

PENERAPAN ALGORITMA *FORWARD KINEMATIK* DAN *INVERS KINEMATIK* PADA ROBOT LENGAN *ORION ROBOTIC 5 DOF*

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



ARDYANTO DWI KURNIAWAN

NIM. 115060300111022-63

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2016

RINGKASAN

Ardyanto Dwi Kurniawan, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Januari 2016, *Penerapan Algoritma Forward Kinematic dan Invers Kinematic pada Robot Lengan Orion Robotic 5 DOF*, Dosen Pembimbing : Goegoes Dwi Nusantoro dan Rahmadwati.

Saat ini perkembangan teknologi sudah sangat pesat dan telah mencakup di segala bidang. Perkembangan yang juga terjadi di bidang industri menuntut penggunaan alat bantu yang dapat mengoptimalkan sumber-sumber daya yang ada agar dapat bersaing di pasar bebas. Robot merupakan salah satu alat bantu yang dalam kondisi tertentu sangat diperlukan dalam industri. Diantara robot yang sering digunakan dalam dunia industri adalah robot lengan.

Robot lengan diharapkan dapat diprogram ulang secara fleksibel oleh pengguna, maka yang dibutuhkan adalah antarmuka antara robot lengan dengan pengguna melalui komputer. Melalui sistem antarmuka, penerapan metode *invers/forward* pada robot lengan akan lebih mudah dalam menghitung posisi *end-effector*. Penggunaan analisa pendekatan geometri akan diterapkan pada *invers kinematic* untuk mempermudah pencarian solusi dari posisi *end-effector*.

Pada pengujian *forward kinematic* dapat dilihat rata-rata *error* posisi pada sumbu x yaitu sebesar ± 0.11321 cm, pada sumbu y yaitu sebesar ± 0.10571 cm dan pada sumbu z yaitu ± 0.94785 cm. Pada pengujian *invers kinematic* dapat dilihat rata-rata *error* sudut pada servo *base* yaitu sebesar ± 0.571 derajat, pada servo *shoulder* yaitu sebesar ± 0.785 derajat, pada servo *elbow* yaitu sebesar ± 1.285 derajat, pada servo *wrist* yaitu sebesar ± 1.035 derajat.

Kata kunci : *Orion Robotic Arm*, Robot Lengan, *Forward Kinematic*, *Invers Kinematic*

SUMMARY

Ardyanto Dwi Kurniawan, Department of Electrical Engineering, Faculty of Engineering University of Brawijaya, January 2016, *Application of Algorithm on the Forward Kinematic and Inverse Kinematic at Orion Robotic Arm Robot 5 DOF*, Academic Supervisor : Goegoes Dwi Nusantoro dan Rahmadwati.

At this time, the development of technology is gain rapidly. It also happens in industrial field, that requires the aid tools. The tools able to support existing resources, so the product can be compete in free market. At this moment, robot is needed in industry. Commonly, the arm robot is very popular used in industry.

The robot arm is expected to be re-programmed flexibly by the user, so the interface between the robot arm with the user through a computer is needed. Through the user interface system, the application of the method of inverse / forward on the robot arm will be easier in order to calculate the end-effector's position. Analytical geometry will be applied in the inverse kinematic solution to facilitate the search of the end-effector's position.

The results from testing, forward kinematic average can be seen on the x-axis position error is equal to ± 0.11321 cm, on the y-axis is equal to ± 0.10571 cm and the z axis is ± 0.94785 cm. In testing the inverse kinematic can be seen the average error on the servo base angle is equal to ± 0571 degrees, the shoulder servo is equal to ± 0785 degrees, the servo elbow is equal to $\pm 1,285$ degrees, the servo wrist that is equal to ± 1.035 degree.

Keywords : Orion Robotic Arm, Robot Arm, Forward Kinematic, Invers Kinematic

DAFTAR ISI

| | Halaman |
|---|---------|
| PENGANTAR | i |
| DAFTAR ISI | iii |
| DAFTAR GAMBAR | v |
| DAFTAR TABEL | vi |
| DAFTAR LAMPIRAN | ix |
| RINGKASAN | xi |
| | |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 2 |
| 1.3 Batasan Masalah | 3 |
| 1.4 Tujuan | 3 |
| 1.5 Sistematika Pembahasan | 3 |
| BAB II TINJAUAN PUSTAKA | 5 |
| 2.1 Kinematika Robot Lengan | 5 |
| 2.1.1 Rotasi | 5 |
| 2.1.2 Translasi | 7 |
| 2.1.3 Matriks Transformasi Homogen | 8 |
| 2.1.4 <i>Denavit-Hartenberg (D-H) Parameter</i> | 9 |
| 2.1.5 <i>Forward Kinematic</i> | 10 |
| 2.1.6 <i>Invers Kinematic</i> | 11 |
| 2.2 Trigonometri | 12 |
| 2.3 Sistem Robot Lengan | 12 |
| 2.4 <i>Orion Arm Robotic</i> | 15 |
| 2.5 Motor Servo | 16 |
| 2.6 Arduino UNO | 18 |
| 2.6.1 Daya | 19 |
| 2.6.2 Pemrograman | 19 |
| 2.7 <i>Processing</i> | 20 |

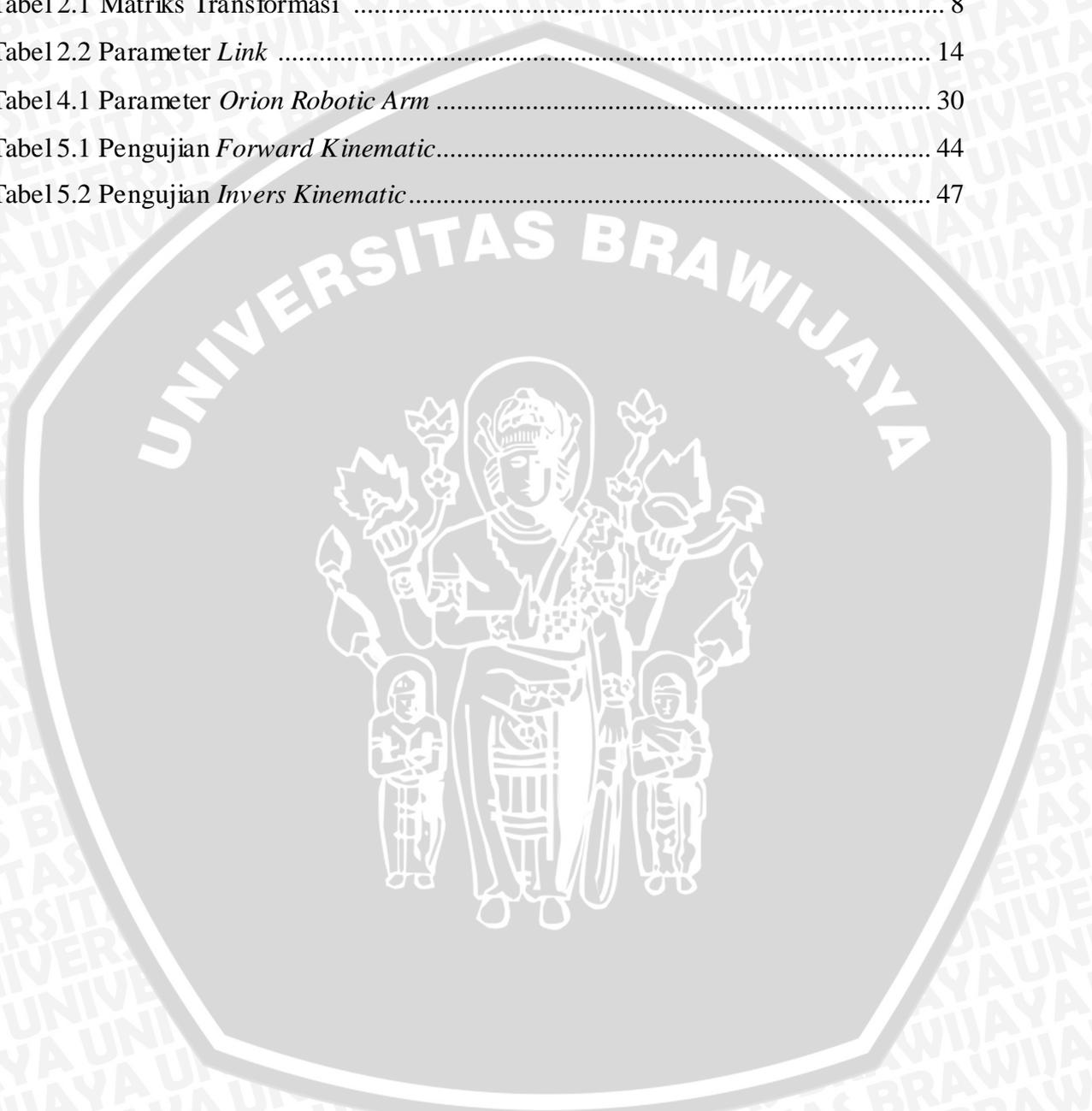
| | |
|--|----|
| BAB III METODE PENELITIAN | 23 |
| 3.1 Perancangan Sistem | 23 |
| 3.2 Realisasi Pembuatan Sistem | 23 |
| 3.3 Pengujian dan Analisa Data | 24 |
| 3.4 Pengambilan Kesimpulan dan Saran | 25 |
| BAB IV PERANCANGAN SISTEM DAN PEMBUATAN PROGRAM | 27 |
| 4.1 Perancangan Sistem | 27 |
| 4.1.1 Diagram Balok dan Prinsip Kerja Robot Lengan | 27 |
| 4.1.2 Deskripsi <i>Frame</i> Robot Lengan | 29 |
| 4.2 Perancangan Kinematika Robot Lengan | 31 |
| 4.2.1 <i>Forward Kinematic</i> | 31 |
| 4.2.2 <i>Invers Kinematic</i> | 33 |
| 4.3 Perancangan Pergerakan Robot Lengan | 37 |
| 4.3.1 <i>Workspace</i> Robot Lengan | 37 |
| 4.3.2 Skema Gerak Robot Lengan | 39 |
| 4.3.3 Perancangan Perangkat Lunak | 40 |
| BAB V PENGUJIAN DAN ANALISIS SISTEM | 43 |
| 5.1 Pengujian <i>Forward Kinematic</i> | 43 |
| 5.2 Pengujian <i>Invers Kinematic</i> | 46 |
| BAB VI PENUTUP | 53 |
| 6.1 Kesimpulan | 51 |
| 6.2 Saran | 52 |
| DAFTAR PUSTAKA | |
| LAMPIRAN | |

DAFTAR GAMBAR

| No. | Judul | Halaman |
|-------------|--|---------|
| Gambar 2.1 | Rotasi pada Tiap-tiap Sumbu Koordinat Kartesian..... | 5 |
| Gambar 2.2 | Gambaran Umum Penempatan Frame pada <i>Forward Kinematic</i> | 11 |
| Gambar 2.3 | Mekanik dari Robot Lengan | 13 |
| Gambar 2.4 | Pendeklarasian Frame pada Robot Lengan | 14 |
| Gambar 2.5 | <i>Orion Robotics Digital Servo HV-220</i> | 16 |
| Gambar 2.6 | Konfigurasi Pin Pengkabelan Motor Servo | 16 |
| Gambar 2.7 | Pengaturan Sudut Motor Servo | 17 |
| Gambar 2.8 | Arduino UNO | 18 |
| Gambar 2.9 | Processing 2.1.1 | 20 |
| Gambar 2.10 | Tampilan Utama Processing 2.1.1 | 21 |
| Gambar 4.1 | Diagram Balok Perangkat Keras | 27 |
| Gambar 4.2 | Diagram Balok Sistem Pengaturan | 28 |
| Gambar 4.3 | <i>Orion Robotic Arm</i> | 28 |
| Gambar 4.4 | Modul Arduino UNO | 28 |
| Gambar 4.5 | Pendeklarasian Frame pada Robot Lengan Orion..... | 30 |
| Gambar 4.6 | Tampilan Perhitungan 0T_4 | 32 |
| Gambar 4.7 | Segitiga pada Sumbu X dan Y..... | 34 |
| Gambar 4.8 | Tampilan Perhitungan A_0 dan W_T | 34 |
| Gambar 4.9 | Penyederhanaan di Bidang Sumbu W dan Z dengan 3DOF | 35 |
| Gambar 4.10 | Arah <i>Gripper</i> dengan Permisalan Sudut | 35 |
| Gambar 4.11 | Tampilan Perhitungan W_2 dan Z_2 | 35 |
| Gambar 4.12 | Penyederhanaan di Bidang Sumbu W dan Z dengan 2DOF | 36 |
| Gambar 4.13 | Tampilan Perhitungan L_{12} dan A_{12} | 36 |
| Gambar 4.14 | Tampilan Perhitungan A_1 , W_1 , Z_1 , A_2 dan A_3 | 37 |
| Gambar 4.15 | <i>Workspace</i> Robot Lengan | 38 |
| Gambar 4.16 | Rancangan Medan Robot Lengan | 40 |
| Gambar 4.17 | <i>Flowchart</i> Program <i>Invers Kinematic</i> | 41 |
| Gambar 4.18 | <i>Flowchart</i> Program <i>Forward Kinematic</i> | 42 |
| Gambar 5.1 | Balok Pengujian <i>Forward Kinematic</i> | 44 |
| Gambar 5.2 | Balok Pengujian <i>Invers Kinematic</i> | 46 |

DAFTAR TABEL

| No. | Judul | Halaman |
|-----------|--|---------|
| Tabel 2.1 | Matriks Transformasi | 8 |
| Tabel 2.2 | Parameter <i>Link</i> | 14 |
| Tabel 4.1 | Parameter <i>Orion Robotic Arm</i> | 30 |
| Tabel 5.1 | Pengujian <i>Forward Kinematic</i> | 44 |
| Tabel 5.2 | Pengujian <i>Invers Kinematic</i> | 47 |



DAFTAR LAMPIRAN

| No. | Judul | Halaman |
|------------|-----------------------|---------|
| Lampiran 1 | Foto Alat | 55 |
| Lampiran 2 | Listing Program | 57 |
| Lampiran 3 | Datasheet | 70 |



BAB I PENDAHULUAN

Skripsi ini membahas tentang suatu sistem yang dapat digunakan sebagai alat bantu kecerdasan untuk berinteraksi antara robot lengan dengan manusia. Sistem robot lengan dibuat dengan menggabungkan kombinasi antara mekanik (*Orion Robotic Arm*) dan rangkaian elektronika (Arduino Uno). Melalui persamaan pemodelan kinematika *Orion Robotic Arm*, sistem ini mampu menghasilkan nilai sudut dan koordinat tiap join lengan robot (*base, shoulder, elbow, wrist*). Permodelan kinematika (kinematik maju dan kinematik balik) menggunakan metode Denavit-Hartenberg akan diuji melalui implementasi *hardware (Orion Robotic Arm)* dan implementasi *software (Processing)*.

1.1 Latar Belakang

Pada saat ini, perkembangan teknologi sudah sangat pesat. Perkembangannya telah mencakup di segala bidang, terutama pada bidang teknologi komputer, elektronika, dan kontrol. Salah satu contoh perkembangan pada bidang komputer dan bidang elektronika dan kontrol adalah penggunaan robot. Penggunaan robot kini semakin meningkat dari waktu ke waktu untuk menangani berbagai tugas. Baik tugas yang tidak bisa ditangani manusia ataupun tugas-tugas yang telah umum dilakukan oleh manusia (Hamidah, 2008).

Di sisi lainnya, perkembangan industri menuntut penggunaan alat bantu yang dapat mengoptimalkan sumber-sumber daya (*resources*) yang ada agar dapat bersaing di pasar bebas. Robot merupakan salah satu alat bantu yang dalam kondisi tertentu sangat diperlukan dalam industri. Terdapat kondisi-kondisi tertentu dalam industri yang tidak mungkin ditangani oleh manusia seperti kebutuhan akan akurasi yang tinggi, tenaga yang besar, kecepatan yang tinggi atau resiko yang tinggi. Keadaan-keadaan ini dapat diatasi dengan penggunaan robot (Masykuri, 2010).

Di antara robot yang sering digunakan dalam dunia industri adalah robot lengan. Robot lengan merupakan robot yang meniru gerak lengan pada manusia sehingga mampu melakukan pemindahan barang dari satu tempat ke tempat lain. Robot lengan diharapkan dapat diprogram ulang secara fleksibel oleh pengguna. Pemrograman ulang pada robot lengan membuat pengguna membutuhkan antarmuka antara robot lengan dengan pengguna melalui komputer. Dengan adanya antarmuka tersebut diharapkan robot lengan akan lebih mudah diprogram ulang sesuai kebutuhan (Seif, 2012).

Hingga saat ini sudah ada dua penelitian mengenai lengan robot di Teknik Elektro Universitas Brawijaya, yaitu:

1. Lengan robot Daggu Hi-Tech oleh Seif NuraUrfin pada tahun 2012. Dengan mikrokontroler Atmega 168 dan Visual Basic 6.0. Lengan robot ini mencakup *workspace* dalam ruang dua dimensi dan 3 derajat kebebasan.
2. Lengan robot AX-12A oleh Dina Caysar pada tahun 2014. Dengan mikrokontroler Atmega1280 dan Qt Framework. Lengan robot ini mencakup *workspace* dalam ruang dua dimensi dan 2 derajat kebebasan.

Pada penelitian ini lengan robot yang digunakan adalah *Orion Robotic Arm 5 DOF*, dengan menggunakan mikrokontroler Arduino Uno dan Processing 1.5.1 sebagai HMI. Metode *invers* dan *forward* pada *Geometry - Based Kinematic* digunakan dalam penelitian ini untuk mendapatkan posisi *end effector*. Dengan metode *invers* dan *forward* dengan pendekatan *Geometry - Based Kinematic* diharapkan mahasiswa memahami dinamika gerak robot lengan dan pembuatan sistem antarmuka robot lengan yang baik diharapkan dapat digunakan mahasiswa sebagai modul pembelajaran mata kuliah robotika.

1.2 Rumusan Masalah

Berdasarkan Latar belakang dapat disusun rumusan masalah sebagai berikut:

- a) bagaimana membuat pemodelan statis (kinematika maju dan kinematika balik) pada *Orion Robotic Arm* dengan lima derajat kebebasan.
- b) bagaimana hasil dari penerapan algoritma *forward kinematic* dan *invers kinematic* dengan pendekatan *Geometry - Based Kinematic*.

1.3 Batasan Masalah

Dengan mengacu pada permasalahan yang telah dirumuskan, maka hal-hal yang berkaitan dengan alat akan diberi batasan sebagai berikut :

- a) Pemodelan statis kinematika robot lengan *Orion Robotic Arm* dengan *Forward Kinematics* dan *Inverse Kinematics*.
- b) Pembahasan ditekankan pada kinematika gerak robot.
- c) Lintasan gerakan *end-effector* robot lengan ditentukan sebelumnya.
- d) Menggunakan Mikrokontroler Arduino UNO sebagai kontroler pergerakan robot lengan.
- e) Sistem yang digunakan adalah *open loop*.
- f) Realisasi lengan robot akan dihitung dan diukur tingkat akurasi dari *end effector*.

1.4 Tujuan

Merancang sistem pergerakan robot lengan *Orion Robotic Arm* melalui pendekatan *Geometry - Based Kinematic* yang akan dikontrol dengan Arduino UNO dan perangkat lunak *Processing*.

1.5 Sistematika Pembahasan

Skripsi ini terdiri atas enam bab dengan sistematika pembahasan sebagai berikut:

BAB I Pendahuluan

Membahas latar belakang, rumusan masalah, batasan masalah, tujuan dan sistematika pembahasan.

BAB II Tinjauan Pustaka

Membahas teori-teori yang mendukung dalam perencanaan dan pembuatan sistem.

BAB III Metode Penelitian

Membahas tentang metode yang digunakan dalam penyusunan skripsi ini.

BAB IV Perancangan dan Pembuatan Alat

Menjelaskan tentang perancangan dan pembuatan alat yang meliputi prinsip kerja alat, perancangan perangkat keras dan perangkat lunak.

BAB V Pengujian dan Analisis

Memuat hasil pengujian menggunakan metode *invers kinematic* dan *forward kinematic* terhadap ketepatan posisi dari robot lengan (*end-effector*).

BAB VI Penutup

Memuat kesimpulan dan saran-saran untuk pengembangan lebih lanjut dari sistem yang telah dibuat.

BAB II

TINJAUAN PUSTAKA

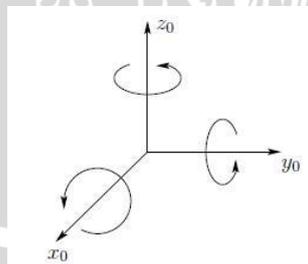
Skripsi ini membahas tentang sistem pengaturan *Orion Robotic Arm 5 DOF* melalui pendekatan *Geometry - Based Kinematic* untuk mendapatkan *goal position*. Berikut adalah teori-teori penunjang dalam penyelesaian skripsi ini.

2.1 Kinematika Robot Lengan

Salah satu dasar dari ilmu robotika adalah pemahaman mengenai kinematika dan dinamika robot. Dinamika robot merupakan pengetahuan atau teori tentang pergerakan objek dengan memperhitungkan gaya-gaya yang dapat menyebabkan objek bergerak, contohnya torsi pada aktuator, percepatan yang dialami oleh benda pejal dan distribusi massa. Sedangkan kinematika robot merupakan pengetahuan atau teori tentang pergerakan objek tanpa memperhitungkan gaya-gaya yang menyebabkan benda itu bergerak. Kinematika robot terdiri atas rotasi dan translasi. Kinematika robot secara umum terbagi menjadi dua, yaitu kinematika maju (*forward*) dan kinematika balik (*invers*). Kinematika maju juga dikenal sebagai kinematika langsung (*direct*), di mana panjang dari tiap link dan sudut dari tiap sendi diberikan dan dihitung posisi robot. Pada kinematika inversi, panjang dari tiap link dan posisi diberikan, lalu dihitung sudut dari tiap sendinya (Budiharto, 2010).

2.1.1 Rotasi

Rotasi yaitu gerakan berputar pada sebuah sumbu yang tetap, gerakan tersebut dapat berputar pada sumbu x , y maupun z . seperti yang terlihat dalam Gambar 2.1.



Gambar 2.1 Rotasi pada Tiap-tiap Sumbu Koordinat Kartesian (Spong, Mark W, 2007)

Z_0 adalah sumbu utama rotasi pada arah Z , Y_0 adalah sumbu utama rotasi pada arah Y , X_0 adalah sumbu utama rotasi pada arah X

Matrik transformasi rotasi pada tiap-tiap sumbu terlihat dalam persamaan di bawah ini (Persamaan 2.1, 2.2 dan 2.3), θ untuk sumbu z , α untuk sumbu x dan ϕ untuk sumbu y .

$$R_{z,\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \dots\dots\dots(2.1)$$

$$R_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \dots\dots\dots(2.2)$$

$$R_{y,\phi} = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \dots\dots\dots(2.3)$$

Dimana

$R_{z,\theta}$ = Matriks transformasi rotasi pada sumbu z sepanjang θ

$R_{x,\alpha}$ = Matriks transformasi rotasi pada sumbu x sepanjang α

$R_{y,\phi}$ = Matriks transformasi rotasi pada sumbu y sepanjang ϕ

Jadi untuk merepresentasikan sebuah posisi (p_1) dari sumbu koordinat yang telah berotasi ke dalam posisi (p_0) sumbu koordinat semula dirumuskan dalam Persamaan 2.4 sampai dengan 2.6 (Unikom, 2009).

$$p_0 = \begin{bmatrix} p_{0x} \\ p_{0y} \\ p_{0z} \end{bmatrix} \dots\dots\dots(2.4)$$

$$p_1 = \begin{bmatrix} p_{1x} \\ p_{1y} \\ p_{1z} \end{bmatrix} \dots\dots\dots(2.5)$$

$$p_0 = R_0^1 p_1 \dots\dots\dots(2.6)$$

Dimana :

p_0 = posisi sumbu koordinat semula.

p_{0x} = posisi sumbu koordinat x semula.

p_{0y} = posisi sumbu koordinat y semula.

p_{0z} = posisi sumbu koordinat z semula.



p_{1x} = posisi sumbu koordinat x yang telah mengalami translasi pada jarak tertentu dari sumbu koordinat semula.

p_{1y} = posisi sumbu koordinat y yang telah mengalami translasi pada jarak tertentu dari sumbu koordinat semula.

p_{1z} = posisi sumbu koordinat z yang telah mengalami translasi pada jarak tertentu dari sumbu koordinat semula.

R_0^1 = sumbu koordinat semula yang telah mengalami rotasi dengan sudut tertentu.

d_0^1 = sumbu koordinat semula yang telah mengalami pergeseran sejauh jarak tertentu.

H = matrik transformasi homogen yang terdiri dari gerakan rotasi dan translasi.

2.1.2 Translasi

Translasi artinya terdapat pergeseran sumbu koordinat pada jarak tertentu dari sumbu koordinat semula. Jadi untuk merepresentasikan posisi

(p_1) dari sumbu koordinat translasi ke dalam sumbu koordinat semula dirumuskan dalam Persamaan 2.7 dan 2.8.

$$p_0 = p_1 + d_0^1 \dots\dots\dots(2.7)$$

$$\begin{bmatrix} p_{0x} \\ p_{0y} \\ p_{0z} \end{bmatrix} = \begin{bmatrix} p_{1x} \\ p_{1y} \\ p_{1z} \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} \dots\dots\dots(2.8)$$

Dimana :

d_x = translasi sejauh jarak tertentu pada sumbu x

d_y = translasi sejauh jarak tertentu pada sumbu y

d_z = translasi sejauh jarak tertentu pada sumbu z

Apabila terjadi dua gerakan yaitu rotasi dan translasi secara bersamaan, artinya sumbu koordinat semula diputar (rotasi) dengan sudut tertentu (R_0^1) kemudian digeser (translasi) sejauh jarak tertentu (d_0^1). Dapat diselesaikan dengan menggabungkan persamaan rotasi dan translasi (Persamaan 2.6 dan 2.7) seperti dalam persamaan 2.9 (Unikom, 2009).

$$p_0 = R_0^1 p_1 + d_0^1 \dots\dots\dots(2.9)$$



2.1.3 Matriks Transformasi Homogen

Matriks transformasi homogen merupakan matriks yang digunakan untuk mengonversikan lokasi suatu titik pada koordinat sumbu j ke koordinat sumbu k. Simbol yang biasa dipakai adalah T_j^k untuk merepresentasikan transformasi suatu titik pada koordinat sumbu j menjadi suatu titik yang sama pada koordinat k. Simbol F_j^k juga digunakan untuk merepresentasikan transformasi koordinat sumbu j menjadi searah dengan koordinat sumbu k (Budiharto, 2010).

Matrik Rotasi (3 x 3)

$$T_1 = \begin{bmatrix} C1 & -S1 & 0 \\ S1 & C1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \dots\dots\dots(2.10)$$

Vektor Translasi (3 x 1)

$$R_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \dots\dots\dots(2.11)$$

Matrik Homogen (4 x 4)

$$T_H = \begin{bmatrix} C1 & -S1 & 0 & x_1 \\ S1 & C1 & 0 & y_1 \\ 0 & 0 & 1 & z_1 \\ \hline 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots(2.12)$$

Tabel 2.1 Matriks Transformasi

| F_j^k | T_j^k | Nama Transformasi |
|---|---|-------------------|
| Translasi koordinat k, pada sumbu x sejauh a, pada sumbu y sejauh b, pada sumbu z sejauh c. | $\begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$ | Trans(a,b,c) |
| Rotasi koordinat k, pada sumbu x sebesar θ | $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ | Rot x(θ) |
| Rotasi koordinat k, pada sumbu y sebesar θ | $\begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ \sin \theta & 1 & -\cos \theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ | Rot y(θ) |



| | | |
|---|---|-------------------|
| Rotasi koordinat k, pada sumbu z sebesar θ | $\begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ | Rot z(θ) |
|---|---|-------------------|

2.1.4 Denavit -Hartenberg (D-H) Parameter

Representasi bentuk matrik H yang terdiri dari matrik rotasi dan translasi disebut sebagai *homogenous transformation*. *Homogenous transformation* dapat memudahkan dalam perhitungan untuk mendapatkan posisi suatu titik yang diperlihatkan dari koordinat *frame i* kedalam bentuk koordinat *frame j* yang telah dilakukan gerakan translasi dan rotasi.

$$p_0 = R_0^1 p_1 + d_0^1$$

$$\begin{bmatrix} p_0 \\ 1 \end{bmatrix} = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ 1 \end{bmatrix}$$

$$H = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} \dots\dots\dots(2.13)$$

Pada umumnya untuk menentukan *frame* atau koordinat memiliki aturan-aturan tertentu agar nantinya memudahkan dalam menganalisa posisi dari pergerakan robot tersebut, diantaranya adalah *Denavit – Hartenberg parameter*. Pada *D-H parameter*, suatu matrik *homogenous transformation* A_i merepresentasikan hasil dari empat transformasi dasar. Secara matematis ditunjukkan dalam Persamaan 2.11 sampai 2.14 berikut ini.

$$A_i = Rot_{z_i, \theta_i} Trans_{z_i, d_i} Trans_{x_i, a_i} Rot_{x_i, \alpha_i} \dots\dots\dots(2.14)$$

$$A_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots(2.15)$$

$$A_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \cos \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & \cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots(2.16)$$

$$T_0^n = \begin{bmatrix} n_x & s_x & a_x & P_x \\ n_y & s_y & a_y & P_y \\ n_z & s_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots(2.17)$$

Dimana :



- $Rot_{z\theta}$ = Matrik rotasi sumbu koordinat z semula yang telah mengalami rotasi dengan sudut θ tertentu.
- $Rot_{x\alpha}$ = Matrik rotasi sumbu koordinat z semula yang telah mengalami rotasi dengan sudut α tertentu.
- $Trans_{zd}$ = Matrik translasi sumbu koordinat semula yang telah mengalami translasi sejauh jarak d tertentu.
- $Trans_{xd}$ = Matrik translasi sumbu koordinat semula yang telah mengalami rotasi translasi sejauh jarak d tertentu.
- A = Matrik transformasi homogen yang terdiri atas empat transformasi dasar transformasi untuk menentukan frame / koordinat.

Pada *D-H parameter*, terdapat beberapa parameter yaitu a_i , α_i , d_i , dan θ_i . a_i disebut sebagai *length*, α_i disebut sebagai *twist*, d_i disebut sebagai *offset*, dan θ_i disebut sebagai *angle* (Unikom, 2009).

2.1.5 Forward kinematic

Apabila menghendaki untuk menghitung posisi (dari koordinat dasar) robot lengan dengan memberikan nilai pada *D-H parameter*, maka hal itu disebut *forward kinematic*. *Forward kinematic* adalah suatu metode perhitungan posisi sebagai fungsi sudut dengan menghitung setiap *link*. Jika memiliki *n-link* maka untuk mencari matriks transformasi homogen bisa melalui persamaan 2.15.

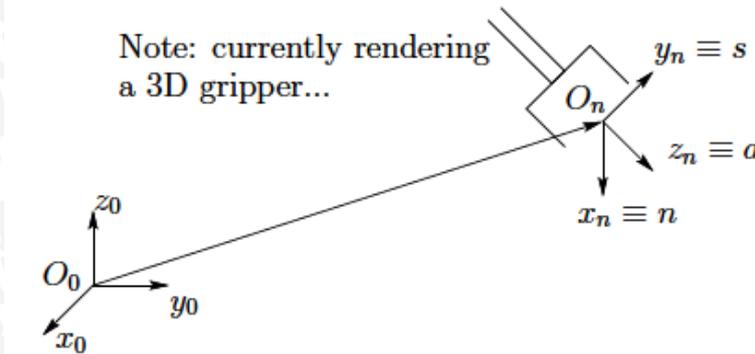
$$T_0^n = A_1 \dots \dots A_n \dots \dots \dots (2.18)$$

Matrik $n = [n_x \quad n_y \quad n_z]^T$ adalah vektor yang merepresentasikan arah sumbu o_1x_1 kedalam $o_0x_0y_0z_0$, n adalah arah normal (keatas).

Matrik $s = [s_x \quad s_y \quad s_z]^T$ adalah vektor yang merepresentasikan arah sumbu o_1y_1 kedalam $o_0x_0y_0z_0$, s adalah arah *sliding* (pergeseran).

Matrik $a = [a_x \quad a_y \quad a_z]^T$ adalah vektor yang merepresentasikan arah sumbu o_1z_1 kedalam $o_0x_0y_0z_0$, a adalah *approach* (pendekatan). Matrik $p = [p_x \quad p_y \quad p_z]^T$ adalah vektor yang merepresentasikan dari sumbu asal o_0 ke sumbu asal o_1 direpresentasikan kedalam sumbu koordinat $o_0x_0y_0z_0$

Dalam Gambar 2.2 berikut ini diperlihatkan gambaran umum penempatan *frame* robot pada bagian dasar dan sebuah *gripper* menggunakan *forward kinematic* (Spong, 2007).



Gambar 2.2 Gambaran Umum Penempatan Frame Pada *Forward Kinematic* (Spong, Mark W, 2007)

2.1.6 *Invers Kinematic*

Invers kinematik merupakan suatu metode analisa untuk melakukan transformasi dari ruang Cartesian ke ruang sendi. Dari persamaan kinematik, dapat diperoleh hubungan antar konsep geometri ruang sendi pada robot dengan konsep koordinat yang biasa digunakan untuk menentukan kedudukan suatu objek. Dengan model kinematik, programmer dapat menentukan konfigurasi referensi masukan yang harus diumpankan kepada masing-masing aktuator agar robot dapat melakukan gerakan secara simultan untuk mencapai posisi yang dikehendaki (Kusuma, 2013).

Langkah awal untuk menentukan sudut pada tiap-tiap link adalah dengan melakukan *invers* pada matrik *forward kinematic*. Persamaan-persamaan yang dihasilkan dari *invers* matrik *forward kinematic* kemudian dicari solusinya dengan melakukan substitusi dan eliminasi berdasarkan dua pendekatan di atas. Atau jika ditulis secara matematis akan menghasilkan

$$A_1^{-1}T_0^n = A_1^{-1}A_1 \dots A_n \dots \dots \dots (2.19)$$

sehingga:

$$A_{n-1}^{-1} \dots A_1^{-1}T_0^n = A_{n-1}^{-1} \dots A_1^{-1}A_1 \dots \dots A_n \dots \dots \dots (2.20)$$

2.2 Trigonometri

Trigonometri diambil dari bahasa Yunani, *trigonos* dan *metron*. *Trigonos* berarti segitiga dan *metron* berarti ukuran. Sehingga trigonometri dikenal sebagai ilmu ukur segitiga (Afriyanti, 2008).

Berikut beberapa fungsi trigonometri yang digunakan dalam perhitungan segitiga dalam *invers kinematik*, rumus trigonometri jumlah dan selisih dua sudut



$$\cos (\theta_1 - \theta_2) = (\cos \theta_1 \cdot \cos \theta_2) + (\sin \theta_1 \cdot \sin \theta_2) \quad \dots\dots(2.21)$$

$$\cos (\theta_1 + \theta_2) = (\cos \theta_1 \cdot \cos \theta_2) - (\sin \theta_1 \cdot \sin \theta_2) \quad \dots\dots(2.22)$$

$$\sin (\theta_1 - \theta_2) = (\sin \theta_1 \cdot \cos \theta_2) - (\cos \theta_1 \cdot \sin \theta_2) \quad \dots\dots(2.23)$$

$$\sin (\theta_1 + \theta_2) = (\sin \theta_1 \cdot \cos \theta_2) + (\cos \theta_1 \cdot \sin \theta_2) \quad \dots\dots(2.24)$$

2.3 Sistem Robot Lengan

Sistem robot lengan pada umumnya merupakan suatu struktur mekanik yang tersusun atas beberapa batang kaku yang sering disebut dengan *link*. *Link* terbuat dari logam, plastik, dan lain sebagainya. Antara *link* satu dengan *link* lainnya dihubungkan oleh persendian yang disebut *joint*. Umumnya *Prismatic Joint* dan *Flat Joint* dapat menghasilkan pergeseran. Sedangkan *Spheris Joint* dan *Revolute Joint* dapat menghasilkan *Degree of Freedom (DOF)* atau derajat kebebasan. *Degree of Freedom (DOF)* atau derajat kebebasan adalah jumlah arah yang independen di mana *actuator* dari sebuah robot dapat bergerak dan menghasilkan gerakan berputar. *DOF* dapat dihitung tiap sendi dan tidak termasuk *end effector*. Sedangkan *end effector* adalah piranti yang terpasang pada lengan robot untuk melaksanakan fungsi-fungsi tertentu. *End effector* dan keseluruhan bagian robot lengan bekerja pada *workspace* tertentu, tergantung kemampuan robot yang digunakan.

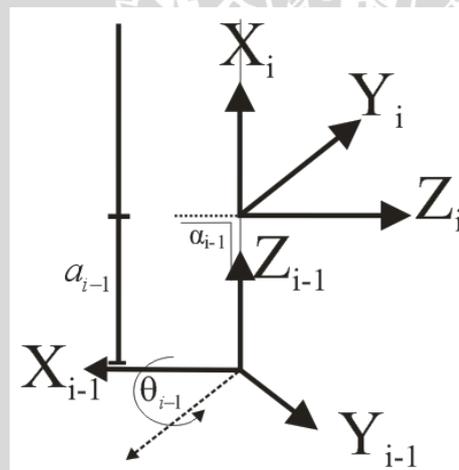
Pada robot *Orion Arm Robotic* memiliki 5 derajat kebebasan ditambah sebuah *gripper*. Bagian-bagian pada robot lengan diantaranya yaitu batang-batang logam alumunium yang berfungsi sebagai *link*, 5 buah servo yang berfungsi sebagai aktuator dan sebuah servo pada bagian *gripper* yang berfungsi sebagai *end effector*. Pengendalian lengan dibantu dengan suatu sistem mikrokontroler dan PC sebagai *interface*. Gambar 2.3 menunjukkan mekanik robot lengan *Orion Robotic Arm*



Gambar 2.3 Mekanik dari robot lengan *Orion Robotic Arm* (nodna.de)

Sistem koordinat robot lengan mengacu pada standar, setiap *link* diberi nomor dimulai dari bagian *basement*. *Basement* disebut *link 0*. Batang yang bergerak dan berhubungan dengan *basement* disebut *link 1*. Batang yang bergerak dan berhubungan dengan *link 1* disebut *link 2* dan seterusnya. Sehingga secara keseluruhan lengan robot terbentuk dari *link 0*, *link 1*, *link 2*, *link 3* dan *end effector*. Antara *link* yang satu dengan *link* tetangganya dihubungkan dengan *joint*. *Joint $i+1$* adalah *joint* yang menghubungkan *link i* dengan *link $i+1$* . Sehingga *joint 1* menghubungkan *link 0* dan *link 1*. Sedangkan *link 1* dan *link 2* dihubungkan dengan *joint 2* dan seterusnya (Spong, 2004).

Dari kelima *joint* yang dimiliki lengan robot lengan, seluruhnya memiliki satu derajat kebebasan. *Frame* merupakan sistem koordinat yang menggambarkan posisi sebuah *link* relatif terhadap *link* lainnya. Sistem koordinat ini melekat pada *link*. Penomoran *frame* sesuai dengan penomoran *link* yang dilekatinya. *Frame* ditempatkan pada *link* sehingga sumbu *Z* dari *frame $\{i\}$* yaitu Z_i , sejajar dengan sumbu putar (poros) dari *joint i* . Pendeklarasian frame ditunjukkan dalam gambar 2.4 berikut :



Gambar 2.4 Pendeklarasian frame pada Robot Lengan

Dari gambar diatas diketahui terdapat 3 parameter, X (arah putar), Z (poros) dan Y (*sum of product*). Beberapa parameter yang digunakan yaitu a_i , d_i , dan θ_i . Jarak dari Z_{i-1} ke Z_i diukur sepanjang X_i disebut a_i . Sudut antara Z_{i-1} dengan Z_i diukur sepanjang X_i disebut θ_i . Parameter d_i adalah jarak dari X_{i-1} ke X_i diukur sepanjang Z_i . Tabel 2.1 berikut ini menunjukkan parameter link dari lengan robot 5 DOF

Tabel 2.2 Parameter *Link*

| Sendi ke-i | a_i (cm) | α_i | d_i (cm) | θ_i |
|------------|------------|------------|------------|------------|
| 1 | a_1 | α_1 | d_1 | θ_1 |
| 2 | a_2 | α_2 | d_2 | θ_2 |
| 3 | a_3 | α_3 | d_3 | θ_3 |
| 4 | a_4 | α_4 | d_4 | θ_4 |

2.4 Orion Arm Robotic

Orion Arm Robotic merupakan robot lengan berbasis motor yang diproduksi di USA. Fitur *Orion Arm Robotic* didesain dengan kecepatan dan torsi yang lebih bagi pengguna yang ingin performa maksimal dari lengan robot mereka. Robot ini memiliki konstruksi bahan rangka yang seluruhnya terbuat dari alumunium yang kokoh dan kuat serta tahan terhadap korosi. Robot ini memiliki lima derajat kebebasan ditambah sebuah *gripper*.

Orion Arm Robotic menggunakan 6 buah motor servo sebagai aktuator. Yakni yang terdiri dari 1 servo untuk base, 1 servo untuk bahu, 1 servo untuk lengan, servo untuk siku, 1 servo pergelangan dan 1 servo *gripper* (penjapit). Secara keseluruhan, spesifikasi dari *Orion Arm Robotic* adalah sebagai berikut :

- Panjang keseluruhan (dari dasar) – 60.5cm
- Ketinggian Base – 12.5cm
- Joint 1 ke Joint 2 – 15.5cm
- Joint 2 ke Joint 3 – 13cm
- Joint 3 ke Joint 4 – 7.5cm
- Axis 4 sampai akhir gripper (terbuka) – 17.5cm
- Axis 4 sampai akhir gripper (ditutup) - 20cm
- Panjang keseluruhan dengan gripper ditutup – 60.5cm
- Max lebar 8cm dengan posisi gripper terbuka
- Memiliki 5 servo *Orion HV220* (4 sumbu + 1 gripper).

2.5 Motor Servo

Motor DC seringkali disebut juga sebagai motor servo walau dalam realitanya berbeda dengan motor DC. Motor servo merupakan motor DC yang mempunyai kualitas tinggi, sudah dilengkapi dengan sistem kontrol di dalamnya. Dalam aplikasi motor servo

sering digunakan sebagai kontrol loop tertutup untuk menangani perubahan posisi secara tepat dan akurat. Begitu juga dengan pengaturan kecepatan dan percepatan (Hughes, 2006).

Motor servo mampu bekerja dua arah (*CW* dan *CCW*) di mana arah dan sudut pergerakan rotornya dapat dikendalikan hanya dengan memberikan pengaturan *duty cycle* sinyal *PWM* pada bagian pin kontrolnya. Gambar 2.5 menunjukkan gambar motor servo.

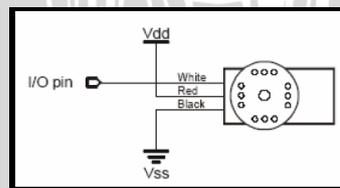


Gambar 2.5 Orion Robotics Digital Servo HV220 (Sumber : nodna.de)

Motor servo merupakan motor yang berputar lambat, di mana biasanya ditunjukkan oleh *rate* putarannya yang lambat, namun demikian memiliki torsi yang kuat karena *internal gear*-nya. Karakteristik motor servo adalah sebagai berikut :

- Memiliki 3 jalur kabel : *power*, *ground*, dan *control*.
- Pin *Control* untuk mengendalikan posisi.
- Operasional dari servo motor dikendalikan oleh sebuah pulsa selebar ± 20 ms, di mana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari *range* sudut maksimum.
- Konstruksi didalamnya meliputi *internal gear*, *potensiometer*, dan *feedback control*.

Gambar 2.6 menunjukkan konfigurasi pin pengkabelan motor servo.



Gambar 2.6 Konfigurasi Pin Pengkabelan Motor Servo (Sumber : Parallax, Inc.)

Secara umum terdapat dua jenis motor servo, yaitu :

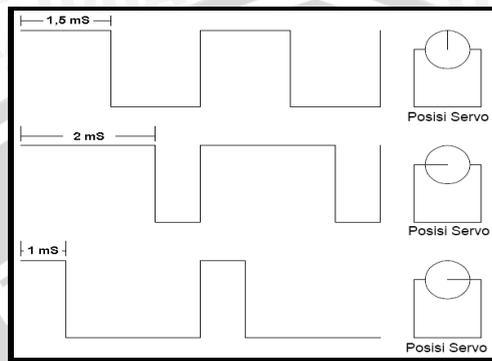
- Motor Servo *Standar 180°*

Motor servo jenis ini merupakan motor yang hanya mampu bergerak dua arah (*CW* dan *CCW*) dan mempunyai *defleksi* masing-masing sudut mencapai 90° sehingga total *defleksi* sudut dari kanan – tengah – kiri adalah 180° .

- Motor Servo *Continuous*

Motor servo jenis ini mampu bergerak dua arah (*CW* dan *CCW*) dan tanpa batasan defleksi sudut putar (dapat berputar secara kontinyu) sehingga motor ini berputar 360°.

Pengaturan sudut motor servo diperlukan untuk mengetahui gerakan dari motor servo dan pulsa yang harus diberikan untuk bergerak ke kanan atau bergerak ke kiri. Gambar 2.7 menunjukkan teknik PWM untuk mengatur sudut motor servo (Pitowarno, 2006).



Gambar 2.7 Pengaturan Sudut Motor Servo (Sumber : Parallax, Inc.)

Dalam Gambar 2.7 diasumsikan bahwa saat diberikan sinyal periodik dengan lebar 1 ms maka motor servo akan bergerak dengan sudut 0°, jika diberi sinyal 1,5 ms maka motor servo akan bergerak dengan sudut 90°, dan jika diberi sinyal 2 ms maka motor servo akan bergerak dengan sudut 180°. Dari kondisi tersebut bisa ditarik kesimpulan bahwa untuk mengatur sudut putaran motor servo diperlukan sinyal dengan lebar pulsa seperti pada persamaan berikut:

$$S = D \left(\frac{1000}{180} \right) + 1000 \mu s \quad \dots\dots(2.25)$$

Dimana:

S = Lebar pulsa dalam μs

D = Sudut putar servo dalam derajat

2.6 Arduino UNO

Arduino UNO adalah sebuah *board* mikrokontroler yang didasarkan pada ATmega328. Arduino UNO mempunyai 14 pin digital *input/output* (6 di antaranya dapat digunakan sebagai *output* PWM), 6 *input* analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah *power jack*, sebuah ICSP *header*, dan sebuah tombol reset. Arduino UNO memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah computer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya.

Spesifikasinya sebagai berikut:



| | |
|-----------------------------|------------------------------------|
| Microcontroller | ATmega328 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limits) | 6-20V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 40 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (ATmega328) |
| SRAM | 2 KB (ATmega328) |
| EEPROM | 1 KB (ATmega328) |
| Clock Speed | 16 MHz |



Gambar 2.8 ARDUINO UNO (Sumber: protostack.com)

2.6.1 Daya

Pin-pin dayanya adalah sebagai berikut:

- VIN. Tegangan input ke Arduino *board* ketika board sedang menggunakan sumber suplai eksternal (seperti 5 Volt dari koneksi USB atau sumber tenaga lainnya yang diatur). Dapat menyuplai tegangan melalui pin ini, atau jika penyuplaian tegangan melalui *power jack*, aksesnya melalui pin ini.
- 5V. Pin *output* ini merupakan tegangan 5 Volt yang diatur dari regulator pada *board*. *Board* dapat disuplai dengan salah satu suplai dari DC *power jack* (7-12V), USB *connector* (5V), atau pin VIN dari *board* (7-12). Penyuplaian tegangan melalui pin 5V atau 3,3V *bypass* regulator, dan dapat membahayakan *board*. Hal itu tidak dianjurkan.

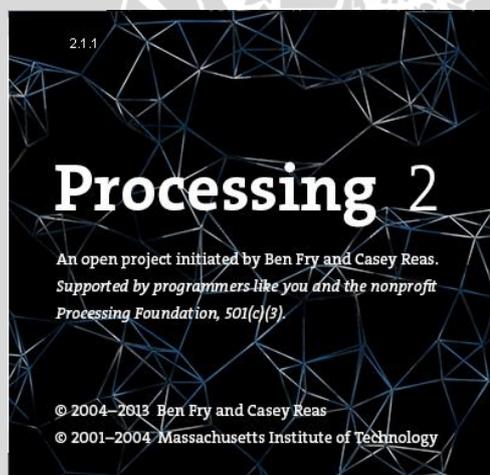
- 3V3. Sebuah suplai 3,3 Volt dihasilkan oleh regulator pada *board*. Arus maksimum yang dapat dilalui adalah 50 mA.
- GND. Pin *ground*.

2.6.2 Pemrograman

Arduino UNO dapat diprogram dengan software *Arduino*. *ATmega 328* pada *Arduino UNO* dilengkapi dengan *bootloader* yang memungkinkan untuk meng-*upload* kode baru untuk tanpa menggunakan programmer *hardware* eksternal (Hendriono, 2014).

2.7 Processing 2.1.1.

Processing adalah bahasa pemrograman dan lingkungan pemrograman (*development environment*) *open source* untuk memrogram gambar, animasi, game dan interaksi. *Processing* digunakan untuk belajar membuat prototipe dan produksi. *Processing* juga digunakan untuk mengajarkan dasar-dasar pemrograman komputer dalam konteks rupa dan berfungsi sebagai buku sketsa perangkat lunak dan *tools* produksi profesional. Secara singkatnya, *Processing* adalah sebuah aplikasi perangkat lunak yang memungkinkan penggunanya untuk dapat menulis, mengedit, meng-*compile*, serta menjalankan *Java Code*.

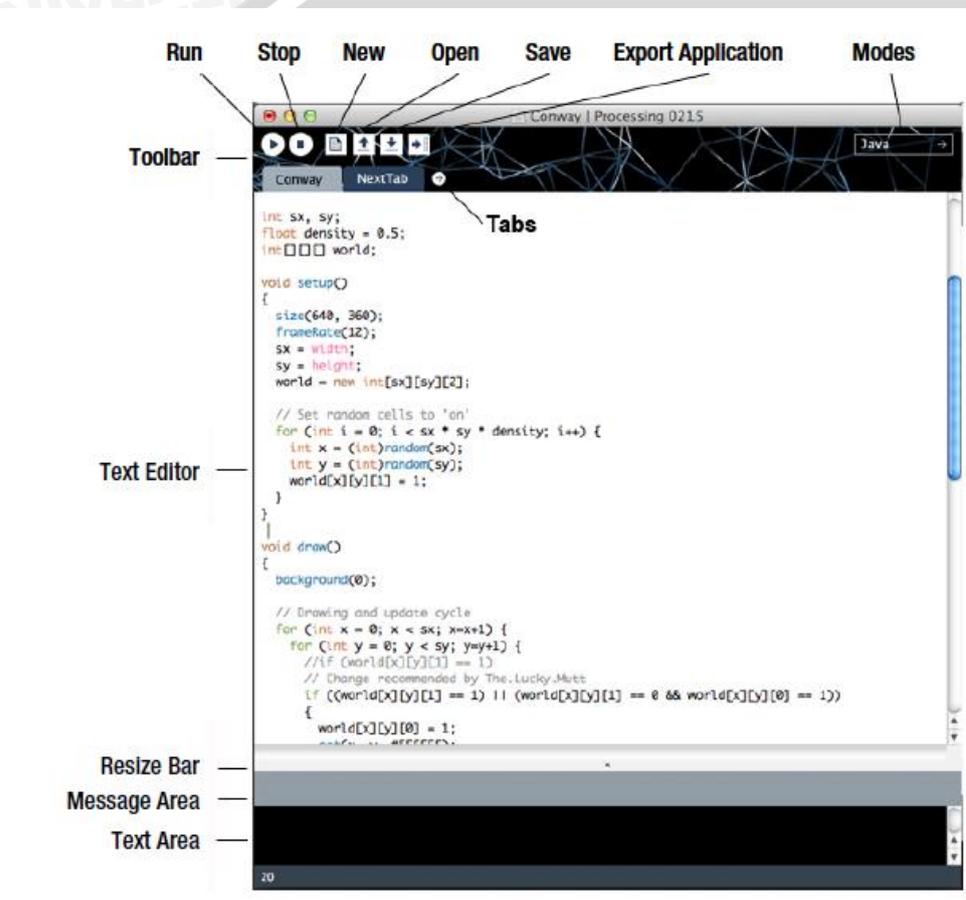


Gambar 2.9 Processing 2.1.1 (Sumber : processing.org)

Sistem antarmuka yang dibangun menggunakan *Processing* dapat menyederhanakan proses pemrograman. Akan tetapi hal ini tidak dimaksudkan untuk mengurangi kompleksitas dari proses berkreasi dan kreativitas pengguna. *Processing* merupakan aplikasi yang *portable*. Pengguna dapat menampilkannya hanya dengan melakukan double klik dan dapat langsung memulai *coding*. Banyak bahasa pemrograman

dan lingkungan pemrograman (*development environment*) yang membutuhkan integrasi yang cukup rumit dengan pengaturan sistem operasi untuk dapat bekerja. *Processing* sangat kontras jika dibandingkan dengan banyak bahasa pemrograman dan lingkungan pemrograman (*development environment*) yang umum digunakan karena tidak terlalu membutuhkan kompleksitas integrasi dan *coding* (Nugroho, 2012).

Processing merupakan bahasa pemrograman yang berfokus pada sebuah pemrograman kreatif. Dengan desain tampilan yang minimalis, *Processing* merupakan sebuah teks editor yang sederhana. Berikut ini adalah deskripsi mengenai elemen-elemen *toolbar* yang ada pada *Processing*:



Gambar 3.0 Tampilan utama Processing 2.1.1 (Sumber : Greenberg, Ira. 2013 : 13)

BAB III

METODE PENELITIAN

Kajian dalam skripsi ini merupakan penelitian yang bersifat aplikatif, yaitu merancang suatu sistem pengaturan pergerakan robot lengan *Orion Robotic Arm* melalui pendekatan *Geometry - Based Kinematic* dengan mikrokontroler *Arduino UNO* dan antarmuka menggunakan *Processing*. Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang akan dibuat adalah perancangan sistem, realisasi pembuatan sistem, pengujian dan analisa data, pengambilang kesimpulan dan saran.

3.1. Perancangan Sistem

Perancangan sistem dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektronik pendukungnya, hal ini dimaksudkan agar sistem pengendalian pergerakan robot lengan *Orion Arm Robotic* dapat berjalan sesuai dengan deskripsi awal yang telah direncanakan. Perancangan sistem yang dilakukan menggunakan peralatan di Laboratorium Mekatronika dan Robotika serta Laboratorium Sistem Kontrol Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, meliputi:

Penentuan sistem yang digunakan.

- Penentuan spesifikasi sistem yang akan dibuat, meliputi :
 - a. Penentuan deskripsi kerja sistem secara keseluruhan.
 - b. Rangkaian elektronik pendukung.
- Perancangan program pada *Arduino UNO* sebagai pusat pengontrol sistem.

Pelaksanaan perancangan sistem dilakukan di Laboratorium Mekatronika dan Robotika dan Laboratorium Sistem Kontrol Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.

3.2. Realisasi Pembuatan Sistem

Untuk merealisasikan alat atau sistem yang telah dirancang sebelumnya maka perlu diperhatikan beberapa tahapan dalam pembuatan alat tersebut, diantaranya yaitu :

- Penyempurnaan mekanik robot lengan *Orion Arm Robotic* secara keseluruhan yang meliputi penentuan batasan area kerja robot lengan *Orion Arm Robotic*.
- Pembuatan program pengontrolan menggunakan *Arduino UNO* dan *Processing*.

3.3. Pengujian dan Analisa Data

Setelah semua komponen pada alat sudah terhubung sesuai dengan diagram blok sistem yang telah dirancang dan perangkat lunak untuk mendukung sistem telah dibuat, maka diadakan pengujian dan analisa alat. Metode pengujian alat adalah sebagai berikut:

1. Pengujian pada motor servo yang digunakan.

Pengujian motor servo dilakukan dengan cara mencatat dan membandingkan hasil pembacaan sudut motor servo yang terukur menggunakan busur dengan sudut motor servo yang diinginkan dalam controller Arduino Uno.

2. Pengujian pada tiap-tiap blok rangkaian.

Pengujian tersebut meliputi pengujian *forward kinematic* dan *invers kinematic*.

3. Mengadakan pengujian sistem secara keseluruhan.

Pengujian ini dilakukan dengan cara menggabungkan semua bagian alat yang dibuat dan melihat kinerja alat. Pengujian ini bertujuan untuk mengetahui kinerja alat yang dibuat dan memberikan analisa terhadap kinerja alat. Mekanismenya, pengujian dilakukan dengan beberapa macam posisi yang berbeda-beda. Pengujian tersebut dilakukan dengan menggunakan algoritma *invers kinematic* dan *forward kinematic* yang sudah ditentukan sebelumnya untuk melihat apakah sistem bisa berjalan tanpa ada *error* atau dengan *error* seminimal mungkin dengan penggunaan algoritma yang sama dengan cara mengubah posisi awal.

4. Mengevaluasi hasil pengujian sistem secara keseluruhan.

Evaluasi dilakukan untuk mengetahui kinerja alat secara keseluruhan untuk mengetahui apa yang menjadi kekurangan dalam sistem tersebut.

3.4. Pengambilan Kesimpulan dan Saran

Kesimpulan diambil berdasarkan data yang diperoleh dari pengujian sistem secara keseluruhan. Jika hasil yang didapatkan telah sesuai dengan yang direncanakan sebelumnya, maka sistem kendali tersebut telah berhasil memenuhi harapan dan tentunya memerlukan pengembangan lebih lanjut untuk penyempurnaan.

BAB IV

PERANCANGAN SISTEM DAN PEMBUATAN PROGRAM

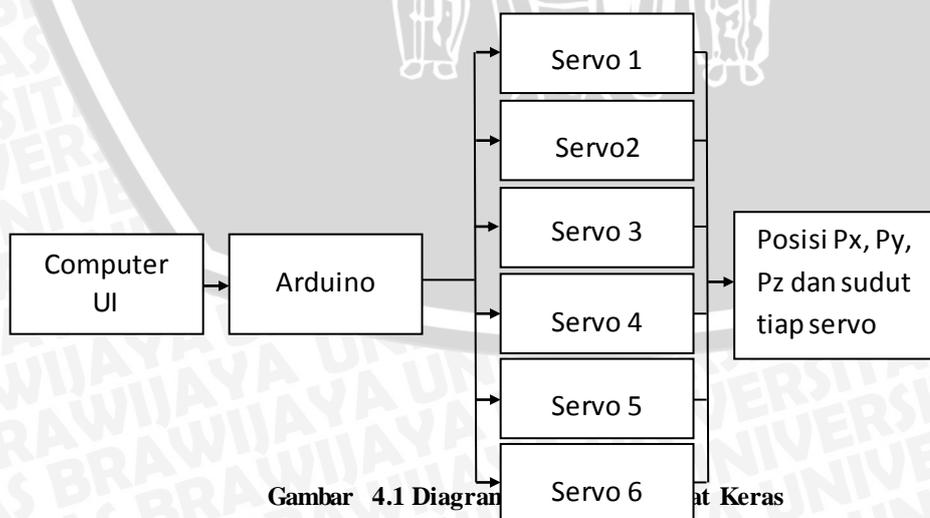
4.1 Perancangan Sistem

Perancangan sistem terlebih dahulu dilakukan agar memudahkan pemahaman tentang pengendalian dan prinsip kerja dari robot lengan. Perancangan sistem robot lengan meliputi:

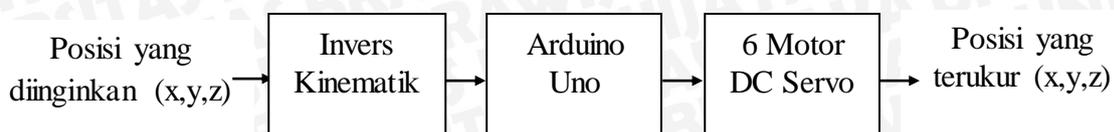
- Diagram balok dan prinsip kerja robot lengan.
- Deskripsi *frame* robot lengan

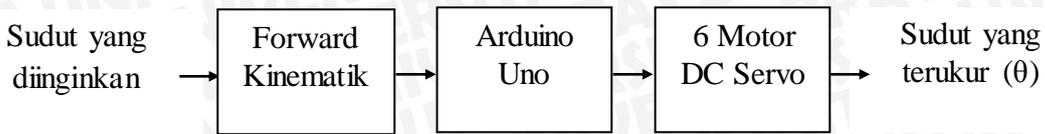
4.1.1 Diagram Balok dan Prinsip Kerja Robot Lengan

Robot lengan ini pada dasarnya memiliki lima *degree of freedom* (DOF) yang akan dimanfaatkan untuk menerapkan metode *forward* dan *invers* pada skripsi ini. Gambar 4.1 adalah diagram balok perangkat keras sedangkan Gambar 4.2 adalah diagram balok sistem pengaturan. Komputer sebagai antarmuka dengan pengguna. Pengguna dapat menggunakan dua metode yaitu *forward kinematic* dan *invers kinematic*. Blok mikrokontroler yaitu sebuah modul Arduino UNO (seperti tampak dalam Gambar 4.1) yang memiliki fungsi sebagai *driver* untuk mengatur posisi sudut motor servo. Motor servo sebagai aktuator penggerak *end-effector*. *End-effector* adalah perangkat pada ujung robot lengan, yang dirancang untuk berinteraksi dengan lingkungan (berupa penjepit). Posisi *end-effector* dapat dilihat berdasarkan papan penunjuk koordinat *end-effector*.



Gambar 4.1 Diagram Perangkat Keras





Gambar 4.2 Diagram Balok Sistem Pengaturan

Dalam Gambar 4.3 dan Gambar 4.4 berikut ini diperlihatkan mekanik robot lengan *Orion Arm Robotic* dan *Arduino UNO*



Gambar 4.3 Orion Arm Robotic AX-12A (Sumber: nodna.de)



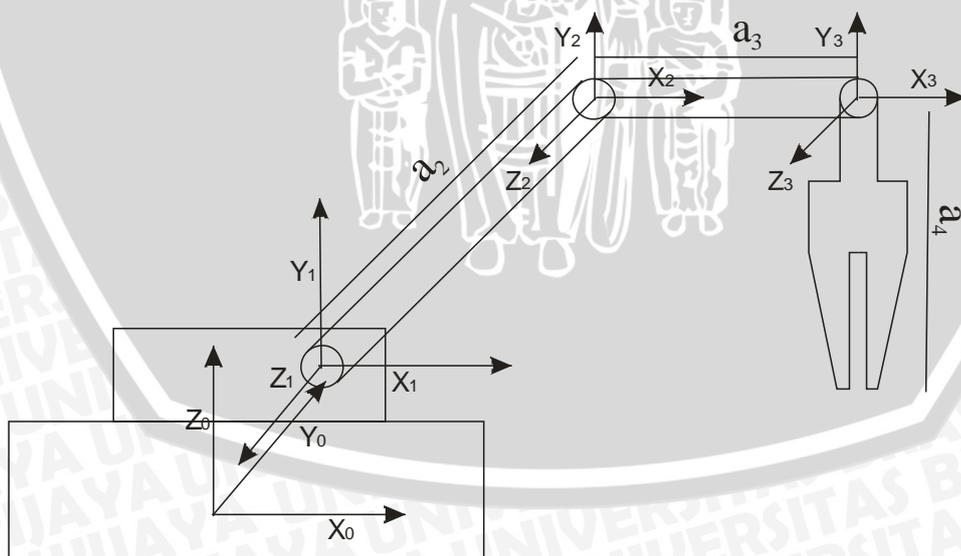
Gambar 4.4 Modul Arduino UNO (protostack.com)

4.1.2 Deskripsi *Frame Robot Lengan*

Perancangan program dilakukan dengan terlebih dahulu mendefinisikan *frame* pada tiap-tiap *link* pada robot lengan, berikut pendefinisian *frame*. Pada robot *Orion Arm Robotic*. Bagian-bagian pada robot lengan diantaranya batang-batang logam aluminium

yang berfungsi sebagai link, enam buah motor servo yang berfungsi sebagai aktuator dan sebuah motor servo berfungsi sebagai penjapit pada *end effector*. Pengendalian lengan dibantu dengan suatu sistem mikrokontroler dan PC sebagai *interface*.

Pada skripsi ini hanya memanfaatkan lima derajat kebebasan. Sistem koordinat robot lengan mengacu pada standar yang digunakan Mark W. Spong, 2004, setiap *link* diberi nomor dimulai dari bagian *basement* yang bergerak. *Basement* disebut *link 0*. Batang yang bergerak dan berhubungan dengan *basement* disebut *link 1*. Batang yang bergerak dan berhubungan dengan *link 1* disebut *link 2* dan seterusnya. Sehingga secara keseluruhan lengan robot terbentuk dari *link 0*, *link 1*, *link 2*, dan *end-effector*. Antara *link* yang satu dengan *link* tetangganya dihubungkan dengan *joint*. *Joint i+1* adalah *joint* yang menghubungkan *link i* dengan *link i+1*. Sehingga *joint 1* menghubungkan *link 0* dan *link 1*. Sedangkan *link 1* dan *link 2* dihubungkan dengan *joint 2* dan seterusnya. Dari keempat *joint* yang dimiliki lengan robot lengan, seluruhnya memiliki satu derajat kebebasan. *Frame* merupakan sistem koordinat yang menggambarkan posisi sebuah *link* relatif terhadap *link* lainnya. Sistem koordinat ini melekat pada *link*. Penomoran *frame* sesuai dengan penomoran *link* yang dilekatinya. *Frame* ditempatkan pada *link* sehingga sumbu *Z* dari *frame {i}* yaitu Z_i , sejajar dengan sumbu putar (poros) dari *joint i*. Sesuai dengan gambar 2.4, berikut ini pendeklarasian *frame* robot lengan seperti yang tampak pada Gambar 4.5 :



Gambar 4.5 Pendeklarasian *Frame* Pada Robot Lengan Orion

Beberapa parameter yang digunakan yaitu a_i , α_i , d_i , dan θ_i . Jarak dari Z_{i-1} ke Z_i diukur sepanjang X_i disebut a_i . Sudut antara Z_{i-1} dengan Z_i diukur sepanjang X_i disebut α_i .

Parameter d_i adalah jarak dari X_{i-1} ke X_i diukur sepanjang Z_i . Parameter θ_i adalah sudut antara X_{i-1} dengan X_i diukur sepanjang Z_i . Sesuai Tabel 2.2, Tabel 4.1 menunjukkan parameter *link* dari lengan robot sesuai dengan peletakan *frame* yang ditunjukkan dalam Gambar 4.5.

Tabel 4.1 Parameter *Orion Robotic Arm*

| Sendi ke-i | a_i (cm) | α_i | d_i (cm) | θ_i |
|------------|------------|------------|------------|------------|
| 1 | 0 | 90 | 0 | θ_1 |
| 2 | 15.5 | 0 | 0 | θ_2 |
| 3 | 13 | 0 | 0 | θ_3 |
| 4 | 19.5 | 0 | 0 | θ_4 |

Sesuai tabel 2.1, didapatkan matrik homogen sebagai berikut :

$${}^0A_1 = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ \sin\theta_1 & 0 & -\cos\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots(4.1)$$

$${}^1A_2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2 \cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2 \sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots(4.2)$$

$${}^2A_3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & a_3 \cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & a_3 \sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots(4.3)$$

$${}^3A_4 = \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & a_4 \cos\theta_4 \\ \sin\theta_4 & \cos\theta_4 & 0 & a_4 \sin\theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots(4.4)$$

4.2 Perancangan Kinematika Robot Lengan

Kinematika robot lengan merupakan pengetahuan atau teori mengenai pergerakan objek tanpa memperhitungkan gaya-gaya yang menyebabkan objek itu bergerak. Kinematika robot lengan terdiri dari pergerakan rotasi dan translasi. Pada penelitian ini, tinjauan kinematika akan digunakan dalam menentukan perhitungan *Forward Kinematic* dan *Invers Kinematic*.

4.2.1 Forward Kinematic

Secara garis besar *forward kinematic* adalah memberikan sudut kepada motor servo setelah itu didapatkan posisi *end-effector*. Dalam perancangan *forward kinematic*, *homogenous* matrik semua *link* dihitung terlebih dahulu sesuai pada Tabel 2.1. Berikut ini adalah penyusunan matrik transformasi homogen dari matrik homogen di tiap-tiap link yang akan dipergunakan dalam perhitungan *forward kinematic*.

$${}^0T_4 = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 = \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \dots(4.5)$$

```
2 - sin_teta1=sym('sin_teta1')
3 - sin_teta2=sym('sin_teta2')
4 - sin_teta3=sym('sin_teta3')
5 - sin_teta4=sym('sin_teta4')
6
7 - cos_teta1=sym('cos_teta1')
8 - cos_teta2=sym('cos_teta2')
9 - cos_teta3=sym('cos_teta3')
10 - cos_teta4=sym('cos_teta4')
11
12 - a2_cos_teta2=sym('a2_cos_teta2')
13 - a2_sin_teta2=sym('a2_sin_teta2')
14 - a3_cos_teta3=sym('a3_cos_teta3')
15 - a3_sin_teta3=sym('a3_sin_teta3')
16 - a4_cos_teta4=sym('a4_cos_teta4')
17 - a4_sin_teta4=sym('a4_sin_teta4')
18
19 - A1=[cos_teta1 0 sin_teta1 0 ;
20 -     sin_teta1 0 -cos_teta1 0 ;
21 -     0 1 0 0;
22 -     0 0 0 1]
23
24 - A2=[cos_teta2 -sin_teta2 0 a2_cos_teta2;
25 -     sin_teta2 cos_teta2 0 a2_sin_teta2;
26 -     0 0 1 0;
27 -     0 0 0 1]
28
29 - A3=[cos_teta3 -sin_teta3 0 a3_cos_teta3;
30 -     sin_teta3 cos_teta3 0 a3_sin_teta3;
31 -     0 0 1 0;
32 -     0 0 0 1]
33
34 - A4=[cos_teta4 -sin_teta4 0 a4_cos_teta4;
35 -     sin_teta4 cos_teta4 0 a4_sin_teta4;
36 -     0 0 1 0;
37 -     0 0 0 1]
38
39 - A1A2A3=A1*A2*A3*A4
40
```

Gambar 4.6 Tampilan perhitungan 0T_4

Dengan mengacu pada persamaan trigonometri dasar tentang jumlah dan selisih sesuai persamaan 2.18 hingga 2.21, maka didapatkan :

$$\begin{aligned} n_x &= \cos \theta_4 \cos \theta_1 (\cos (\theta_2 + \theta_3)) - \sin \theta_4 \cos \theta_1 (\sin (\theta_2 + \theta_3)) \\ n_y &= \cos \theta_4 \sin \theta_1 (\cos (\theta_2 + \theta_3)) - \sin \theta_4 \sin \theta_1 (\sin (\theta_2 + \theta_3)) \\ n_z &= \cos \theta_4 (\sin (\theta_2 + \theta_3)) + \sin \theta_4 (\cos (\theta_2 + \theta_3)) \end{aligned} \dots\dots\dots(4.6)$$

$$\begin{aligned} o_x &= \cos \theta_4 \cos \theta_1 (\sin (\theta_2 + \theta_3)) + \sin \theta_4 \cos \theta_1 (\cos (\theta_2 + \theta_3)) \\ o_y &= \cos \theta_4 \sin \theta_1 (\sin (\theta_2 + \theta_3)) + \sin \theta_4 \sin \theta_1 (\cos (\theta_2 + \theta_3)) \\ o_z &= \sin \theta_4 (\sin (\theta_2 + \theta_3)) - \cos \theta_4 (\cos (\theta_2 + \theta_3)) \end{aligned} \dots\dots\dots(4.7)$$

$$\begin{aligned} a_x &= \sin \theta_1 \\ a_y &= -\cos \theta_1 \\ a_z &= 0 \end{aligned} \dots\dots\dots(4.8)$$

$$\begin{aligned} P_x &= a_4 \cos \theta_4 \cos \theta_1 (\cos (\theta_2 + \theta_3)) - a_4 \sin \theta_4 \cos \theta_1 (\sin (\theta_2 + \theta_3)) + \\ &\quad a_3 \cos \theta_1 (\cos (\theta_2 + \theta_3)) + a_2 \cos \theta_2 \cos \theta_1 \\ P_y &= a_4 \cos \theta_4 \sin \theta_1 (\cos (\theta_2 + \theta_3)) - a_4 \sin \theta_4 \sin \theta_1 (\sin (\theta_2 + \theta_3)) + \\ &\quad a_3 \sin \theta_1 (\cos (\theta_2 + \theta_3)) + a_2 \cos \theta_2 \sin \theta_1 \\ P_z &= (a_4 \cos \theta_4 \sin (\theta_2 + \theta_3)) + (a_4 \sin \theta_4 \cos (\theta_2 + \theta_3)) + \\ &\quad (a_3 \sin (\theta_2 + \theta_3)) + (a_2 \sin \theta_2) \end{aligned} \dots\dots\dots(4.9)$$

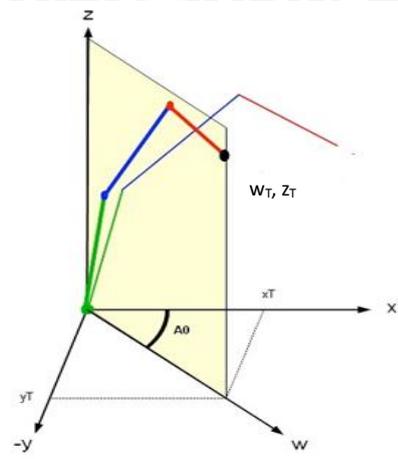
Matrik *forward* memberikan informasi orientasi dan posisi dari *end-effector*. Pada skripsi ini tidak membahas mengenai orientasi *end-effector* hanya posisi *end-effector*. Dapat dilihat pada persamaan diatas, posisi koordinat (Px, Py, Pz) *end-effector* dapat dihitung dengan memberikan masukan sudut pada motor servo untuk setiap *link*. Pergerakan robot merupakan kombinasi perubahan θ_1 , θ_2 , θ_3 , dan θ_4 secara bersamaan. Namun melihat kondisi yang paling memungkinkan terkait keterbatasan fisik robot maka perubahan pada masing-masing sudut θ_1 , θ_2 , θ_3 , dan θ_4 dibatasi agar tidak merusak robot.

4.2.2 *Invers Kinematic*

Pada *invers kinematic*, perhitungan menggunakan aturan *trigonometry* untuk mendapatkan solusi sudut pada posisi tertentu *end-effector*. Aturan *trigonometry* berlaku untuk sudut dalam segitiga, jadi langkah awal yang perlu dilakukan adalah membentuk segitiga pada tiap *link*.



- a. Pada mulanya yang dilakukan adalah pembentukan segitiga pada sumbu x dan sumbu y. Didapatkan sudut dasar rotate (A_0) sebagai fungsi arctan (\tan^{-1}) sederhana dari target aktuator yang diinginkan X_T dan Y_T koordinat.



Gambar 4.7 Segitiga pada Sumbu X dan Y

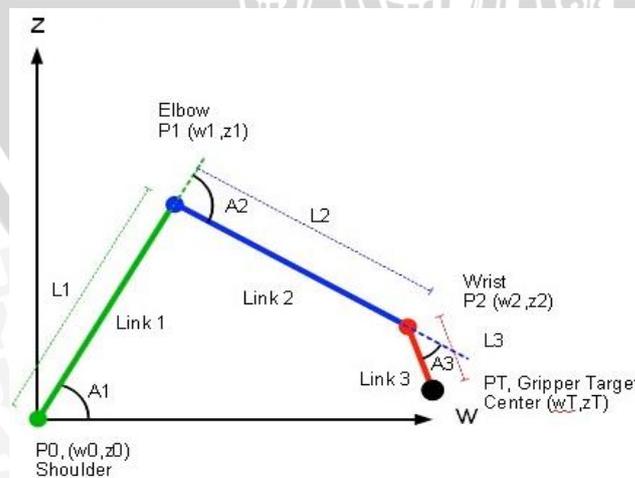
$$A_0 = \tan^{-1}(Y_T / X_T) \quad \dots\dots(4.10)$$

$$W_T = \sqrt{Y_T^2 + X_T^2} \quad \dots\dots(4.11)$$

```
void inverse(int xT, int yT, int zT, int sudutgripper)
{
    wT = (sqrt(sq(yT)+sq(xT)));
    sudutbase = round( atan2 (yT, xT) * konversi2 );
}
```

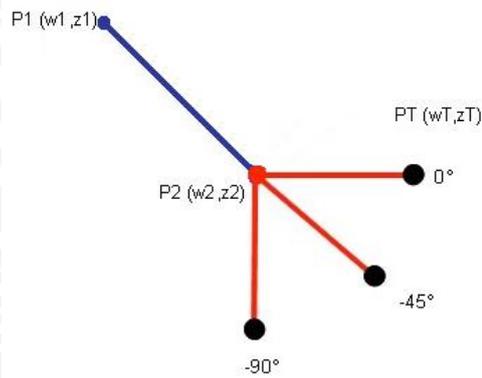
Gambar 4.8 Tampilan perhitungan A_0 dan W_T

- b. Setelah didapatkan nilai W_T , perhitungan dilakukan di bidang dua dimensi. Gambar 4.7 berikut menunjukkan lengan robot disederhanakan untuk 3 derajat kebebasan di bidang dua dimensi.



Gambar 4.9 Penyederhanaan di Bidang Sumbu W dan Z dengan 3 DOF

Posisi Gripper dapat ditentukan arahnya menggunakan permisalan sudut.



Gambar 4.10 Arah Gripper dengan Permisalan Sudut

- c. Jika sudah menentukan koordinat gripper sasaran, W_T , Z_T , dan sudut yang diinginkan untuk link aktuator (sudut gripper) maka dapat menggunakan trigonometri sederhana untuk menghitung koordinat W_2, Z_2

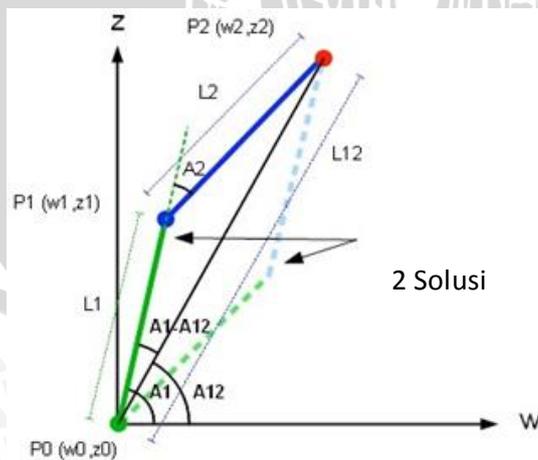
$$W_2 = W_T - \text{Link3} \cos(\text{posisi gripper}) \quad \dots\dots(4.12)$$

$$Z_2 = Z_T - \text{Link3} \sin(\text{posisi gripper}) \quad \dots\dots(4.13)$$

```
w2 = wT - link3*cos(-sudutgripper*konversil);
z2 = zT - link3*sin(-sudutgripper*konversil);
```

Gambar 4.11 Tampilan perhitungan W_2 dan Z_2

- d. Sekarang disederhanakan menjadi 2 DOF ditampilkan dalam diagram berikut. Di mana koordinat akhir Link 2 (W_2, Z_2) dan koordinat awal Link 1 (W_0, Z_0) dan hanya perlu menghitung sudut untuk kedua link.



Gambar 4.12 Penyederhanaan di Bidang Sumbu W dan Z dengan 2 DOF



A_1 , A_2 dan A_{12} adalah tiga sudut yang dibutuhkan. Menggunakan hukum cosinus untuk mendapatkan sudut A_1 - A_{12} karena memiliki panjang tiga sisi segitiga Link1, Link2 dan Link12.

$$L_{12} = \sqrt{W_2^2 + Z_2^2} \quad \dots\dots(4.14)$$

$$A_{12} = \tan^{-1}\left(\frac{Z_2}{W_2}\right) \quad \dots\dots(4.15)$$

```
link12 = sqrt(sq(w2)+sq(z2));
sudut12 = round( atan2 (z2, w2) * konversi2 );
```

Gambar 4.13 Tampilan perhitungan L_{12} dan A_{12}

$$A_1 = \cos^{-1} \frac{link1^2 + link12^2 - link2^2}{2 \times link1 \times link12} + A_{12} \quad \dots\dots(4.16)$$

$$W_1 = link1 \times \cos(A_1) \quad \dots\dots(4.17)$$

$$Z_1 = link1 \times \sin(A_1) \quad \dots\dots(4.18)$$

$$A_2 = \tan^{-1}\left(\frac{Z_2 - Z_1}{W_2 - W_1}\right) - A_1 \quad \dots\dots(4.19)$$

$$A_3 = \text{Posisi Gripper} - A_1 - A_2 \quad \dots\dots(4.20)$$

```
sum1 = (sq(link1) + sq(link12) - sq(link2));
sum2 = (2*link1*link12);
double sum3;
sum3 = sum1 / sum2;

sudut1 = (round (acos (sum3) * konversi2)) + sudut12;

w1 = link1*cos(sudut1*konversil);
z1 = link1*sin(sudut1*konversil);

sum4 = (z2-z1);
sum5 = (w2-w1);
double sum6;
sum6 = sum4 / sum5;

sudut2 = (round( atan(sum6) * konversi2 )) - sudut1;

sudut3 = -sudutgripper - sudut1 - sudut2;
```

Gambar 4.14 Tampilan perhitungan A_1 , W_1 , Z_1 , A_2 dan A_3

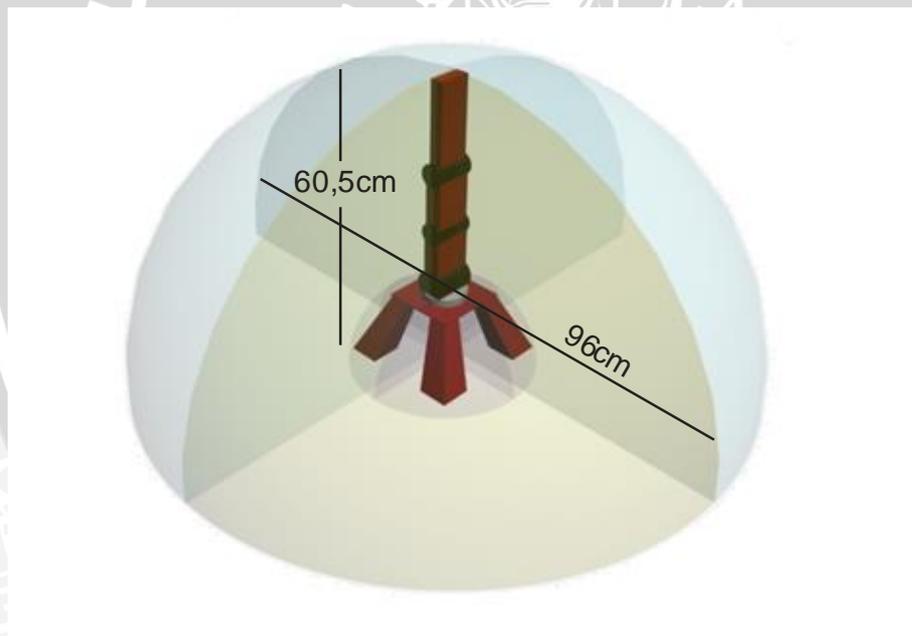
4.3 Perancangan Pergerakan Robot Lengan



Perancangan pergerakan robot lengan meliputi penentuan area kerja robot lengan yang dipergunakan. Selain itu diperlukan juga penentuan desain dimensi atau ukuran medan robot lengan. Hal ini dimaksudkan agar pengguna dapat mengetahui daerah jangkauan minimum mdan maksimum gerakan robot lengan pada medan yang telah ditentukan.

4.3.1 *Workspace Robot Lengan*

Workspace robot lengan adalah total luas yang memungkinkan terlewati (tersapu) oleh gerakan robot lengan. *Workspace* dapat ditentukan dengan cara menggerakkan tiap motor servo secara berurutan sehingga dapat mengetahui jangkauan robot lengan dengan keterbatasan fisik robot.. Gambar 4.9 menunjukkan *workspace Orion Robotic Arm* beserta panjang *link* dari posisi *base* hingga *gripper*. Sedangkan Gambar 4.10 memperlihatkan *workspace* robot lengan dengan jangkauan minimum dan maksimum pergerakan robot lengan tersebut.

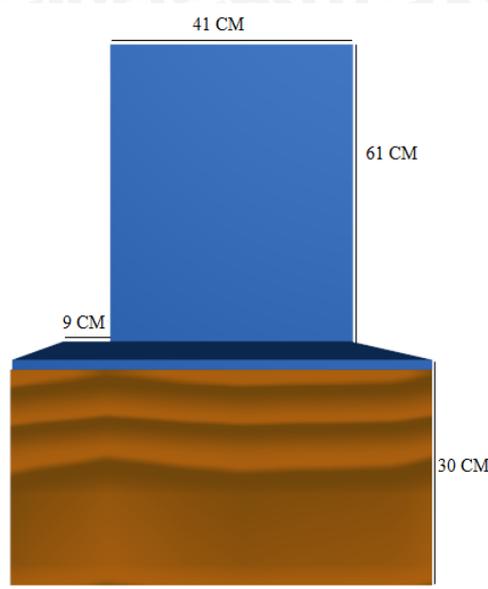


Gambar 4.15 *Workspace Robot Lengan (Perancangan)*

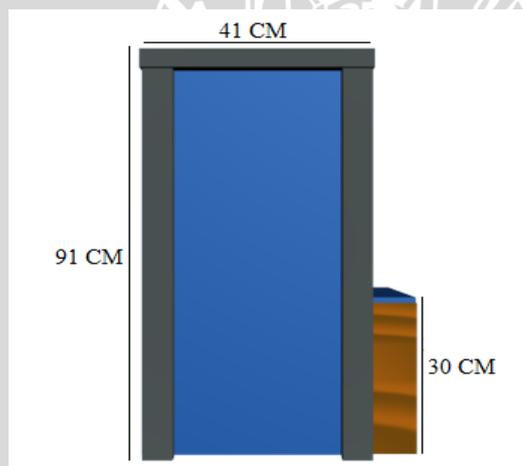
Motor servo yang digunakan memiliki keterbatasan sudut yaitu antara -90 derajat sampai 90 derajat. Motor servo *shoulder* digerakkan (dari sudut min hingga max) maka akan terbentuk setengah lingkaran dengan jari-jari 48 cm.

Dalam Gambar 4.11, menunjukan *workspace* robot lengan dengan batasan medan. Setelah diketahui *workspace* robot lengan maka dapat melakukan perancangan medan robot lengan untuk mengambil objek. Medan robot lengan kami sesuaikan dengan

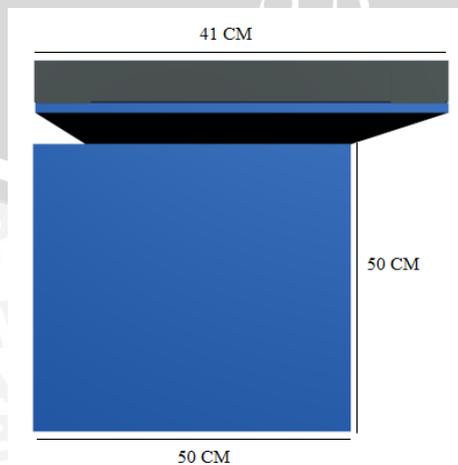
kemampuan jangkauan robot lengan. Penempatan papan difungsikan untuk mengukur posisi dari *end-effector*.



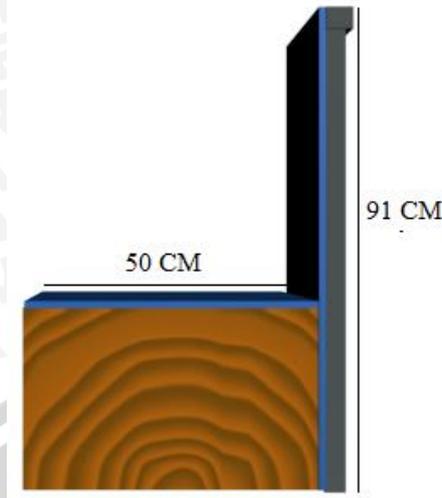
(a) Tampak depan



(b) Tampak belakang



(c) Tampak atas



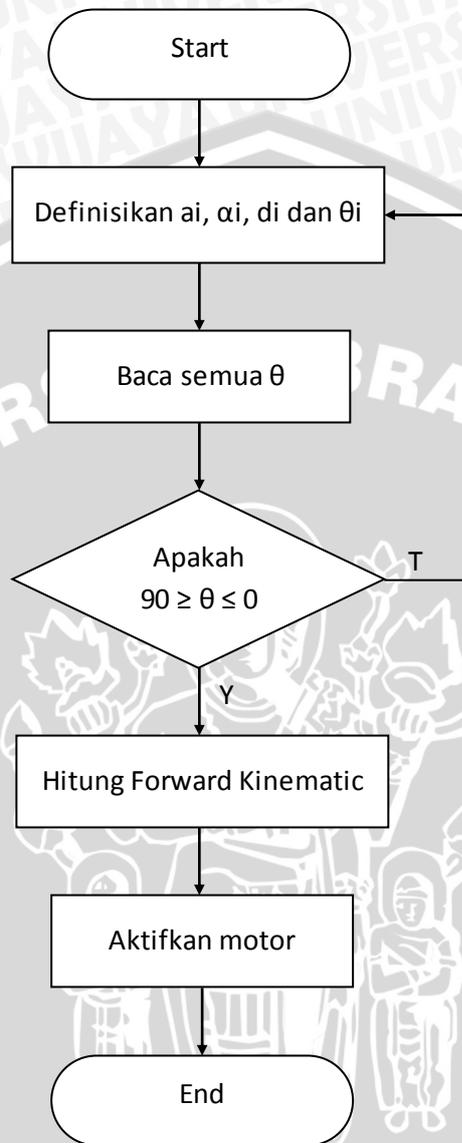
(d) Tampak samping

Gambar 4.16 Rancangan Medan Robot Lengan (Perancangan)

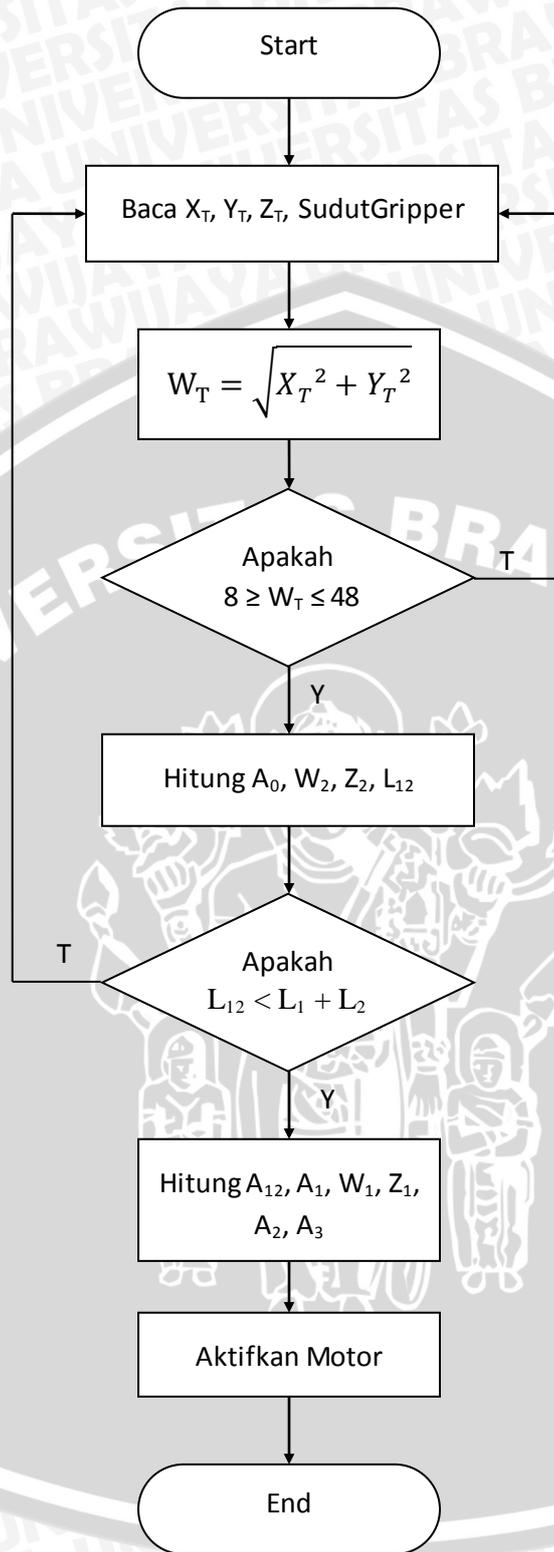
4.3.2 Skema Gerak Robot Lengan

Gerak robot lengan sangat dipengaruhi kemampuan motor servo robot lengan dan juga jangkauan posisi yang mungkin dituju oleh metode *invers kinematic* dengan solusi geometri (segitiga). Perancangan *invers kinematic* dengan solusi geometri (segitiga) diketahui memiliki jangkauan posisi antara 28.6 cm sampai 40 cm dari koordinat dasar, selama objek berada pada jangkauan tersebut maka robot lengan akan mampu mengambil objek tersebut. Objek diletakkan pada posisi dalam jangkauan, posisi penghalang dan objek diatur sesuai dengan kemampuan robot lengan. Posisi objek dapat ditaruh pada semua posisi dalam area solusi *invers kinematic*.

4.3.3 Perancangan Perangkat Lunak



Gambar 4.17 Flowchart Program Forward Kinematic



Gambar 4.18 Flowchart Program Invers Kinematic

BAB V

PENGUJIAN DAN ANALISIS SISTEM

Tujuan pengujian sistem ini adalah untuk menentukan apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perancangan. Pengujian pada sistem ini meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk menemukan letak kesalahan dan mempermudah analisis pada sistem apabila alat tidak bekerja sesuai dengan perancangan.

5.1 Pengujian *Forward Kinematic*

a. Tujuan

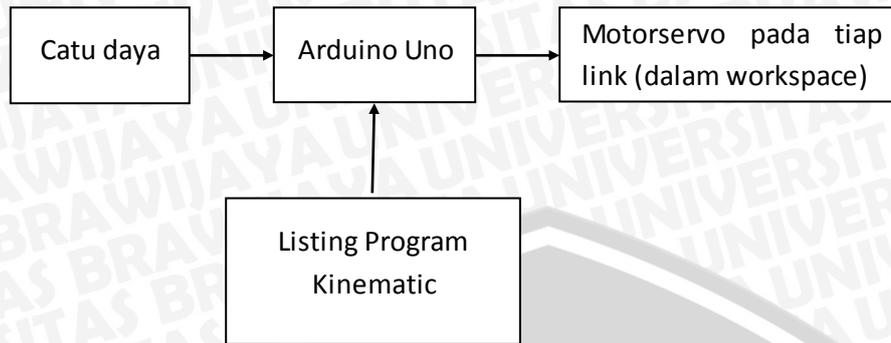
Untuk mengetahui ketelitian posisi *end-effector* dengan metode *forward kinematic*.

b. Peralatan yang digunakan

- *Orion Robotic Arm 5 DOF*
- *Arduino Uno*
- Catu daya 6 – 10 volt
- Kabel komunikasi serial *USB Type A to B* Komputer/laptop
- Penggaris siku
- Medan robot lengan

c. Langkah pengujian

1. Rangkai peralatan seperti dalam Gambar 5.1.
2. *Kinematic Software* dinyalakan dan diberikan nilai sudut yang diinginkan pada masing-masing *link*.
3. Perhitungan secara teoritis akan dilakukan oleh *software*
4. Pengukuran posisi *end-effector* dengan dua buah penggaris siku dan papan penunjuk posisi



Gambar 5.1 Blok Pengujian *Forward Kinematic*

d. Hasil pengujian

Hasil pengujian *forward kinematic* ditunjukkan dalam Tabel 5.1

Tabel 5.1 Pengujian *Forward Kinematic*

| No | Sudut (derajat) | | | | Posisi Pengukuran | | | Posisi Teori | | | Error | | |
|----|-----------------|------------|------------|------------|-------------------|---------|---------|--------------|---------|---------|---------|---------|---------|
| | θ_1 | θ_2 | θ_3 | θ_4 | Px (cm) | Py (cm) | Pz (cm) | Px (cm) | Py (cm) | Pz (cm) | Px (cm) | Py (cm) | Pz (cm) |
| 1 | 45 | 90 | -45 | -45 | 20.2 | 20.2 | 24.1 | 20.15 | 20.15 | 24.69 | -0.05 | -0.05 | 0.59 |
| | | | | | 20.3 | 20.3 | 24.0 | | | | -0.15 | -0.15 | 0.69 |
| | | | | | 20.2 | 20.2 | 24.2 | | | | -0.05 | -0.05 | 0.49 |
| | | | | | 20.2 | 20.1 | 24.2 | | | | -0.05 | 0.05 | 0.49 |
| | | | | | 20.2 | 20.2 | 24.2 | | | | -0.05 | -0.05 | 0.49 |
| | | | | | 20.3 | 20.1 | 24.1 | | | | -0.15 | 0.05 | 0.59 |
| | | | | | 20.2 | 20.2 | 24.2 | | | | -0.05 | -0.05 | 0.49 |
| 2 | 45 | 60 | -60 | -30 | 28.2 | 28.2 | 2.1 | 28.61 | 28.61 | 3.67 | 0.41 | 0.41 | 1.57 |
| | | | | | 28.1 | 28.1 | 2.2 | | | | 0.51 | 0.51 | 1.47 |
| | | | | | 28.1 | 28.1 | 2.1 | | | | 0.51 | 0.51 | 1.57 |
| | | | | | 28.2 | 28.2 | 2.0 | | | | 0.41 | 0.41 | 1.67 |
| | | | | | 28.1 | 28.1 | 2.1 | | | | 0.51 | 0.51 | 1.57 |
| | | | | | 28.1 | 28.1 | 2.1 | | | | 0.51 | 0.51 | 1.57 |
| | | | | | 28.1 | 28.1 | 2.1 | | | | 0.51 | 0.51 | 1.57 |
| 3 | 90 | 45 | -45 | -30 | 40.6 | 0.1 | 0.3 | 40.85 | 0 | 1.21 | 0.25 | -0.1 | 0.91 |
| | | | | | 40.7 | 0.2 | 0.2 | | | | 0.15 | -0.2 | 1.01 |
| | | | | | 40.7 | 0.1 | 0.2 | | | | 0.15 | -0.1 | 1.01 |
| | | | | | 40.8 | 0.1 | 0.2 | | | | 0.05 | -0.1 | 1.01 |
| | | | | | 40.8 | 0 | 0.1 | | | | 0.05 | 0 | 1.11 |
| | | | | | 40.7 | 0 | 0.2 | | | | 0.15 | 0 | 1.01 |
| | | | | | 40.8 | 0.2 | 0.1 | | | | 0.05 | -0.2 | 1.11 |



| No | Sudut (derajat) | | | | Posisi Pengukuran | | | Posisi Teori | | | Error | | |
|----|-----------------|------------|------------|------------|-------------------|---------|---------|--------------|---------|---------|---------|---------|---------|
| | θ_1 | θ_2 | θ_3 | θ_4 | Px (cm) | Py (cm) | Pz (cm) | Px (cm) | Py (cm) | Pz (cm) | Px (cm) | Py (cm) | Pz (cm) |
| 4 | 0 | 30 | -30 | 0 | 0.1 | 45.8 | 7.1 | 0 | 45.92 | 7.75 | -0.1 | 0.12 | 0.65 |
| | | | | | 0 | 45.9 | 7.0 | | | | 0 | 0.02 | 0.75 |
| | | | | | 0.1 | 45.8 | 7.1 | | | | -0.1 | 0.12 | 0.65 |
| | | | | | 0.1 | 45.8 | 7.2 | | | | -0.1 | 0.12 | 0.55 |
| | | | | | 0 | 45.9 | 7.1 | | | | 0 | 0.02 | 0.65 |
| | | | | | 0.1 | 45.8 | 7.1 | | | | -0.1 | 0.12 | 0.65 |
| | | | | | 0.1 | 45.9 | 7.1 | | | | -0.1 | 0.02 | 0.65 |

Dari hasil pengujian dapat dilihat rata-rata *error* posisi pada sumbu x yaitu sebesar ± 0.11321 cm, pada sumbu y yaitu sebesar ± 0.10571 cm, sedangkan rata-rata *error* yang terjadi pada sumbu z yaitu ± 0.94785 cm.

Contoh perhitungan tabel 5.1 (lihat tabel 4.1 dan persamaan 4.13):

$$a_2 = 15.5, a_3 = 13, a_4 = 19.5$$

$$\theta_1 = 45^\circ$$

$$\theta_2 = 90^\circ$$

$$\theta_3 = -45^\circ$$

$$\theta_4 = -45^\circ$$

$$P_x = a_2 \cos \theta_2 \cos \theta_1 + a_3 \cos \theta_1 (\cos (\theta_2 + \theta_3)) + a_4 \cos \theta_4 \cos \theta_1 (\cos (\theta_2 + \theta_3)) - a_4 \sin \theta_4 \cos \theta_1 (\sin (\theta_2 + \theta_3))$$

$$P_y = a_2 \cos \theta_2 \sin \theta_1 + a_3 \sin \theta_1 (\cos (\theta_2 + \theta_3)) + a_4 \cos \theta_4 \sin \theta_1 (\cos (\theta_2 + \theta_3)) - a_4 \sin \theta_4 \sin \theta_1 (\sin (\theta_2 + \theta_3))$$

$$P_z = a_2 \sin \theta_2 + a_3 (\sin (\theta_2 + \theta_3)) + a_4 \cos \theta_4 (\sin (\theta_2 + \theta_3)) + a_4 \sin \theta_4 (\cos (\theta_2 + \theta_3))$$

$$P_x = 15.5 \cos 90 \cos 45 + 13 \cos 45 (\cos(90 - 45)) + 19.5 \cos (-45) \cos 45 (\cos(90 - 45)) - 19.5 \sin (-45) \cos 45 (\sin(90 - 45)) = 20.15 \approx 20$$

$$P_y = 15.5 \cos 90 \sin 45 + 13 \sin 45 (\cos(90 - 45)) + 19.5 \cos (-45) \sin 45 (\cos(90 - 45)) - 19.5 \sin (-45) \sin 45 (\sin(90 - 45)) = 20.15 \approx 20$$

$$P_z = 15.5 \sin 90 + 13 (\sin (90 - 45)) + 19.5 \cos (-45) (\sin (90 - 45)) + 19.5 \sin (-45) (\cos (90 - 45)) = 24.69 \approx 25$$

Untuk selanjutnya, nilai perhitungan secara teoritis dihitung dengan program sederhana.

5.2 Pengujian *Invers Kinematic*

a. Tujuan

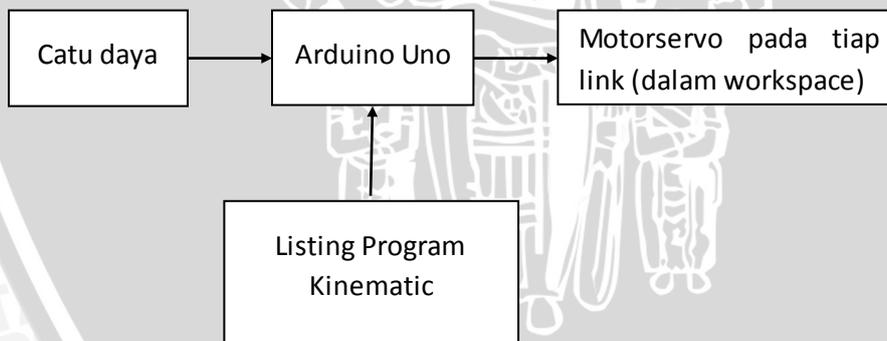
Untuk mengetahui ketelitian posisi *end-effector* dengan metode *invers kinematic*.

b. Peralatan yang digunakan

- *Orion Robotic Arm 5 DOF*
- *Arduino Uno*
- Catu daya 6 – 10 volt
- Kabel komunikasi serial *USB Type A to B* Komputer/laptop
- Busur
- Medan lengan robot

c. Langkah pengujian

1. Rangkai peralatan seperti dalam Gambar 5.2.
2. *Kinematic Software* dinyalakan dan diberikan nilai posisi yang diinginkan pada *end-effector*.
3. Perhitungan secara teoritis akan dilakukan oleh *software*.
4. Ukur sudut tiap link dengan busur.



Gambar 5.2 Blok Pengujian *Invers Kinematic*

c. Hasil pengujian

Pada Tabel 5.2 Menunjukkan hasil pengujian *invers kinematic*

Tabel 5.2 Pengujian *Invers Kinematic*

| No | Posisi Tujuan | | | Solusi Sudut | | | | Sudut Terukur | | | | Error | | | |
|----|---------------|---------|---------|--------------|------------|------------|------------|---------------|------------|------------|------------|------------|------------|------------|------------|
| | Px (cm) | Py (cm) | Pz (cm) | θ_1 | θ_2 | θ_3 | θ_4 | θ_1 | θ_2 | θ_3 | θ_4 | θ_1 | θ_2 | θ_3 | θ_4 |
| 1 | 20 | 20 | 25 | 45 | 90 | -45 | -45 | 46 | 91 | -46 | -46 | -1 | -1 | 1 | 1 |
| | | | | | | | | 46 | 91 | -47 | -46 | -1 | -1 | 2 | 1 |
| | | | | | | | | 45 | 89 | -47 | -46 | 0 | 1 | 2 | 1 |
| | | | | | | | | 45 | 92 | -47 | -45 | 0 | -2 | 2 | 0 |
| | | | | | | | | 45 | 89 | -47 | -46 | 0 | -1 | 2 | 1 |
| | | | | | | | | 45 | 90 | -46 | -46 | 0 | 0 | 1 | 1 |
| | | | | | | | | 46 | 91 | -47 | -45 | -1 | -1 | 2 | 0 |
| 2 | 28.61 | 28.61 | 3.67 | 45 | 40 | -25 | -45 | 46 | 42 | -27 | -46 | -1 | -2 | 2 | 1 |
| | | | | | | | | 45 | 42 | -27 | -45 | 0 | -2 | 2 | 0 |
| | | | | | | | | 45 | 41 | -28 | -46 | 0 | -1 | 3 | 1 |
| | | | | | | | | 46 | 40 | -27 | -47 | -1 | 0 | 2 | 2 |
| | | | | | | | | 45 | 40 | -28 | -45 | 0 | 0 | 3 | 0 |
| | | | | | | | | 45 | 41 | -28 | -45 | 0 | -1 | 3 | 0 |
| | | | | | | | | 45 | 40 | -28 | -45 | 0 | 0 | 3 | 0 |
| 3 | 40.85 | 0 | 1.21 | 90 | 45 | -45 | -30 | 91 | 43 | -45 | -30 | -1 | 2 | 0 | 0 |
| | | | | | | | | 92 | 43 | -44 | -32 | -2 | 2 | -1 | 2 |
| | | | | | | | | 90 | 42 | -45 | -32 | 0 | 3 | 0 | 2 |
| | | | | | | | | 90 | 42 | -44 | -30 | 0 | 3 | -1 | 0 |
| | | | | | | | | 90 | 42 | -44 | -30 | 0 | 3 | -1 | 0 |
| | | | | | | | | 91 | 43 | -44 | -30 | -1 | 2 | -1 | 0 |
| | | | | | | | | 90 | 42 | -44 | -31 | 0 | 3 | -1 | 1 |
| 4 | 0 | 45.92 | 7.75 | 0 | 30 | -30 | 0 | 0 | 28 | -32 | -2 | 0 | 2 | 2 | 2 |
| | | | | | | | | 0 | 29 | -31 | -1 | 0 | 1 | 1 | 1 |
| | | | | | | | | 2 | 28 | -30 | -2 | -2 | 2 | 0 | 2 |
| | | | | | | | | 1 | 28 | -32 | -3 | -1 | 2 | 2 | 3 |
| | | | | | | | | 0 | 29 | -32 | -3 | 0 | 1 | 2 | 3 |
| | | | | | | | | 2 | 27 | -32 | -2 | -2 | 3 | 2 | 2 |
| | | | | | | | | 2 | 28 | -32 | -2 | -2 | 2 | 2 | 2 |

Dari hasil pengujian dapat di lihat bahwa *error* sudut servo *base* (θ_1) yaitu sebesar ± 0.571 derajat, pada servo *shoulder*(θ_2) yaitu sebesar ± 0.785 derajat, pada servo *elbow*(θ_3) yaitu sebesar ± 1.285 derajat, dan pada servo *wrist*(θ_4) yaitu sebesar ± 1.035 derajat.

Contoh perhitungan tabel 5.2 (lihat sub bab 4.2.2)

$$A0 = \tan^{-1}(yt / xt)$$

$$A0 = \tan^{-1}\left(\frac{20}{20}\right) = 45^\circ$$

$$wT = \sqrt{xt^2 + yt^2}$$

$$wT = \sqrt{(20)^2 + (20)^2} = 28.28$$

$$w2 = wT - L3 \cos(\text{posisi gripper})$$

$$w2 = 28.28 - 19.5 \cos(0) = 8.78$$

$$z2 = zT - L3 \sin(\text{posisi gripper})$$

$$z2 = 25 - 19.5 \sin(0) = 25$$

$$L12 = \sqrt{w2^2 + z2^2}$$

$$L12 = \sqrt{(8.78)^2 + (25)^2} = 26.50$$

$$A12 = \tan^{-1}\left(\frac{z2}{w2}\right)$$

$$A12 = \tan^{-1}\left(\frac{28.5}{9.0}\right) = 70$$

$$A1 = \cos^{-1} \frac{\text{link}1^2 + \text{link}12^2 - \text{link}2^2}{2 \times \text{link}1 \times \text{link}12} + A12$$

$$A1 = \cos^{-1} \frac{15.5^2 + 29.88^2 - 13^2}{2 \times 15.5 \times 29.88} + 70 = 90$$

$$w1 = \text{link}1 \times \cos(A1)$$

$$w1 = 15.5 \times \cos(90) = -0.27$$

$$z1 = \text{link}1 \times \sin(A1)$$

$$z1 = 15.5 \times \sin(90) = 15.5$$

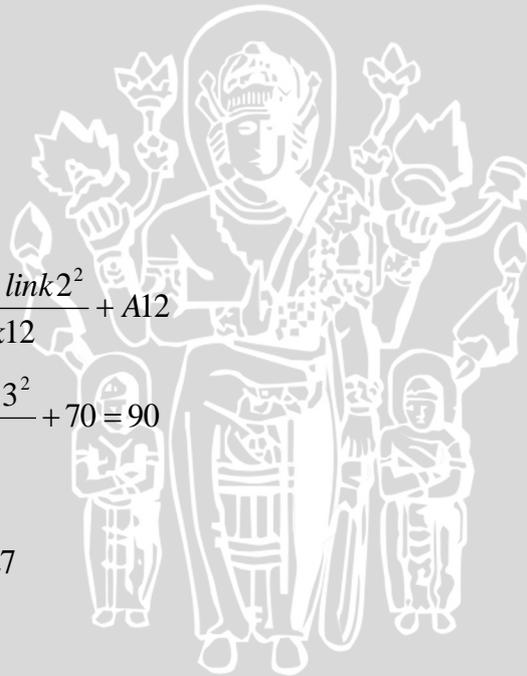
$$A2 = \tan^{-1}\left(\frac{z2 - z1}{w2 - w1}\right) - A1$$

$$A2 = \tan^{-1}\left(\frac{25 - 15.5}{8.78 - (-0.27)}\right) - 90 = -45$$

$$A3 = \text{Posisi Gripper} - A1 - A2$$

$$A3 = 0 - 90 - (-45) = -45$$

Untuk selanjutnya, nilai perhitungan secara teoritis dihitung dengan program sederhana.



BAB VI PENUTUP

6.1 Kesimpulan

Dari perancangan, pengujian dan pengamatan yang telah dilakukan pada *Orion Robotic Arm 5 DOF* maka dapat diambil kesimpulan sebagai berikut:

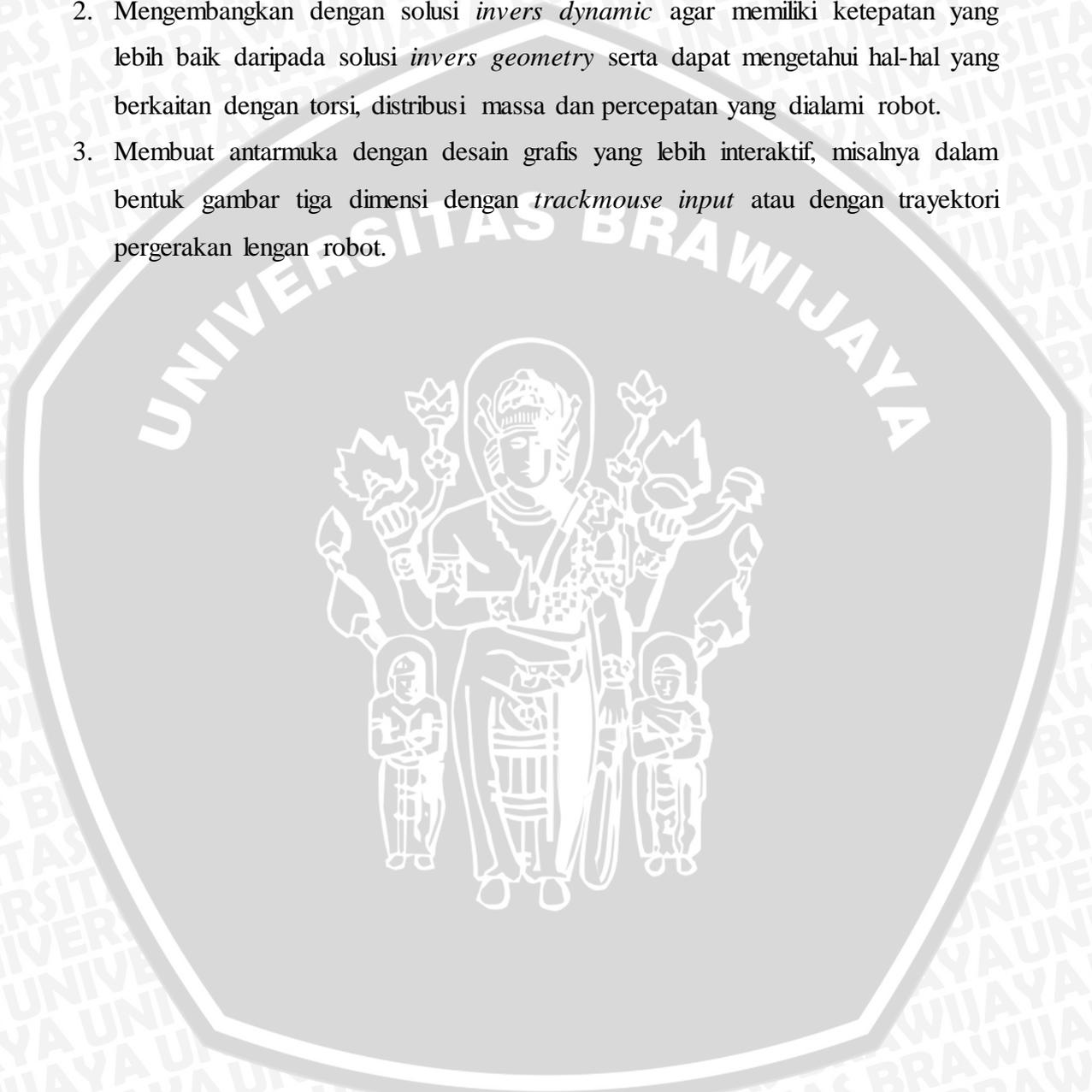
1. Dalam membuat pemodelan statis kinematika maju adalah dengan membuat pendeskripsian frame beserta parameter DH. Setelah penentuan tersebut, didapatkan matrik *homogenus* dan nilai-nilai dari *end-effector* dalam sumbu x, y dan z. Untuk pemodelan statis kinematika terbalik, perhitungan dimulai dari servo base dalam sumbu x dan y. Penentuan sudut gripper untuk penyederhanaan dimensi dalam perhitungan sudut servo shoulder dan elbow.
2. Hasil penerapan metode *forward kinematic* dan *invers kinematic* menggunakan pendekatan *Geometry - Based Kinematic* pada robot lengan. Pada pengujian *forward kinematic* dapat dilihat rata-rata error posisi pada sumbu x yaitu sebesar ± 0.11321 cm, pada sumbu y yaitu ± 0.10571 cm dan pada sumbu z yaitu ± 0.94785 cm. Pada pengujian *invers kinematic* dapat dilihat rata-rata error sudut servo *base* yaitu sebesar ± 0.571 derajat, pada servo *shoulder* yaitu sebesar ± 0.785 derajat, pada servo *elbow* yaitu sebesar ± 1.285 derajat, dan pada servo *wrist* yaitu sebesar ± 1.035 derajat.



6.2 Saran

Dalam perancangan dan pembuatan alat ini masih terdapat kelemahan. Untuk memperbaiki kinerja alat dan pengembangan lebih lanjut disarankan:

1. Menggunakan power supply yang *ajustable* dengan display agar distribusi tegangan dan arus dapat diatur dan terlihat.
2. Mengembangkan dengan solusi *invers dynamic* agar memiliki ketepatan yang lebih baik daripada solusi *invers geometry* serta dapat mengetahui hal-hal yang berkaitan dengan torsi, distribusi massa dan percepatan yang dialami robot.
3. Membuat antarmuka dengan desain grafis yang lebih interaktif, misalnya dalam bentuk gambar tiga dimensi dengan *trackmouse input* atau dengan trayektori pergerakan lengan robot.



DAFTAR PUSTAKA

- Budiharto, Widodo. 2010. *Robotika Teori dan Implementasi*. Yogyakarta : Andi.
- Caysar, Dina. 2014. *Pengaturan Pergerakan Robot Lengan Smart Arm Robotic Ax – 12a melalui Pendekatan Geometry Based Kinematics Menggunakan Arduino*. Malang. Fakultas Teknik Universitas Brawijaya.
- Dass, H.K., 2002. *Introduction to Engineering Mathematics*. New Deldi (India) : S.Chand & Company LTD.
- Hamidah, Syarifah. 2008. *Pengendalian Robot Lengan dengan Menggunakan Pemrograman Visual Basic*. Jurnal Ilmu Komputer Fakultas MIPA Universitas Tanjungpura
- Koyuncu, Baki, Mehmet Guzel. 2007. *Software Development for kinematic Analysis of a Lynx 6 Robot Arm*. Journal of computer engineering.
- Nurdinsidiq, Muh. 2004. *Pengendalian Lengan Robot ROB3 Berbasis Mikrokontroler AT89C51 Menggunakan Transduser Ultrasonik*. Tugas Akhir S1 Teknik Elektro UGM : Yogyakarta.
- Nura, Seif Urfin. 2012. *Pengendalian Lintasan End Effector Robot Lengan dengan Pendekatan Geometri Based Kinematics*. Malang. Fakultas Teknik Universitas Brawijaya.
- Selig, J.M. 1992. *Introductory Robotics*. United State of America : Prentice Hall.
- Senoaji. 2010. *Modeling Moving Object Control System for Motorized Arm Robot Using Adam Series*. Jurnal Teknik Elektronika Fakultas Teknik Politeknik Elektronika Negeri Surabaya.
- Spong, Mark W. 2007. *Robot Dynamics and Control*. John Wiley & Sons : New York.
- Syaifulloh, Masykuri. 2010. *Sistem Pengendalian Lengan Robot dengan Interfacing Java Berbasis ATMEGA 8535*. Naskah Publikasi STMIK AMIKOM Yogyakarta.
- <https://www.arduino.cc/en/Main/ArduinoBoardUno>

UNIVERSITAS BRAWIJAYA

