

**APLIKASI ANDROID UNTUK KENDALI ROBOT WAYANG  
MELALUI SISTEM KOMUNIKASI BLUETOOTH**

**Tugas Akhir**

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



Oleh :

Jimi Adriansyah

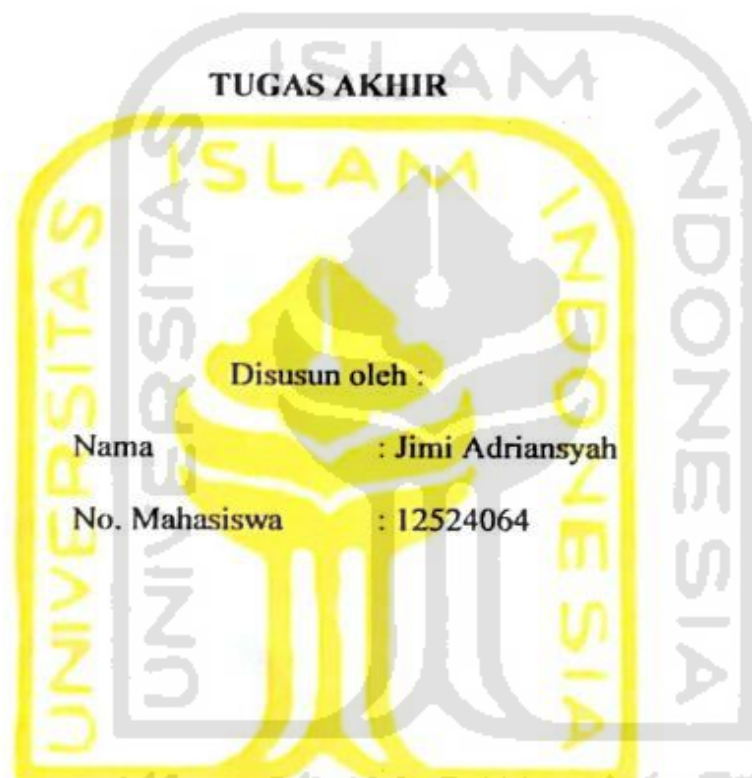
12524064

**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA**

**2017**

**LEMBAR PENGESAHAN DOSEN PEMBIMBING**

**APLIKASI ANDROID UNTUK KENDALI ROBOT WAYANG  
MELALUI SISTEM KOMUNIKASI BLUETOOTH**



**TUGAS AKHIR**

Disusun oleh :

Nama : Jimi Adriansyah

No. Mahasiswa : 12524064

Yogyakarta, Agustus 2017

Laporan ini disetujui oleh

Dosen pembimbing

A handwritten signature in black ink, appearing to be 'Sisdarmanto Adinandra'.

Sisdarmanto Adinandra, ST, M.Sc, Ph.D

**LEMBAR PENGESAHAN DOSEN PENGUJI**  
**APLIKASI ANDROID UNTUK KENDALI ROBOT WAYANG**  
**MELALUI SISTEM KOMUNIKASI BLUETOOTH**

**TUGAS AKHIR**

Disusun oleh :

Nama : Jimi Adriansyah

No. Mahasiswa : 12524064

Telah Dipertimbangkan di Depan Sidang Penguji Sebagai Salah Satu Syarat Untuk Memproleh Gelar Sarjana Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia.

Yogyakarta, Agustus 2017

Tim penguji,

Sisdarmanto Adinandra, ST, M.Sc, Ph.D  
Ketua

Almira Budiyanto, ST.,M.Eng.  
Anggota I

Hendra Setiawan, Dr.Eng.,MT.,ST  
Anggota II



Mengetahui,

Ketua Program Studi Teknik Elektro

Universitas Islam Indonesia



Dr. Eng. Hendra Setiawan, ST., MT

## LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini :

Nama : Jimi Adriansyah

Nim : 12524064

Menyatakan dengan jujur bahwa tugas akhir ini adalah hasil pekerjaan saya sendiri, tanpa adanya niat untuk menjiplak atau melakukan plagiat dari karya orang lain. Adapun bagian-bagian tertentu dari sumber referensi yang saya ambil, sesuai dengan ketentuan dan tata tertib yang berlaku dalam penulisan karya ilmiah. Jika terbukti ada sesuatu hal yang melanggar, saya siap bertanggung jawab sepenuhnya.

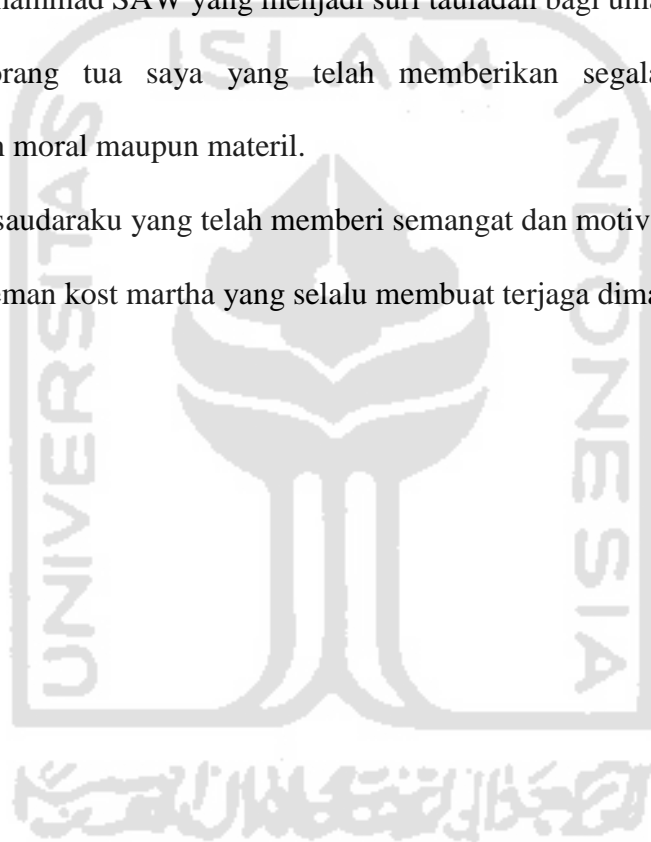
Yogyakarta, Agustus 2017



## HALAMAN PERSEMBAHAN

Dalam tugas akhir ini saya persembahkan kepada :

1. Allah SWT yang telah memberikan saya kekuatan, petunjuk dan karunia-Nya.
2. Nabi Muhammad SAW yang menjadi suri tauladan bagi umat manusia.
3. Kedua orang tua saya yang telah memberikan segala-galanya baik dukungan moral maupun materil.
4. Saudara-saudaraku yang telah memberi semangat dan motivasi.
5. Teman-teman kost martha yang selalu membuat terjaga di malam hari.



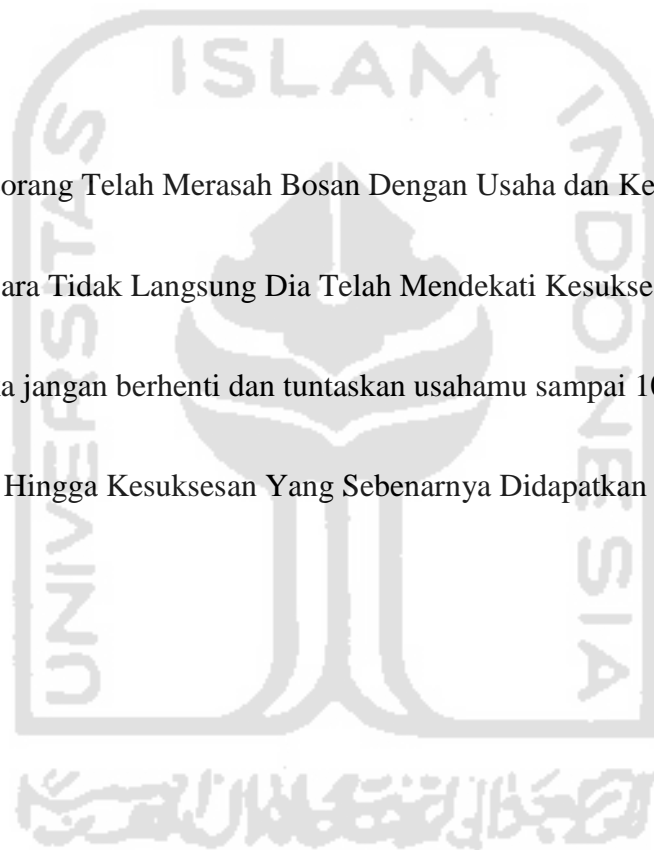
## HALAMAN MOTTO

Apabila Seseorang Telah Merasah Bosan Dengan Usaha dan Kerja Kerasnya

Secara Tidak Langsung Dia Telah Mendekati Kesuksesan

Maka jangan berhenti dan tuntaskan usahamu sampai 100%

Hingga Kesuksesan Yang Sebenarnya Didapatkan



## KATA PENGANTAR

Syukur Alhamdulillah tak henti-hentinya kami haturkan kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga tugas akhir ini dapat diselesaikan. Shalawat dan salam semoga tercurah kepada Rasulullah Muhammad SAW beserta para keluarganya, sahabat dan para pengikutnya hingga akhir zaman. Semoga kita menjadi umat-umatnya yang mendapatkan syafaat beliau di yaumul akhir nanti.

Alhamdulillah penulis ucapkan, karena akhirnya dapat menyelesaikan tugas akhir yang berjudul “Aplikasi Android Untuk Kendali Robot Wayang Melalui Sistem Komunikasi Bluetooth”. Sungguh banyak kisah dan pengalaman yang sangat mengesankan selama mengerjakan tugas akhir ini. Adapun tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Ucapan terima kasih yang sebanyak-banyaknya kepada berbagai pihak yang telah memberikan bantuan, bimbingan, dukungan, kerjasama, fasilitas dan kemudahan lainnya kepada saya dalam penyelesaian tugas akhir ini.

1. Bapak Dr. Eng. Hendra Setiawan, ST., MT selaku ketua prodi Teknik Elektro Universitas Islam Indonesia.
2. Bapak Sisdarmanto Adinandra, Ph.D. selaku pembimbing tugas akhir yang telah meluangkan waktunya untuk memberikan bimbingan sampai terselesaikannya tugas akhir ini.
3. Kedua orang tua saya yang telah memberikan segala-galanya baik dukungan moral maupun materil.

4. Seluruh Dosen Jurusan Teknik Elektro Universitas Islam Indonesia, terima kasih atas bimbingan selama menempuh kuliah di Jurusan Teknik Elektro.
5. Teman-teman; Robi Hatmanto, Bagus Loco, Ripal Aktoba, Fajar Aga, Fadli Adhan, Ridho Dhinata serta teman-teman mahasiswa Teknik Elektro Universitas Islam Indonesia lainnya yang telah memberikan dukungan dan semangat.
6. Dan banyak pihak lain yang tidak dapat penulis sebutkan seluruhnya yang telah membantu dalam penyelesaian tugas akhir ini.

Penulis menyadari sepenuhnya bahwa tugas akhir ini masih jauh dari kesempurnaan. Hal ini disebabkan karena keterbatasan kemampuan dan kurangnya pengetahuan yang dimiliki penulis. Oleh karena itu penulis mengharapkan kritikan dan saran yang mendukung dari semua pembaca untuk kebaikan dan kesempurnaan tugas akhir ini. Semoga tugas akhir ini dapat bermanfaat bagi siapa saja yang membutuhkan.

Yogyakarta, Agustus 2017

Penulis



## Daftar Isi

LEMBAR PENGESAHAN DOSEN PEMBIMBING.....	ii
LEMBAR PENGESAHN DOSEN PENGUJI.....	iii
Lembar Pernyataan Keaslian.....	iv
Halaman Persembahan.....	v
Motto.....	vi
Kata Pengantar.....	vii
Daftar Isi.....	ix
Daftar Gambar.....	xi
Daftar Tabel.....	xii
Abstrak.....	xiii
<b>BAB I</b> Pendahuluan.....	1
1.1 Latar Belakang Masalah.....	3
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Penulisan.....	4
<b>BAB II</b> Tinjauan Pustaka.....	6
2.1 Penelitian Sejenis.....	6
2.2 Dasar Teori.....	8
2.2.1 Kinematika Lengan Robot.....	8
2.2.2 Android.....	9
2.2.3 Arduino.....	10

2.2.4 Modul Bluetooth HC-05.....	11
<b>BAB III Desain Sistem.....</b>	<b>12</b>
3.1 Pemodelan Sistem.....	12
3.2 Diagram Alir.....	13
3.2.1 Diagram Alir Aplikasi Android.....	13
3.2.2 Diagram Alir Arduino.....	16
3.3 <i>Kinematic Chain</i> .....	18
3.4 Perancangan dan Konfigurasi Madul Bluetoth.....	23
3.5 Perancangan <i>User Interface</i> Pada Android.....	24
3.5.1 Perancangan <i>Paired</i> Bluetooth HC-05.....	24
3.5.2 Perancangan Mode <i>Forward Kinematic</i> .....	29
3.5.3 Perancangan Mode <i>Inverse Kinematic</i> .....	32
<b>BAB IV Pengujian, Analisis dan Pembahasan.....</b>	<b>36</b>
4.1 Pengujian Komunikasi Bluetooth HC-05.....	36
<b>BAB V Penutup.....</b>	<b>53</b>
5.1 Kesimpulan.....	53
5.2 Saran.....	53
<b>Daftar Pustaka.....</b>	<b>54</b>
<b>Lampiran.....</b>	<b>55</b>

## Daftar Gambar

Gambar 2.1 Skematik Dari <i>Forward Kinematic</i> .....	8
Gambar 2.2 Skematik Dari <i>Inverse Kinematic</i> .....	9
Gambar 2.3 Modul Bluetooth Hc-05.....	11
Gambar 3.1 Blok Diagram Sistem Kendali Robot Wayang.....	12
Gambar 3.2 Diagram Alir Aplikasi Android.....	14
Gambar 3.3 Diagram Alir Program Pada Arduino.....	16
Gambar 3.4 <i>Kinematic chain</i> dari robot wayang.....	18
Gambar 3.5 <i>Kinematic Chain</i> Pertama Tampak Atas.....	19
Gambar 3.6 <i>Kinematic Chain</i> Bagian Kedua Tampak Samping.....	19
Gambar 3.7 Pengkabelan Modul Bluetooth Hc-05.....	23
Gambar 3.8 <i>Layout</i> Koneksi Bluetooth Hc-05.....	24
Gambar 3.9 <i>Listing</i> Mengaktifkan Bluetooth.....	25
Gambar 3.10 <i>Listing Paired</i> Bluetooth.....	26
Gambar 3.11 <i>Listing</i> Bluetooth <i>Device</i> .....	27
Gambar 3.12 <i>Listing</i> Bluetooth <i>Socket</i> .....	27
Gambar 3.13 <i>Listing</i> Fungsi <i>Throw IOException</i> .....	28
Gambar 3.14 <i>Listing</i> Pindah <i>Layout</i> .....	28
Gambar 3.15 Tampilan <i>Layout Forward Kinematic</i> .....	29
Gambar 3.16 <i>Listing</i> Inisialisasi.....	29
Gambar 3.17 <i>Listing</i> ID <i>Edit Text</i> dan ID <i>Button</i> .....	30
Gambar 3.18 <i>Listing</i> Fungsi <i>Reset</i> .....	31
Gambar 3.19 <i>Listing</i> Send Data.....	31
Gambar 3.20 Tampilan <i>Layout Forward Kinematic</i> .....	32
Gambar 3.21 <i>Listing</i> Inisialisasi.....	32
Gambar 3.22 <i>Listing</i> ID <i>Edit Text</i> dan ID <i>Button</i> .....	33
Gambar 3.23 <i>Listing</i> Fungsi <i>Reset</i> .....	29
Gambar 3.24 <i>Listing</i> Send Data.....	34
Gambar 4.1 Pengiriman Data Ke Robot Wayang.....	36
Gambar 4.2 Durasi Pengiriman Data Ke Robot Wayang.....	52

## Daftar Tabel

Tabel 2.1 Perkembangan Versi Android.....	10
Tabel 3.1 Parameter Denavit-Hartenberg Robot Wayang.....	18
Tabel 4.1 Tabel Pengujian Sistem <i>Forward Kinematic</i> dengan Jarak 1 Meter.....	37
Tabel 4.2 Tabel Pengujian Sistem <i>Forward Kinematic</i> dengan Jarak 2 Meter.....	39
Tabel 4.3 Tabel Pengujian Sistem <i>Forward Kinematic</i> dengan Jarak 3 Meter.....	40
Tabel 4.4 Tabel Pengujian Sistem <i>Forward Kinematic</i> dengan Jarak 4 Meter.....	41
Tabel 4.5 Tabel Pengujian Sistem <i>Forward Kinematic</i> dengan Jarak 5 Meter.....	41
Tabel 4.6 Tabel Pengujian Sistem <i>Forward Kinematic</i> dengan Jarak 6 Meter.....	42
Tabel 4.7 Tabel Pengujian Sistem <i>Forward Kinematic</i> dengan Jarak 7 Meter.....	42
Tabel 4.8 Tabel Pengujian Sistem <i>Forward Kinematic</i> dengan Jarak 8 Meter.....	43
Tabel 4.9 Tabel Pengujian Sistem <i>Forward Kinematic</i> dengan Jarak 9 Meter.....	43
Tabel 4.10 Tabel Pengujian Sistem <i>Forward Kinematic</i> dengan Jarak 10 Meter.....	44
Tabel 4.11 Hasil Perhitungan RMSE Pada Pengujian <i>Forward Kinematic</i> .....	44
Tabel 4.12 Tabel Pengujian Sistem <i>Inverse Kinematic</i> dengan Jarak 1 Meter.....	45
Tabel 4.13 Tabel Pengujian Sistem <i>Inverse Kinematic</i> dengan Jarak 2 Meter.....	46
Tabel 4.14 Tabel Pengujian Sistem <i>Inverse Kinematic</i> dengan Jarak 3 Meter.....	48
Tabel 4.15 Tabel Pengujian Sistem <i>Inverse Kinematic</i> dengan Jarak 4 Meter.....	48
Tabel 4.16 Tabel Pengujian Sistem <i>Inverse Kinematic</i> dengan Jarak 5 Meter.....	49
Tabel 4.17 Tabel Pengujian Sistem <i>Inverse Kinematic</i> dengan Jarak 6 Meter.....	49
Tabel 4.18 Tabel Pengujian Sistem <i>Inverse Kinematic</i> dengan Jarak 7 Meter.....	50
Tabel 4.19 Tabel Pengujian Sistem <i>Inverse Kinematic</i> dengan Jarak 8 Meter.....	50
Tabel 4.20 Tabel Pengujian Sistem <i>Inverse Kinematic</i> dengan Jarak 9 Meter.....	51
Tabel 4.21 Tabel Pengujian Sistem <i>Forward Kinematic</i> dengan Jarak 10 Meter.....	51
Tabel 4.22 Hasil Perhitungan RMSE Pada Pengujian <i>Forward Kinematic</i> .....	52

## Abstrak

Pada tugas akhir ini dirancang suatu desain aplikasi android untuk robot lengan (*articulated robot*) berbentuk wayang 4 DOF (*Degrees of Freedom*) dengan konfigurasi *joint* RRRR. Desain aplikasi pada *smartphone* android digunakan sebagai antarmuka serta perhitungan *forward kinematic* dan *inverse kinematic*. Data hasil penghitungan kinematika robot kemudian dikirimkan ke arduino melalui sistem komunikasi bluetooth. Selanjutnya arduino memproses data yang dikirim dari android dan memberi aksi ke *joint* robot.

Pengujian yang dilakukan yaitu proses pengiriman data dari android melalui komunikasi bluetooth HC-05. Pengiriman data dilakukan dalam mode *forward kinematic* dan *inverse kinematic*. Pengujian komunikasi bluetooth HC-05 ini dilakukan 10 kali percobaan dengan nilai yang berbeda-beda. Di setiap percobaannya dilakukan pengulangan sebanyak 3 kali. Dari hasil 3 kali pengulangan tersebut dihitung nilai rata-ratanya untuk memastikan tingkat akurasi dari hasil yang diperoleh. Kemudian pengujian ini dilakukan pada jarak 1 meter hingga 10 meter.

Dengan menggunakan metode *Root-Mean-Square Error* (RMSE), diperoleh *error* posisi untuk sumbu x, y, z dan orientasi  $\psi$  secara berurutan untuk pengujian komunikasi bluetooth HC-05 dengan mode *forward kinematic* sebesar 0,967 cm; 1,853 cm; 0,887 cm; 1,14°. Untuk pengujian komunikasi bluetooth HC-05 dengan mode *inverse kinematic* diperoleh *error* posisi untuk sumbu x, y, z dan orientasi  $\psi$  secara berurutan sebesar 2,221 cm; 2,829 cm; 2,359 cm; 2,19°.

Kata Kunci : *articulated robot*, Android, Arduino, *forward kinematic*, *inverse kinematic*

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Istilah robot berasal dari bahasa Cheko “robota” yang berarti kuli atau pekerja yang tidak mengenal lelah dan bosan [1]. Robot merupakan teknologi yang saat ini dipakai dalam banyak hal terutama untuk membantu pekerjaan manusia. Robotika berkembang seiring dengan perubahan kebutuhan hidup manusia. Robot dinilai berguna karena dapat mengurangi kesalahan-kesalahan pekerjaan yang umumnya dilakukan manusia. Robot diaplikasikan dalam berbagai hal misalnya bidang industri, pendidikan dan kesehatan.

Robot industri dan kesehatan umumnya dirancang untuk melakukan pekerjaan yang membutuhkan tingkat akurasi dan presisi yang tinggi serta dengan desain yang rumit. Sejak tahun 2010 permintaan robot industri meningkat secara cepat akibat adanya tren menuju otomasi dan banyaknya perkembangan inovatif pada robot industri. Diantara tahun 2010 hingga 2014, penjualan robot meningkat 17% per tahun. Adanya peningkatan penjualan pada tiap tahun menunjukkan tingginya minat pada robot perindustrian. Pada tahun 2014 sendiri penjualan robot meningkat sebesar 29% menjadi 229.261 unit [2].

Berbeda dengan robot industri, robot pendidikan dirancang dengan desain yang sederhana dan mudah digunakan. Salah satu jenis robot yang digunakan untuk pendidikan adalah robot lengan. Untuk menggerakkan robot lengan diperlukan pengendali. Salah satu jenis pengendali yang dapat digunakan adalah

*smartphone* berbasis android. *Smartphone* android digunakan karena kemudahan dalam pengembangan. Karena sistem operasi android dapat diunduh dan dikembangkan secara gratis.

Dari penelitian terdahulu yang dilakukan oleh Marsiano [3], robot wayang masih dikendalikan melalui perangkat komputer/laptop serta pengiriman data antara komputer dengan robot wayang melalui kabel USB. Penggunaan kabel USB dinilai kurang fleksibel. Diantaranya pengiriman data dari komputer ke robot wayang terbatas dengan panjang kabel USB. Sedangkan menggunakan komunikasi bluetooth mencapai jarak maksimum hingga 10 meter.

Ber macam jenis robot lengan tersedia pada laboratorium FTI UII dengan konfigurasi yang berbeda-beda. Robot lengan tersebut masih dikendalikan melalui perangkat komputer/laptop serta pengiriman data antara komputer dengan robot wayang melalui kabel USB. *Smartphone* android dapat digunakan untuk memenuhi kebutuhan tersebut karena telah tersedia fasilitas yang lengkap. Sensor dan koneksi nirkabel menjadi standar fasilitas *smartphone*. Fasilitas *smartphone* yang sering digunakan sebagai pengendali robot adalah sensor posisi dan koneksi bluetooth. Bluetooth digunakan karena sudah dapat dikembangkan menggunakan fungsi-fungsi yang disediakan oleh android versi 2.0. Maka penulis tertarik untuk melakukan penelitian dengan judul aplikasi android untuk kendali robot wayang melalui komunikasi bluetooth.

## 1.2 Rumusan Masalah

Berdasar dari latar belakang masalah, didapatkan rumusan masalah pada tugas akhir ini yaitu perancangan suatu aplikasi android untuk mengendalikan robot wayang dengan metode pengendali *forward* dan *inverse kinematic* melalui komunikasi bluetooth.

## 1.3 Tujuan Penelitian

- Mengetahui cara pembuatan aplikasi android menggunakan perangkat *Java Android* untuk mengendalikan robot wayang.
- Mengetahui performa sistem pengendalian robot wayang dengan mode *forward kinematic* dan *inverse kinematic* menggunakan *smartphone* android.

## 1.4 Manfaat Penelitian

Manfaat dari penelitian tugas akhir ini adalah :

- Menjadi model acuan untuk melakukan pengembangan penelitian perancangan pengendalian robot menggunakan aplikasi android.
- Sebagai sarana pembelajaran lengan robot dengan 4 DOF (*Degree of Freedom*).

## 1.5 Batasan Masalah

- Aplikasi dapat berjalan pada *smartphone* yang memiliki sistem operasi minimal android 4.4 (Kitkat).
- Jarak *smartphone* android ke robot wayang maksimal 10 meter (dengan penghalang).



## **1.6 Sistematika Penulisan**

Sistematika penulisan dan pembahasan laporan tugas akhir ini dapat dijelaskan sebagai berikut :

### **BAB I PENDAHULUAN**

Bab ini membahas tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan penelitian yang diangkat menjadi materi tugas akhir aplikasi android untuk kendali robot wayang melalui sistem komunikasi bluetooth.

### **BAB II TINJAUAN PUSTAKA**

Bab kedua berisi tentang teori-teori yang berhubungan dengan penelitian. Memuat kajian secara induktif yang berisikan hasil penelitian yang pernah dilakukan sebelumnya dan tentunya berhubungan dengan penelitian yang dilakukan. Disamping itu juga memuat kajian secara deduktif yang berisikan konsep dan prinsip dasar yang diperlukan untuk memecahkan masalah penelitian, dasar-dasar teori untuk mendukung penelitian yang akan dilakukan.

### **BAB III PERANCANGAN SISTEM**

Bagian ini menjelaskan perancangan sistem yang digunakan, implementasi rancangan, cara kerja dari rancangan dan cara pengujian sistem (indikator kinerja dan cara mengukurnya).

### **BAB IV PENGUJIAN, ANALISIS DAN PEMBAHASAN**

Bagian ini berisi penjelasan analisis hasil pengujian sistem yang dibuat dibandingkan dengan kriteria (indikator kinerja) hasil pengujian yang telah

ditentukan. Selain itu bab ini juga menjelaskan dengan detail langkah-langkah yang dilakukan sebelum, selama, dan setelah percobaan dilakukan.

## **BAB V KESIMPULAN DAN SARAN**

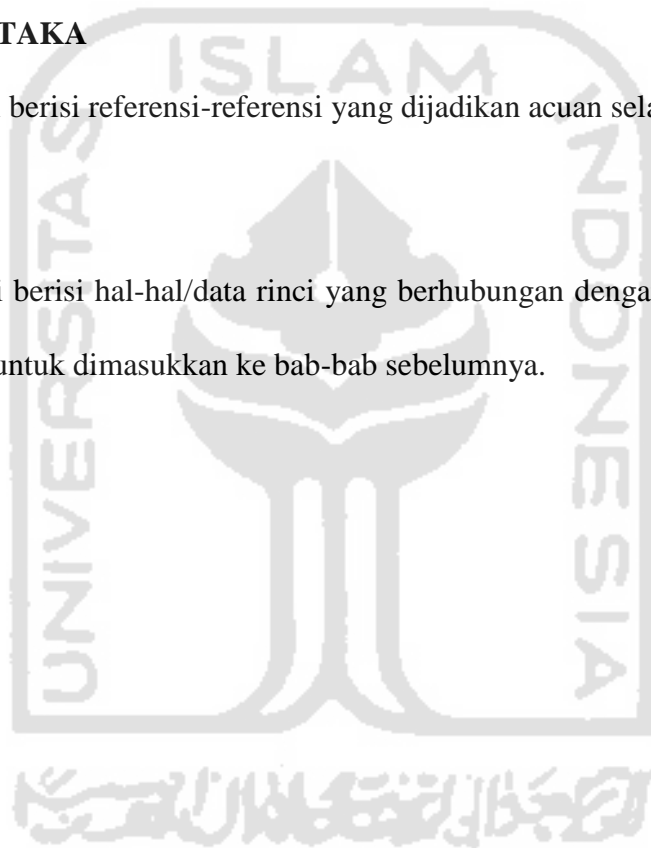
Bab ini menyampaikan kesimpulan dari penelitian yang telah dilaksanakan serta saran-saran untuk pengembangan lebih lanjut.

## **DAFTAR PUSTAKA**

Bagian ini berisi referensi-referensi yang dijadikan acuan selama penelitian.

## **LAMPIRAN**

Bagian ini berisi hal-hal/data rinci yang berhubungan dengan skripsi, tetapi terlalu panjang untuk dimasukkan ke bab-bab sebelumnya.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Sejenis

Penelitian yang mengenai sistem pengendalian robot lengan adalah “Perancangan Kontroler Lengan Robot Hastobot Menggunakan Android Dan Arduino Dengan Komunikasi Bluetooth” oleh Wiratama [4]. Hastobot terbuat dari akrilik dan digerakkan dengan 4 buah motor servo. Hastobot memiliki 4 (empat) derajat kebebasan. Penelitian ini difokuskan terhadap pembuatan kontroler untuk lengan robot Hastobot menggunakan arduino dan android dengan komunikasi bluetooth. Arduino terhubung dengan modul bluetooth HC-05 sebagai *slave* untuk menerima data dari aplikasi android. Aplikasi android dibuat dengan *Eclipse Integrated Development Environment*. Motor servo digerakkan melalui mode geser *touch screen* pada aplikasi android dengan nilai koordinat x dan y. Format data yang dikirimkan aplikasi android menggunakan *header* data untuk mengidentifikasi pergerakan masing-masing motor servo.

Penelitian lain dengan judul “Sistem Kendali Gerak *Continuous Path Tracking* Dengan Menggunakan *Cubic Trajectory Planning* Pada Robot Manipulator 4 DOF” oleh Nurhakim [5]. Pada penelitian ini dirancang sebuah *prototipe* robot manipulator berbentuk lengan 4 DOF dengan konfigurasi sendi PRRR. Sendi robot manipulator ini digerakkan dengan motor DC dan motor servo yang dikendalikan secara *closed-loop* menggunakan mikrokontroler Atmega 2560. Potensiometer digunakan untuk mengamati posisi servo pada tiap *joint rotasi* dan *rotary encoder* untuk mengamati perubahan panjang lengan.

Pergerakan sendi dan *end-effector* robot menggunakan *trajectory planning* dan dapat diamati secara *real-time* oleh software komputer berbasis .Net.

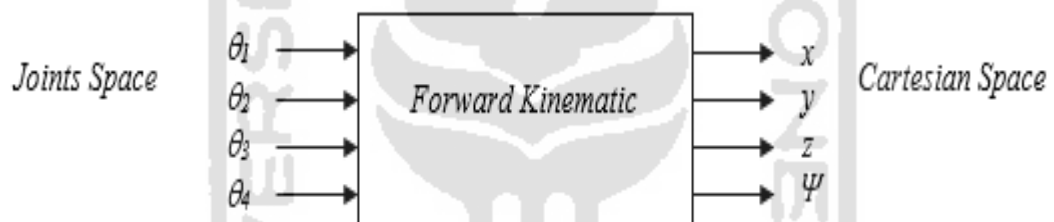
Penelitian lain mengenai perancangan aplikasi android sebagai pengendali adalah penelitian oleh Saefullah [6] yang berjudul “Sistem Kontrol Robot Pemindah Barang Menggunakan Aplikasi Android Berbasis Arduino Uno”. *Smartphone* android digunakan untuk menjalankan aplikasi untuk mengendalikan robot. Modul bluetooth HC-06 yang merupakan perangkat yang dapat menerima dan mengirim sinyal untuk komunikasi antara *smartphone* android dengan mikrokontroler. Aduino IDE sebagai *platform* untuk memasukkan program ke dalam mikrokontroler Atmega328. *Power Supply* merupakan catu daya untuk memberikan tegangan kepada arduino dan motor servo yang merupakan alat penggerak yang bergerak sesuai perintah mikrokontroler.

Robot Pemindah Barang ini memiliki beberapa fungsionalitas diantaranya, menggunakan 5 servo yaitu 1 servo untuk poros bergerak horizontal, 3 servo untuk kekuatan bergerak vertical, dan 1 servo untuk *gripper*. *Gripper* robot untuk mencengkram barang berupa dus dengan berat beban yang diangkat 4 kg dan lebar 5 cm.

## 2.2 Dasar Teori

### 2.2.1 Kinematika Lengan Robot

Kinematika adalah studi yang mempelajari pergerakan robot tanpa memperhatikan gaya yang mempengaruhi pergerakan lengan robot. Terdapat dua kinematika yaitu kinematika maju (*forward kinematic*) dan kinematika mundur (*inverse kinematic*). *Forward kinematic* adalah suatu metode untuk menentukan posisi ( $x$ ,  $y$  dan  $z$ ) dan orientasi ( $\Psi$ ) dari *end-effector* suatu robot dari variabel *joint* (sudut untuk *joint* rotasional dan jarak untuk *joint* prismatic) antar link dapat dilihat pada Gambar 2.1. [7]



Gambar 2.1 Skematik Dari *Forward Kinematic*

Berkebalikan dengan *forward kinematic*, *inverse kinematic* adalah suatu metode untuk menentukan variabel dari tiap *joint* (sudut atau panjang) untuk suatu posisi dan orientasi dari *end-effector* robot. Berbeda dengan *forward kinematic*, *inverse kinematic* tidak memiliki solusi yang unik. Untuk mendapatkan solusi dari *inverse kinematic* secara umum terbagi menjadi 2 metode. Metode tersebut adalah metode analitik dan metode numerik. Metode analitik adalah metode pencarian solusi yang berdasarkan pada karakteristik geometri dari robot yang bisa diselesaikan secara aljabar. Metode kedua adalah metode numerik, metode ini melakukan iterasi dengan metode tertentu untuk mendapat solusi yang paling

optimal. Dibanding dengan metode analitik, metode numerik dapat mencakup banyak konfigurasi dari lengan robot, namun metode ini tidak dapat menemukan semua solusi *inverse kinematic* sekaligus. Selain itu, karena metode ini berbasis iterasi maka solusi yang didapat relatif lebih lambat daripada metode analitik. Adapun skemaatik dari *inverse kinematic* dapat dilihat pada Gambar 2.2.



Gambar 2.2 Skematik Dari *Inverse Kinematik*

### 2.2.2 Android

Android adalah sistem operasi yang berbasis linux untuk telepon seluler seperti *smartphone* dan komputer tablet. Android bersifat *open source*. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi sesuai keinginan mereka sendiri yang digunakan untuk berbagai macam piranti bergerak.

Awalnya, Google Inc. Membeli android Inc., pendatang baru yang membuat piranti lunak untuk ponsel. Kemudian untuk mengembangkan android, dibentuklah open handset alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi, termasuk google, HTC, intel, motorola, qualcomm, t-mobile dan nvidia.

Perkembangan versi android ditunjukkan pada Tabel 2.1 sebagai berikut [8]:

Tabel 2.1 Perkembangan Versi Android

Versi	Nama Kode	Tanggal Rilis	Level API
7.0	Nougat	22 Agustus 2016	24
6.0	Marshmallow	19 Agustus 2015	23
5.x	Lollipop	15 Oktober 2014	21
4.4.x	Kitkat	31 Oktober 2013	19
4.1.x	Jelly Bean	9 Juli 2012	16
4.0.3 - 4.0.4	Ice Cream Sandwich	16 Desember 2011	15
3.1	Honeycomb	10 Mei 2011	12
2.3 - 2.3.2	Gingerbread	6 Desember 2010	9
2.2	Froyo	20 Mei 2010	8
2.0 - 2.1	Eclair	26 Oktober 2009	7
1.6	Donut	15 September 2009	4
1.5	Cupcake	30 April 2009	3

### 2.2.3 Arduino

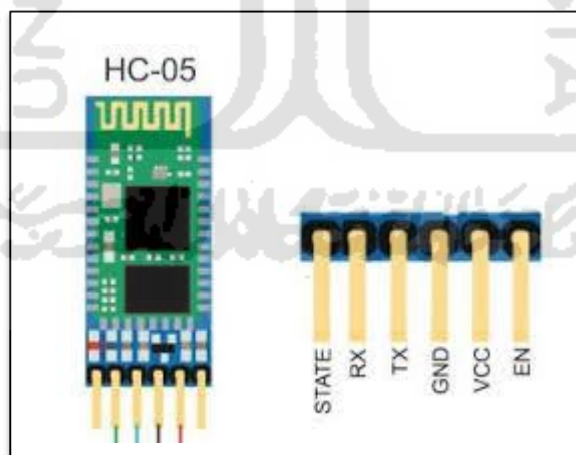
Arduino adalah suatu perangkat *open source* berbasis mikrokontroler yang dapat digunakan sebagai basis pembuatan berbagai proyek elektronika. Tujuan awal arduino dikembangkan adalah penyedia suatu perangkat dengan harga terjangkau dan mudah digunakan baik oleh pemula maupun profesional dalam perancangan suatu perangkat menggunakan kombinasi aktuator dan sensor. Sistem arduino menyediakan bermacam *port input/output* baik digital maupun analog. Beberapa sistem komunikasi yang digunakan oleh arduino adalah USB

dan port serial. Baik *port I/O* maupun komunikasi dapat ditambahkan pada arduino dengan modul-modul yang disebut *shield*.

Arduino menggunakan bahasa pemrograman yang disebut *processing*. Bahasa pemrograman ini berbasis bahasa C dan C++. Banyak *library* yang dapat digunakan untuk mempercepat proses pengembangan proyek berbasis arduino.

#### 2.2.4 Modul Bluetooth HC-05

Modul HC-05 merupakan tipe modul bluetooth yang dapat diubah dari *master* ke *slave* maupun sebaliknya. Modul bluetooth dapat digunakan ketika perangkat *master* telah melakukan *pairing* dengan perangkat *slave*. Komunikasi tidak akan terjadi ketika semua perangkat adalah *master* atau semua perangkat adalah *slave*. Komunikasi yang terjadi antara modul bluetooth *master* dan modul bluetooth *slave* seperti komunikasi serial dengan adanya TXD dan RXD dapat dilihat pada Gambar 2.3.



Gambar 2.3 Modul Bluetooth HC-05

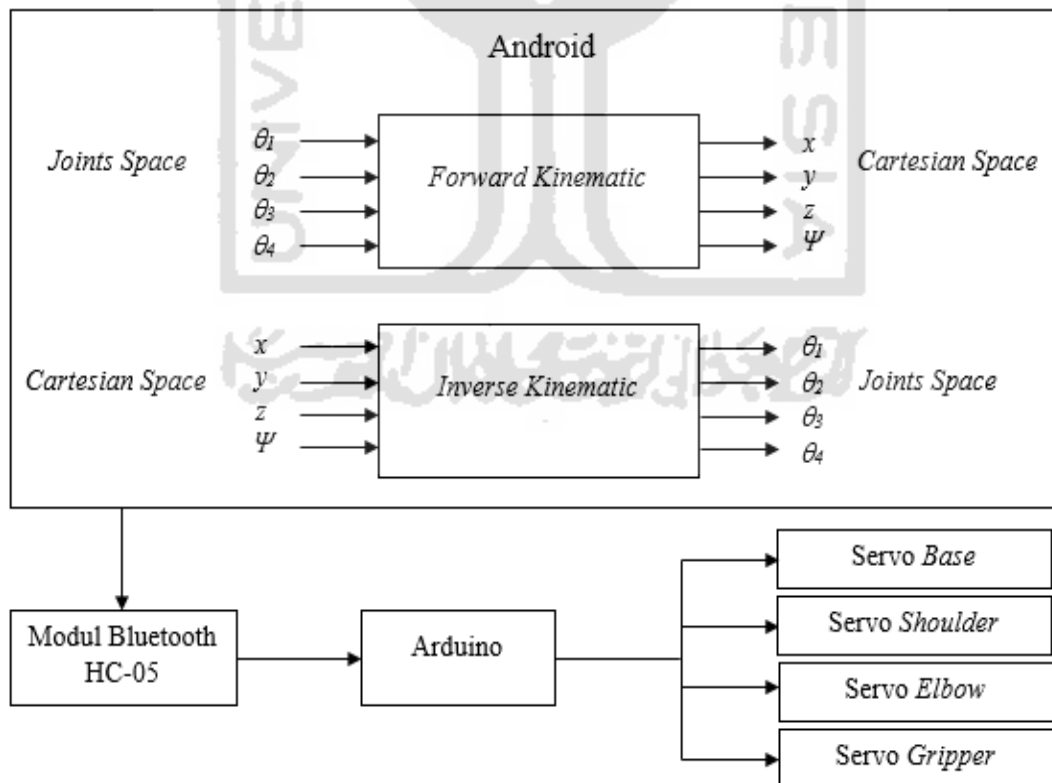


## BAB III

### DESAIN SISTEM

#### 3.1 Pemodelan Sistem

Blok diagram sistem dari Gambar 3.1 terdiri dari antarmuka pada android yang mengatur kinematika kendali robot secara *forward* maupun *inverse* yang kemudian mengatur besar sudut dari setiap *joint* robot. Untuk metode *forward kinematic* berupa pengaturan sudut putar dari *joint* secara langsung yang kemudian dikirim ke robot wayang melalui komunikasi bluetooth menuju arduino. Sedangkan untuk metode *inverse kinematic* berupa pengaturan posisi sumbu (x, y, z) dan orientasi ( $\Psi$ ) pada *end-effector* robot wayang. Robot wayang ini semua sendinya merupakan sendi rotasi yang digerakkan menggunakan servo.

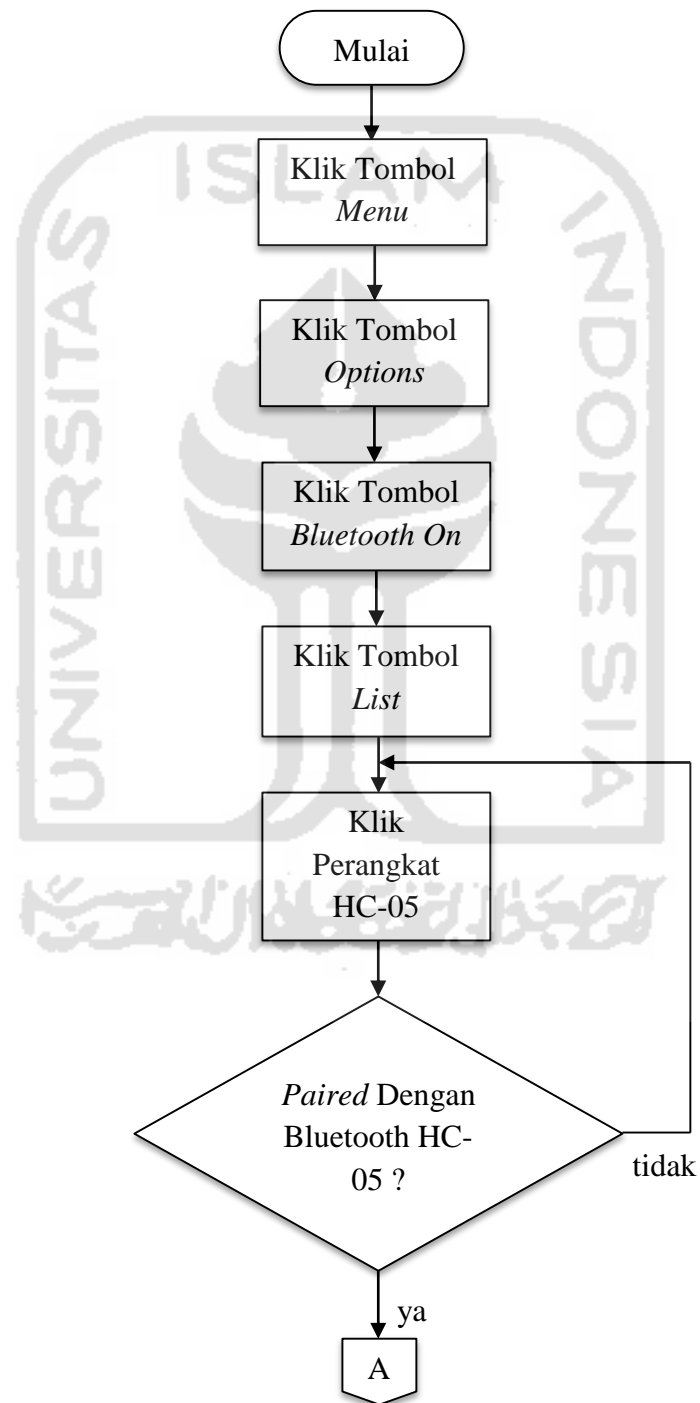


Gambar 3.1 Blok Diagram Sistem Kendali Robot Wayang

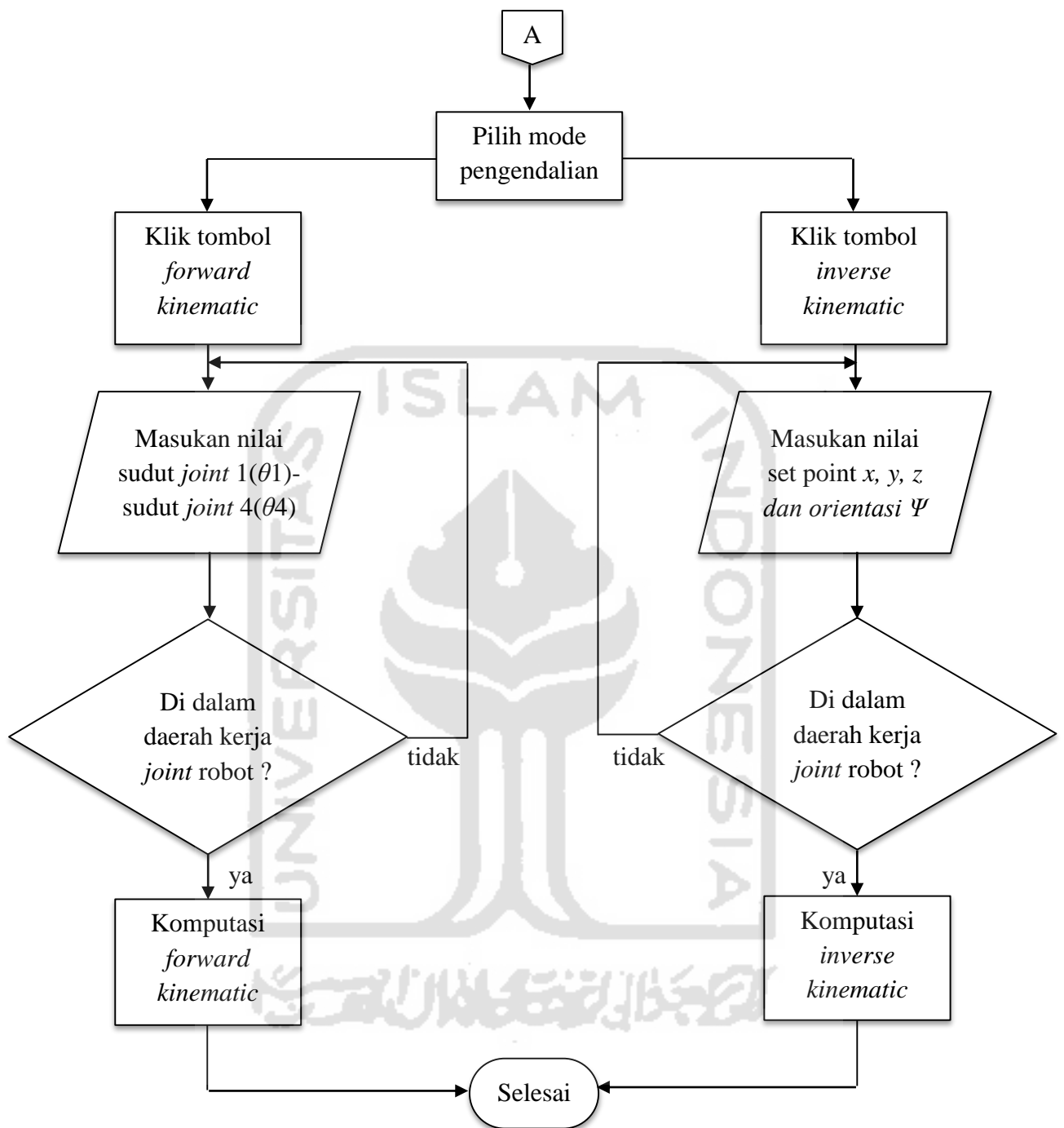
## 3.2 Diagram Alir

### 3.2.1 Diagram Alir Aplikasi Android

Diagram alir dari aplikasi android dalam mengoperasikan robot wayang dapat dilihat pada Gambar 3.2 sebagai berikut.



Gambar 3.2 (a) Diagram Alir Aplikasi Android



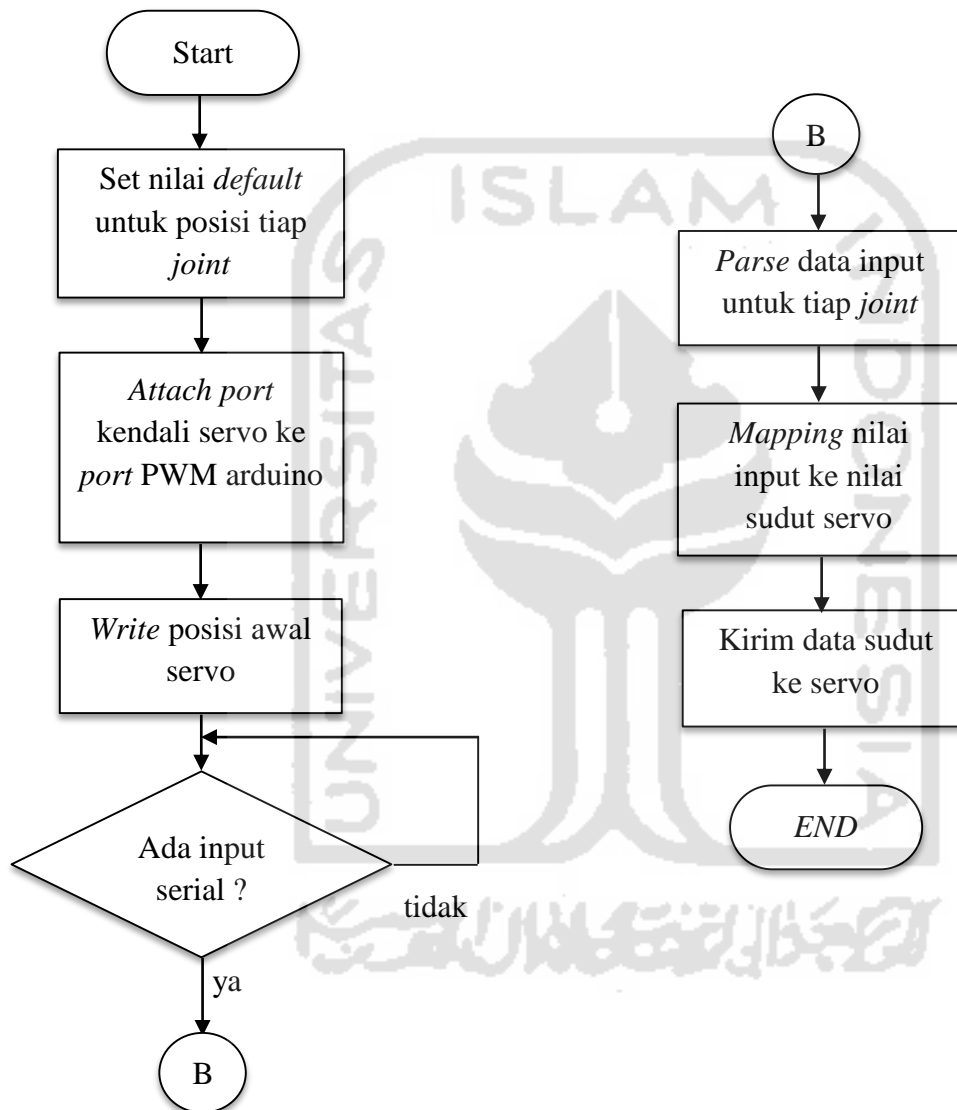
Gambar 3.2 (b) Diagram Alir Aplikasi Android

Pada Gambar 3.2 diagram alir aplikasi android dimulai dengan klik tombol menu utama pada halaman *splashscreen*. Setelah klik tombol menu akan tampil pilihan tiga submenu yaitu *move*, *gallery* dan *options*. Selanjutnya klik tombol *options* akan tampil pilihan untuk mengaktifkan bluetooth. Setelah bluetooth aktif, klik tombol *list* akan tampil pilihan perangkat bluetooth yang akan dipairing. Setelah android *paired* dengan perangkat bluetooth HC-05 aplikasi masuk ke *layout move*. Pada *layout move* terdiri dari metode *forward kinematic* dan *inverse kinematic*. Pada *forward kinematic*, tampilan *interface* terdiri dari masukan nilai  $\theta_1 - \theta_4$  sebagai masukan dari *joint base* hingga *wrist* pada robot serta tombol *run* dan *reset*. Nilai masukan  $\theta_1 - \theta_4$  akan dibandingkan dengan nilai maksimal sudut operasi tiap servo sehingga masukan nilai memenuhi *configuration space* dari robot. Hasil penghitungan yang didapat dibandingkan dengan posisi aktual dari *end-effector* robot.

Pada metode *inverse kinematic*, tampilan *interface* terdiri dari *set point* berupa posisi dan orientasi *end-effector* yang diinginkan yang kemudian diproses menjadi keluaran posisi sudut dari tiap *joint* robot. Sesuai dengan konfigurasi robot, untuk mendapatkan posisi akhir dari *end-effector* diperlukan *setpoint* berupa koordinat  $x$ ,  $y$ , dan  $z$  dari titik akhir *end-effector*. Sedangkan untuk mendapatkan orientasi akhir digunakan nilai sudut *end-effector* terhadap bidang  $x$  dan  $y$ .

### 3.2.2 Diagram Alir Arduino

Diagram alir untuk program arduino dalam mengoperasikan robot wayang dapat dilihat pada Gambar 3.3 sebagai berikut.



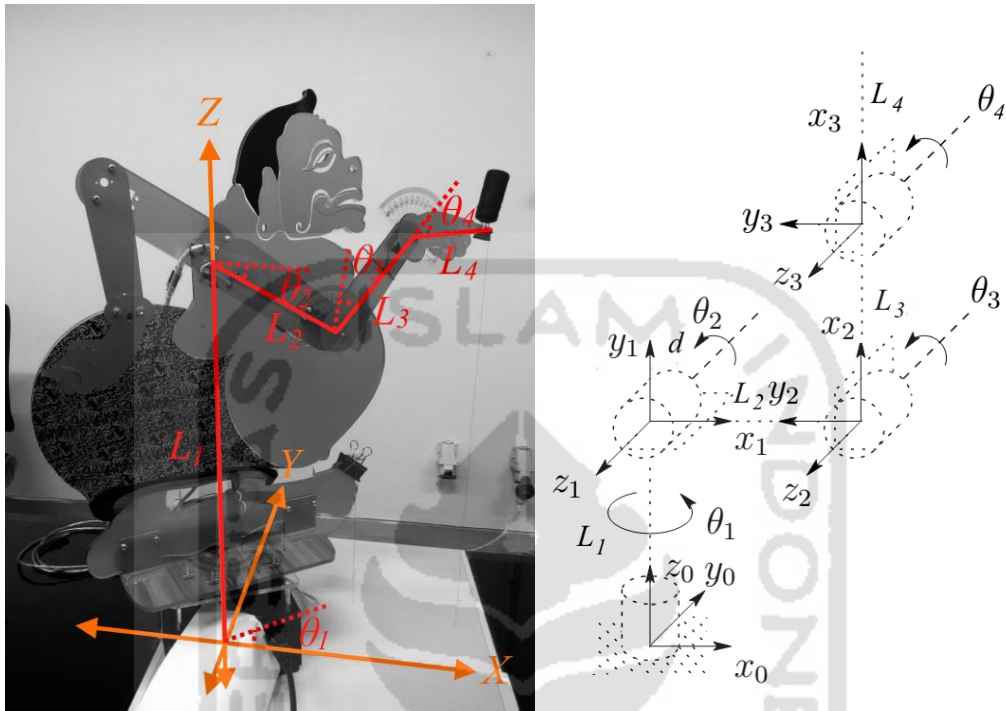
Gambar 3.3 Diagram Alir Program pada Arduino

Program pada arduino berfungsi untuk menerima data dari hasil komputasi kinematik dari android berupa sudut *joint* yang kemudian diolah arduino untuk diteruskan ke servo. Program menggunakan *library varspeedservo.h* untuk mengendalikan servo robot. *Library* ini berfungsi untuk mempermudah memanipulasi servo yang terkoneksi ke arduino, yaitu fungsi pembacaan dan penulisan posisi servo, penyetoran kecepatan putar servo dan pembuatan sekuens pergerakan servo. Posisi *Zero/Default* pada servo perlu diinisialisasi pada pengaturan pin ke servo yang diinginkan untuk menghindari pergerakan servo yang terlalu cepat pada *startup*. Hal ini terjadi karena ketika servo menerima masukan pada *startup* robot pertama kali maka servo akan mengasumsikan posisi servo adalah pada posisi *default* servo (yang tidak sama dengan posisi *zero kinematic chain* robot) sehingga servo akan mengoreksi posisi dengan kecepatan tertinggi.

Program akan selalu mengecek masukan pada *port* serial selama program berjalan. Ketika program menerima masukan pada *port* serial, maka program akan melakukan *parsing* data untuk tiap *joint* robot. Data hasil *parsing* kemudian dibandingkan dengan nilai batas operasional servo, kemudian nilai yang melebihi batas operasional servo akan dibatasi untuk menjaga kondisi mekanik servo.

### 3.3 Kinematic Chain

Kinematic chain dari robot yang digunakan adalah sebagai berikut



Gambar 3.4 Kinematic chain dari robot wayang

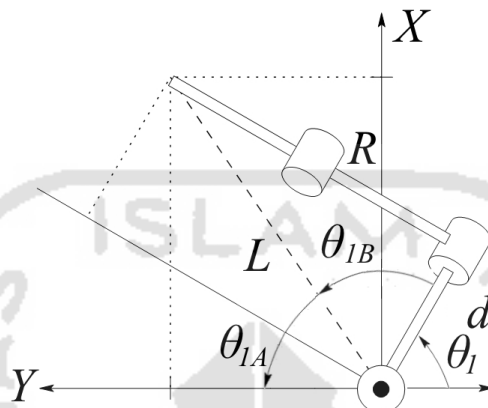
Dari Gambar 3.4 dapat diturunkan parameter Denavit-Hartenberg dari robot yaitu :

Tabel 3.1 Parameter Denavit-Hartenberg Robot Wayang

Joint $i$	$\alpha_i$	$a_i$	$d_i$	$\theta_i$	Coordinate offset
1	$90^\circ$	0	46,5	$\theta_1$	0
2	0	13,5	9,5	$\theta_2$	$-20^\circ$
3	0	13,5	0	$\theta_3$	$110^\circ$
4	0	10,5	0	$\theta_4$	0

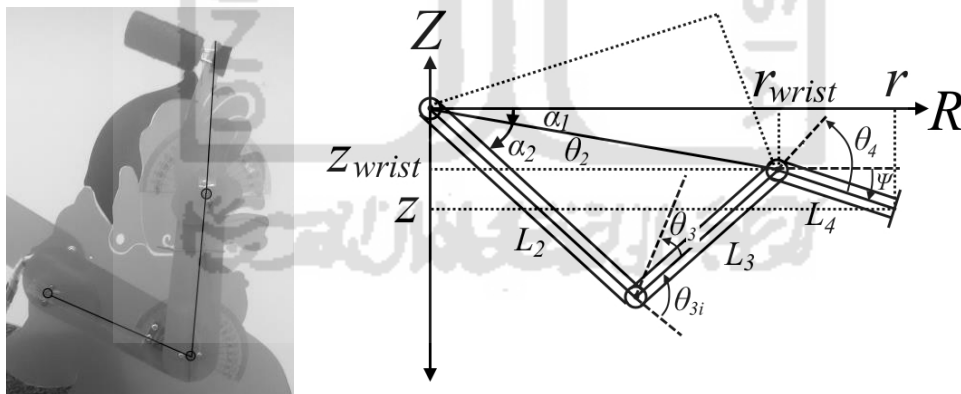
Dari Tabel 3.1 didapat bahwa parameter panjang lengan robot yang digunakan adalah sebesar 46,5 cm, 13,5 cm, 13,5 cm, dan 10,5 cm untuk lengan  $L_1, L_2, L_3$ , dan  $L_4$  dengan *offset* pada *joint* 1 sebesar 9,5 cm. Parameter *frame* orientasi tiap *joint* dapat dilihat pada Gambar 3.4.

Untuk mendapatkan persamaan kinematika dari lengan robot, *kinematic chain* dapat dibagi menjadi 2 bagian. *Kinematic chain* untuk bagian pertama adalah dari *base* hingga *offset* tangan kanan pada akhir *link* 1 dapat dilihat pada Gambar 3.5.



Gambar 3.5 *Kinematic Chain* Pertama Tampak Atas

Bagian kedua yaitu terdiri dari *link* 2, 3, dan 4, sehingga *kinematic chain* kedua dapat dilihat pada Gambar 3.6.



Gambar 3.6 *Kinematic Chain* Bagian Kedua Tampak Samping

### 3.3.1 Persamaan *Forward Kinematic*

Untuk mendapatkan posisi *end effector* persamaan *forward kinematic* perlu dibagi menjadi 2 bagian. Bagian pertama dapat dilihat pada Gambar 3.5. Pada *kinematic chain* ini persamaan kinematika sama dengan suatu lengan robot 3R



linear, namun karena adanya *offset* sudut dan berbedanya titik acuan yang digunakan persamaan *forward kinematic* yang digunakan sedikit berubah. Untuk menghitung panjang lengan robot  $R$  dan  $z$  digunakan trigonometri pada tiap *link* robot

$$r_2 = L_2 \cos(\theta_2 - 20^\circ) \quad (3.1)$$

$$r_3 = L_3 \sin(\theta_2 + \theta_3) \quad (3.2)$$

$$r_4 = L_4 \sin(\theta_2 + \theta_3 + \theta_4) \quad (3.3)$$

$$R = r_2 - r_3 - r_4 \quad (3.4)$$

$$z_2 = L_2 \sin(\theta_2 - 20^\circ) \quad (3.5)$$

$$z_3 = L_3 \cos(\theta_2 + \theta_3) \quad (3.6)$$

$$z_4 = L_4 \cos(\theta_2 + \theta_3 + \theta_4) \quad (3.7)$$

$$z = z_2 + z_3 + z_4 + L_1 \quad (3.8)$$

Kemudian untuk menghitung posisi  $x$  dan  $y$  dapat dilihat *kinematic chain* kedua pada Gambar 3.5. Komponen  $R$  pada penghitungan *kinematic chain* pertama kemudian dicari panjang pada sumbu  $x$  dan  $y$  dengan menggunakan hukum trigonometri dengan memperhatikan adanya *offset* tangan kanan pada robot.

$$L = \sqrt{R^2 + d^2} \quad (3.9)$$

$$\theta_{1B} = \tan^{-1}(r/d) \quad (3.10)$$

$$x = L \sin(\theta_1 + \theta_{1B}) \quad (3.11)$$

$$y = L \cos(\theta_1 + \theta_{1B}) \quad (3.12)$$

$$\Psi = \theta_2 + \theta_3 + \theta_4 + 90^\circ \quad (3.13)$$

### 3.3.2 Persamaan *Inverse Kinematic*

Pada penghitungan *inverse kinematic* robot bagian pertama dapat dilihat pada Gambar 3.5. Pada gambar tersebut dapat dilihat bahwa untuk menghitung sudut  $\theta_{1A}$  dapat digunakan segitiga yang terbentuk dari titik  $x, y$ , dan *base*

$$\theta_{1A} = \tan^{-1} \frac{x}{y} \quad (3.14)$$

Untuk sudut  $\theta_{1B}$  dapat dihitung dengan menggunakan segitiga yang terbentuk dari garis *offset*  $d$  dan garis  $r$

$$L = \sqrt{x^2 + y^2} \quad (3.15)$$

$$\theta_{1B} = \cos^{-1} d/L \quad (3.16)$$

Diketahui bahwa jumlah sudut antara ketiga sudut tersebut adalah  $180^\circ$ , sehingga besar sudut  $\theta_1$  dapat dihitung

$$\theta_1 = \pi - (\theta_{1A} + \theta_{1B}) \quad (3.17)$$

Persamaan *inverse kinematics* untuk *kinematic chain* kedua dapat dihitung sebagai suatu lengan robot 3R linear [3]. Komponen  $L$  pada *kinematic chain* ini merupakan komponen  $L$  yang didapat dari penghitungan *kinematic chain* sebelumnya. Dapat dihitung sudut *joint*  $\theta_2$  dengan menggunakan selisih sudut antara 2 segitiga yang terbentuk antara titik *base*,  $r_{wrist}$ , dan titik  $L_4$ , serta segitiga yang terbentuk dari garis  $L_2, L_3$ , dan garis penghubung frame *base* dan *wrist*.

$$R = \sqrt{L^2 - d^2} \quad (3.18)$$

$$r_{wrist} = R - L_4 \cos \Psi \quad (3.19)$$

$$z_{wrist} = z - (L_4 \sin \Psi + l_0) \quad (3.20)$$

$$\alpha_1 = \tan^{-1} \frac{z_{wrist}}{r_{wrist}} \quad (3.21)$$

$$\alpha_2 = \cos^{-1}\left(\frac{r_{wrist}^2 + z_{wrist}^2 + L_2^2 - L_3^2}{2l_1\sqrt{r_{wrist}^2 + z_{wrist}^2}}\right) \quad (3.22)$$

$$\theta_{2i} = \alpha_1 \pm \alpha_2 \begin{cases} \alpha_1 + \alpha_2 \text{ for } \theta_{3i} < 0 \\ \alpha_1 - \alpha_2 \text{ for } \theta_{3i} > 0 \end{cases} \quad (3.23)$$

Kemudian, karena  $\cos(180 + \theta_3) = -\cos\theta_3$ , maka dengan menggunakan hukum cosinus sudut  $\theta_4$  dapat dihitung. Adanya 2 set solusi akibat hasil penghitungan ini adalah 2 set solusi konfigurasi *joint*, yaitu untuk konfigurasi *elbow up* dan *elbow down*. Namun karena keterbatasan daerah kerja (*configuration space*) pada *joint* kedua maka konfigurasi yang digunakan hanya konfigurasi *elbow down*

$$r_{wrist}^2 + z_{wrist}^2 = L_2^2 + L_3^2 - 2L_2L_3 \cos(180 + \theta_{3i}) \quad (3.24)$$

$$\theta_{3i} = \cos^{-1}\left(\frac{r_{wrist}^2 + z_{wrist}^2 - L_2^2 - L_3^2}{2l_1l_2}\right) \quad (3.25)$$

$$\theta_{2i} = \alpha_1 - \alpha_2 \quad (3.26)$$

Diketahui bahwa besar sudut pada ujung akhir suatu bidang merupakan jumlah dari sudut-sudut komponennya, sehingga jumlah dari ketiga sudut  $\theta$  adalah orientasi dari *link* terakhir frame ini.

$$\Psi = \theta_{2i} + \theta_{3i} + \theta_4 \quad (3.27)$$

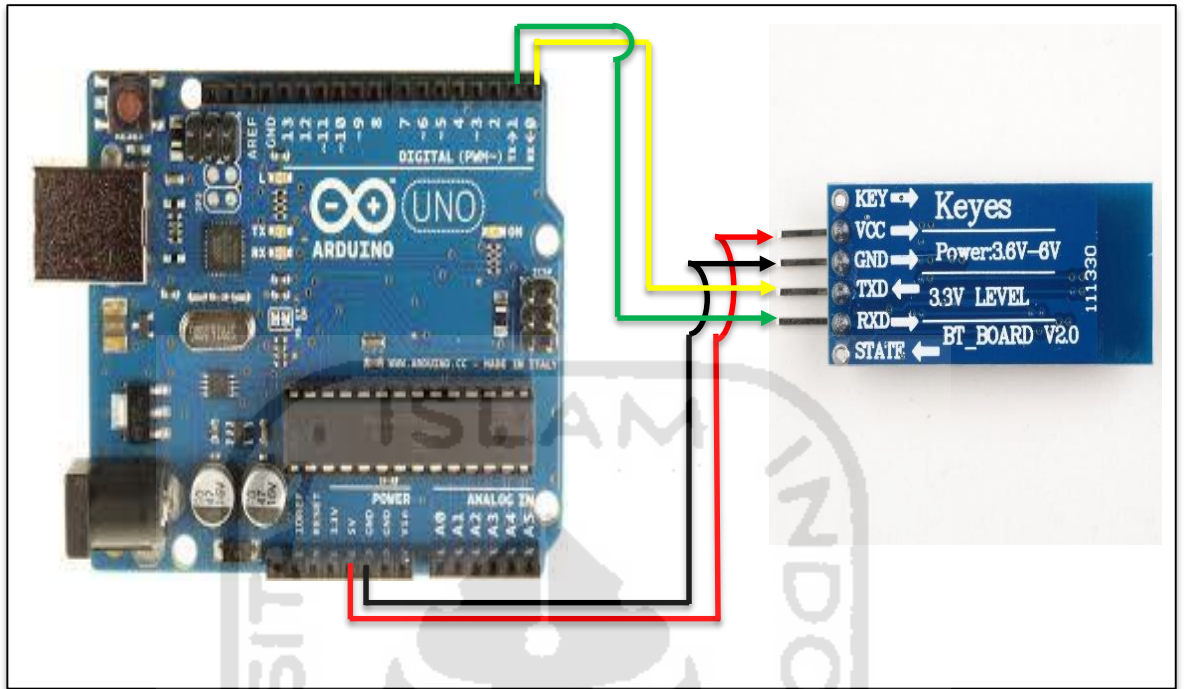
$$\theta_4 = \Psi - \theta_{2i} - \theta_{3i} \quad (3.28)$$

Untuk menyesuaikan orientasi dari *kinematic chain* dasar 3R ke *kinematic chain* maka besar sudut  $\theta_2$  dan  $\theta_3$  perlu ditambahkan *offset* pada *joint* tersebut, sehingga persamaan akhir untuk  $\theta_2$  dan  $\theta_3$  adalah

$$\theta_2 = \theta_{2i} + 20^\circ \quad (3.29)$$

$$\theta_3 = \theta_{3i} - 110^\circ \quad (3.30)$$

### 3.4 Perancangan dan Konfigurasi Modul Bluetooth



Gambar 3.7 Pengkabelan Modul Bluetooth HC-05

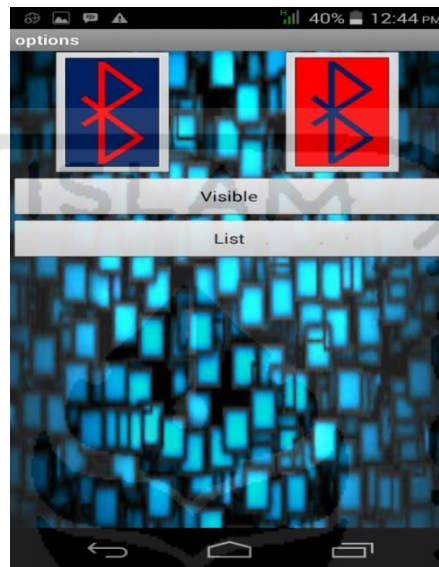
Parameter Gambar 3.7 adalah sebagai berikut :

1. Modul bluetooth pin TX dihubungkan dengan pin RX pada Arduino Uno.
2. Modul bluetooth pin RX dihubungkan dengan pin TX pada Arduino Ino.
3. Modul bluetooth pin VCC dihubungkan dengan pin 5V pada Arduino Uno.
4. Modul bluetooth pin GND dihubungkan dengan pin GND pada Arduino Uno.

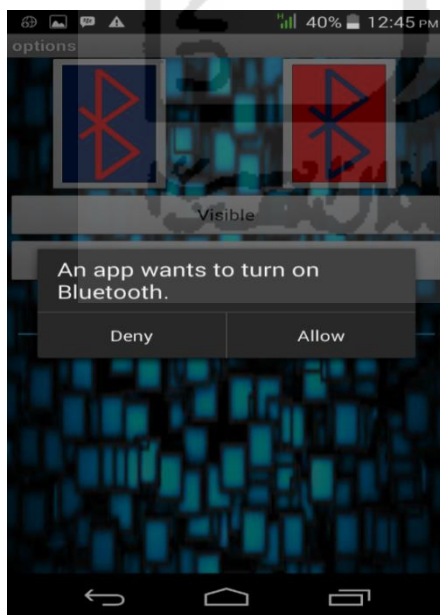
### 3.5 Desain *User Interface* Pada Android

#### 3.5.1 Desain *Paired Bluetooth HC-05*

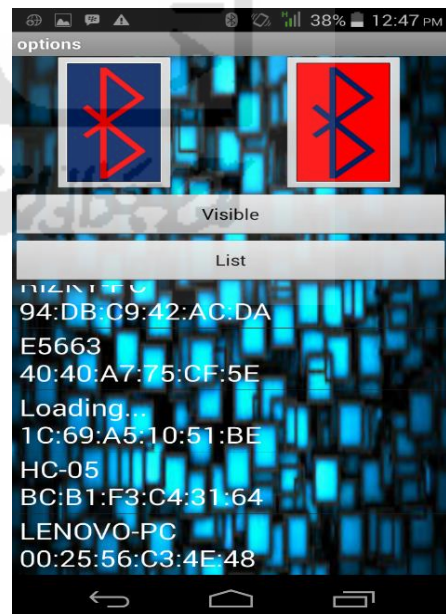
Desain *user interface* untuk *paired* bluetooth HC-05 dapat dilihat pada Gambar 3.8 sebagai berikut.



(a)



(b)



(c)

Gambar 3.8 *Layout* Koneksi Bluetooth HC-05

Dalam aplikasi pengendalian robot wayang menggunakan operasi android melalui komunikasi bluetooth, berbagai fungsi digunakan untuk perancangan komunikasi bluetoothnya. fungsi yang pertama adalah bluetooth adapter. Fungsi ini merupakan fungsi yang diinisialisasikan pertama kali. Bluetooth adapter digunakan untuk melakukan pengecekan apakah perangkat *smartphone* yang akan digunakan telah memiliki komunikasi bluetooth atau tidak.

```
//listing 3
Private BluetoothAdapter btAdapter;
import android.bluetooth.BluetoothAdapter;
if (btAdapter.isEnabled()) {
    btAdapter.disable();
    Statusoff();
} else if (!btAdapter.isEnabled()) {
    Intent openBT = new
    Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
    startActivityForResult(openBT, OPEN_BLUETOOTH);
} else if (btAdapter == null) {
    showToast("Tidak Support Bluetooth");
}
```

Gambar 3.9 Listing Mengaktifkan Bluetooth

Code pada Gambar 3.9 berfungsi untuk mengaktifkan dan mematikan bluetooth pada *smartphone* yang memiliki komunikasi bluetooth yang sesuai dengan kebutuhan aplikasi pengendali robot wayang. Jika *smartphone* tidak memiliki komunikasi bluetooth atau versi bluetooth tidak sesuai yang dibutuhkan aplikasi maka akan muncul peringatan bahwa *smartphone* tidak didukung dengan komunikasi bluetooth. Bluetooth adapter juga dapat mendeteksi apakah bluetooth sedang dalam kondisi aktif atau tidak.

Setelah menangani pengaktifan bluetooth, bluetooth adapter melakukan proses *scanning* dan *pairing*. Listing program untuk proses *scanning* dan *pairing* adalah sebagai berikut.

```

//listing 35
scanBT.setOnClickListener(new OnClickListener() {
    public void onClick(View arg0) {
        btAdapter.startDiscovery();
    }
}
pairedBT.setOnClickListener(new OnClickListener() {
    private string address = "20:14:11:28:24:28";
    public void onClick(View arg0) {
        if (progresPairing.getBondState() == BluetoothDevice.BOND_BONDED){
            unpairDevice(progresPairing);
            USD_1146 = prosesPairing;
        } else {
            pairDevice(progresPairing);
        }
    }
});

```

Gambar 3.10 Listing Paired Bluetooth

Code pada Gambar 3.7 berfungsi untuk melakukan proses *pairing* ketika perangkat dengan alamat “20:14:11:28:24:28” terdeteksi setelah proses *scanning*. Dalam program ini dibatasi hanya perangkat bluetooth SPP (*Serial Port Profile*) dengan alamat “20:14:11:28:24:28” yang dapat berkomunikasi dengan aplikasi kendali robot wayang. Alamat tersebut merupakan alamat modul bluetooth HC-05 yang terhubung dengan arduino.

Code yang kedua adalah code bluetooth *device*. Kelas ini menampung informasi mengenai nama dan alamat perangkat bluetooth yang terdeteksi dan yang dapat berkomunikasi dengan *smartphone*. Selain itu, bluetooth *device* dapat mendeteksi apakah perangkat yang terdeteksi sedang dalam proses *pairing*, sudah *ter-paired* atau belum *ter-paired*. Listing program untuk penggunaan kelas bluetooth *device* ditunjukkan pada Gambar 3.11 sebagai berikut.

```

//listing 6
if (state == BluetoothDevice.BOND_BONDED
    && prevState == BluetoothDevice.BOND_BONDED)
{
    showToast("paired");
    statusPaired.setText("to USD_115114046");
    pairedBT.setText("Unpair");
    connctBT.setEnabled(true);
} else if (State == BluetoothDevice.BOND_NONE && prevState
    == BluetoothDevice.BOND_BONDED)
{ showToast("Unpaired");
  pairedBT.setText("Unpair");
  statusPaired.setText("belum paired");
}
};

```

Gambar 3.11 Listing Bluetooth Device

Code pada Gambar 3.8 menunjukkan fungsi bluetooth *socket*. Bluetooth *socket* berperan dalam membuka jalur yang akan digunakan untuk melakukan komunikasi antara *smartphone* dengan perangkat bluetooth yang ter-*paired*. Listing program untuk penggunaan kelas bluetooth *socket* adalah sebagai berikut.

```

//listing 7
private BluetoothSocket mmSocket;
void closeBT() throws IOException {
    UUID uuid = UUID.fromString("00001101-0000-100-8000-0085F9B34FB");
    mmSocket = USD_1146.createRfcommSocketToServiceRecord(uuid);
    mmSocket.close();
}
Void openBT() throws IOException {
    UUID uuid = UUID.fromString("00001101-0000-100-8000-0085F9B34FB");
    mmSocket = USD_1146.createRfcommSocketToServiceRecord(uuid);
    mmSocket.connect();
    mmOutputStream = mmSocket.getOutputStream();
    statusConnect.setText("Bluetooth opened" + "\n");
}

```

Gambar 3.12 Listing Bluetooth Socket

UUID adalah *Universally Unique Identifier* yang merupakan kode standar untuk komunikasi SPP (*Serial Port Profile*). Jalan komunikasi tidak akan terbuka jika UUID yang digunakan tidak sesuai dengan yang dibutuhkan dapat dilihat pada Gambar 3.12.



Bluetooth *socket* juga mengatur pengiriman dan penerimaan data. Dalam aplikasi pengendalian robot wayang yang dibuat, aplikasi hanya dapat mengirimkan data saja. Untuk setiap pengiriman data, android mewajibkan pengembangan untuk menggunakan *IOException*. Berikut *listing* program pengiriman data aplikasi pengendalian robot wayang.

```
//listing 8
private OutputStream mmOutputStream;
void sendData(String msg0) throws IOException {
    msg0 += "\n";
    mmOutputStream.write(msg0.getBytes());
    StatusConnect.setText("Data Send");
}
```

Gambar 3.13 *Listing* Fungsi *Throw IOException*

Code pada Gambar 3.13 yaitu fungsi *MmOutputStream* merupakan objek yang dibentuk untuk dapat mengirimkan data. Semua program yang digunakan untuk membuka, menutup, mengirimkan dan menerima data pasti dijalankan dengan metode *throws IOException* sesuai dengan aturan *android application programming interface* yang terdapat di website [developer.android.com](http://developer.android.com).

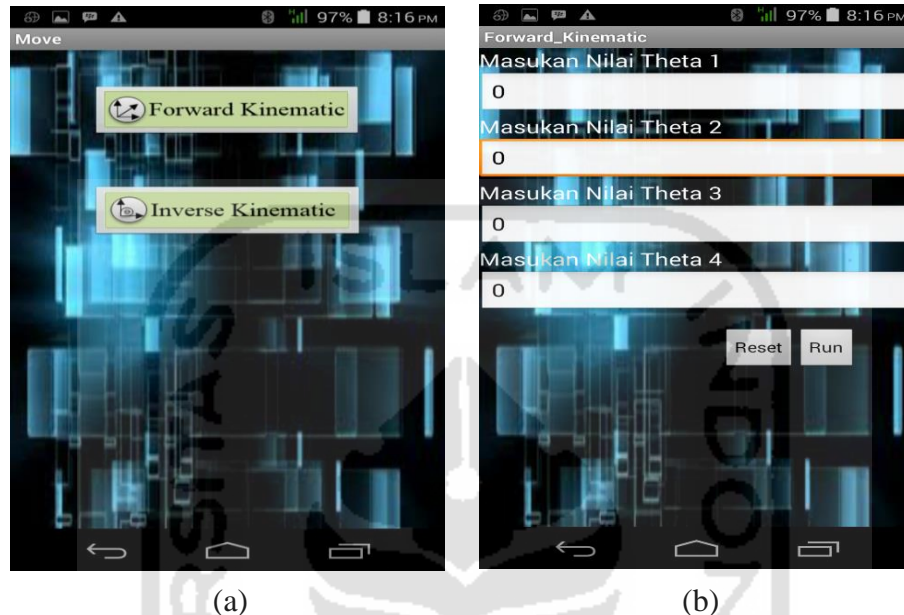
Terakhir dibutuhkan *listing* untuk pindah ke *layout* selanjutnya setelah terjadi *paired* dengan perangkat bluetooth HC-05. *Listing* program berpindah *layout* adalah sebagai berikut.

```
Intent i = new Intent(options.this, move.class);
Toast.makeText(getApplicationContext(), address,
    Toast.LENGTH_SHORT).show();
i.putExtra(EXTRA_DEVICE_ADDRESS, address);
startActivity(i);
}
});
```

Gambar 3.14 *Listing* Pindah *Layout*

### 3.5.2 Desain Mode *Forward Kinematic*

Desain *user interface* untuk mode *forward kinematic* dapat dilihat pada Gambar 3.15 sebagai berikut.



Gambar 3.15 Tampilan *Layout Forward Kinematic*

```
static final String TAG = "bluetooth";
private BluetoothAdapter btAdapter = null;
private BluetoothSocket btSocket = null;
private OutputStream outputStream = null;
private static final UUID MY_UUID =
    UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");
Button Run, reset;
EditText theta1;
EditText theta2;
EditText theta3;
EditText theta4;
private static String address;
```

Gambar 3.16 *Listing inialisasi*

Pertama melakukan inialisasi jalur komunikasi bluetooth untuk pengiriman data masukan nilai *forward kinematic*. Selanjutnya membuka jalur pengiriman data masukan nilai *forward kinematic* melalui bluetooth *socket*. Kemudian melakukan inialisasi *edit text* untuk masukan nilai serta inialisasi untuk tombol *run* dan tombol *reset* dapat dilihat pada Gambar 3.16.

```

theta1 = (EditText) findViewById(R.id.nilaix);
theta2 = (EditText) findViewById(R.id.nilaiy);
theta3 = (EditText) findViewById(R.id.nilaiz);
theta4 = (EditText) findViewById(R.id.nilaiw);
Run = (Button) findViewById(R.id.join2);
Run.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View arg0) {
String nilaix = theta1.getText().toString();
String nilaiy = theta2.getText().toString();
String nilaiz = theta3.getText().toString();
String nilaiw = theta4.getText().toString();
SendData(nilaix + ";" + nilaiy + ";" + nilaiz + ";" + nilaiw);
Toast.makeText(getApplicationContext(),
"Sending Data", Toast.LENGTH_SHORT).show();
}
});

```

Gambar 3.17 listing ID Edit Text dan ID Button

Selanjutnya setelah inisialisasi *edit text*, tentukan ID untuk setiap masing-masing *edit text* agar ketika memberikan masukan nilai sesuai dengan *edit text* yang dituju. Begitu juga untuk ID tombol *run* dan tombol *reset*. Kemudian setelah *edit text* diberi masukan nilai, tombol *run* diklik akan mengirimkan data yang telah diberikan pada *edit text* dengan format pengiriman `SendData(nilaix + ";" + nilaiy + ";" + nilaiz + ";" + nilaiw)`; serta akan tampil text sending data setelah tombol run ditekan seperti yang ditunjukkan pada Gambar 3.17.

```

reset = (Button)findViewById(R.id.reset2);
reset.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View arg0) {
theta1.setText("0");
theta2.setText("0");
theta3.setText("0");
theta4.setText("0");
}
});

```

Gambar 3.18 Listing Fungsi Reset

Code pada Gambar 3.18 adalah fungsi tombol reset. Untuk nilai masukan sebelumnya dapat direset menggunakan tombol *reset* dengan nilai awal yaitu angka 0. Kemudian dilakukan pengiriman data kembali untuk robot wayang sebelum melakukan masukan nilai selanjutnya. Hal itu dilakukan agar nilai yang terkirim ke robot wayang adalah nilai yang diinginkan.

```

private void sendData(String message) {
byte[] msgBuffer = message.getBytes();
Log.d(TAG, "...SEND DATA: " + message + "...");
try{
outStream.write(msgBuffer);
}catch (IOException e) {
String msg = "in onResume() and an exception occurred
during write; "
+ e.getMessage();
if(address.equals("00:00:00:00:00:00"))
msg = msg + ".\n\nUpdate your server address
from 00:00:00:00:00:00 " +
"to the correct address on line 35
in the java code";
msg = msg + ".\n\nCheck that the SPP UUID: " +
MY_UUID.toString() +
"exists on server.\n\n";
errorExit("Fatal Error", msg);
}
}

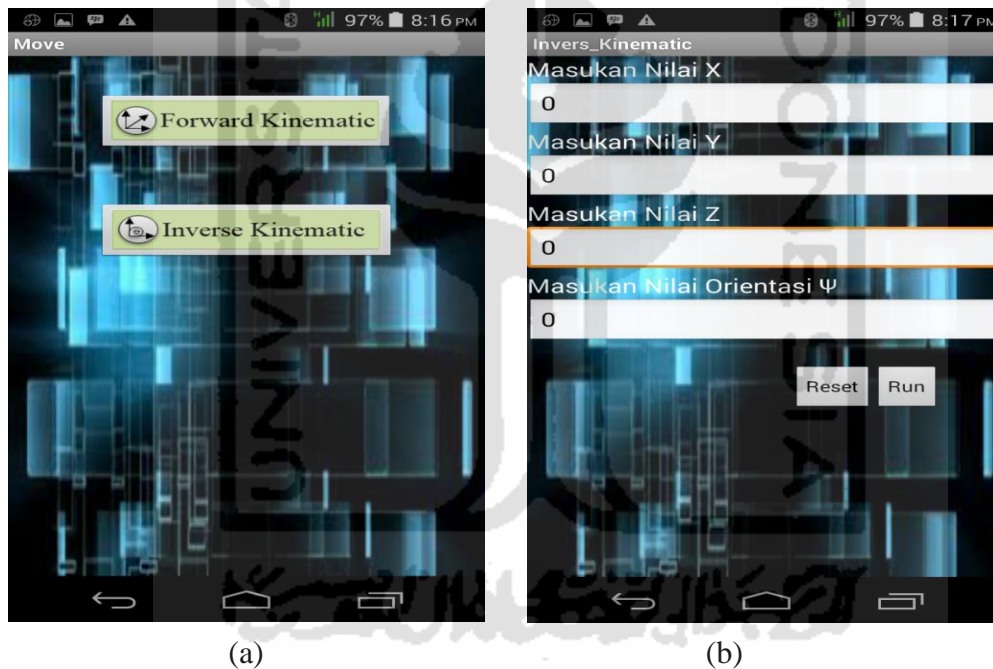
```

Gambar 3.19 Listing Send Data

Code pada gambar 3.19 adalah fungsi pengiriman data ke arduino melalui serial bluetooth. Aplikasi android akan mengecek ID bluetooth HC-05 apakah dalam kondisi *paired* atau tidak. Apabila telah *paired* maka fungsi *IOException* akan mengirim semua data yang diinginkan. Pada aplikasi android studio akan menampilkan info log.cat yang berisi format data yang terkirim.

### 3.5.3 Desain Mode *Inverse Kinematic*

Desain *user interface* untuk mode *inverse kinematic* dapat dilihat pada Gambar 3.20 sebagai berikut.



Gambar 3.20 Tampilan *Layout Inverse Kinematic*

```

private BluetoothAdapter btAdapter = null;
private BluetoothSocket btSocket = null;
private OutputStream outputStream = null;
private static final UUID MY_UUID =
    UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");
Button Run, Reset;
EditText nilai_x;
EditText nilai_y;
EditText nilai_z;
EditText nilai_psi;
private static String address;

```

Gambar 3.21 Listing inisialisasi

Code pada gambar 3.21 berfungsi untuk melakukan inisialisasi jalur komunikasi bluetooth untuk pengiriman data masukan nilai *inverse kinematic*. Selanjutnya membuka jalur pengiriman data masukan nilai *inverse kinematic* melalui bluetooth *socket*. Kemudian melakukan inisialisasi *edit text* untuk masukan nilai serta inisialisasi untuk tombol *run* dan tombol *reset* dapat dilihat pada Gambar 3.20 (b).

```

nilai_x = (EditText) findViewById(R.id.nilaix);
nilai_y = (EditText) findViewById(R.id.nilaiy);
nilai_z = (EditText) findViewById(R.id.nilaiz);
nilai_psi = (EditText) findViewById(R.id.nilaipsi);
Run = (Button) findViewById(R.id.aseem);
Run.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View arg0) {
String nilaix = nilai_x.getText().toString();
String nilaiy = nilai_y.getText().toString();
String nilaiz = nilai_z.getText().toString();
String nilaipsi = nilai_psi.getText().toString();
SendData(nilaix + ";" + nilaiy + ";" + nilaiz + ";" + nilaipsi);
Toast.makeText(getApplicationContext(),
    "Sending Data", Toast.LENGTH_SHORT).show();
}
});

```

Gambar 3.22 Listing ID Edit Text dan ID Button

Selanjutnya setelah inisialisasi *edit text*, tentukan ID untuk setiap masing-masing *edit text* agar ketika memberikan nilai *set point* sesuai dengan *edit text* yang dituju. Begitu juga untuk ID tombol *run* dan tombol *reset* ditentukan. Kemudian setelah *edit text* diberi masukan nilai, tombol *run* diklik akan mengirimkan data yang telah diberikan pada *edit text* dengan format pengiriman `SendData(nilaix + ";" + nilaiy + ";" + nilaiz + ";" + nilaipsi)`; serta akan tampil *text sending data* setelah tombol *run* ditekan seperti yang ditunjukkan pada Gambar 3.22.

```
Reset = (Button) findViewById(R.id.reset);
Reset.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View arg0) {
        nilai_x.setText("0");
        nilai_y.setText("0");
        nilai_z.setText("0");
        nilai_psi.setText("0");
    }
});
```

Gambar 3.23 *Listing Fungsi Reset*

Code pada Gambar 3.23 adalah fungsi tombol reset. Untuk nilai masukan sebelumnya dapat *direset* menggunakan tombol *reset* dengan nilai awal yaitu angka 0. Kemudian dilakukan pengiriman data kembali untuk robot wayang sebelum memberikan nilai *set point* selanjutnya. Hal itu dilakukan agar nilai yang terkirim ke robot wayang adalah nilai yang diinginkan.

```

private void sendData(String message) {
    byte[] msgBuffer = message.getBytes();
    Log.d(TAG, "...SEND DATA: " + message + "...");
    try{
        outputStream.write(msgBuffer);
    } catch (IOException e) {
        String msg = "in onResume() and an exception occurred
during write; "
+ e.getMessage();
        if(address.equals("00:00:00:00:00:00"))
            msg = msg + ".\n\nUpdate your server address
from 00:00:00:00:00:00 " +
"to the correct
in the java code";
            msg = msg + ".\n\nCheck that the SPP UUID: " +
MY_UUID.toString() +
"exists on server.\n\n";
            errorExit("Fatal Error", msg);
    }
}

```

Gambar 3.24 Listing Send Data

Code pada Gambar 3.24 berfungsi untuk mengecek ID bluetooth HC-05 apakah dalam kondisi *paired* atau tidak. Apabila telah *paired* maka fungsi *IOException* akan mengirim semua data yang diinginkan. Pada aplikasi android akan menampilkan info yang berisi format data yang terkirim ke arduino.

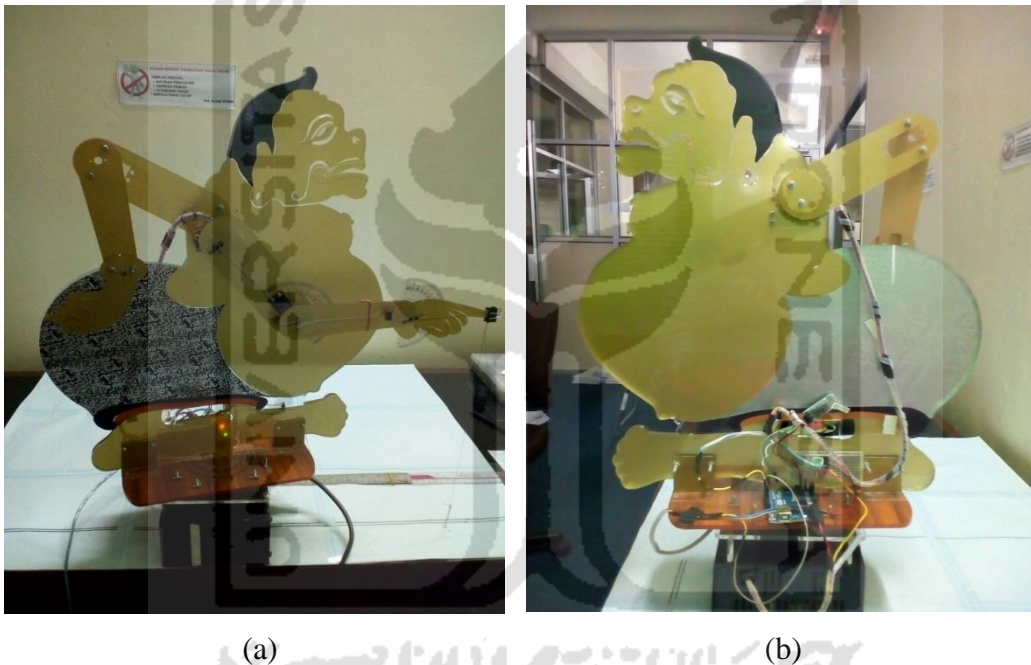


## BAB IV

### PENGUJIAN, ANALISIS, DAN PEMBAHASAN

#### 4.1 Pengujian Komunikasi Bluetooth HC-05

Pada pengujian kali ini, dilakukan pengujian komunikasi bluetooth HC-05 terhadap *smartphone* yang telah terpasang perangkat aplikasi pengendali robot wayang. Pengujian pertama dilakukan proses pengiriman data dengan jarak 1 meter hingga jarak 10 meter dapat dilihat pada Gambar 4.1.



Gambar 4.1 (a) Pengiriman data dengan jarak 1 meter (b) Pengiriman data dengan jarak 10 meter

Setelah dilakukan proses *pairing*, dilakukan pengujian dengan menggunakan metode *forward kinematic*. Pengujian tersebut dilakukan untuk melihat *performa* robot wayang berdasarkan jarak pengiriman data. Dimulai dengan jarak 1 meter hingga jarak maksimal 10 meter. Di setiap jarak pengukuran dilakukan 10 kali percobaan dan setiap percobaannya dilakukan 3 kali

pengulangan. Pengujian pengiriman data pada robot wayang dengan jarak 1 meter hingga 10 meter dilakukan tanpa penghalang.

Dari hasil 3 kali pengulangan tersebut dihitung nilai rata-ratanya untuk memastikan tingkat akurasi dari hasil yang diperoleh. Adapun metode pengukuran dilakukan dengan menggunakan seutas benang yang ditempelkan pada ujung *end-effector* yang diberikan pemberat (jarum) pada ujungnya. Posisi jarum kemudian akan menunjuk satu posisi pada *milimeter block*. Posisi ini menunjukkan koordinat  $x$  dan  $y$  *end-effector*. Posisi  $z$  bidang kartesian dari ujung *end-effector* didapat dengan mengukur panjang benang dari ujung *end-effector* ke dasar bidang dengan menggunakan meteran. Orientasi  $\Psi$  diukur dengan menggunakan busur pada pangkal *end-effector*.

Adapun hasil pengujian untuk pengiriman data ke robot wayang dengan jarak 1 meter ditunjukkan pada Table 4.1 sebagai berikut.

Tabel 4.1 Tabel Pengujian Sistem *Forward Kinematic* dengan Jarak 1 Meter

No.	Input				Hasil Perhitungan				Nilai Rata-Rata Hasil Pengukuran			
	$\theta_1$ ( $^\circ$ )	$\theta_2$ ( $^\circ$ )	$\theta_3$ ( $^\circ$ )	$\theta_4$ ( $^\circ$ )	$x$ (cm)	$y$ (cm)	$z$ (cm)	$\Psi$ ( $^\circ$ )	$x$ (cm)	$y$ (cm)	$z$ (cm)	$\Psi$ ( $^\circ$ )
1	10	5	-50	50	22,97	5,59	63,01	95	21,54	8,33	62,5	94
2	20	10	-40	45	19,53	-3,01	65,99	105	19,33	-4,67	65	103
3	30	15	-30	40	15,57	-1,97	67,83	115	16,43	-3,33	67	115
4	40	20	-20	35	11,83	-2,47	68,66	125	12,53	-3,50	67,8	125
5	50	25	-10	30	8,90	-4,16	68,12	135	10,66	-4,50	67,5	133
6	70	-30	50	-25	10,62	-1,42	59,30	85	10,5	-2	59	85
7	40	-35	40	-20	13,21	-1,31	59,03	75	12,5	-4	59,5	75
8	30	-40	30	-15	16,45	-1,46	57,60	65	15,33	-3,5	58,5	65
9	40	-45	20	-10	19,44	3,91	55,10	55	18,67	1,67	56,4	55
10	50	-50	10	-5	20,57	9,74	51,58	45	21,33	7,67	53	43

Dengan menggunakan metode *Root-Mean-Square Error* (RMSE), yaitu

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\text{Perhitungan}_i - (\text{Nilai Rata - Rata Hasil Pengukuran}_i))^2}{n}} \quad (4.1)$$

Maka didapatkan hasil rata-rata error untuk posisi x, y, z dan orientasi  $\Psi$ . Adapun perhitungan RSME untuk pengujian pada jarak 1 meter adalah sebagai berikut:

$$x_{rmse} = \sqrt{\frac{(22,97 - 21,54)^2 + (19,55 - 19,53)^2 + \dots + (20,57 - 21,33)^2}{10}} \quad (4.2)$$

$$x_{rmse} = 0,967$$

$$y_{rmse} = \sqrt{\frac{(5,59 - 8,33)^2 + (-3,01 + 4,67)^2 + \dots + (9,74 - 7,67)^2}{10}} \quad (4.3)$$

$$y_{rmse} = 1,853$$

$$z_{rmse} = \sqrt{\frac{(63,01 - 62,5)^2 + (65,99 - 65)^2 + \dots + (51,58 - 53)^2}{10}} \quad (4.4)$$

$$z_{rmse} = 0,887$$

$$\Psi_{rmse} = \sqrt{\frac{(95 - 94)^2 + (105 - 103)^2 + \dots + (45 - 43)^2}{10}} \quad (4.5)$$

$$\Psi_{rmse} = 1,14$$

Dari hasil perhitungan nilai RMSE bahwa untuk nilai x, y, z, dan  $\Psi$  didapat sebesar 0,967 cm; 1,853 cm; 0,887 cm; 1,14°.

Selanjutnya pengujian dengan jarak 2 meter. Robot wayang dikondisikan pada posisi awal sebelum dilakukan pengiriman data ke robot wayang. Hal yang sama dilakukan untuk jarak 3 meter hingga 10 meter. Adapun hasil pengujian untuk pengiriman data ke robot wayang dengan jarak 2 meter ditunjukkan pada Table 4.2 sebagai berikut.

Tabel 4.2 Tabel Pengujian Sistem *Forward Kinematic* dengan Jarak 2 Meter

No.	Input				Hasil Perhitungan				Nilai Rata-Rata Hasil Pengukuran			
	$\theta_1$ ( $^\circ$ )	$\theta_2$ ( $^\circ$ )	$\theta_3$ ( $^\circ$ )	$\theta_4$ ( $^\circ$ )	x (cm)	y (cm)	z (cm)	$\Psi$ ( $^\circ$ )	x (cm)	y (cm)	z (cm)	$\Psi$ ( $^\circ$ )
1	10	5	-50	50	22,97	5,59	63,01	95	21,54	8,33	62,5	94
2	20	10	-40	45	19,53	-3,01	65,99	105	19,33	-4,67	65	103
3	30	15	-30	40	15,57	-1,97	67,83	115	16,43	-3,33	67	115
4	40	20	-20	35	11,83	-2,47	68,66	125	12,53	-3,50	67,8	125
5	50	25	-10	30	8,90	-4,16	68,12	135	10,66	-4,50	67,5	133
6	70	-30	50	-25	10,62	-1,42	59,30	85	10,5	-2	59	85
7	40	-35	40	-20	13,21	-1,31	59,03	75	12,5	-4	59,5	75
8	30	-40	30	-15	16,45	-1,46	57,60	65	15,33	-3,5	58,5	65
9	40	-45	20	-10	19,44	3,91	55,10	55	18,67	1,67	56,4	55
10	50	-50	10	-5	20,57	9,74	51,58	45	21,33	7,67	53	43

$$x_{rmse} = \sqrt{\frac{(22,97 - 21,54)^2 + (19,53 - 19,53)^2 + \dots + (20,57 - 21,33)^2}{10}} \quad (4.6)$$

$$x_{rmse} = 0,967$$

$$y_{rmse} = \sqrt{\frac{(5,59 - 8,33)^2 + (-3,01 + 4,67)^2 + \dots + (9,74 - 7,67)^2}{10}} \quad (4.7)$$

$$y_{rmse} = 1,853$$

$$z_{rmse} = \sqrt{\frac{(63,01 - 62,5)^2 + (65,99 - 65)^2 + \dots + (51,58 - 53)^2}{10}} \quad (4.8)$$

$$z_{rmse} = 0,887$$

$$\Psi_{rmse} = \sqrt{\frac{(95 - 94)^2 + (105 - 103)^2 + \dots + (45 - 43)^2}{10}} \quad (4.9)$$

$$\Psi_{rmse} = 1,14$$

Dari hasil perhitungan nilai RMSE bahwa untuk nilai  $x$ ,  $y$ ,  $z$  dan  $\Psi$  didapat sebesar 0,93 cm; 2,02 cm; 0,78 cm; 1,14°. Dari hasil perhitungan nilai RMSE menunjukkan bahwa tidak ada perubahan nilai yang terjadi pada sumbu  $x$ ,  $y$ ,  $z$  dan nilai orientasi *end-effector*  $\Psi$ . Dikarenakan hasil pengujian dengan jarak 2 meter dan 1 meter menghasilkan nilai yang sama dapat dilihat pada Tabel 4.1 dan Tabel 4.2.

Pengujian selanjutnya dilakukan dengan jarak 3 meter hingga 10 meter. Namun hasil percobaan yang didapatkan sama dengan hasil percobaan dengan jarak 1 meter. Hal tersebut menunjukkan bahwa dalam pengujian yang dilakukan dengan merubah jarak pengiriman data dari satu titik ke titik lain tidak mempengaruhi hasil percobaan dan tingkat presisi.

Adapun data percobaan dengan jarak 3 meter hingga 10 meter ditunjukkan pada Tabel 4.3-4.10 sebagai berikut.

Tabel 4.3 Tabel Pengujian Sistem *Forward Kinematic* dengan Jarak 3 Meter

No.	Input				Hasil Perhitungan				Nilai Rata-Rata Hasil Pengukuran			
	$\theta_1$ (°)	$\theta_2$ (°)	$\theta_3$ (°)	$\theta_4$ (°)	$x$ (cm)	$y$ (cm)	$z$ (cm)	$\Psi$ (°)	$x$ (cm)	$y$ (cm)	$z$ (cm)	$\Psi$ (°)
1	10	5	-50	50	22,97	5,59	63,01	95	21,54	8,33	62,5	94
2	20	10	-40	45	19,53	-3,01	65,99	105	19,33	-4,67	65	103
3	30	15	-30	40	15,57	-1,97	67,83	115	16,43	-3,33	67	115
4	40	20	-20	35	11,83	-2,47	68,66	125	12,53	-3,50	67,8	125
5	50	25	-10	30	8,90	-4,16	68,12	135	10,66	-4,50	67,5	133
6	70	-30	50	-25	10,62	-1,42	59,30	85	10,5	-2	59	85
7	40	-35	40	-20	13,21	-1,31	59,03	75	12,5	-4	59,5	75
8	30	-40	30	-15	16,45	-1,46	57,60	65	15,33	-3,5	58,5	65
9	40	-45	20	-10	19,44	3,91	55,10	55	18,67	1,67	56,4	55
10	50	-50	10	-5	20,57	9,74	51,58	45	21,33	7,67	53	43

Tabel 4.4 Tabel Pengujian Sistem *Forward Kinematic* dengan Jarak 4 Meter

No.	Input				Hasil Perhitungan				Nilai Rata-Rata Hasil Pengukuran			
	$\theta_1$ ( $^\circ$ )	$\theta_2$ ( $^\circ$ )	$\theta_3$ ( $^\circ$ )	$\theta_4$ ( $^\circ$ )	x (cm)	y (cm)	z (cm)	$\Psi$ ( $^\circ$ )	x (cm)	y (cm)	z (cm)	$\Psi$ ( $^\circ$ )
1	10	5	-50	50	22,97	5,59	63,01	95	21,54	8,33	62,5	94
2	20	10	-40	45	19,53	-3,01	65,99	105	19,33	-4,67	65	103
3	30	15	-30	40	15,57	-1,97	67,83	115	16,43	-3,33	67	115
4	40	20	-20	35	11,83	-2,47	68,66	125	12,53	-3,50	67,8	125
5	50	25	-10	30	8,90	-4,16	68,12	135	10,66	-4,50	67,5	133
6	70	-30	50	-25	10,62	-1,42	59,30	85	10,5	-2	59	85
7	40	-35	40	-20	13,21	-1,31	59,03	75	12,5	-4	59,5	75
8	30	-40	30	-15	16,45	-1,46	57,60	65	15,33	-3,5	58,5	65
9	40	-45	20	-10	19,44	3,91	55,10	55	18,67	1,67	56,4	55
10	50	-50	10	-5	20,57	9,74	51,58	45	21,33	7,67	53	43

Tabel 4.5 Tabel Pengujian Sistem *Forward Kinematic* dengan Jarak 5 Meter

No.	Input				Hasil Perhitungan				Nilai Rata-Rata Hasil Pengukuran			
	$\theta_1$ ( $^\circ$ )	$\theta_2$ ( $^\circ$ )	$\theta_3$ ( $^\circ$ )	$\theta_4$ ( $^\circ$ )	x (cm)	y (cm)	z (cm)	$\Psi$ ( $^\circ$ )	x (cm)	y (cm)	z (cm)	$\Psi$ ( $^\circ$ )
1	10	5	-50	50	22,97	5,59	63,01	95	21,54	8,33	62,5	94
2	20	10	-40	45	19,53	-3,01	65,99	105	19,33	-4,67	65	103
3	30	15	-30	40	15,57	-1,97	67,83	115	16,43	-3,33	67	115
4	40	20	-20	35	11,83	-2,47	68,66	125	12,53	-3,50	67,8	125
5	50	25	-10	30	8,90	-4,16	68,12	135	10,66	-4,50	67,5	133
6	70	-30	50	-25	10,62	-1,42	59,30	85	10,5	-2	59	85
7	40	-35	40	-20	13,21	-1,31	59,03	75	12,5	-4	59,5	75
8	30	-40	30	-15	16,45	-1,46	57,60	65	15,33	-3,5	58,5	65
9	40	-45	20	-10	19,44	3,91	55,10	55	18,67	1,67	56,4	55
10	50	-50	10	-5	20,57	9,74	51,58	45	21,33	7,67	53	43

Tabel 4.6 Tabel Pengujian Sistem *Forward Kinematic* dengan Jarak 6 Meter

No.	Input				Hasil Perhitungan				Nilai Rata-Rata Hasil Pengukuran			
	$\theta_1$ ( $^\circ$ )	$\theta_2$ ( $^\circ$ )	$\theta_3$ ( $^\circ$ )	$\theta_4$ ( $^\circ$ )	x (cm)	y (cm)	z (cm)	$\Psi$ ( $^\circ$ )	x (cm)	y (cm)	z (cm)	$\Psi$ ( $^\circ$ )
1	10	5	-50	50	22,97	5,59	63,01	95	21,54	8,33	62,5	94
2	20	10	-40	45	19,53	-3,01	65,99	105	19,33	-4,67	65	103
3	30	15	-30	40	15,57	-1,97	67,83	115	16,43	-3,33	67	115
4	40	20	-20	35	11,83	-2,47	68,66	125	12,53	-3,50	67,8	125
5	50	25	-10	30	8,90	-4,16	68,12	135	10,66	-4,50	67,5	133
6	70	-30	50	-25	10,62	-1,42	59,30	85	10,5	-2	59	85
7	40	-35	40	-20	13,21	-1,31	59,03	75	12,5	-4	59,5	75
8	30	-40	30	-15	16,45	-1,46	57,60	65	15,33	-3,5	58,5	65
9	40	-45	20	-10	19,44	3,91	55,10	55	18,67	1,67	56,4	55
10	50	-50	10	-5	20,57	9,74	51,58	45	21,33	7,67	53	43

Tabel 4.7 Tabel Pengujian Sistem *Forward Kinematic* dengan Jarak 7 Meter

No.	Input				Hasil Perhitungan				Nilai Rata-Rata Hasil Pengukuran			
	$\theta_1$ ( $^\circ$ )	$\theta_2$ ( $^\circ$ )	$\theta_3$ ( $^\circ$ )	$\theta_4$ ( $^\circ$ )	x (cm)	y (cm)	z (cm)	$\Psi$ ( $^\circ$ )	x (cm)	y (cm)	z (cm)	$\Psi$ ( $^\circ$ )
1	10	5	-50	50	22,97	5,59	63,01	95	21,54	8,33	62,5	94
2	20	10	-40	45	19,53	-3,01	65,99	105	19,33	-4,67	65	103
3	30	15	-30	40	15,57	-1,97	67,83	115	16,43	-3,33	67	115
4	40	20	-20	35	11,83	-2,47	68,66	125	12,53	-3,50	67,8	125
5	50	25	-10	30	8,90	-4,16	68,12	135	10,66	-4,50	67,5	133
6	70	-30	50	-25	10,62	-1,42	59,30	85	10,5	-2	59	85
7	40	-35	40	-20	13,21	-1,31	59,03	75	12,5	-4	59,5	75
8	30	-40	30	-15	16,45	-1,46	57,60	65	15,33	-3,5	58,5	65
9	40	-45	20	-10	19,44	3,91	55,10	55	18,67	1,67	56,4	55
10	50	-50	10	-5	20,57	9,74	51,58	45	21,33	7,67	53	43

Tabel 4.8 Tabel Pengujian Sistem *Forward Kinematic* dengan Jarak 8 Meter

No.	Input				Hasil Perhitungan				Nilai Rata-Rata Hasil Pengukuran			
	$\theta_1$ ( $^\circ$ )	$\theta_2$ ( $^\circ$ )	$\theta_3$ ( $^\circ$ )	$\theta_4$ ( $^\circ$ )	x (cm)	y (cm)	z (cm)	$\Psi$ ( $^\circ$ )	x (cm)	y (cm)	z (cm)	$\Psi$ ( $^\circ$ )
1	10	5	-50	50	22,97	5,59	63,01	95	21,54	8,33	62,5	94
2	20	10	-40	45	19,53	-3,01	65,99	105	19,33	-4,67	65	103
3	30	15	-30	40	15,57	-1,97	67,83	115	16,43	-3,33	67	115
4	40	20	-20	35	11,83	-2,47	68,66	125	12,53	-3,50	67,8	125
5	50	25	-10	30	8,90	-4,16	68,12	135	10,66	-4,50	67,5	133
6	70	-30	50	-25	10,62	-1,42	59,30	85	10,5	-2	59	85
7	40	-35	40	-20	13,21	-1,31	59,03	75	12,5	-4	59,5	75
8	30	-40	30	-15	16,45	-1,46	57,60	65	15,33	-3,5	58,5	65
9	40	-45	20	-10	19,44	3,91	55,10	55	18,67	1,67	56,4	55
10	50	-50	10	-5	20,57	9,74	51,58	45	21,33	7,67	53	43

Tabel 4.9 Tabel Pengujian Sistem *Forward Kinematic* dengan Jarak 9 Meter

No.	Input				Hasil Perhitungan				Nilai Rata-Rata Hasil Pengukuran			
	$\theta_1$ ( $^\circ$ )	$\theta_2$ ( $^\circ$ )	$\theta_3$ ( $^\circ$ )	$\theta_4$ ( $^\circ$ )	x (cm)	y (cm)	z (cm)	$\Psi$ ( $^\circ$ )	x (cm)	Y (cm)	z (cm)	$\Psi$ ( $^\circ$ )
1	10	5	-50	50	22,97	5,59	63,01	95	21,54	8,33	62,5	94
2	20	10	-40	45	19,53	-3,01	65,99	105	19,33	-4,67	65	103
3	30	15	-30	40	15,57	-1,97	67,83	115	16,43	-3,33	67	115
4	40	20	-20	35	11,83	-2,47	68,66	125	12,53	-3,50	67,8	125
5	50	25	-10	30	8,90	-4,16	68,12	135	10,66	-4,50	67,5	133
6	70	-30	50	-25	10,62	-1,42	59,30	85	10,5	-2	59	85
7	40	-35	40	-20	13,21	-1,31	59,03	75	12,5	-4	59,5	75
8	30	-40	30	-15	16,45	-1,46	57,60	65	15,33	-3,5	58,5	65
9	40	-45	20	-10	19,44	3,91	55,10	55	18,67	1,67	56,4	55
10	50	-50	10	-5	20,57	9,74	51,58	45	21,33	7,67	53	43



Tabel 4.10 Tabel Pengujian Sistem *Forward Kinematic* dengan Jarak 10 Meter

No.	Input				Hasil Perhitungan				Nilai Rata-Rata Hasil Pengukuran			
	$\theta_1$ ( $^\circ$ )	$\theta_2$ ( $^\circ$ )	$\theta_3$ ( $^\circ$ )	$\theta_4$ ( $^\circ$ )	x (cm)	Y (cm)	z (cm)	$\Psi$ ( $^\circ$ )	x (cm)	y (cm)	z (cm)	$\Psi$ ( $^\circ$ )
1	10	5	-50	50	22,97	5,59	63,01	95	21,54	8,33	62,5	94
2	20	10	-40	45	19,53	-3,01	65,99	105	19,33	-4,67	65	103
3	30	15	-30	40	15,57	-1,97	67,83	115	16,43	-3,33	67	115
4	40	20	-20	35	11,83	-2,47	68,66	125	12,53	-3,50	67,8	125
5	50	25	-10	30	8,90	-4,16	68,12	135	10,66	-4,50	67,5	133
6	70	-30	50	-25	10,62	-1,42	59,30	85	10,5	-2	59	85
7	40	-35	40	-20	13,21	-1,31	59,03	75	12,5	-4	59,5	75
8	30	-40	30	-15	16,45	-1,46	57,60	65	15,33	-3,5	58,5	65
9	40	-45	20	-10	19,44	3,91	55,10	55	18,67	1,67	56,4	55
10	50	-50	10	-5	20,57	9,74	51,58	45	21,33	7,67	53	43

Tabel 4.11 Hasil Perhitungan RMSE Pada Pengujian *Forward Kinematic*

Jarak	x RMSE	y RMSE	Z RMSE	$\Psi$ RMSE
1 meter	0,967 cm	1,853 cm	0,887 cm	1,14 $^\circ$
2 meter	0,967 cm	1,853 cm	0,887 cm	1,14 $^\circ$
3 meter	0,967 cm	1,853 cm	0,887 cm	1,14 $^\circ$
4 meter	0,967 cm	1,853 cm	0,887 cm	1,14 $^\circ$
5 meter	0,967 cm	1,853 cm	0,887 cm	1,14 $^\circ$
6 meter	0,967 cm	1,853 cm	0,887 cm	1,14 $^\circ$
7 meter	0,967 cm	1,853 cm	0,887 cm	1,14 $^\circ$
8 meter	0,967 cm	1,853 cm	0,887 cm	1,14 $^\circ$
9 meter	0,967 cm	1,853 cm	0,887 cm	1,14 $^\circ$
10 meter	0,967 cm	1,853 cm	0,887 cm	1,14 $^\circ$

Dari hasil perhitungan RMSE menunjukkan bahwa pada jarak 1 meter hingga 10 meter memiliki nilai RMSE yang sama dapat dilihat pada Tabel 4.11. Hal tersebut dikarenakan saat melakukan percobaan dengan nilai masukan yang sama untuk jarak 1 meter hingga 10 meter, posisi *end-effector* jatuh pada posisi

yang sama. Dari penjelasan tersebut bahwa data yang dikirim aplikasi android dalam metode *forward kinematic* menuju arduino pada robot wayang tidak mengalami perubahan meskipun jarak pengiriman berbeda-beda.

Selanjutnya pengujian yang kedua dilakukan dengan metode *inverse kinematic* dengan jarak 1 meter hingga 10 meter (tanpa penghalang). Di setiap jarak pengukuran dilakukan 10 kali percobaan dan setiap kali percobaan dilakukan 3 kali pengulangan.

Adapun hasil pengujian untuk pengiriman data ke robot wayang dengan jarak 1 meter ditunjukkan pada Table 4.12 sebagai berikut.

Tabel 4.12 Tabel Pengujian Sistem *Inverse Kinematic* dengan Jarak 1 Meter

No	Set Point				Nilai Rata-Rata Hasil Pengukuran			
	x (cm)	y (cm)	z (cm)	$\Psi$ (°)	x (cm)	y (cm)	Z (cm)	$\Psi$ (°)
1	15	4	70	90	18,17	5,47	65,77	87,5
2	13	23	60	20	16,17	22,5	57,87	21
3	21	11	61	50	21,07	10,57	60,85	50
4	9	1	64	70	11,27	3,87	65,01	70
5	9	-4	55	170	9,67	-4,2	57,55	165
6	16	12	63	80	15,33	14,67	58,67	80
7	25	11	40	90	21,57	5,47	39,85	90
8	22	19	37	30	21,17	13,55	39,15	26
9	20	-22	60	10	19,5	-21,55	59,5	10
10	21	8	53	70	24,22	9,55	57,5	70

$$x_{rmse} = \sqrt{\frac{(15 - 18,17)^2 + (13 - 16,17)^2 + \dots + (21 - 24,22)^2}{10}} \quad (4.10)$$

$$x_{rmse} = 2,221$$

$$y_{rmse} = \sqrt{\frac{(4 - 5,46)^2 + (23 - 22,51)^2 + \dots + (8 - 9,53)^2}{10}} \quad (4.11)$$

$$y_{rmse} = 2,829$$

$$z_{rmse} = \sqrt{\frac{(70 - 65,77)^2 + (60 - 57,87)^2 + \dots + (53 - 57,5)^2}{10}} \quad (4.12)$$

$$z_{rmse} = 2,359$$

$$\Psi_{rmse} = \sqrt{\frac{(90 - 87,5)^2 + (20 - 21)^2 + \dots + (70 - 70)^2}{10}} \quad (4.13)$$

$$\Psi_{rmse} = 2,19$$

Dari hasil perhitungan nilai RMSE bahwa untuk nilai x, y, z, dan  $\Psi$  didapat sebesar 2,221 cm; 2,829 cm; 2,359 cm; 2,19°.

Selanjutnya pengujian dengan jarak 2 meter. Robot wayang dikondisikan pada posisi awal sebelum dilakukan pengiriman data ke robot wayang. Hal yang sama dilakukan untuk pengiriman data dengan jarak 3 meter hingga 10 meter. Adapun hasil pengujian dengan jarak 2 meter ditunjukkan pada Table 4.13.

Tabel 4.13 Tabel Pengujian Sistem *Inverse Kinematic* dengan Jarak 2 Meter

No	Set Point				Nilai Rata-Rata Hasil Pengukuran			
	x (cm)	y (cm)	z (cm)	$\Psi$ (°)	x (cm)	y (cm)	Z (cm)	$\Psi$ (°)
1	15	4	70	90	18,17	5,47	65,77	87,5
2	13	23	60	20	16,17	22,5	57,87	21
3	21	11	61	50	21,07	10,57	60,85	50
4	9	1	64	70	11,27	3,87	65,01	70
5	9	-4	55	170	9,67	-4,2	57,55	165
6	16	12	63	80	15,33	14,67	58,67	80
7	25	11	40	90	21,57	5,47	39,85	90
8	22	19	37	30	21,17	13,55	39,15	26
9	20	-22	60	10	19,5	-21,55	59,5	10
10	21	8	53	70	24,22	9,55	57,5	70

$$x_{rmse} = \sqrt{\frac{(15 - 18,17)^2 + (13 - 16,17)^2 + \dots + (21 - 24,22)^2}{10}} \quad (4.14)$$

$$x_{rmse} = 2,221$$

$$y_{rmse} = \sqrt{\frac{(4 - 5,46)^2 + (23 - 22,51)^2 + \dots + (8 - 9,53)^2}{10}} \quad (4.15)$$

$$y_{rmse} = 2,829$$

$$z_{rmse} = \sqrt{\frac{(70 - 65,77)^2 + (60 - 57,87)^2 + \dots + (53 - 57,5)^2}{10}} \quad (4.16)$$

$$z_{rmse} = 2,359$$

$$\Psi_{rmse} = \sqrt{\frac{(90 - 87,5)^2 + (20 - 21)^2 + \dots + (70 - 70)^2}{10}} \quad (4.17)$$

$$\Psi_{rmse} = 2,19$$

Dari hasil perhitungan nilai RMSE bahwa untuk nilai x, y, z, dan  $\Psi$  didapat sebesar 2,221 cm; 2,829 cm; 2,359 cm; 2,19°. Dari hasil perhitungan nilai RMSE menunjukkan bahwa tidak ada perubahan nilai yang terjadi pada sumbu x, y, z dan nilai orientasi *end effector*  $\Psi$ . Dikarenakan hasil pengujian dengan jarak 2 meter dan 1 meter menghasilkan nilai yang sama dapat dilihat pada tabel 4.11 dan tabel 4.12.

Pengujian selanjutnya dilakukan dengan jarak 3 meter hingga 15 meter. Namun hasil percobaan yang didapatkan sama dengan hasil percobaan dengan jarak 1 meter. Hal tersebut menunjukkan bahwa dalam pengujian yang dilakukan dengan merubah jarak pengiriman data dari satu titik ke titik lain tidak mempengaruhi hasil percobaan dan tingkat presisi.

Adapun hasil pengujian untuk pengiriman data ke robot wayang dengan jarak 3 meter hingga 10 meter ditunjukkan pada Tabel 4.14-4.21 sebagai berikut.

Tabel 4.14 Tabel Pengujian Sistem *Inverse Kinematic* dengan Jarak 3 Meter

No	Set Point				Nilai Rata-Rata Hasil Pengukuran			
	x (cm)	y (cm)	z (cm)	$\Psi$ (°)	x (cm)	y (cm)	Z (cm)	$\Psi$ (°)
1	15	4	70	90	18,17	5,47	65,77	87,5
2	13	23	60	20	16,17	22,5	57,87	21
3	21	11	61	50	21,07	10,57	60,85	50
4	9	1	64	70	11,27	3,87	65,01	70
5	9	-4	55	170	9,67	-4,2	57,55	165
6	16	12	63	80	15,33	14,67	58,67	80
7	25	11	40	90	21,57	5,47	39,85	90
8	22	19	37	30	21,17	13,55	39,15	26
9	20	-22	60	10	19,5	-21,55	59,5	10
10	21	8	53	70	24,22	9,55	57,5	70

Tabel 4.15 Tabel Pengujian Sistem *Inverse Kinematic* dengan Jarak 4 Meter

No	Set Point				Nilai Rata-Rata Hasil Pengukuran			
	x (cm)	y (cm)	z (cm)	$\Psi$ (°)	x (cm)	y (cm)	Z (cm)	$\Psi$ (°)
1	15	4	70	90	18,17	5,47	65,77	87,5
2	13	23	60	20	16,17	22,5	57,87	21
3	21	11	61	50	21,07	10,57	60,85	50
4	9	1	64	70	11,27	3,87	65,01	70
5	9	-4	55	170	9,67	-4,2	57,55	165
6	16	12	63	80	15,33	14,67	58,67	80
7	25	11	40	90	21,57	5,47	39,85	90
8	22	19	37	30	21,17	13,55	39,15	26
9	20	-22	60	10	19,5	-21,55	59,5	10
10	21	8	53	70	24,22	9,55	57,5	70

Tabel 4.16 Tabel Pengujian Sistem *Inverse Kinematic* dengan Jarak 5 Meter

No	Set Point				Nilai Rata-Rata Hasil Pengukuran			
	x (cm)	y (cm)	z (cm)	$\Psi$ (°)	x (cm)	y (cm)	Z (cm)	$\Psi$ (°)
1	15	4	70	90	18,17	5,47	65,77	87,5
2	13	23	60	20	16,17	22,5	57,87	21
3	21	11	61	50	21,07	10,57	60,85	50
4	9	1	64	70	11,27	3,87	65,01	70
5	9	-4	55	170	9,67	-4,2	57,55	165
6	16	12	63	80	15,33	14,67	58,67	80
7	25	11	40	90	21,57	5,47	39,85	90
8	22	19	37	30	21,17	13,55	39,15	26
9	20	-22	60	10	19,5	-21,55	59,5	10
10	21	8	53	70	24,22	9,55	57,5	70

Tabel 4.17 Tabel Pengujian Sistem *Inverse Kinematic* dengan Jarak 6 Meter

No	Set Point				Nilai Rata-Rata Hasil Pengukuran			
	x (cm)	y (cm)	z (cm)	$\Psi$ (°)	x (cm)	y (cm)	Z (cm)	$\Psi$ (°)
1	15	4	70	90	18,17	5,47	65,77	87,5
2	13	23	60	20	16,17	22,5	57,87	21
3	21	11	61	50	21,07	10,57	60,85	50
4	9	1	64	70	11,27	3,87	65,01	70
5	9	-4	55	170	9,67	-4,2	57,55	165
6	16	12	63	80	15,33	14,67	58,67	80
7	25	11	40	90	21,57	5,47	39,85	90
8	22	19	37	30	21,17	13,55	39,15	26
9	20	-22	60	10	19,5	-21,55	59,5	10
10	21	8	53	70	24,22	9,55	57,5	70

Tabel 4.18 Tabel Pengujian Sistem *Inverse Kinematic* dengan Jarak 7 Meter

No	Set Point				Nilai Rata-Rata Hasil Pengukuran			
	x (cm)	y (cm)	z (cm)	$\Psi$ (°)	x (cm)	y (cm)	Z (cm)	$\Psi$ (°)
1	15	4	70	90	18,17	5,47	65,77	87,5
2	13	23	60	20	16,17	22,5	57,87	21
3	21	11	61	50	21,07	10,57	60,85	50
4	9	1	64	70	11,27	3,87	65,01	70
5	9	-4	55	170	9,67	-4,2	57,55	165
6	16	12	63	80	15,33	14,67	58,67	80
7	25	11	40	90	21,57	5,47	39,85	90
8	22	19	37	30	21,17	13,55	39,15	26
9	20	-22	60	10	19,5	-21,55	59,5	10
10	21	8	53	70	24,22	9,55	57,5	70

Tabel 4.19 Tabel Pengujian Sistem *Inverse Kinematic* dengan Jarak 8 Meter

No	Set Point				Nilai Rata-Rata Hasil Pengukuran			
	x (cm)	y (cm)	z (cm)	$\Psi$ (°)	x (cm)	y (cm)	Z (cm)	$\Psi$ (°)
1	15	4	70	90	18,17	5,47	65,77	87,5
2	13	23	60	20	16,17	22,5	57,87	21
3	21	11	61	50	21,07	10,57	60,85	50
4	9	1	64	70	11,27	3,87	65,01	70
5	9	-4	55	170	9,67	-4,2	57,55	165
6	16	12	63	80	15,33	14,67	58,67	80
7	25	11	40	90	21,57	5,47	39,85	90
8	22	19	37	30	21,17	13,55	39,15	26
9	20	-22	60	10	19,5	-21,55	59,5	10
10	21	8	53	70	24,22	9,55	57,5	70

Tabel 4.20 Tabel Pengujian Sistem *Inverse Kinematic* dengan Jarak 9 Meter

No	Set Point				Nilai Rata-Rata Hasil Pengukuran			
	x (cm)	y (cm)	z (cm)	$\Psi$ (°)	x (cm)	y (cm)	Z (cm)	$\Psi$ (°)
1	15	4	70	90	18,17	5,47	65,77	87,5
2	13	23	60	20	16,17	22,5	57,87	21
3	21	11	61	50	21,07	10,57	60,85	50
4	9	1	64	70	11,27	3,87	65,01	70
5	9	-4	55	170	9,67	-4,2	57,55	165
6	16	12	63	80	15,33	14,67	58,67	80
7	25	11	40	90	21,57	5,47	39,85	90
8	22	19	37	30	21,17	13,55	39,15	26
9	20	-22	60	10	19,5	-21,55	59,5	10
10	21	8	53	70	24,22	9,55	57,5	70

Tabel 4.21 Tabel Pengujian Sistem *Inverse Kinematic* dengan Jarak 10 Meter

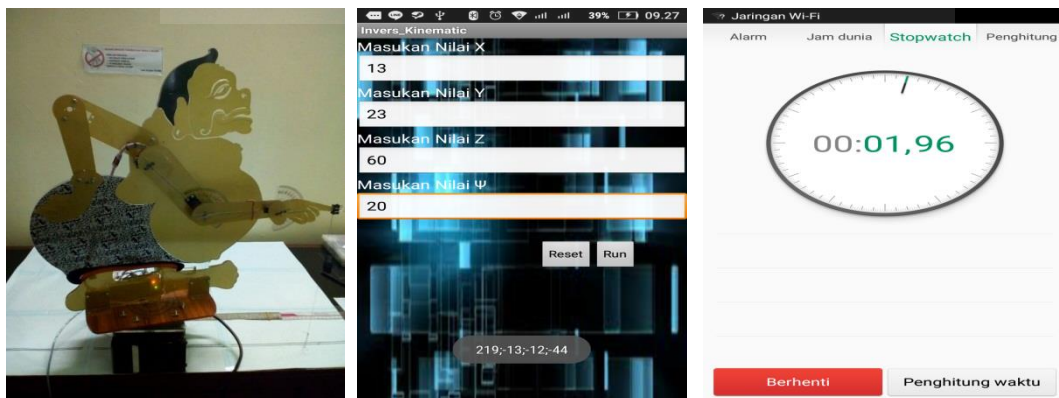
No	Set Point				Nilai Rata-Rata Hasil Pengukuran			
	x (cm)	y (cm)	z (cm)	$\Psi$ (°)	x (cm)	y (cm)	Z (cm)	$\Psi$ (°)
1	15	4	70	90	18,17	5,47	65,77	87,5
2	13	23	60	20	16,17	22,5	57,87	21
3	21	11	61	50	21,07	10,57	60,85	50
4	9	1	64	70	11,27	3,87	65,01	70
5	9	-4	55	170	9,67	-4,2	57,55	165
6	16	12	63	80	15,33	14,67	58,67	80
7	25	11	40	90	21,57	5,47	39,85	90
8	22	19	37	30	21,17	13,55	39,15	26
9	20	-22	60	10	19,5	-21,55	59,5	10
10	21	8	53	70	24,22	9,55	57,5	70



Tabel 4.22 Hasil Perhitungan RMSE Pada Pengujian *Inverse Kinematic*

Jarak	x RMSE	y RMSE	Z RMSE	$\Psi$ RMSE
1 meter	2,221 cm	2,829 cm	2,359 cm	2,19°
2 meter	2,221 cm	2,829 cm	2,359 cm	2,19°
3 meter	2,221 cm	2,829 cm	2,359 cm	2,19°
4 meter	2,221 cm	2,829 cm	2,359 cm	2,19°
5 meter	2,221 cm	2,829 cm	2,359 cm	2,19°
6 meter	2,221 cm	2,829 cm	2,359 cm	2,19°
7 meter	2,221 cm	2,829 cm	2,359 cm	2,19°
8 meter	2,221 cm	2,829 cm	2,359 cm	2,19°
9 meter	2,221 cm	2,829 cm	2,359 cm	2,19°
10 meter	2,221 cm	2,829 cm	2,359 cm	2,19°

Dari hasil perhitungan RMSE menunjukkan bahwa pada jarak 1 meter hingga 10 meter memiliki nilai RMSE yang sama dapat dilihat pada Tabel 4.22. Hal tersebut dikarenakan saat melakukan percobaan dengan nilai masukan yang sama untuk jarak 1 meter hingga 10 meter, posisi *end effector* jatuh pada posisi yang sama. Dari penjelasan tersebut bahwa data yang dikirim aplikasi android dalam metode *inverse kinematic* menuju arduino pada robot wayang tidak mengalami perubahan meskipun jarak pengiriman berbeda-beda. Adapun durasi pengiriman data dari android menuju robot wayang selama 1,96 detik.



Gambar 4.2 Durasi Pengiriman Data Ke Robot Wayang

## BAB V

### PENUTUP

#### 5.3 Kesimpulan

- Dari hasil pengujian *forward* dan *inverse kinematic* didapatkan kesimpulan bahwa aplikasi android dapat digunakan sebagai antarmuka dan perhitungan *forward kinematic* dan *inverse kinematic* untuk robot wayang melalui komunikasi bluetooth.
- Pada pengujian *forward kinematic* rata-rata *error* posisi untuk  $x$ ,  $y$ ,  $z$ , dan  $\Psi$  secara berurutan sebesar 0,967 cm; 1,853 cm; 0,887 cm; 1,14°. Untuk metode *inverse kinematic* rata-rata *error* posisi untuk  $x$ ,  $y$ ,  $z$ , dan  $\Psi$  secara berurutan sebesar 2,221 cm; 2,829 cm; 2,359 cm; 2,19°.

#### 5.4 Saran

- Tipe data yang digunakan dalam pengolahan data masih menggunakan *integer* (bilangan bulat). Agar tingkat akurasi dari robot wayang semakin tinggi sebaiknya tipe data diganti kedalam bentuk *double* (bilangan desimal).
- Servo pada *joint base* sebaiknya diganti dengan servo yang memiliki *rating* torsi yang lebih tinggi karena memiliki beban paling besar diantara semua servo.

## DAFTAR PUSTAKA

- [1] A. Juliansah, *Perancangan Sistem Control Robot Pemindah Barang Menggunakan Aplikasi Android Berbasis Arduino Uno*. Skripsi, tidak diterbitkan, Jurusan Sistem Computer Universitas Raharja, 2014.
- [2] Zpryme Research Survey, “The new hire: How a new generation of robots is transforming manufacturing”, 2014.
- [3] A. F. D. Marsiano, *Kendali Robot Wayang 4-DOF Dengan Modul Robotic Labview*. Skripsi, tidak diterbitkan, Jurusan Teknik Elektro Universitas Islam Indonesia, 2016.
- [4] Y. A. Wiratama, *Perancangan Kontroler Lengan Robot Hastobot Menggunakan Android Dan Arduino Dengan Komunikasi Bluetooth*. Skripsi tidak diterbitkan, Fakultas Sains dan Teknologi Industri Universitas Sanata Dharma, 2015.
- [5] H. R. Nurhakim, *Sistem Kendali Gerak Continuous Path Tracking Dengan Menggunakan Cubic Trajectory Planning Pada Robot Manipulator 4-DOF*. Skripsi, tidak diterbitkan, Fakultas Teknologi Industri Universitas Indonesia, 2010.
- [6] A. Saefullah, *Sistem Kontrol Robot Pemindah Barang Menggunakan Aplikasi Android Berbasis Arduino Uno*. Skripsi, tidak diterbitkan, Jurusan Teknik Elektro Universitas Sanata Dharma, 2015.
- [7] B. Siciliano, L. Sciavicco, G. Villani, G. Oriolo : “Robotic: Modelling, Planning And Control (3rd Edition)”, Springer, 2009.
- [8] G. Andreas dan R. Cuartielles, *Professional Android Open Accessory Programming With Arduino*. Indianapolis: John Wiley and Sons, Inc, 2013.

## LAMPIRAN

### Lampiran 1

```
Program Arduino
#include <VarSpeedServo.h>

VarSpeedServo servo1;
VarSpeedServo servo2;
VarSpeedServo servo3;
VarSpeedServo servo4;

int zero1=100, zero2=80, zero3=84,
zero4=75;
int s1=0,s2=0,s3=0,s4=0,r1,r2,r3,r4,
r1ms,r2ms,r3ms,r4ms;
int zz;
byte dataread;

int pos=0;

void setup() {
  Serial.begin(9600);
  servo4.write(75); //wrist
  servo4.attach(3);
  servo3.write(84);
  servo3.attach(6); //elbow
  servo2.write(80);
  servo2.attach(5); //shoulder
  servo1.write(100);
  servo1.attach(11); //base

  servo4.write(zero4,10,true);
  servo3.write(zero3,10,true);
  servo2.write(zero2,10,true);
  servo1.write(zero1,10,true);
}

void loop() {

  if(Serial.available() >> 0){
    //dataread = Serial.read();
    s1 = Serial.parseInt();
    s2 = Serial.parseInt();
    s3 = Serial.parseInt();
    s4 = Serial.parseInt();

    if(s1 > 80){s1=80;}
    if(s2 > 80){s2=80;}
    if(s3 > 80){s3=80;}
    if(s4 > 80){s4=80;}

    if(s1 < -70){s1=-70;}
    if(s2 < -80){s2=-80;}
    if(s3 < -80){s3=-80;}
    if(s4 < -80){s4=-80;}

    //BASE
    servo1.write(map(s1,80,-
80,20,180),10,false);

    //SHOULDER
    servo2.write(map(s2,-
80,80,10,150),10,false);

    //ELBOW
    servo3.write(map(s3,80,-
80,20,148),10,false);

    //WRIST
    servo4.write(map(s4,-
80,80,0,150),10,false);

    r1 = servo1.read();
    r2 = servo2.read();
    r3 = servo3.read();
    r4 = servo4.read();

    Serial.print(s1);
    Serial.print(",");
    Serial.print(s2);
    Serial.print(",");
    Serial.print(s3);
    Serial.print(",");
    Serial.print(s4);
    Serial.println();
  }
}
```

## Lampiran 2

### Kode penghitungan *forward kinematic*

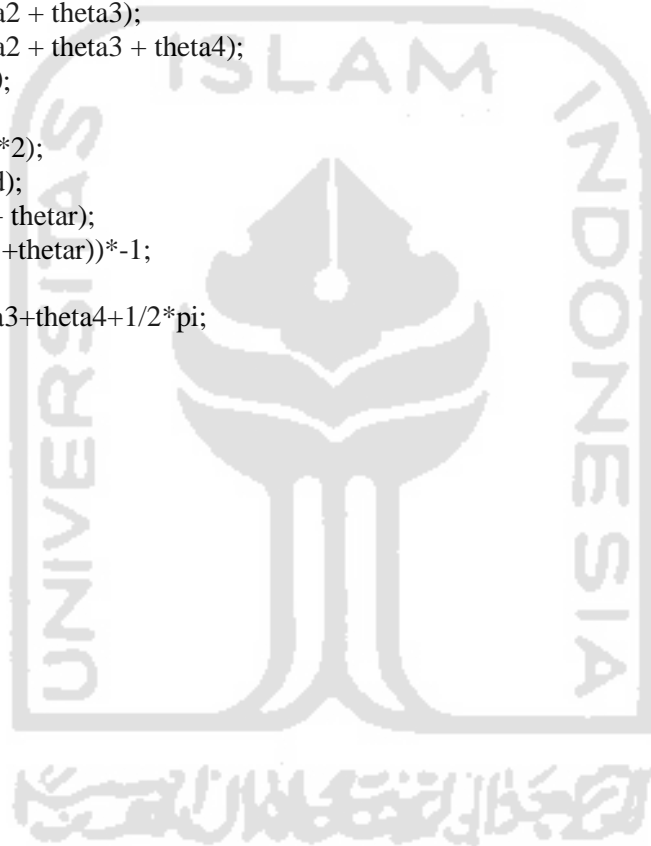
```
float r1, r2, r3, L, z1, z2, z3, z, r, thetar;
```

```
r1 = L1*cos(theta2 - 2/18*pi);  
r2 = L2*sin(theta2 + theta3);  
r3 = L3*sin(theta2 + theta3 + theta4);  
L = r1-(r2+r3);
```

```
z1 = L1*sin(theta2 - 2/18*pi);  
z2 = L2*cos(theta2 + theta3);  
z3 = L3*cos(theta2 + theta3 + theta4);  
z = z1+z2+z3+L0;
```

```
r = sqrt(L**2+d**2);  
thetar = atan2(L,d);  
x = r*sin(theta1 + thetar);  
y = (r*cos(theta1 + thetar))*-1;
```

```
psi = theta2+theta3+theta4+1/2*pi;
```



### Lampiran 3

#### Kode penghitungan *inverse kinematic*

```
float r, rwr, zwr, theta2i, theta3i, theta4, beta, alpha, cth3i, calpha, unreachable, yoff;
float theta1a, theta1b, theta3e, L;
float oldtheta1, oldtheta2, oldtheta3, oldtheta4;
x = x, y = y, z = z, psi = psi;

L = sqrt(x**2+y**2);
theta1a = atan2(x,y);
theta1b = acos(Loff/L);

r = sqrt(L**2-Loff**2);
rwr = r - L3*cos(psi);
zwr = z - L0 - L3*sin(psi);

cth3i = (rwr**2+zwr**2-L1**2-L2**2)/(2*L1*L2);
calpha = (rwr**2+zwr**2+L1**2-L2**2)/(2*L1*(sqrt(rwr**2+zwr**2)));
if ((cth3i>1) || (calpha>1))
{
    unreachable = 1;
    theta2 = oldtheta2; theta3 = oldtheta3; theta4 = oldtheta4;
}
else
{
    unreachable = 0;
    theta1 = pi - theta1a - theta1b;
    theta3i = acos((rwr**2+zwr**2-L1**2-L2**2)/(2*L1*L2));
    beta = atan2 (zwr,rwr);
    alpha = acos((rwr**2+zwr**2+L1**2-L2**2)/(2*L1*(sqrt(rwr**2+zwr**2))));
    if (theta3i>=0)
    {
        theta2i = beta-alpha;
        theta2 = theta2i+(2/18*pi);
        theta3 = theta3i - (pi*11/18);
    }
    else
    {
        theta2i = beta+alpha;
        theta2 = theta2i-(2/18*pi);
        theta3 = (pi*11/18) - theta3i;
    }

    theta4 = psi - (theta3i+theta2i);
    oldtheta1 = theta1; oldtheta2 = theta2; oldtheta3 = theta3; oldtheta4 = theta4;
}
}
```

Tabel Pengukuran *Forward Kinematic* untuk

No	Input				Pengukuran 1				Pengukuran 2				Pengukuran 3			
	$\theta_1$	$\theta_2$	$\theta_3$	$\theta_4$	x	y	z	$\Psi$	x	y	z	$\Psi$	x	y	z	$\Psi$
1	10	5	-50	50	21,6	8,4	62,5	94	21,5	8,3	62,5	94	21,5	8,4	62,5	94
2	20	10	-40	45	19,3	-4,8	65	103	19,35	-4,6	65	103	19,35	-4,8	65	103
3	30	15	-30	40	16,4	-3,4	67	115	16,45	-3,3	67	115	16,45	-3,3	67	115
4	40	20	-20	35	12,5	-3,6	67,8	125	12,55	-3,45	67,8	125	12,55	-3,45	67,8	125
5	50	25	-10	30	10,8	-4,5	67,5	133	10,6	-4,5	67,5	133	10,6	-4,5	67,5	133
6	70	-30	50	-25	10,5	-2	59	85	10,5	-2	59	85	10,5	-2	59	85
7	40	-35	40	-20	12,5	-4	59,5	75	2,5	-4	59,5	75	2,5	-4	59,5	75
8	30	-40	30	-15	15,3	-3,5	5,5	65	15,35	-3,5	5,5	65	15,35	-3,5	5,5	65
9	40	-45	20	-10	18,8	1,7	56,4	55	18,6	1,65	56,4	55	18,6	1,65	56,4	55
10	50	-50	10	-5	21,4	7,7	53	43	21,3	7,65	53	43	21,3	7,65	53	43

Tabel Pengukuran *Inverse Kinematic*

No	Set point				Pengukuran 1				Pengukuran 2				Pengukuran 3			
	x	y	z	$\Psi$	x	y	z	$\Psi$	x	Y	z	$\Psi$	X	y	z	$\Psi$
1	15	4	70	90	18,1	5,5	56,75	87,5	18,2	5,45	56,8	87,5	18,2	5,45	56,8	87,5
2	13	23	60	20	16,2	22,5	57,9	21	16,15	22,5	57,85	21	16,15	22,5	57,85	21
3	21	11	61	50	21,1	10,55	60,85	50	21,05	10,6	60,85	50	21,05	10,6	60,85	50
4	9	1	64	70	11,3	3,85	65	70	11,25	3,9	65	70	11,25	3,9	65	70
5	9	-4	55	170	9,7	-4,2	57,55	165	9,65	-4,2	57,55	165	9,65	-4,2	57,55	165
6	16	12	63	80	15,3	14,7	58,7	80	15,35	14,65	58,65	80	15,35	14,65	58,65	80
7	25	11	40	90	21,6	5,5	39,85	90	21,55	5,45	39,85	90	21,55	5,45	39,85	90
8	22	19	37	30	21,2	13,55	39,15	26	21,2	13,55	39,15	26	21,2	13,55	39,15	26
9	20	-22	60	10	19,5	-21,55	59,5	10	19,5	-21,55	59,5	10	19,5	-21,55	59,5	10
10	21	8	53	70	24,25	9,55	57,5	70	24,2	9,55	57,5	70	24,2	9,55	57,5	70