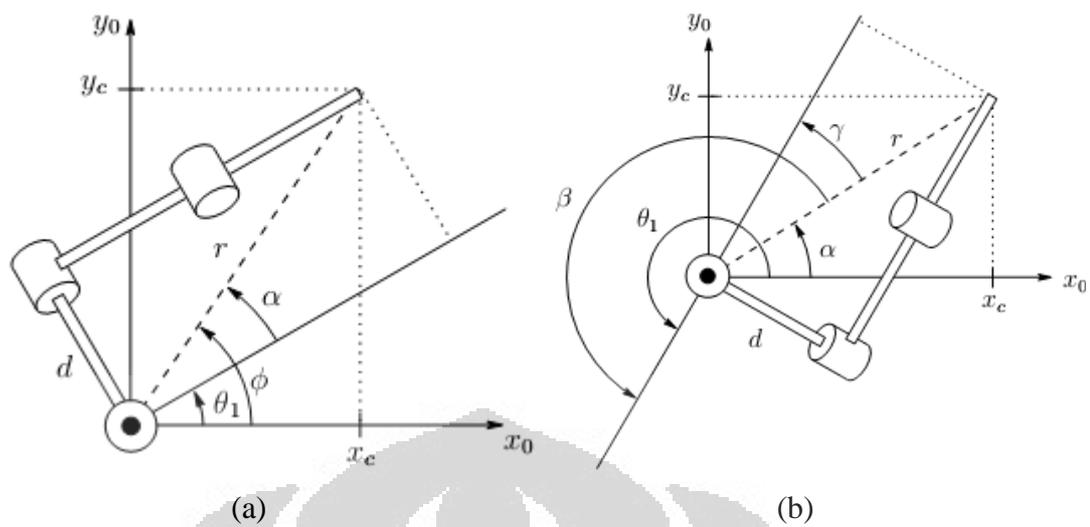
Dengan adanya determinan tersebut, matrik inverse jacobian tidak bisa diimplementasikan ketika determinan matrik tersebut sama dengan nol. Determinan matrik tersebut sama dengan nol ketika θ_2 bernilai $k\pi$, dengan $k = 0, 1, 2, 3, \dots$ dst. Hal ini disebut sebagai *singular*. Berikut ini ilustrasi saat *singularity* terjadi.



Gambar 2.4. Konfigurasi Singular

Sumber: Spong, Mark. *Robot Dynamics and Control*.

Berdasarkan penjelasan *Forward kinematics*, pengendalian manipulator dapat dilakukan. Pengendalian manipulator yang lebih baik adalah dengan mengkombinasikan antara *forward kinematics*, *inverse kinematics*, *jacobian*, dan *inverse jacobian* sehingga menghasilkan gerakan yang mulus.



Gambar 2.6. Dua Solusi pada Inverse Kinematics

Sumber: Spong, Mark, et all. *Robot Dynamics and Control*.

Untuk mendapatkan persamaan *inverse kinematics*, diperlukan acuan terhadap gerakan manipulator yang diinginkan. Untuk mendapatkan posisi seperti yang terlihat pada Gambar 2.4, dilakukan *inverse cosine* untuk mendapatkan θ_2 . Berikut adalah persamaan *inverse kinematics* dari Gambar 2.4.

$$\cos(\theta_2) = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2} = D \quad [2.12]$$

$$\theta_2 = \cos^{-1}(D)$$

Dari persamaan tersebut, dapat diperoleh θ_2 . Namun, cara yang lebih baik untuk menentukan θ_2 adalah dengan memperhatikan $\cos(\theta_2)$ dan $\sin(\theta_2)$ sehingga menghasilkan $\tan(\theta_2)$. Dari fungsi tangen, diperoleh θ_2 . Berikut adalah persamaan tersebut.

$$\sin(\theta_2) = \pm\sqrt{1-D^2}$$

$$\theta_2 = \tan^{-1} \frac{\pm\sqrt{1-D^2}}{D} \quad [2.13]$$

Setelah mendapatkan θ_2 , dapat ditentukan θ_1 sebagai berikut.

$$\theta_1 = \tan^{-1}(y/x) - \tan^{-1}\left(\frac{l_2 \sin \theta_2}{l_1 + l_2 \cos \theta_2}\right) \quad [2.14]$$

Karena adanya *multiple solution* pada penggunaan *inverse kinematics*, *inverse kinematics* sulit diterapkan pada manipulator yang bergerak secara kontinyu. Pergerakan manipulator yang kontinyu menyebabkan adanya kecepatan pergerakan yang disebut sebagai *velocity kinematics*. *Velocity kinematics* memungkinkan adanya perubahan posisi sudut dari manipulator sehingga posisi dari *end-effector* dapat ditentukan. Persamaan ini dapat ditentukan dengan pendekatan persamaan *differential*.

2.3 Development board

2.3.1. Hardware

Development Board yang digunakan pada skripsi ini adalah dua buah board dari DMP Electronics, yaitu RB-100 dan RB-110. Kedua board dapat beroperasi seperti komputer biasa. *Development Board* ini kompatibel dengan berbagai sistem operasi seperti Windows dan Linux. *Development Board* ini berfungsi sebagai *controller* pada manipulator ini.



Gambar 2.7. RB-100

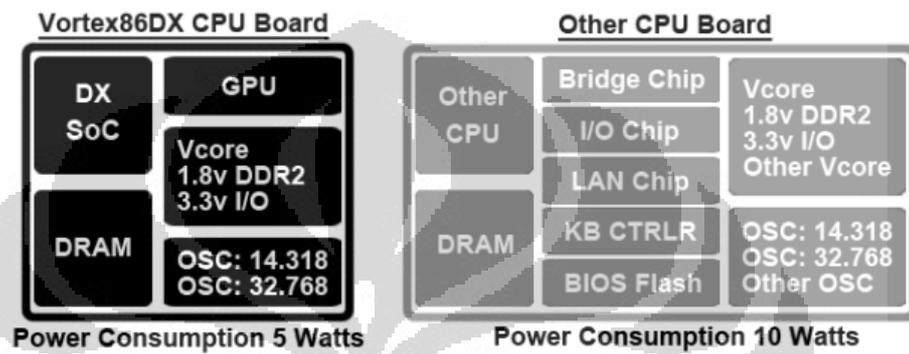
Sumber: Datasheet RB-100

Kedua board ini menggunakan Vortex86DX, sebuah Single on Chip (SoC) yang berbasis prosesor 32 bit dari keluarga intel x86. SoC yang digunakan memiliki clock frekuensi sebesar 1GHz dan memiliki DRAM (Dynamic Random Access Memory) sebesar 256MB. Penggunaan SoC dapat mengurangi kompleksitas rangkaian dan konsumsi energi. Hal ini karena *Development Board* ini dapat mengurangi penggunaan IC lain secara signifikan. Vortex86DX mengintegrasikan 4 buah chip yang umumnya terpisah yaitu CPU, North Bridge, SouthBridge, dan BIOS ke dalam sebuah chip BGA.



Gambar 2.8. Vortex86DX Single On Chip (SoC)

Sumber: Datasheet RB 100



Gambar 2.9. Perbandingan RB-100 dengan CPU Board Lain

Sumber: Datasheet RB 100

RB 100 dan RB 110 memiliki banyak kelebihan dibanding CPU Board lain dalam hal fitur yang dapat digunakan di bidang robotika. Fitur-fitur tersebut dapat dilihat pada tabel spesifikasi RB 100 dan RB 110. Berikut tabel spesifikasi RB 100 (yang dicetak tebal adalah fitur yang digunakan).

Table 2.1. Spesifikasi RB-100 dan RB-110

Specification	RB-100	RB-110
CPU	DM&P VORTEX86DX - 1000 MHz	DM&P VORTEX86DX - 1000 MHz
BIOS	AMI BIOS	AMI BIOS
MEMORY	256 MB DDR2 on-board	256 MB DDR2 on-board
ADC	Analog to Digital Converter AD - 7918 10-bit	Analog to Digital Converter AD - 7918 10-bit
Hi-Speed UART	-	FTDI FT2232HL Hi-Speed UART
I/O INTERFACE	Micro SD Slot	Micro SD Slot
	USB PORT (2.0)	USB PORT (2.0)
CONNECTOR	PWM/GPIO x 24	PWM/GPIO x 16
	RS-232 x 1	RS-232 x 1
	USB x 1	USB x 1
	RS-485 x 1	RS-485 x 1
	TTL Serial x 2	TTL Serial x 2
	SPI dan I2C x1	SPI dan I2C x1
	A/D x 1	A/D x 1
	LAN x 1	LAN x 1
	MIC-in x 1	MIC-in x 1
	Line-out x 1	Line-out x 1
	JTAG x 1	JTAG x 1
	PCI Card	PCI Card
	-	Hi-Speed Serial x 2
RESOLUTION	SPI: 10 Mbps to 150 Mbps Half-Duplex;	SPI: 10 Mbps to 150 Mbps Half-Duplex;
	PWM : 20ns	PWM : 20ns
	I2C : 1 Kbps - 3.3 Mbps	I2C : 1 Kbps - 3.3 Mbps
	Serial : 115200 bps	Serial : 115200 bps/750 Kbps (COM 1, 2, 3, & 4)
		Hi-Speed Serial : up to 12Mbps (COM5 & 6)
POWER CONSUMPTION	+5VDC @400mA	+5VDC @400mA
POWER INPUT	DC-in 6V - 24 V	DC-in 6V - 24 V
DIMENSION	96 mm x 56 mm	96 mm x 56 mm
WEIGHT	40 g	40 g

Spesifikasi RB-110 hampir sama dengan RB-100. Perbedaan yang terdapat pada RB-110 adalah banyaknya jumlah *port* PWM sebanyak 16 *port*, dan penambahan 2 buah *port* komunikasi, yaitu FTDI. *Port* FTDI dapat digunakan untuk komunikasi serial TTL dengan *baudrate* mencapai 12Mbps. Pada RB-110, *power supply* dipisah antara penggunaan power untuk system sebesar 6-24 V, dan

power untuk servo tak terbatas, namun direkomendasi dengan *power supply* kurang dari 24 VDC. Hal ini bertujuan untuk menghindari adanya *power interference* pada sistem dan servo.

2.3.2. Software

Dalam hal software, Kedua board ini menyediakan library C++ untuk pengembangan software dalam penggunaan board ini. Dengan adanya library tersebut, pengembang software tidak perlu bersentuhan dengan register-register low level pada board. Selain itu, Roboard dapat diinstal sistem operasi, seperti Linux Ubuntu, Debian, Windows XP, Windows CE, dan DOS. Penanaman Sistem Operasi ini dimaksudkan pada penggunaan Roboard yang lebih kompleks seperti sistem Pengamanan, Robotika, dan lain-lain.

Pemilihan system operasi yang tepat untuk aplikasi bilateral teleoperation merupakan hal yang sangat penting. Hal ini karena sistem operasi memiliki spesifikasi minimum requirement hardware yang sama dengan spesifikasi yang dimiliki system. Selain itu, dalam pengembangan system *embedded*, diperlukan system operasi yang tidak terlalu berat sehingga proses dari system dapat diminimalisasi. Selain itu, system operasi juga harus memiliki kernel yang mampu menjadi antarmuka antara software aplikasi dengan hardware yang digunakan. Selain itu, system operasi juga harus dapat dimanipulasi secara bebas. Hal ini bertujuan untuk memudahkan pengembangan selanjutnya sehingga pengembangan dapat dilakukan tanpa melanggar lisensi tertentu.

Sistem operasi yang digunakan pada skripsi ini adalah linux. Hal ini karena Linux merupakan system operasi berbasis *open source* dengan tujuan untuk memperoleh akses untuk pengembangan yang lebih luas bila dibandingkan dengan system operasi berbayar. Distro linux yang digunakan disebut Lubuntu LTS 10.04. Lubuntu LTS 10.04 adalah salah satu jenis linux yang cocok digunakan untuk *embedded system* karena sistemnya tidak terlalu kompleks dan tidak membutuhkan kapasitas RAM dan hard disk yang besar. Selain itu, Lubuntu 10.04 LTS juga memiliki GUI sehingga memudahkan pengguna dalam menggunakannya. Walaupun ukuran file Lubuntu cukup kecil, perintah-perintah yang disediakan Lubuntu 10.04 LTS cukup lengkap. Sistem operasi ini strukturnya mirip dengan Linux Distro Ubuntu yang sering digunakan oleh orang sebagai

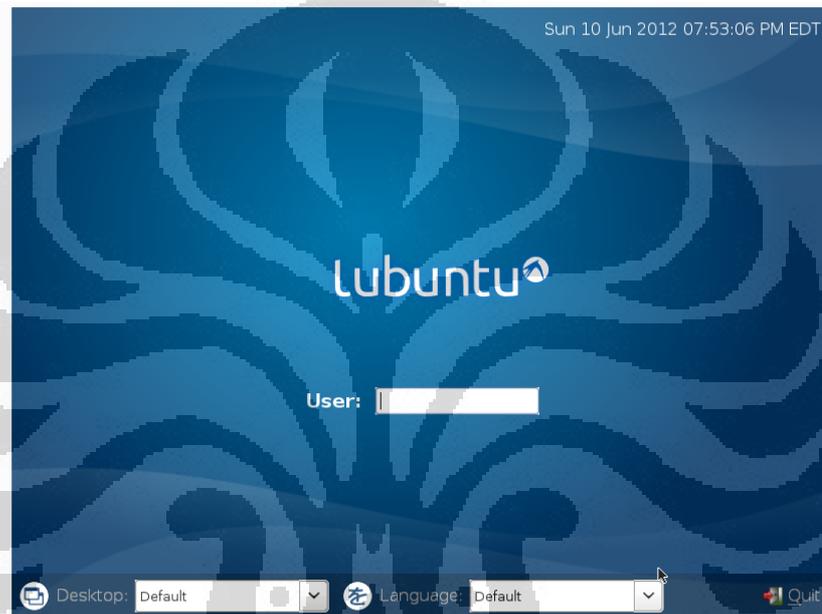
sistem operasi di komputer sehingga sistem operasi ini cukup familiar untuk digunakan.

Instalasi Lubuntu 10.04 LTS dilakukan dengan bantuan software virtual box. Untuk dapat menginstal Lubuntu 10.04 LTS diperlukan Micro SD Card dengan minimum kapasitas memori sebesar 4 GB. Kemudian, download dan simpan file iso installer Lubuntu 10.04 LTS di halaman http://www.roboard.com/download_ml.htm. Berikut ini adalah prosedur untuk instalasi Lubuntu 10.04 LTS pada Micro SD Card yang akan digunakan pada roboard.

1. Buat sebuah virtual mesin pada virtual box
2. Set memori ke 256MB agar sama seperti memori pada RB-100/RB-110
3. Hilangkan tanda centang pada pilihan *Boot Hard Disk*, klik *continue* jika muncul *dialog box*. Kemudian, klik *create*. Tutup Virtual Box
4. Lepas semua sambungan usb di komputer. Sambungkan Micro SD Card ke computer dengan menggunakan *card reader*.
5. Buka *command line*, gunakan *run as administrator* pada windows 7
6. Masuk ke folder virtual box dengan mengetikkan pada *command line* C:\Program Files\oracle\virtualbox
7. Pasang USB device sebagai virtual box raw disk dengan menjalankan perintah berikut.
`Vboxmanage internalcommands createrawvmdk -filename <folder untuk output file> -rawdisk \\.\physicaldrive1 -register`
 Instruksi yang digunakan pada skripsi ini adalah sebagai berikut.
`Vboxmanage internalcommands createrawvmdk -filename "D:\Akademik\roboard.vmdk" -rawdisk \\.\physicaldrive1 -register`
8. Buka virtual box dan pilih virtual mesin yang dibuat pada langkah 1
9. Pilih storage, kemudian mount file ISO Lubuntu 10.04 LTS sebagai CDROM IDE Primary Master, dan roboard.vmdk yang dibuat pada langkah 6 sebagai IDE Primary Slave
10. Klik start dan virtualbox akan menjalankan instalasi Lubuntu 10.04 LTS
11. Ketika terlihat clonezilla boot screen, pilih opsi 1 dan tekan ENTER

12. Tekan ENTER, kemudian “y” sebanyak 2 kali. Program akan menjalankan proses *restore image* ke Micro SD Card. Tunggu hingga proses selesai.
13. Setelah proses *restore image* selesai, tekan ENTER, kemudian pilih “0” untuk poweroff virtual machine dan tekan ENTER sebanyak 2 kali.
14. Lubuntu 10.04 LTS telah ter-install di Micro SD Card.

Berikut adalah tampilan Lubuntu 10.04 LTS yang telah terinstall Micro SD yang dijalankan melalui roboard dengan menggunakan VGA Card yang tersambung ke monitor.



Gambar 2.10. Tampilan Lubuntu 10.04 LTS yang Dijalankan Oleh Roboard Pada Monitor

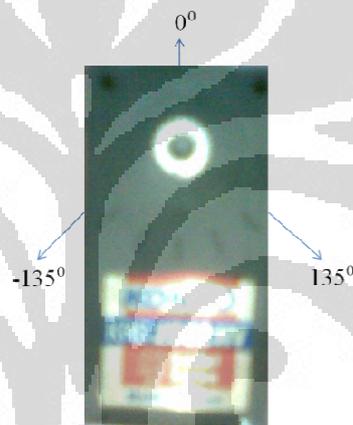
Login dapat dilakukan dengan menggunakan USER ID sebagai roboard dan password “roboard”.

2.4 Motor Serial Servo Kondo KRS 2552 HV

Pada skripsi ini, digunakan motor serial servo agar mempermudah dalam pengendalian servo. Motor serial servo sendiri merupakan motor servo yang ditambahkan unit mikrokontroler agar mampuk melakukan fitur-fitur tambahan dan dapat dikendalikan melalui komunikasi serial. Hal ini berbeda dengan motor servo standar yang pengendaliannya dengan menggunakan PWM. Motor serial servo cukup dengan menggunakan komunikasi serial dalam pengendaliannya dengan pengiriman suatu instruksi tertentu. Pengendalian motor serial servo tidak

hanya menggerakkan servo saja, melainkan dapat dilakukan pengendalian fitur-fitur yang lain seperti umpan balik posisi, temperatur, limitasi, beban, tegangan, dan lain-lain. Dengan menggunakan motor serial servo, beban kerja prosesor dapat dikurangi karena tugas pembacaan posisi dan lain-lain dibebankan kepada mikrokontroler yang terdapat di dalam motor serial servo.

Motor serial servo yang digunakan pada skripsi ini merupakan servo Kondo KRS 2552 HV. Kondo KRS 2552 HV merupakan salah satu serial servo produksi Kondo, perusahaan yang mengkhususkan diri pada pengembangan robot humanoid. Serial servo ini beroperasi pada tegangan 9 – 12V dengan sudut putaran efektif 270°, yaitu -135° ~ 135° dan sudut putaran maksimum 360°. Berikut adalah ilustrasi dari sudut operasi servo Kondo KRS 2552 HV.



Gambar 2.11. Sudut Operasi Servo Kondo KRS 2552 HV

Servo Kondo KRS 2552 HV menyediakan koneksi multi-drop dengan total servo terkoneksi sampai 31 servo. Servo ini menggunakan metal gear pada sistem gearnya sehingga lebih tahan lama dan tidak cepat rusak.



Gambar 2.12. Servo Kondo KRS 2552 HV

Sumber: <http://robosavvy.com/store/images/KRS-2552HV.jpg>

Serial servo ini menggunakan protokol khusus dari Kondo yang disebut dengan ICS 3.0. Protokol ini mendukung 3 jenis pilihan *baudrate* yaitu 115200, 625000, dan 1,25Mbps. Namun, Servo KRS 2552 HV hanya bisa menggunakan *baudrate* 115200. Pengaturan komunikasi yang digunakan oleh serial servo ini adalah 1 start bit, 8 bit data, 1 stop bit, no flow control, dan even parity.

Protokol yang digunakan oleh serial servo ini relatif lebih mudah dan lebih singkat. Protokol serial servo yang digunakan ditunjukkan pada tabel berikut.

Table 2.2. Paket Instruksi Serial Servo Kondo KRS 2552 HV

TX	1 (CMD)	2 (SC)	3 - N(Data)
	Command + ID	SubCommand	Data

CMD merupakan data serial servo dengan ketentuan Command + ID servo yang diinginkan. SC merupakan SubCommand, yaitu bagian protokol yang menentukan fungsi yang dijalankan servo. Dan Data merupakan parameter yang dibaca atau ditulis pada servo. Berikut ini adalah tabel Command pada protokol ICS 3.0.

Table 2.3. Command pada Protokol ICS 3.0

100xxxxx	Pengaturan posisi
101xxxxx	Pembacaan parameter (jenis parameter tergantung sub command)
110xxxxx	Penulisan parameter (jenis parameter tergantung sub command)
111xxxxx	Pengaturan ID

xxxxxx merupakan bentuk biner dari ID servo yang akan diproses. Misalnya untuk servo dengan ID = 2, biner yang dikirim adalah 00010. ID maksimum yang dapat digunakan adalah 0b11111 atau 0x1F atau 31 (decimal).

SubCommand yang dapat digunakan pada Kondo KRS 2552 HV terdapat 4 jenis SubCommand. Berikut adalah tabel jenis SubCommand pada Kondo KRS 2552 HV.

Table 2.4. Jenis SubCommand pada Kondo KRS 2552 HV

Parameter	Nilai	Keterangan
EEPROM	0x00	Akses ke EEPROM
STRC	0x01	Manipulasi stretching pulsa
SPD	0x02	Kontrol kecepatan motor
CUR	0x03	Kontrol arus pada motor

Setelah menerima paket instruksi, serial servo akan memberikan tanggapan yang disebut paket status. Bentuk paket status dari protokol ICS adalah sebagai berikut.

Table 2.5. Paket Status pada Protokol ICS 3.0

RX	1	2	3	4	5	6
	Loopback Command Sent			R_CMD	SC	Data

Berikut adalah penjelasan dari tabel 2.5. Loopback Command Sent berisi nilai yang sama dengan nilai yg dikirimkan pada saat TX. Loopback command sent memastikan bahwa data yang dikirim telah sampai ke servo dengan benar. R_CMD dan SC adalah command dan sub command yang sama dengan command dan sub command pada saat pengiriman. Data merupakan nilai terakhir yang dimiliki oleh servo sebelum nilai tersebut diubah oleh data yang baru.

Pada skripsi ini, dilakukan pengaturan posisi Servo Kondo KRS 2552 HV untuk menggerakannya, pembacaan data status posisi Servo untuk mendapatkan data posisi actual, dan pembacaan data torque untuk melihat sensasi force yang dirasakan servo. Pengaturan posisi Servo Kondo KRS 2552 HV sedikit berbeda dengan protocol data sebelumnya. Terdapat perbedaan pula pada protocol untuk membaca data status posisi servo. Sedangkan untuk pembacaan data torque, protocol yang digunakan mengikuti protocol data yang telah dijelaskan sebelumnya dengan register tertentu. Berikut adalah penjelasan dalam pengaturan posisi, pembacaan posisi, dan pembacaan data torque pada servo.

2.4.1. Pengaturan posisi pada Servo Kondo KRS 2552 HV

Pada kondo KRS 2552 HV, protokol untuk menggerakkan servo dapat dilakukan lebih mudah. Pada pengaturan posisi, protokol tidak membutuhkan SubCommand dan Data. Protokol yang digunakan memenuhi tabel berikut.

Table 2.6. Pengaturan Posisi Servo Kondo KRS 2552 HV

TX	1 (CMD)	2	3
	100xxxxx + ID	POS_H	POS_L

Sudut operasi yang umumnya digunakan adalah berkisar antara 3500 – 11500 desimal dengan sudut operasi $-135^{\circ} \sim 135^{\circ}$. Sudut operasi tersebut dipisah ke dalam POS_H dan POS_L. POS_L merupakan bagian Low dari sudut operasi, yaitu 7 bit dari kanan. Dan POS_H merupakan bagian High dari sudut operasi, yaitu sisa dari bit setelah 7 bit dari kanan. Berikut adalah tabel dari POS_H dan POS_L.

Table 2.7. Format POS_H dan POS_L

MSB	POS_H / POS_L						LSB
7	6	5	4	3	2	1	0
0	x	x	x	x	x	x	x

Berdasarkan tabel 2.7, untuk menggerakkan sudut 90° (sudut 0° berdasarkan Gambar 2.13), dikirim data 7500 desimal. Nilai 7500 desimal jika dikonversi menjadi bilangan biner akan menjadi 1110101001100. Bilangan ini kemudian didistribusikan ke dalam POS_H dan POS_L dengan syarat MSB (most significant bit) masing-masing variabel harus bernilai 0. Dengan ketentuan tersebut maka nilai POS_L adalah 7 bit dari kanan ditambah bit 0 pada MSB menjadi 01001100 biner atau 4C hexa. Sedangkan nilai POS_H adalah sisa bit ditambah bit 0 pada MSB menjadi 00111010 biner atau 3A hexadesimal.

Untuk menggerakkan sudut 90° pada servo ID 1, user harus mengirimkan data 7500 desimal ke servo dengan mengikuti protocol yang terdapat pada tabel 2.7. Berikut adalah table dari pengiriman paket instruksi servo Kondo untuk menggerakkan sudut 90° pada servo ID 1.

Table 2.8. Paket Instruksi Sudut 90° pada Servo ID 1

TX	1 (CMD)	2	3
	81	3A	4C

Command : 0b10000001 atau 0x81

POS_H : 0b00111010 atau 0x3A

POS_L : 0b01001100 atau 0x4C

Setelah paket instruksi dikirim ke servo, servo akan mengirimkan paket status. Berikut adalah tabel paket status yang dikirimkan servo.

Table 2.9. Paket Status Pengiriman Instruksi Sudut 90° pada ID 1

RX	1	2	3	4	5	6
	81	3A	4C	81	0D	AC

3 byte pertama yang pada paket status merupakan data yang paket instruksi yang dikirimkan ke servo. 1 byte selanjutnya merupakan Command+ID untuk pengaturan posisi servo ID 1. Dan 2 byte terakhir adalah posisi terakhir servo sebelum bergerak. Posisi terakhir servo berdasarkan table 2.9 dapat diketahui dengan menggeser 8 bit ke kiri dari POS_H kemudian dilakukan operasi bit OR dengan data POS_L. berikut operasi OR untuk mendapatkan Data posisi terakhir.

POS_H : 00001101 atau 0x0D

POS_L : 10101100 atau 0xAC

Data : 0000110110101100 atau 0x0DAC atau 3500

2.4.2. Pembacaan Status Posisi pada Servo Kondo KRS 2552 HV

Untuk membaca status posisi Servo Kondo KRS 2552 HV, dapat dilakukan pemberian instruksi *free run*. *Free run* merupakan instruksi yang memungkinkan servo tidak memberikan torsi dan mengunci posisi terakhir. Untuk menggunakan *free run mode*, *user* cukup memberikan instruksi 0000. Berikut adalah table untuk menggunakan *free run mode*.

Table 2.10. Instruksi Freerun Mode pada Servo ID 1

TX	1	2	3
	0x81	0	0

Setelah pemberian data free run mode, servo akan memberikan status packets berisi data posisi saat itu. Status Packet yang diberikan servo ketika servo berada di posisi 90° adalah sebagai berikut.

Table 2.11. Packet Status Servo Kondo KRS 2552 HV

RX	1	2	3	4	5	6
	0x81	0	0	0x81	0x1D	0x4C

2.4.3. Pembacaan data Torque pada servo Kondo KRS 2552 HV

Untuk membaca torque pada servo kondo KRS 2552 HV, dapat dilakukan dengan menggunakan protocol pada table 2.2 dengan Command untuk pembacaan parameter servo, yaitu 0b101xxxxx dan SubCommand untuk data EEPROM Kontrol speed, yaitu 0x02. Paket instruksi yang dikirim ke servo adalah sebagai berikut.

Table 2.12. Pembacaan Speed pada Servo ID 1

TX	1	2
	0xA1	0x02

Setelah pengiriman paket instruksi, servo akan memberikan paket status sebagai berikut.

Table 2.13. Paket Status Pembacaan Speed pada Servo

RX	1	2	3	4	5
	0xA1	0x02	0xA1	0x02	Data

Data yang diperoleh berkisar antara 0 – 127. Data ini dapat digunakan untuk mendeteksi torsi yang dialami oleh servo. Ketika servo tertahan pada suatu nilai speed tertentu, gaya yang bekerja pada servo dapat diartikan sama dengan gaya yang diberikan servo.

Untuk mendapatkan sensasi *haptic*, digunakan protocol penulisan parameter pada servo sesuai dengan table 2.2 dengan penulisan parameter pada servo dengan protocol Command 0b110xxxxx dan Subcommand untuk Kontrol speed, yaitu 0x02. Kemudian, data yang dikirim berisi speed untuk servo pada kisaran 0 – 127. Berikut adalah contoh pengiriman data speed pada servo.

Table 2.14. Penulisan Speed pada Servo ID 1 dengan Data Speed Sebesar 2

TX	1	2	3
	0xA1	0x02	0x02

2.5 Komunikasi Data dengan Socket Programming

Komunikasi data antara master dan slave pada robot ini menggunakan Socket programming dalam pengaplikasiannya. Socket sendiri dapat diartikan sebagai suatu cara yang disediakan pada jaringan divais untuk melakukan koneksi antarproses. Socket merupakan sebuah cara untuk berkomunikasi dengan

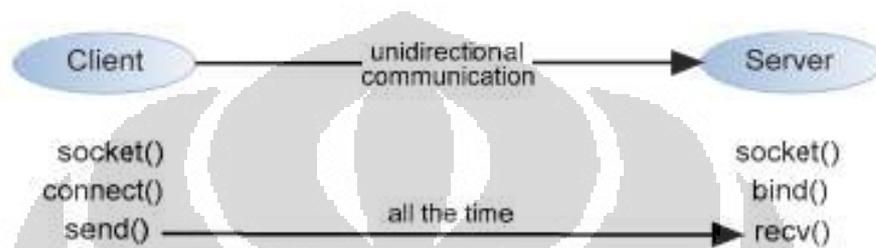
program atau node lain menggunakan file deskriptor. Komunikasi dengan program lain semudah membaca dan menulis file deskriptor. Socket programming pada umumnya sering digunakan oleh teknisi komputer dalam penggunaan jaringan komputer. Karena penggunaan socket ini berhubungan dengan jaringan, socket programming dapat disebut juga sebagai network programming. Dengan menggunakan socket, file dapat dieksekusi dari divais lain karena telah terhubung suatu koneksi.

Pada dasarnya, terdapat dua macam komunikasi data yang sering digunakan melalui program jaringan, yaitu TCP dan UDP (User Datagram Protocol). TCP (Transmission Communication Protocol) atau bisa juga disebut sebagai *socket stream* merupakan suatu protokol yang berada di lapisan transport, yaitu lapisan pertukaran data, yang berorientasi sambungan (*connection-oriented*) dan dapat diandalkan (*reliable*). TCP merupakan koneksi yang reliable yang memastikan data datang tanpa cacat dan ditransmisikan kembali jika terjadi *lost*, dan mengeliminasi duplikasi. Sedangkan User Datagram Protocol (UDP) merupakan protocol data yang bersifat *connectionless*, memancar, dan bersifat *unreliable*. Hal ini karena UDP tidak memastikan apakah paket data telah sampai ke tujuan atau tidak. UDP tidak menjamin bahwa paket akan sampai sesuai dengan instruksinya. UDP biasanya digunakan untuk aplikasi seperti streaming media (audio dan video, dll) di mana waktu TCP memerlukan untuk pengiriman ulang dan pemesanan ulang mungkin tidak tersedia, atau untuk *query* sederhana atau respon aplikasi di mana overhead dari menyiapkan koneksi yang handal adalah tidak proporsional besar.

Komunikasi yang digunakan pada system ini adalah UDP dengan melalui koneksi wifi. Komunikasi ini digunakan karena kecepatan data yang dikirimkan menggunakan media wireless dengan frekuensi 2.4 GHz. Kecepatan transfer data dengan menggunakan komunikasi ini dapat mencapai 54 MBPS. Dengan UDP, data yang master tidak perlu menunggu data yang dikirimkan slave sehingga waktu menunggu tidak lama sehingga proses bilateral teleoperation cukup baik dengan waktu transmisi data yang lebih cepat daripada TCP.

Sebagaimana telah dijelaskan sebelumnya, komunikasi UDP tidak bersifat *connectionless*. Bagian server membuat *socket* pada *port* komunikasi yang

mengakibatkan server selalu siap untuk menerima paket yang datang. Bagian client cukup membuka socket dan membuat koneksi dengan fungsi *connect()* untuk mengirim paket secara langsung dengan fungsi *send()*. Karena hubungan antara client dan server bersifat searah, client tidak akan tahu apakah data yang dikirim telah berhasil diterima oleh server atau tidak. Hal ini dapat diatasi dengan membuat dua jalur komunikasi UDP dan membuat suatu *acknowledgement* di *application layer* pada bagian *client* dan *server*.



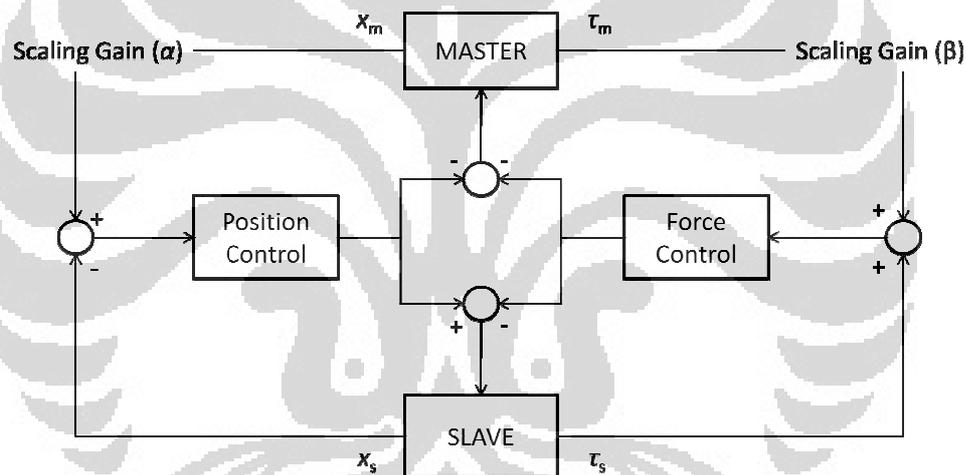
Gambar 2.13. Bentuk Komunikasi UDP

BAB 3 PERANCANGAN SISTEM

Perancangan Sistem Bilateral Teleoperation ini dibagi menjadi 4 bagian, yaitu Konsep Bilateral Teleoperation, Perancangan Perangkat Keras, Perangkat Lunak, dan Integrasi Sistem. Berikut ini penjelasan dari masing-masing bagian tersebut.

3.1 Konsep Bilateral Teleoperation

Algoritma dari sistem *bilateral teleoperation* yang dirancang dalam skripsi ini merupakan hasil dari studi literatur dan berusaha mencakup dari tujuan pembuatan skripsi ini. Blok diagram sistem *bilateral teleoperation* yang dirancang dapat direpresentasikan dalam gambar berikut.



Gambar 3.1. Blok Diagram Sistem Bilateral yang Dirancang

Blok Diagram yang dirancang, seperti terlihat pada Gambar 3.1, merupakan ilustrasi dari matriks hybrid Scaling Bilateral Teleoperation. Berikut adalah matriks hybrid Scaling Bilateral Teleoperation.

$$\begin{bmatrix} \beta \tau_m \\ \alpha x_m \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha x_s \\ \tau_s \end{bmatrix} \quad [3.1]$$

Dengan τ dan x merupakan gaya dan posisi dan indeks m dan s adalah *master* dan *slave*. α merupakan representasi dari faktor skala untuk posisi sedangkan β merupakan representasi dari faktor skala untuk force atau beban yang bekerja pada motor. Karena bilateral teleoperation yang dilakukan pada

skripsi ini merupakan bilateral teleoperation dengan 2-DOF pada master dan 3-DOF pada slave, koordinat posisi (x_m dan x_s) pada persamaan tersebut akan digantikan dengan selisih posisi akhir dengan posisi inialisasi pada master dan slave (dx_m dan dx_s). Sehingga persamaan tersebut menjadi seperti berikut.

$$\begin{bmatrix} \beta\tau_m \\ \alpha(dx_m) \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha(dx_s) \\ \tau_s \end{bmatrix} \quad [3.2]$$

Keadaan ideal system Bilateral adalah ketika $\alpha dx_m = dx_s$ dengan menggunakan sensasi *haptic* yang direpresentasikan oleh $\beta\tau_m = \tau_s$. Sensasi *haptic* membuat operator merasakan gaya yang diterima oleh robot di lingkungan. Hambatan di lingkungan robot akan membuat $\alpha dx_m \neq dx_s$. Hal ini akan membuat adanya error pada dx .

$$dx_{err} = \alpha dx_m - dx_s$$

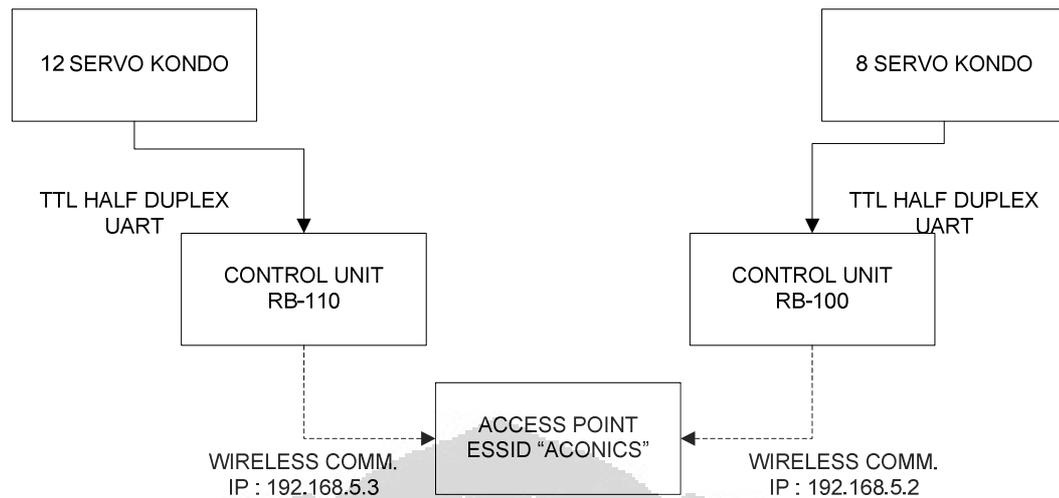
Untuk itu, master akan bergerak sejauh dx_{err} dengan menggunakan $\beta\tau_m$ sebesar τ_s sehingga hal ini akan membuat $\alpha dx_m = dx_s$. Hal ini akan membuat sensasi *haptic* yang dirasakan oleh operator. Kemudian, operator akan memberikan gaya yang lebih besar sehingga membuat $\tau_s \neq \beta\tau_m$ dan $\alpha dx_m \neq dx_s$ sehingga terdapat error gaya di sisi *slave* seperti diberikan oleh persamaan berikut.

$$\tau_{err} = \beta\tau_m - \tau_s$$

Slave akan menaikkan gaya sebesar τ_{err} untuk dapat bergerak sehingga membuat $dx_s = \alpha dx_m$.

3.2 Perangkat Keras

Perangkat keras yang digunakan dalam sistem Bilateral Teleoperation ini meliputi RB-100 dan RB-110 sebagai *control unit*, servo Kondo KRS 2552 HV sebagai aktuator, baterai atau adaptor sebagai *power supply*, dan beberapa perangkat pendukung. Secara umum, sistem yang dirancang memiliki diagram blok sistem sebagai berikut.



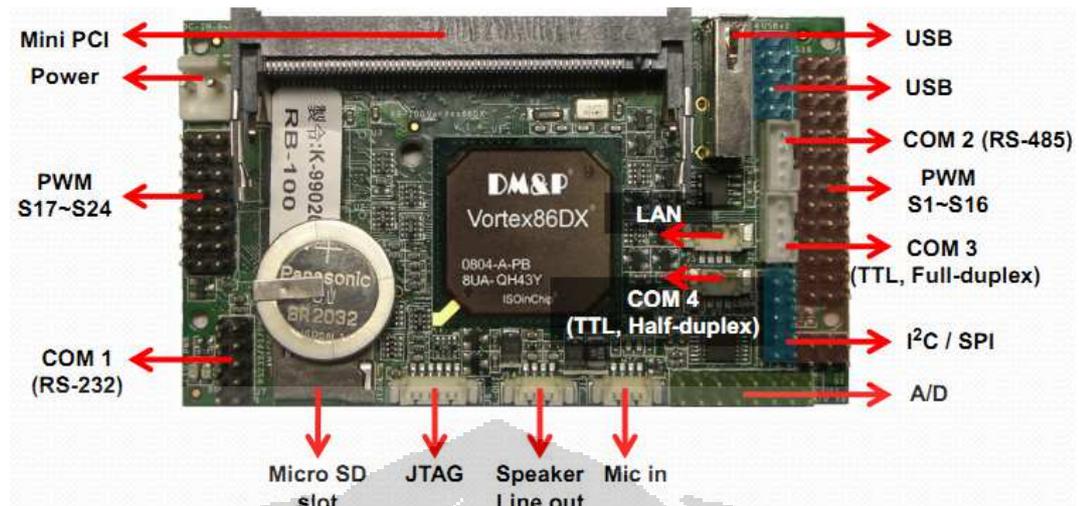
Gambar 3.2. Blok Diagram Secara umum sistem Bilateral Teleoperation pada skripsi ini

Sistem yang dirancang dapat dibagi menjadi dua, yaitu sistem *master* dan sistem *slave*. Sistem yang dirancang hampir sama, hanya saja terdapat 12 servo yang terdapat di *Slave* dan 8 Servo yang terdapat di *master*. Masing-masing sistem menggunakan *port* komunikasi TTL HALF DUPLEX UART untuk mengendalikan servo. Kemudian, masing-masing sistem menggunakan Protokol komunikasi TCP untuk saling berkomunikasi antar Roboard. *Port* Komunikasi ini dipilih karena TCP merupakan serial komunikasi yang paling tahan terhadap *noise* dan memiliki kecepatan transfer data mencapai 54 Mbps.

Dari diagram blok di atas, terlihat bahwa terdapat 2 komponen utama dalam sistem ini, yaitu Roboard, Servo kondo KRS 2552 HV, dan Wireless USB TP-LINK dan Access Point. Selain itu, terdapat pula komponen pendukung, yaitu kabel TTL servo, *Power supply*, dan desain mekanik robot. Berikut ini adalah penjelasan dari perancangan seluruh komponen dari sistem.

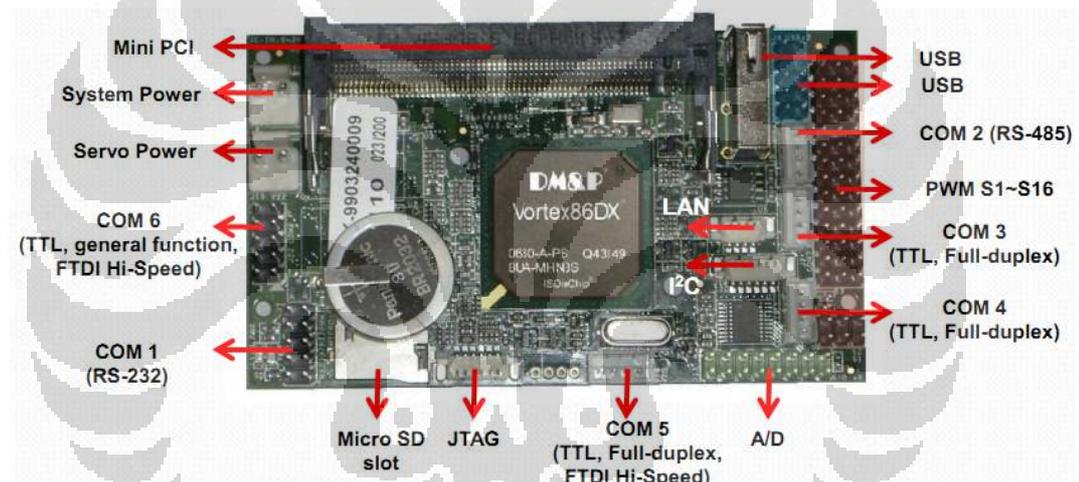
3.2.1 Roboard

Roboard yang digunakan pada sistem ini adalah RB-100 dan RB-110. Karena *port* yang digunakan pada sistem *Master* dan *Slave* sama, tidak ada perubahan dalam penggunaan *port* komunikasi pada masing-masing sistem. Hal yang berbeda adalah tempat *port* komunikasi yang terdapat pada RB-100 dan RB-110. Berikut ini adalah gambar dari *port* komunikasi yang digunakan pada RB-100 dan RB-110.



Gambar 3.3. RB-100

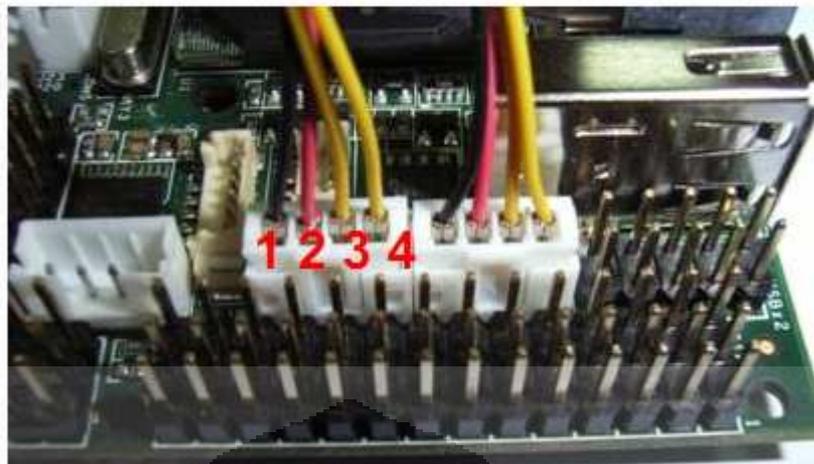
Sumber: datasheet RB-100



Gambar 3.4. RB-110

Sumber: Datasheet RB-110

Terlihat pada Gambar 3.2 dan Gambar 3.3, *port* komunikasi yang digunakan sama, yaitu COM 3 (TTL, Full Duplex UART). Karena servo Kondo KRS 2552 HV hanya dapat digunakan pada Half Duplex UART, COM 3 dapat dibuat menjadi Half Duplex UART. Hal ini dapat dilakukan dengan menyatukan pin TX dan RX pada COM3. Dengan demikian, *port* komunikasi tidak terdapat masalah.



Gambar 3.5. COM 3 (TTL, Full Duplex UART)

Selain *port* komunikasi serial, digunakan pula mini VGA Card untuk disambungkan ke layar monitor sehingga program dapat ditampilkan di layar monitor. Hal yang berbeda dari kedua board ini adalah power yang disuplai ke *board*. Pada RB-100, *power supply* untuk sistem dan motor servo menjadi satu sedangkan pada RB-110, *power supply* untuk sistem dan motor servo dipisah. Oleh karena itu, *power supply* pada RB-110 perlu dilakukan konfigurasi paralel untuk memberikan power ke sistem dan motor servo.

3.2.2 Servo Kondo

Servo kondo yang digunakan pada sistem ini adalah servo KRS 2552 HV. Servo ini digunakan karena penggunaannya cukup mudah dengan hanya memberikan data serial TTL/Half Duplex UART dengan konfigurasi baudrate 115200 bps, 8 bit, 1 stop bit, *no flow control*, dan *even parity*.

Sebelum menggunakan servo kondo, dilakukan studi literatur dengan membaca datasheet dan manual servo kondo. Karena servo kondo menggunakan protokol ICS dengan komunikasi serial Half Duplex UART, diperlukan komunikasi serial Half Duplex UART pada Roboard. Untuk membuat serial Half Duplex UART pada Roboard, hanya diperlukan penyatuan pin TX dan RX pada COM3. Hal ini karena pada Roboard telah terdapat pengaman pada *port* komunikasi Full Duplex UART sehingga aman dalam penggabungan TX dan RX untuk keperluan Half Duplex UART. Pada developmen board yang lain, hal ini dapat dilakukan dengan menambahkan board converter, yaitu 74HC04. Dengan demikian, servo Kondo dapat dikendalikan oleh Roboard. Berikut adalah

ilustrasi dari kabel yang digunakan untuk mengendalikan servo kondo melalui Roboard.



Gambar 3.6. Kabel Servo Kondo pada Roboard

Sumber: Manual DMP Electronics

Setelah dilakukan penyatuan pada pin TX dan RX, pada RB-110, diperlukan *power supply* ke pin motor servo agar servo Kondo memperoleh daya. Dalam menyambungkan kabel servo dengan kabel COM3, penyambungan tidak boleh terbalik karena dapat merusak servo. Setelah hal tersebut dilakukan, dilakukan konfigurasi servo sesuai dengan sistem yang diinginkan. Berikut ini adalah gambar dari sistem bilateral teleoperation.



(a)

(b)

Gambar 3.7. Konfigurasi Servo Kondo, (a) Robot SAR, (b) Sisi Operator

3.2.3 Wireless USB dan Access Point

Wireless USB ini digunakan untuk membentuk suatu jaringan antara master dan slave dengan didukung dengan *Access Point* atau Wireless Router. Wireless Router digunakan untuk menyediakan *access point* untuk membentuk jaringan. *Access Point* ini di-*setting* dengan IP tertentu dan essid tertentu.

Untuk membentuk jaringan UDP, Roboard pada master dan slave menggunakan Wireless USB yang dihubungkan ke essid yang telah dibentuk oleh

access point. Setiap roboard pada *master* dan *slave* memiliki IP tertentu yang berbeda sebagai alamat IP pada masing-masing sistem. Karena digunakan UDP, hanya dibutuhkan nomor *port* untuk transmisi data.



Gambar 3.8. Wireless Router dan Wireless USB

3.2.4 Komponen Pendukung

3.2.4.1 Kabel Serial TTL

Kabel serial TTL yang digunakan merupakan hasil modifikasi dari kabel serial TTL biasa yang telah disediakan oleh Roboard. Kabel ini dimodifikasi dengan menyatukan pin TX dan RX sehingga kabel TTL Full Duplex UART menjadi kabel TTL Half Duplex UART. Hal ini dilakukan karena servo Kondo hanya dapat digunakan dengan menggunakan serial TTL Half Duplex UART. Berikut ini adalah gambar dari kabel serial TTL yang dimodifikasi.



Gambar 3.9. Kabel Serial TTL Half Duplex UART

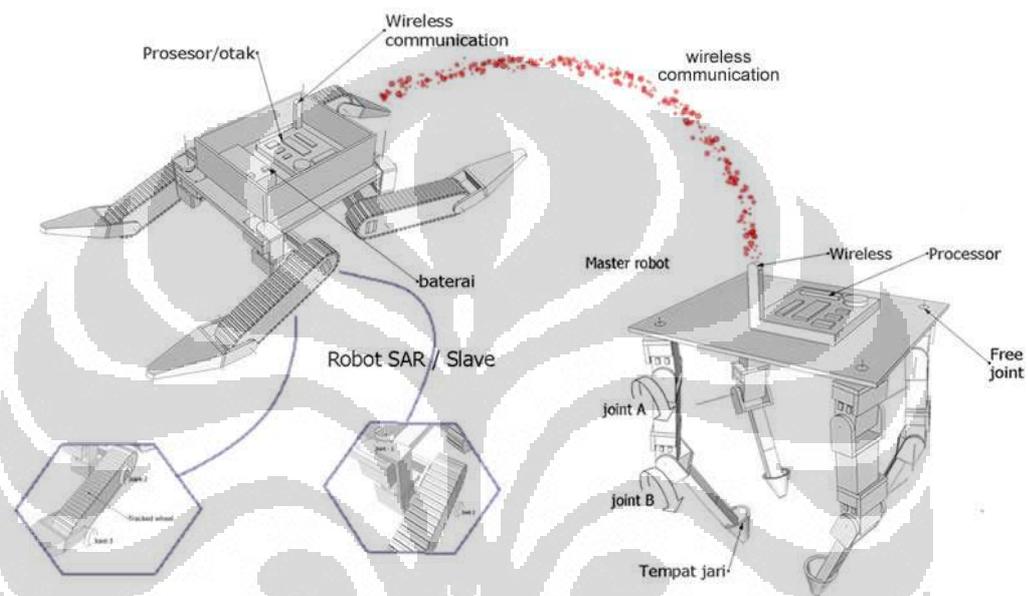
3.2.4.2 *Power supply*

Power supply yang digunakan pada bagian robot bersumber dari baterai LiPo 3 Cell 11.4 V 2200 mAH. Sedangkan pada bagian operator bersumber dari listrik PLN. Digunakan adaptor untuk meregulasi tegangan menjadi 12 VDC. Tegangan ini digunakan untuk menyuplai Roboard dan servo yang digunakan. Adaptor yang digunakan memiliki spesifikasi *input* 85-240 V AC 50/60 Hz

dengan *output* 12 VDC 1200 mA. Pada *power supply* yang digunakan untuk RB-110, kabel diparalel agar dapat menyuplai sistem dan servo.

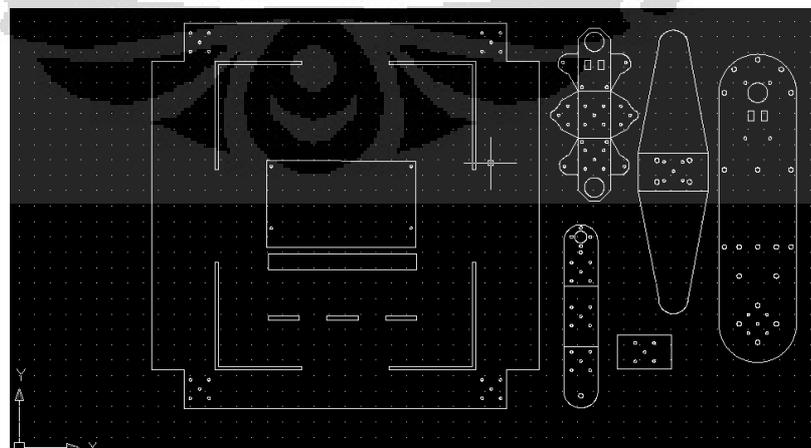
3.2.4.3 Desain Mekanik (Bracket Servo)

Sistem Bilateral Teleoperation pada Robot SAR dan operator ini merupakan pengembangan dari riset yang ada di lingkungan Universitas Indonesia. Desain awal system mekanik Robot SAR dan operator diberikan pada Gambar berikut.



Gambar 3.10. Gambar Sistem Mekanik Bilateral Teleoperation

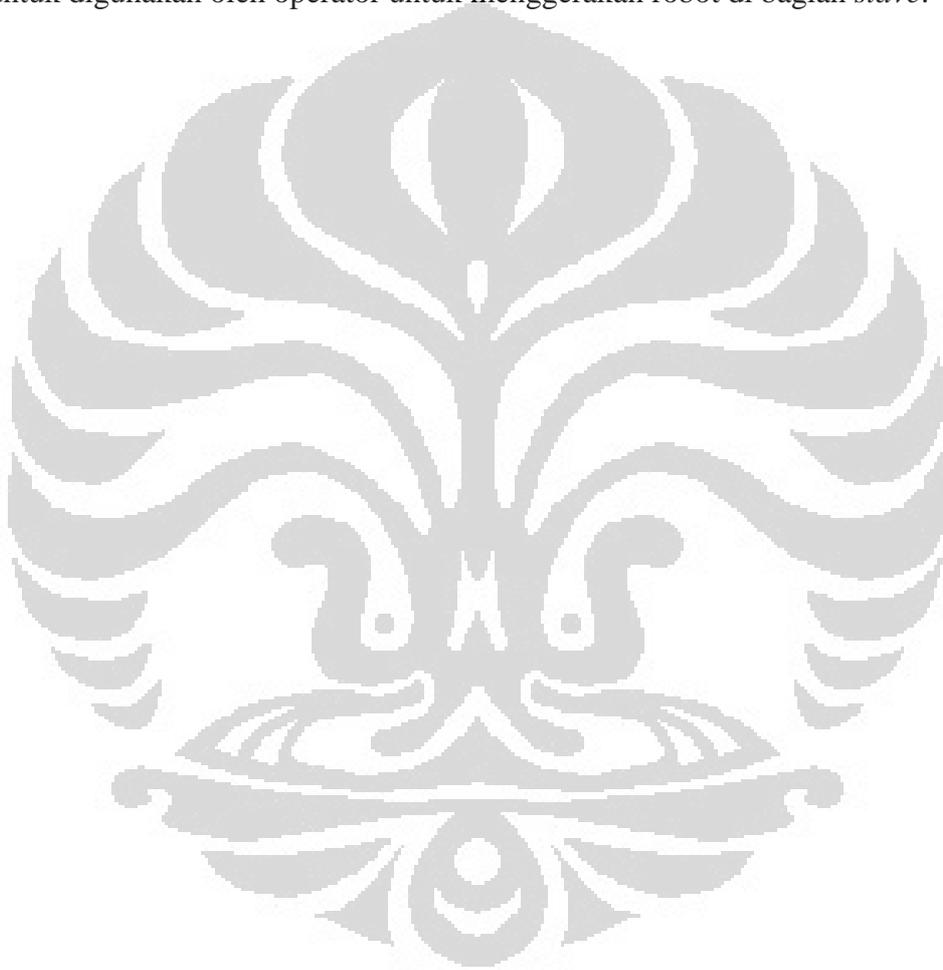
Berdasarkan Gambar 3.9, desain mekanik dibuat secara 2 Dimensi di AutoCad 2007. Desain mekanik menggunakan acrylic sebagai bahan dasar. Kemudian, dilakukan *laser cutting* untuk memproduksi desain mekanik.



Gambar 3.11. Desain Mekanik di AutoCad 2007

Desain mekanik yang berhubungan dengan servo disebut sebagai bracket servo. Kemudian, seluruh desain mekanik dirakit sehingga menghasilkan robot SAR dan robot joystick untuk dikendalikan oleh operator. Bagian desain mekanik yang berhubungan dengan servo disebut sebagai bracket.

Bracket-bracket servo tersebut dirakit bersama dengan bagian kaki dan bodi menjadi suatu robot Quadruped (berkaki empat) dengan motor servo sebagai aktuatornya. Selain robot Quadruped, dibentuk pula system di bagian *master* untuk digunakan oleh operator untuk menggerakkan robot di bagian *slave*.





Gambar 3.12. Desain Mekanik (a) per bagian bracket servo, (b) per bagian kaki robot, (c) Robot SAR, dan (d) Robot Joystick Operator

3.3 Perangkat Lunak

Perancangan Perangkat Lunak dibagi menjadi 3, yaitu fungsi komunikasi antar Roboard, fungsi komunikasi serial kondo, dan algoritma program keseluruhan. Fungsi komunikasi serial dilakukan secara terpisah untuk membuat

modul fungsi komunikasi sehingga memudahkan dalam membuat program. Perangkat Lunak yang dirancang berjalan pada Sistem operasi Ubuntu 10.04 LTS. Berikut ini adalah penjelasan dari perancangan perangkat lunak dari sistem.

3.3.1 Fungsi UDP sebagai komunikasi antar Roboard

Fungsi yang digunakan pada komunikasi antar roboard adalah fungsi UDP. Komunikasi UDP ini dapat dilakukan dengan menggunakan wireless USB. Penggunaan wireless USB ini dapat dilakukan secara otomatis dan dijalankan pada saat pertama kali *booting*. Untuk melakukan hal tersebut, diperlukan beberapa prosedur sebagai berikut.

1. Buka file `interfaces` yang terdapat pada folder `/etc/network/interfaces` melalui terminal. Editor yang dapat digunakan pada terminal biasanya adalah `vi` atau `nano`. Misalnya digunakan editor `vi` sehingga dituliskan pada terminal sebagai berikut.

```
# vi /etc/network/interfaces
```

2. Tambahkan baris di file tersebut dengan mengetikkan baris berikut. (Pada skripsi ini, wireless USB dideteksi oleh system sebagai "ra0").

```
auto ra0
iface ra0 inet 192.168.5.3
iface ra0 netmask 255.255.255.0
pre-up iwconfig ra0 mode managed
pre-up ifconfig ra0 up
pre-up iwconfig ra0 essid "aconics"
```

3. Simpan file tersebut. Kemudian, jalankan instruksi berikut pada terminal.

```
# /etc/init.d/networking restart
```

Karena *Socket programming* yang digunakan pada skripsi ini adalah UDP, diperlukan suatu prosedur dalam pembentukan koneksi antara *master* dan *slave*. Untuk menggunakan UDP, Linux telah menyediakan API (*Application Programming Interfaces*) sebagai antarmuka antara divais dengan programmer. API ini tersedia dalam bentuk fungsi. Fungsi-fungsi yang sering digunakan dalam pemrograman UDP adalah sebagai berikut.

- 1) `socket()`, membuat socket.

- 2) `htonl()`, meng-convert unsigned integer ke network byte order.
- 3) `inet_ntoa()`, ntoa singkatan 'network to ascii', nilai return-nya berupa string dari alamat yang dilewatkan.
- 4) `inet_addr()`, meng-convert string menjadi network byte order.
- 5) `bind()`, digunakan pada pihak server, untuk membuka 'port' komunikasi.
- 6) `write()`, `send()`, bertugas mengirimkan data.
- 7) `read()`, `recv()`, bertugas menerima data.
- 8) `connect()`, digunakan pada pihak client, untuk mengadakan koneksi ke server.
- 9) `listen()`, menunggu koneksi socket.
- 10) `accept()`, menerima koneksi socket.
- 11) `close()`, `shutdown()` menutup atau mengakhiri koneksi.

Dengan menggunakan fungsi-fungsi tersebut, program untuk menjalankan UDP dapat dibuat. Berikut ini adalah langkah prosedur dalam penggunaan UDP yang ditambahkan pada program yang digunakan.

1. Berikut adalah untuk bagian server.

- 1) Inisialisasi variable untuk server. Variable yang digunakan adalah variable-variable untuk *file descriptor*, *stream counter*, *internet socket handling structur*, dan data.
- 2) Membuat *streaming socket* dengan fungsi `socket()`.
- 3) Inisialisasi struct untuk alamat server dan *port*.
- 4) Menentukan nomor *port* untuk socket. Untuk menentukan nomor *port*, digunakan fungsi `bind()`.
- 5) Membuat socket menunggu koneksi dengan menggunakan fungsi `listen()`;
- 6) Inisialisasi variable untuk client. Variable yang digunakan adalah variable-variable untuk *file descriptor*, *internet socket handling structur*, panjang alamat, dan data. Hal ini dilakukan untuk menyimpan data yang diterima dari client.

- 7) Menerima koneksi, membuat *data pipe*. Pada step ini digunakan fungsi `accept()`.
 - 8) Menerima data, dalam hal ini digunakan fungsi `recv()`.
 - 9) Untuk keperluan bilateral teleoperation, terdapat data yang akan dikirimkan kembali ke client. Pengiriman data ini menggunakan fungsi `send()`.
 - 10) Pada mode *listening forever*, yaitu menunggu koneksi selamanya, prosedur dari poin 1.6 hingga poin 1.9 dimasukkan ke dalam *looping*. Kemudian, tutup socket dengan fungsi `close()`;
2. Berikut adalah untuk bagian client.
- 1) Inisialisasi nomor *port* server.
 - 2) Inisialisasi variable untuk server. Variable yang digunakan adalah variable-variable untuk *file descriptor*, *stream counter*, *internet socket handling structur*, dan data.
 - 3) Membuka *socket* dengan fungsi `socket()`.
 - 4) Inisialisasi struct untuk alamat server dan port.
 - 5) Membuat koneksi ke server. Pada step ini digunakan fungsi `connect()`.
 - 6) Mengirim data, dalam hal ini digunakan fungsi `send()`.
 - 7) Untuk keperluan bilateral teleoperation, terdapat data yang akan diterima kembali oleh client. Penerimaan data ini menggunakan fungsi `recv()`.
 - 8) Tutup socket dengan fungsi `close()`;

Data yang diperoleh dari roboard merupakan data array sebanyak 8. Berikut adalah tabel dari data yang dikirim dengan komunikasi UDP.

Table 3.1. Transmisi Data Antar Roboard

TX/RX	1	2	3	4	5	6	7	8
	Dx1	Speed1	Dx2	Speed2	Dx3	Speed3	Dx4	Speed4

Data yang dikirim antara roboard merupakan 8 data yang merepresentasikan *dx* dan *speed* pada masing-masing kaki dari robot. Pengiriman data dari *master* ke *slave* merupakan suatu data variable yang akan ditransformasikan di robot untuk melakukan gerakan tertentu. Pengiriman data dari *slave* ke *master* merupakan suatu data *dx* dan *speed* robot sebagai representasi dari gaya yang dialami oleh robot.

3.3.2 Fungsi Komunikasi serial Servo kondo

Fungsi serial servo kondo merupakan fungsi untuk mengoperasikan servo Kondo. Fungsi ini menggunakan *port* komunikasi serial COM3 (TTL/Full Duplex UART/Half Duplex UART). Untuk melihat jenis *port* pada Roboard, dituliskan perintah pada command Line sebagai berikut.

```
# dmesg | grep tty
```

COM3 terdeteksi pada Roboard sebagai */dev/ttyS2*. Oleh karena itu, dilakukan konfigurasi COM3 di dalam program. Berikut adalah fungsi dari konfigurasi COM3.

1. buat suatu fungsi yang membuat konfigurasi COM3 yang akan digunakan untuk mengendalikan servo Kondo. Di skripsi ini, dibuat fungsi dengan nama `configcom3()`;
2. di dalam `configcom3()`, buka *port* komunikasi COM3. Konfigurasi yang digunakan adalah *read and write*, tanpa CTTY, dan Normal *Blocking Behaviour* (O_NDELAY). Berikut adalah syntax yang digunakan.

```
com3 = open("/dev/ttyS1", O_RDWR | O_NOCTTY | O_NDELAY);
fcntl(com3, F_SETFL, 0);
```

3. dilakukan konfigurasi *port* COM3, konfigurasi yang dilakukan adalah *baudrate* 115200, 8 *data bit*, 1 *stop bit*, *no flow control*, dan *even parity*. Pada fungsi ini, digunakan `kondo_serial_configure()` yang telah tersedia pada `libkondo4` untuk roboard.
4. Konfigurasi serial ini dimasukkan ke dalam variable `com3` untuk dapat digunakan dalam mengirim data. Pengiriman data ke servo kondo mengikuti protokol yang telah ditentukan untuk servo Kondo KRS 2552 HV. Protokol yang telah dibuat, kemudian dikirim dengan menggunakan fungsi `write()`.

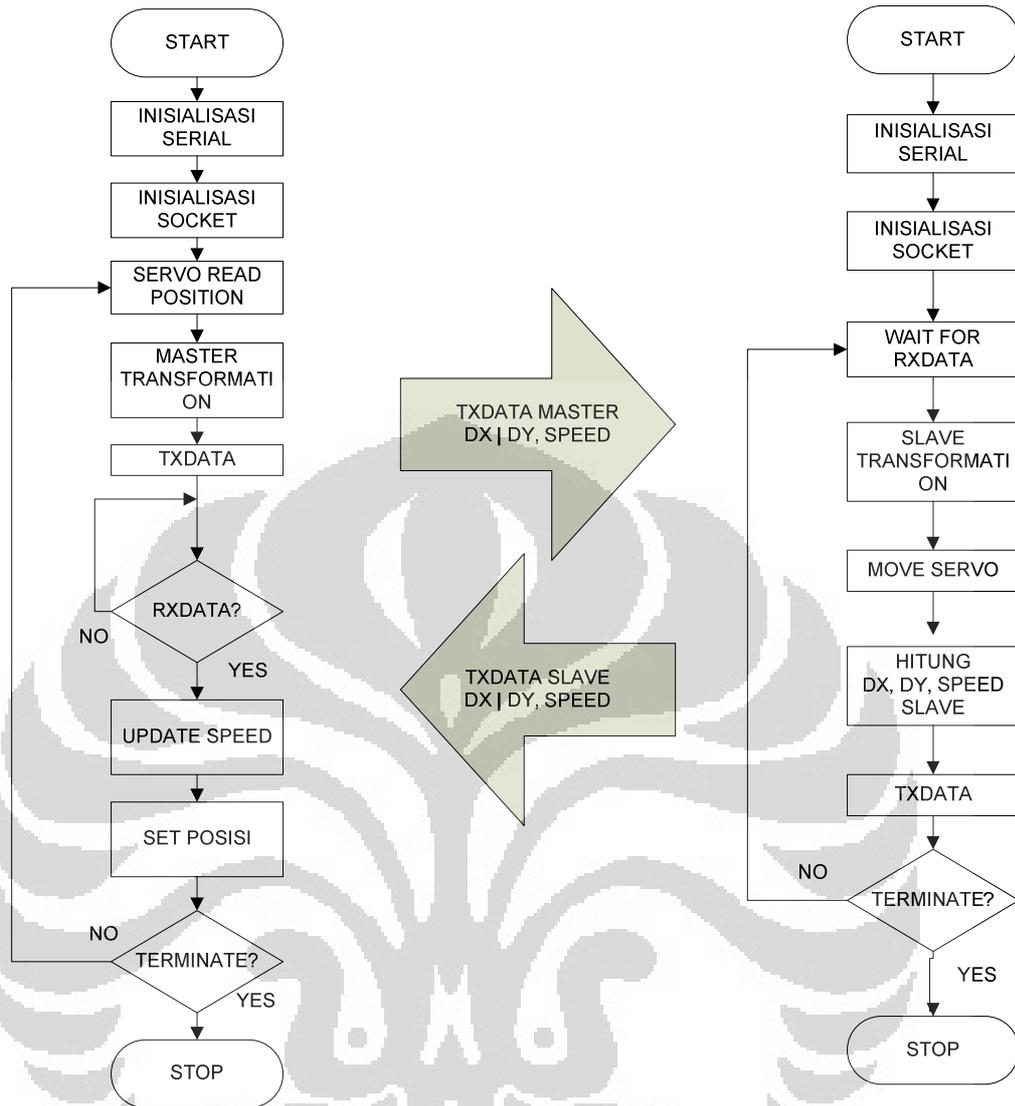
Pada bagian *master*, terdapat fungsi untuk mengetahui sudut servo, yaitu *freerun mode*. *Freerun mode* dibuat untuk mengetahui posisi servo saat itu. Dengan data posisi servo, data tersebut diubah menjadi sudut sehingga dapat dilakukan komputasi *Forward kinematics*.

Dengan demikian, data posisi *end-effector* sistem dapat diperoleh dengan menggunakan fungsi *forward kinematics* dengan sudut yang diperoleh dari servo kondo.

3.3.3 Algoritma

Pada sistem secara keseluruhan, terdapat algoritma dalam melakukan fungsi manipulator sebagai sistem bilateral teleoperation. Berikut adalah algoritma dalam bentuk *flow chart*.





Gambar 3.13. Flow Chart Sistem Bilateral teleoperation

Pada gambar 3.13, digunakan Master-Slave Transformation pada sistem. Program *Master-Slave Transformation* merupakan transformasi yang dilakukan oleh *master* kepada *slave* dan sebaliknya sehingga hasil dari transformasi pada suatu bagian dapat menjadi masukan dengan suatu nilai tertentu pada bagian lainnya. Pada bagian *master*, digunakan *forward Kinematics* yang selanjutnya dihitung selisih posisi dari posisi baru terhadap posisi inisialisasi. Teknik disebut *Master Transformation*. Pada *slave*, terdapat *slave transformation* yang akan mengubah informasi dari *master* untuk kemudian ditentukan respon *slave* terhadap data tersebut. Pada pengiriman data yang dilakukan dari *master* ke *slave*,

data sudut dikirimkan ke *slave* sebanyak 8 byte. Data yang dikirimkan merupakan hasil dari *master transformation* dan *speed* dari masing-masing kaki. Berikut ini adalah table dari pengiriman data dari *master* ke *slave*.

Table 3.2. Pengiriman Data *Master Slave*

TX/RX	1	2	3	4	5	6	7	8
	Dx1	Speed1	Dx2	Speed2	Dx3	Speed3	Dx4	Speed4

Setiap data yang dikirimkan, diolah dari hasil data posisi dari kaki master dengan menggunakan *forward kinematics*.

$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)$$

$$y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$$

Dengan menggunakan data posisi x dan y dari setiap kaki master, dihitung dx dan dy pada setiap kaki. Kemudian, Data dx dan dy diolah ke dalam suatu data binary. Pengolahan data x dan y pada setiap kaki master diberikan pada table berikut.

Table 3.3. Konfigurasi Data Dx dan Dy Pada Setiap Data yang Dikirim

TX/RX	data dx				data dy			
	1	2	3	4	5	6	7	8
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Dari table di atas, terlihat data dx dan dy menyusun suatu data binary 8 bit. Hal ini dilakukan untuk mengakomodasi pengiriman data yang memiliki maksimum 8 *bit* pada setiap *byte*. Dengan demikian, maksimum dari data dx atau dy yang dikirim adalah masing-masing 15 angka desimal atau 4 bit. Berdasarkan maksimum pengiriman data pada setiap koordinat, persamaan *forward kinematics* memiliki nilai l_1 dan l_2 adalah sebesar 1.5 satuan panjang sehingga diperlukan perkalian 10 untuk mengirimkan data koordinat posisi dari *master* ke *slave*. Kemudian, dihitung data dx dan dy dengan menggunakan persamaan.

$$dx = x_{new} - x_{init}$$

$$dy = y_{new} - y_{init}$$

X_{new} dan y_{new} merupakan data koordinat x dan y baru hasil *forward kinematics* terhadap sudut servo yang dibentuk terakhir kali. Dan x_{new} dan y_{new} merupakan data koordinat x dan y hasil *forward kinematics* terhadap sudut servo yang dibentuk pada saat inisialisasi awal. Dengan demikian, tingkat ketelitian koordinat yang digunakan adalah sebesar 0.1 satuan panjang.

Selain data dx dan dy, dikirim juga data *speed master* ke *slave*. Hal ini untuk menentukan *speed* yang akan digunakan oleh *slave* dan sebagai sensasi *haptic* yang dapat dirasakan oleh operator. Pada skripsi ini, pengolahan data ini disebut *master transformation*.

Dengan persamaan di atas, data yang dikirimkan dari *master* ke *slave* adalah data dx dan dy. Data ini akan diproses di sisi *slave* dengan memisahkan data dx dan dy pada setiap byte sesuai dengan table 3.3.

Dari data tersebut, dideteksi kaki mana yang memiliki data dx dan dy. Dari perubahan kaki robot operator tersebut, ditentukan gerakan robot berdasarkan table berikut.

Table 3.4. Gerakan Robot

Motion	Perubahan Leg Master			
	1	2	3	4
Left Walking	x			
Forward Walking		x		
Backward Walking			x	
Right Walking				x
Left Forward	x	x		
Right Forward		x		x
Left Backward	x		x	
Right Backward				x
Standing		x	x	

Kemudian, besarnya nilai dx dan dy dari *master* menentukan seberapa jauh robot bergerak. Nilai dx digunakan untuk banyaknya gerakan robot sedangkan nilai dy digunakan untuk seberapa besar robot dapat berdiri. Data *speed* dari master digunakan untuk menentukan *speed* pada masing-masing kaki robot. Digunakan perhitungan *scaling* untuk menentukan *speed* pada masing-masing kaki robot berdasarkan *speed* master. Persamaan *scaling speed* diberikan sebagai berikut.

$$speed_s = (speed_m \times 5) + 70$$

Dengan menggunakan teknik *scaling*, *speed* pada *master* dapat diimplementasikan pada *slave*. *Slave* akan mengirim data yang diperoleh sebelumnya ke master dengan menggunakan protocol data yang diberikan pada table 3.2. Nilai dx dan dy pada masing-masing kaki robot dijadikan 1 byte dengan menggunakan table 3.3. Nilai *speed* ditransformasikan dengan teknik *scaling* sebagai berikut.

$$speed_{txslave} = (speed_s - 70) / 5$$

Pada skripsi ini, Pengolahan data yang berhubungan dengan transmisi data antar roboard yang dilakukan di *slave* disebut *slave transformation*.

Data yang dikirimkan dari *slave* akan diterima oleh *master*. Data dari *slave* akan diolah dengan menggunakan *master transformation*. Ketika terdapat

perbedaan data dx dan dy dari slave terhadap data dx dan dy sebelumnya, servo di master akan memberikan kenaikan data *speed master*. Kenaikan data *speed* ini akan membuat sensasi *haptic* yang dapat dirasakan oleh operator. Jika operator masih dapat menggerakkan servo, data dx, dy, dan speed baru akan dikirimkan ke slave. Di bagian slave, data speed akan diupdate sehingga slave akan dapat bergerak melewati hambatan. Hal ini terjadi secara terus-menerus hingga *maksimum* kenaikan speed yang ke-10. *Speed master* akan diset secara *maksimum* menandakan *maksimum speed* dari *slave* telah terpenuhi. Operator harus mencari jalan lain untuk menggerakkan robot.

Dengan menggunakan algoritma tersebut, program dibuat dengan menggunakan bahasa C. Program di-transfer ke roboard melalui koneksi wifi, kemudian program di-*compile* dengan menggunakan *compiler gcc* di roboard.

3.4 Sistem Terintegrasi

Perangkat Keras yang digunakan dirakit sesuai dengan blok diagram pada Gambar 3.1. Setelah dirakit, masing-masing sistem ditanam perangkat lunak yang telah dibuat dengan mentransfer data dari laptop ke *storage device* pada Roboard. File hasil transfer tersebut dieksekusi dengan mengetikkan “./<nama_file>” pada direktori yang memiliki file tersebut. Contoh file yang dieksekusi.

```
./main
```

Kemudian, dilakukan *setting* ID servo pada masing-masing sistem. Program telah dibuat untuk memberikan *feedback* kepada *user* agar dapat dengan dinamis mengganti servo pada sistem dengan hanya memasukan ID servo. Dengan demikian, sistem siap dijalankan.



(a)



(b)

Gambar 3.14. ID Servo pada Sistem (a) Sisi Operator (b) Sisi Robot

BAB 4

PENGUJIAN DAN ANALISIS

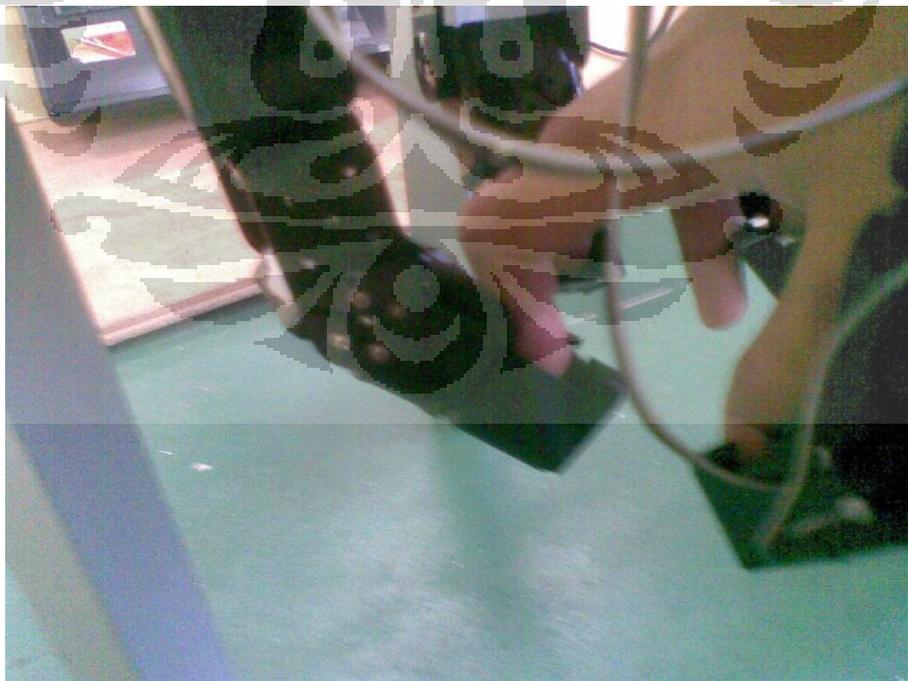
Pengujian sistem dilakukan sebanyak 3 kali, yaitu pengujian bilateral teleoperation 1 (pengujian dengan menggerakkan 1 kaki di sisi operator), pengujian bilateral teleoperation 2 (menggerakkan 1 kaki di sisi operator dengan adanya objek yang menahan robot), dan percobaan bilateral teleoperation 3 (menggerakkan system secara keseluruhan). Berikut adalah detail dari pengujian yang dilakukan.

4.1 Pengujian Bilateral Teleoperation 1

Sistem diuji dengan menggerakkan 1 kaki di sisi operator dan pada *slave* akan bergerak 1 kaki. Pengujian ini dilakukan untuk melihat respon dari *slave* akan data yang diberikan *master*.

4.1.1 Percobaan Bilateral Teleoperation 1

Pada *master*, digerakkan 1 kaki untuk kemudian diperoleh data sudut actual. Data sudut ini ditransformasikan dengan menggunakan *master transformation* untuk dikirimkan melalui socket ke bagian slave, sisi robot, yang akan merespon data yang diberikan oleh master. Berikut adalah Ilustrasi dari pengubahan 1 kaki pada slave.



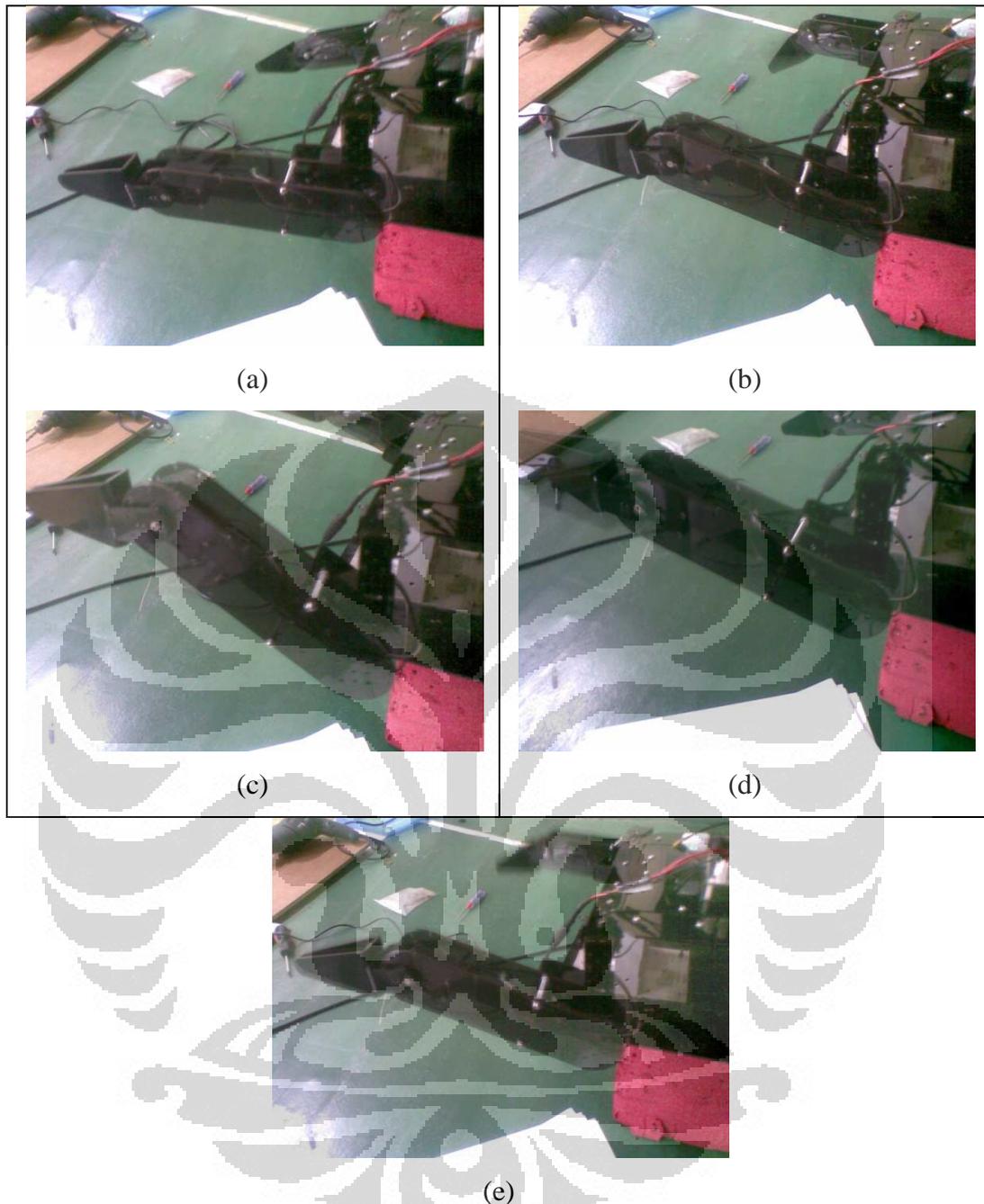
(a)



(b)

Gambar 4.1. Perubahan Posisi 1 Kaki pada Sisi Operator (a) Posisi Awal, dan (b) Posisi Akhir

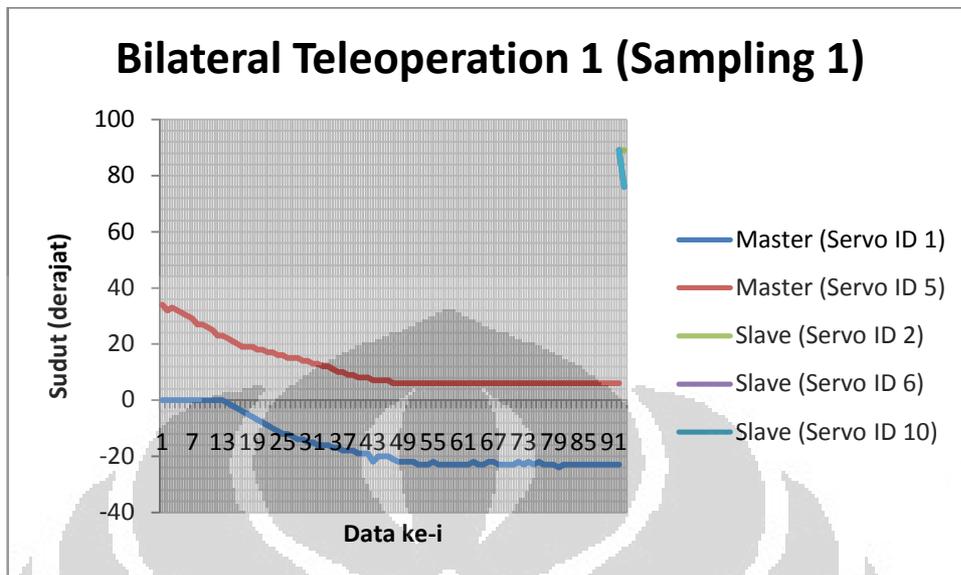
Sistem manipulasi kaki operator ini adalah dengan mengubah secara periodik antara posisi awal, terlihat pada Gambar 4.1. (a), dan posisi akhir, terlihat pada Gambar 4.1. (b). Dengan perubahan ini, diperoleh nilai dx dan dy dari penggunaan master transformation. Data dx dan dy serta speed dikirimkan ke sisi robot sehingga robot akan merespon data dari operator. Di sisi robot telah ditetapkan gerakan untuk merespon nilai dx dan dy yang dikirim dari master, yaitu gerakan *Forward* pada kaki kedua dari sisi robot. Robot akan mengubah kakinya sesuai dengan motion pada table 3.4. Berikut adalah ilustrasi pergerakan kaki robot pada pengujian bilateral teleoperation 1 ini.



Gambar 4.2. Perubahan Posisi Kaki Robot pada Pengujian Bilateral Teleoperation 1 (a) Posisi Awal, (b) Posisi Kedua, (c) Posisi Ketiga, (d) Posisi Keempat, dan (e) Posisi Akhir

Besarnya data dx dan dy yang diterima oleh slave akan mempengaruhi berapa lama robot bergerak. Robot akan bergerak sebanyak nilai dx dan melakukan motion *standing* dengan memanjangkan tungkainya ke tanah berdasarkan nilai dy . Dari pengujian bilateral teleoperation 1 ini, diperoleh data

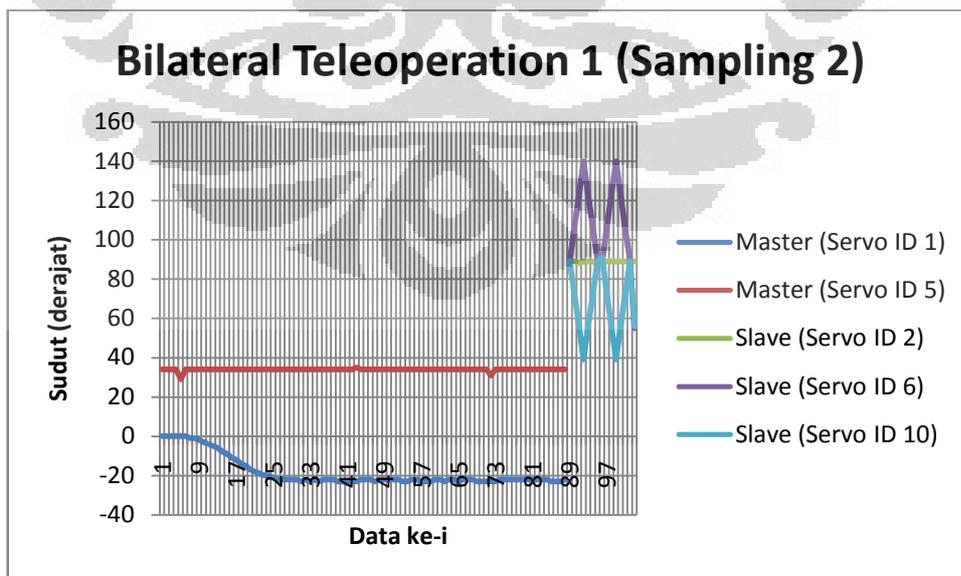
berupa sudut yang dibentuk oleh kaki robot dari masing-masing servo. Data ini diberikan pada grafik berikut.



Gambar 4.3. Grafik Pengujian Bilateral Teleoperation 1 (Sampling 1)

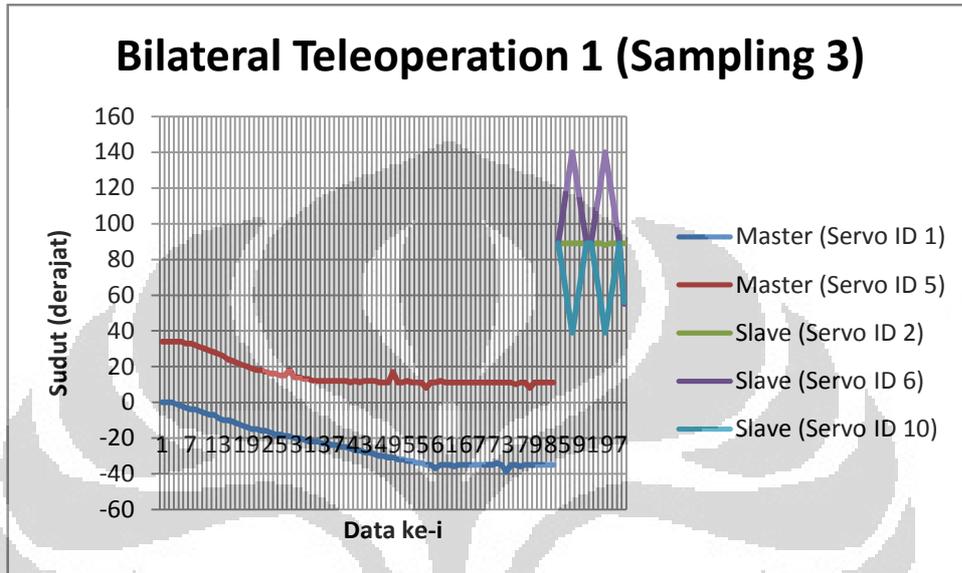
Pada percobaan bilateral teleoperation 1, terdapat 5 kali sampling. Gambar 4.3. merupakan hasil dari gerakan yang dihasilkan oleh robot pada sampling 1. Terlihat bahwa nilai dx bernilai nol sehingga robot tidak bergerak maju. Robot hanya bergerak di tempat karena merespon nilai dy dari master.

Setelah sampling 1, dilakukan pengujian bilateral teleoperation 1 kembali. Berikut ini adalah grafik hasil pengujian tersebut.

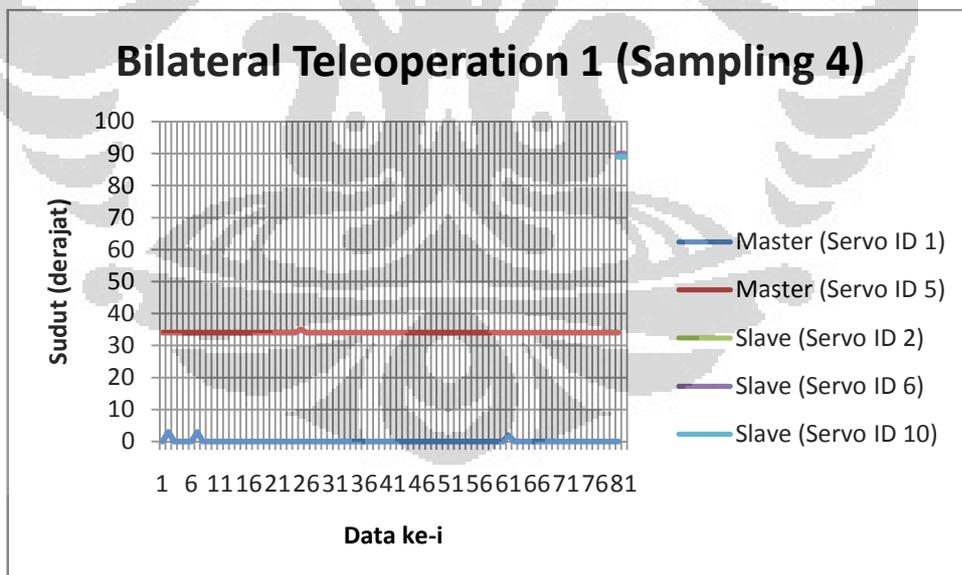


Gambar 4.4. Grafik Pengujian Bilateral Teleoperation 1 (Sampling 2)

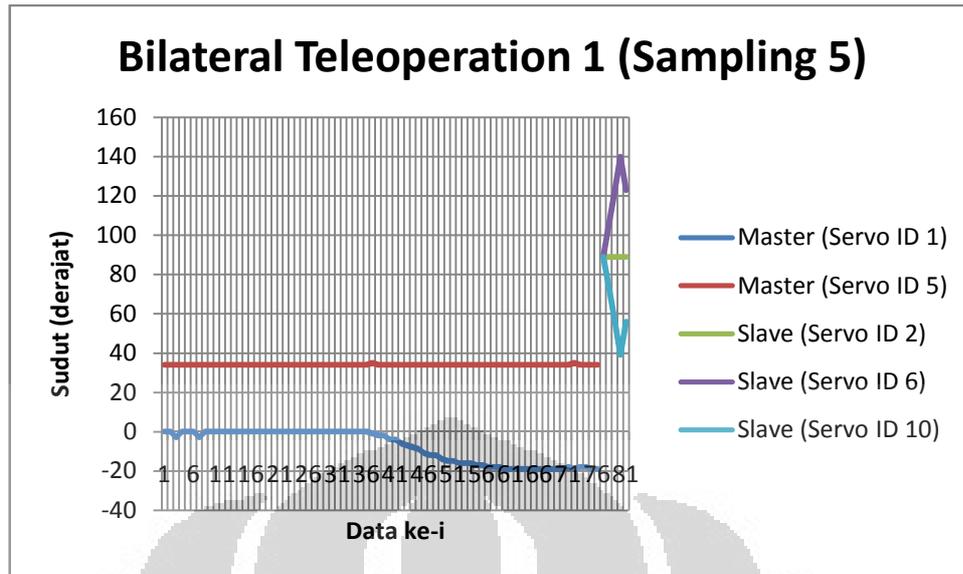
Pada percobaan bilateral teleoperation 1 sampling 2, terlihat bahwa *slave* dapat merespon *master*. Berdasarkan grafik, terlihat *slave* bergerak sesuai dengan besarnya dx dan dy dari master. Slave bergerak dua kali karena besarnya dx , yaitu 2. Kemudian, dilakukan pengujian bilateral teleoperation 1 pada sampling 3, 4, dan 5. Berikut adalah grafik yang diperoleh.



Gambar 4.5. Grafik Pengujian Bilateral Teleoperation 1 (Sampling 3)



Gambar 4.6. Grafik Pengujian Bilateral Teleoperation 1 (Sampling 4)



Gambar 4.7. Grafik Pengujian Bilateral Teleoperation 1 (Sampling 5)

Dari setiap grafik yang diperoleh dari hasil pengujian bilateral teleoperation 1 pada sampling 3, 4, dan 5, terlihat perlakuan berbeda diterapkan pada robot. Pada pengujian bilateral teleoperation 1 sampling 3, perlakuan di sisi operator sama dengan perlakuan ketika sampling 2. Perbedaan antara keduanya adalah besarnya sudut yang dibentuk oleh servo. Terlihat servo *master* memberikan sudut sehingga diperoleh nilai dx dan dy dengan menggunakan *master transformation*.

Pada Pengujian bilateral teleoperation 1 sampling 4, operator tidak menjalankan master dari titik inisialisasinya untuk menguji sistem teleoperation antar system. Terlihat pada grafik bahwa servo slave tidak bergerak. Hal ini karena nilai dx dan dy dari master bernilai nol. Kemudian, pada pengujian bilateral teleoperation 1 sampling 5, operator menjalankan master hanya di satu servo. Hal ini mengakibatkan perubahan nilai dx dan dy yang cukup kecil. Nilai dx yang dihasilkan adalah 1 dan nilai dy yang dihasilkan adalah nol.

4.1.2 Analisis Percobaan 1

Berdasarkan grafik hasil pengujian yang telah dilakukan, terlihat bahwa sistem dapat beroperasi dengan baik. Hal ini karena servo yang digunakan merupakan servo Kondo KRS 2552 HV yang memiliki resolusi 29.62962963 per derajat. Dari sudut -45 derajat hingga 135 derajat pada koordinat cartesian atau -135 derajat hingga 135 derajat pada koordinat sudut servo.

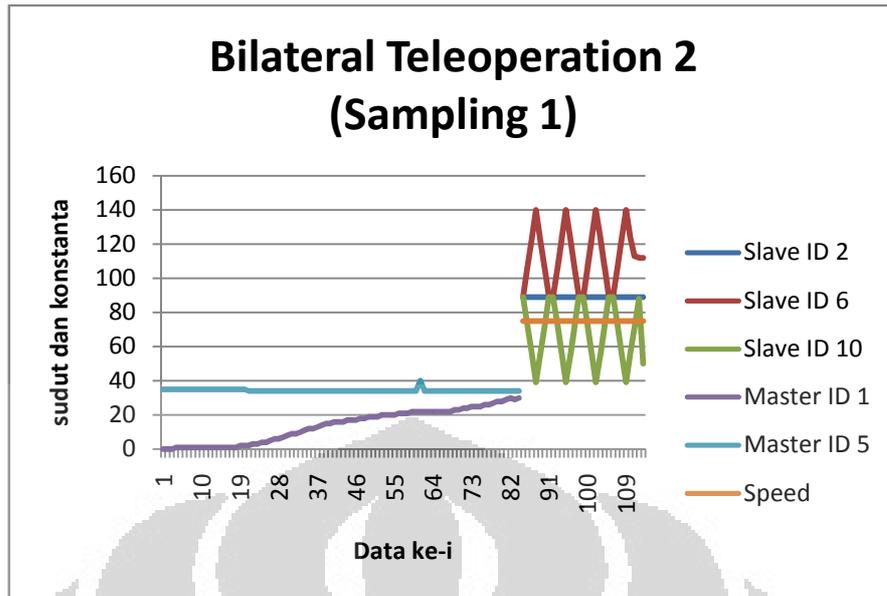
Dari grafik percobaan bilateral teleoperation 1, dapat terlihat servo slave dapat merespon nilai dx dan dy yang diterima dari master. Nilai dx dan dy tersebut ditransformasikan dengan menggunakan *slave transformation* sehingga slave dapat bergerak sesuai dengan perintah operator. Gerakan yang dilakukan oleh slave mungkin masih belum baik. Robot *master* di operator dan robot SAR sebagai *slave* masih belum dapat bergerak secara *mobile* karena keterbatasan system mekanik robot.

4.2 Pengujian Bilateral Teleoperation 2

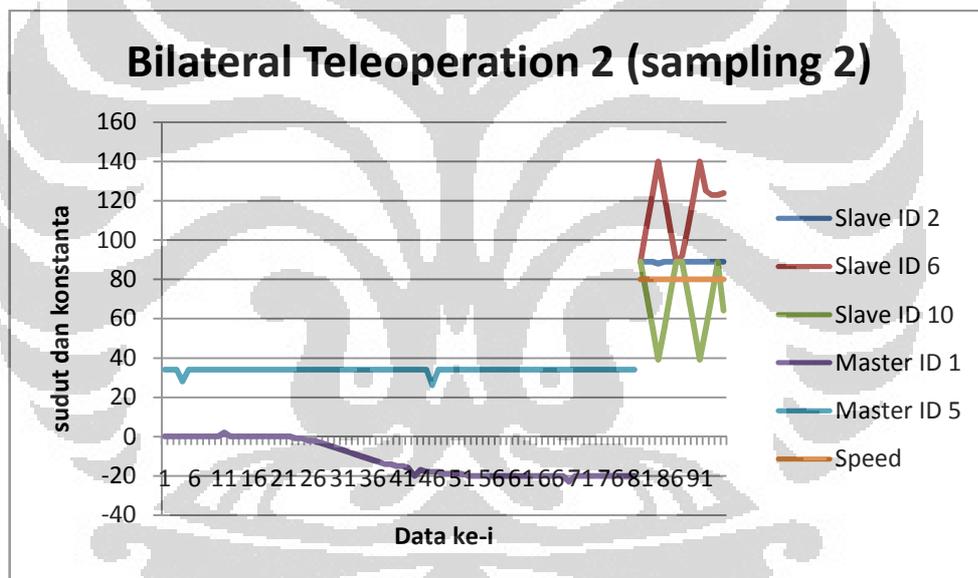
4.2.1 Pengujian Sistem

Pengujian Bilateral Teleoperation 2 dilakukan dengan menggerakkan servo pada 1 kaki di sisi *master*. Pengujian ini dimaksudkan untuk menguji force yang dapat dirasakan oleh operator. Berikut ini adalah ilustrasi dari pengujian yang dilakukan.

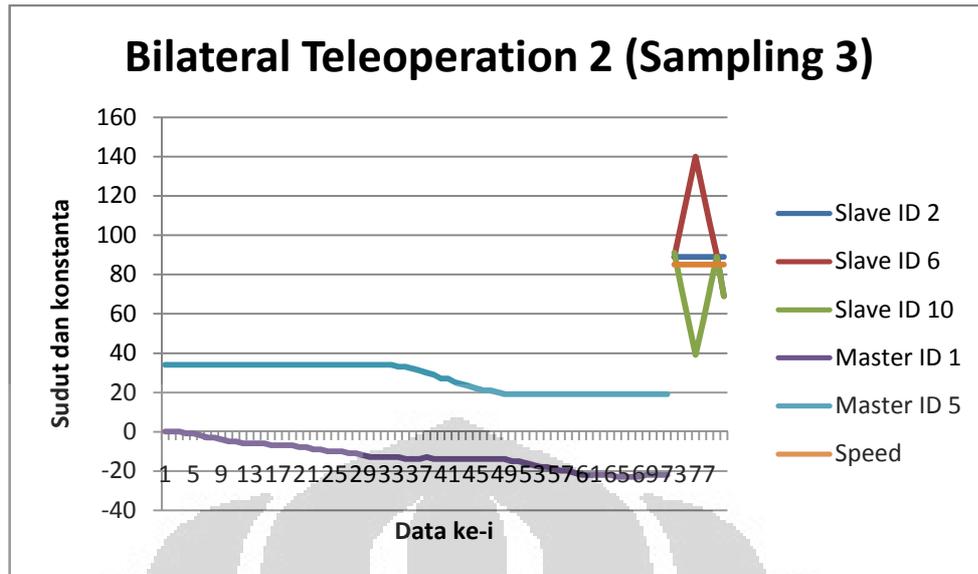
Percobaan 2 dilakukan dengan menggunakan mode *freerun* pada *master*. Percobaan 2 dilakukan dengan mencoba servo pada kaki 1 yang terdapat di operator. Untuk memperoleh sensasi haptic, servo diberi inisialisasi speed sebesar 1. Kemudian, servo digerakan ke sudut inisialisasi, yaitu 7500 desimal, dengan delay 5 ms. Setelah diset posisinya, servo diberikan instruksi *freerun*. Ini digunakan untuk membaca nilai sudut pada servo. Servo dibaca sudutnya dengan delay sebesar 100 ms dengan pengulangan 500 kali. Hal ini dilakukan karena mode *freerun* terdapat kesalahan dalam pengambilan data sekitar 4 – 10 kali. Pengujian dilakukan sebanyak 3 kali. Berikut adalah grafik yang dihasilkan pada pengujian bilateral teleoperation 2.



Gambar 4.8. Grafik Hasil Pengujian Bilateral Teleoperation 2 Sampling 1



Gambar 4.9. Grafik Hasil Pengujian Bilateral Teleoperation 2 Sampling 2



Gambar 4.10. Grafik Pengujian Bilateral Teleoperation 2 sampling 3

4.2.2 Analisis Percobaan 2

Berdasarkan grafik, data yang diperoleh menunjukkan bahwa servo bergerak dengan mengeksekusi suatu motion. Dari Gambar 4.8, terlihat slave bergerak sesuai dengan motion yang dikehendaki. Namun, terlihat pada grafik, di akhir gerakan servo tertahan menyebabkan posisi servo tetap. Dalam kasus ini, *slave* mengirimkan data dx dan dy yang dihitungnya. Karena servo tertahan, perhitungan dx dan dy tersebut memiliki *error* sehingga diketahui oleh sisi *master* bahwa servo slave tertahan. Kemudian, servo *master* menaikkan *speed* sehingga operator dapat merasakan sensasi *haptic* akan tertahannya servo *slave* di robot. *Speed* yang digunakan untuk membuat sensasi *haptic* di sisi *master* akan dikirimkan ke *slave* bersamaan dengan data dx dan dy . Dengan demikian, *speed* pada servo *slave* secara otomatis bertambah.

Pada gambar 4.9, terlihat bahwa data *speed* pada servo slave bertambah. Hal ini mengakibatkan semakin cepat dan kuat dorongan putaran motor servo. Berdasarkan nilai dx dan dy yang diperoleh, servo *slave* bergerak dua kali. Untuk menguji *speed* dari master dan slave, servo secara sengaja ditahan sehingga posisi servo tidak seharusnya. Hal ini dapat terlihat pada akhir grafik pada Gambar 4.9 dimana terdapat garis lurus pada posisi sudut servo. Data dx dan dy serta *speed* servo *slave* dikirim ke *master*. Di sisi *master*, terdeteksi adanya tahanan karena

data dx dan dy dari *slave* tidak sama dengan data dx dan dy *master*. Oleh karena itu, *master* menambah *speed* untuk dapat memberikan sensasi *haptic* bagi operator.

Pada Gambar 4.10, terlihat bahwa *speed* bertambah karena adanya *update speed* dari *master*. Hal ini membuat servo lebih cepat dan kuat dalam melakukan putaran. Dengan demikian, *slave* dapat bergerak sebagaimana mestinya.

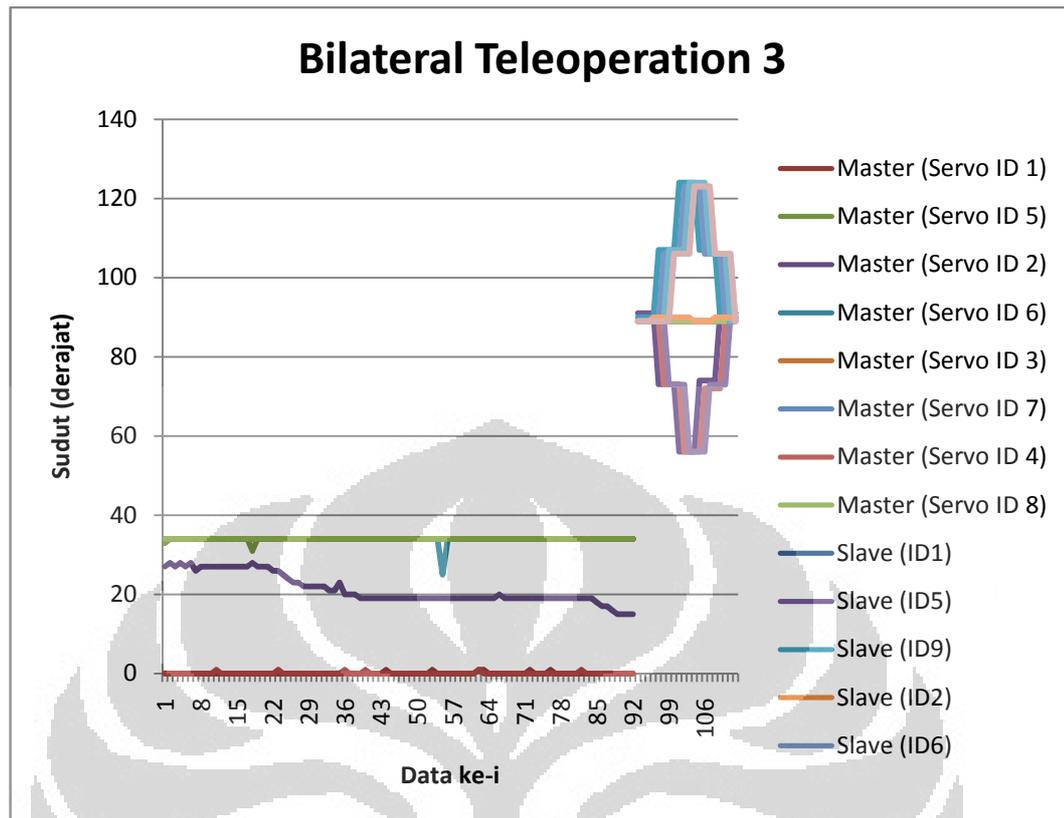
Penambahan *speed* ini dilakukan secara otomatis apabila *slave* terdeteksi terkena hambatan oleh *master*. Penambahan *speed* dilakukan hingga mencapai maksimum *speed* yang dapat diberikan ke servo, yaitu 125.

4.3 Pengujian Bilateral Teleoperation 3

4.3.1 Pengujian Sistem

Pengujian Bilateral Teleoperation 3 dilakukan dengan menggerakkan servo pada seluruh kaki di sisi *master*. Pengujian ini dimaksudkan untuk menguji gerakan robot secara keseluruhan dikontrol oleh operator. Berikut ini adalah ilustrasi dari pengujian yang dilakukan.

Percobaan 3 dilakukan dengan menggunakan mode *freerun* pada *master*. Percobaan 3 dilakukan dengan mencoba keseluruhan servo yang terdapat di operator. Untuk memperoleh sensasi *haptic*, servo diberi *speed* sebesar 1. Kemudian, servo digerakan ke sudut inisialisasi dengan delay 5 ms. Setelah diset posisinya, servo diberikan instruksi *freerun*. Ini digunakan untuk membaca nilai sudut pada servo. Servo dibaca sudutnya dengan delay sebesar 100 ms dengan pengulangan 500 kali. Hal ini dilakukan karena mode *freerun* terdapat kesalahan dalam pengambilan data sekitar 4 – 10 kali. Pembacaan sudut servo berdasarkan leg pada masing-masing servo. Servo ID 1 dibaca terlebih dahulu, kemudian dilanjutkan dengan servo ID 5. Kemudian, pembacaan sudut dilakukan untuk ID 2 dan ID 6 secara berurutan. Selanjutnya, pembacaan sudut dilakukan di leg 3, yaitu ID 3 dan ID 7. Terakhir, pembacaan posisi servo dilakukan di leg 4, yaitu ID 4 dan ID 8.



Gambar 4.11. Grafik Hasil Pengujian Bilateral Teleoperation 3

4.3.2 Analisis Percobaan 2

Berdasarkan grafik, data yang diperoleh menunjukkan bahwa servo bergerak dengan mengeksekusi suatu motion. Terlihat di grafik gerakan tersebut merupakan respon dari dx dan dy dari *leg 2*, yaitu servo master ID 2 dan 6. Terlihat yang terjadi perubahan sudut servo hanya ID 2. Hal ini karena ID 2 digerakan oleh operator. Berdasarkan table 3.4, *motion* yang dilakukan oleh robot adalah FORWARD atau maju. Dengan demikian, operator dapat memberikan instruksi untuk maju ke robot dengan menggerakkan *leg 2* pada *master*.

Dalam hal gerakan, robot masih belum bergerak secara mulus. Hal ini karena adanya keterbatasan system mekanik pada robot. Material yang digunakan sebagai bodi robot juga kurang kuat. Oleh karena itu, pergerakan robot masih belum baik.

BAB 5

KESIMPULAN

Dari keseluruhan pembahasan dari skripsi ini, dapat diperoleh beberapa kesimpulan, yaitu sebagai berikut.

1. Rancang Bangun Robot SAR empat kaki tiga sendi dengan teknik teleoperation menggunakan empat lengan dua sendi dapat direalisasikan dengan menggunakan serial servo Kondo, *Development Board* RB-100 dan RB-110, dan media komunikasi *wireless*.
2. Prinsip kerja Robot SAR empat kaki tiga sendi dengan teknik teleoperation menggunakan empat lengan dua sendi adalah dengan menggunakan *Master-Slave Transformation*.
3. *Master Transformation* menggunakan *Forward Kinematics* yang kemudian dihitung dx dan dy untuk dikirimkan data dx , dy dalam satu *byte* dan *speed* dalam satu *byte* yang lain untuk masing-masing kaki *master* ke sisi *slave*.
4. *Slave Transformation* menggunakan *table* untuk menentukan gerakan robot terhadap data yang diterima oleh robot. Selain itu, dilakukan *scaling* untuk menentukan *speed* bagi robot.
5. Pada Pengujian Bilateral Teleoperation 1, servo slave dapat merespon instruksi dari master dengan baik.
6. Pada Pengujian Bilateral Teleoperation 2, dilakukan *update speed* untuk memberikan sensasi *haptic* di sisi *master* dan untuk menambah gaya dari servo di sisi *slave*.
7. Pada Pengujian Bilateral Teleoperation 3, robot dapat merespon instruksi dari master. Robot menjalankan file gerakan dengan benar. Namun, secara keseluruhan, robot yang dirancang masih memiliki kekurangan yaitu masalah system mekanika robot sehingga gerakan robot masih belum baik.

DAFTAR ACUAN

- Abdul Muis, Kouhei Onishi. (2006, December). *Teleoperation Wwith Soft Sensation Based on Transformer and Gyrator Theorems* , IEEE International Conference on Industrial Technology, 346-351.
- Balweel, F. S. (2011). *Rancang Bangun Sistem Teleoperasi Bilateral 2 DOF Menggunakan Dynamixel AX-12+ dengan Pengendalian Parameter Torsi untuk Meningkatkan Sensasi Haptic* , Skripsi Universitas Indonesia.
- Dewanto, Y. (2011). *Rancang Bangun Bilateral Teleoperation Dengan Teknik Scaling Menggunakan PI Controller* , Skripsi Universitas Indonesia.
- Harneldo, D. (2010). *Rancang Bangun Sistem Bilateral Teleoperation Multi DOF dengan Serial Servo* , Skripsi Universitas Indonesia.
- Mark W Spong, Seth Hutchinson, M. Vidyasagar. (2004). *Robot Dynamics and Control, 2nd Ed.*
- Muis, A. (2007). Dr., S.T., M.Eng. *Friendly Motion Control for Robot as Human Partner Beyond Distance* .

DAFTAR REFERENSI

Buss, S. R. (2009). *Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse, and Damped Least Square*. California: La Jolla.

C. Szep, S.D. Stan, V. Csibi, R. Balan. (2009). Study of Design, Kinematics and Accuracy Modelling of 3 Degrees of Freedom Robot. *MECHANIKA* .

Ching, H. (2006). *Internet Based Bilateral Teleoperation*. Georgia: Georgia Institute of Technology.

Hokayem, Peter F., Mark W Spong. (2006). Bilateral Teleoperation: an Historical Survey. *Journal of Automatica* , 2035-2057.

Marc Alise, Rodney G. Roberts, and Daniel W. Reppenger. (2006). *The Wave Variable Method for Multiple Degree of Freedom Teleoperation Systems With Time Delay* .

Yagmour K, Jon Maters, Gilad ben Yossef, Philippe Gerum. (2008). *Building Embedded Linux Systems, 2nd Ed*. O'Reilly.

LAMPIRAN

