

Analisa *Forward* dan *Inverse Kinematics* pada Simulator *Arm Robot 5 Derajat Kebebasan*

*Budi Utomo¹, Munadi²

¹Mahasiswa Jurusan Teknik Mesin, Fakultas Teknik, Universitas Diponegoro

²Dosen Jurusan Teknik Mesin, Fakultas Teknik, Universitas Diponegoro

Jl. Prof. Sudharto, SH., Tembalang-Semarang 50275, Telp. +62247460059

*E-mail: budiutomo@outlook.com

Abstract

An arm robot simulator 5 dof (degree of freedom) which is equipped with a two-finger gripper is designed to determine the movement of the robot manipulator. To make an arm robot simulator, we used acrylic as a base material, servomotor as a driver and an Arduino Uno SMD as microcontroller. Acrylic was chosen because it is light, strong and durable. Arduino Uno SMD was chosen because it can interact with LabVIEW that will be able to control the movement angle of servomotor manually. The purpose of this final project is to make an arm robot simulator 5 dof which equipped a gripper and continued with analysis forward and inverse kinematics. An arm robot simulator 5 dof can be used as demonstration tool in education. For making the link of simulator, we use the acrylic laser cutting machine to be more precision cutting. Servomotor's movement is controlled by using a program that was created using LabVIEW. Servomotor's angle position error was corrected by using program functions of numerical multiply and numerical divided on LabVIEW. Analysis forward and inverse kinematics used MATLAB software as a tool for the calculation and we used verification with RoboAnalyzer software. For calculation forward kinematics is required notation Denavit-Hartenberg parameters will result the orientation and position of the end-effector. The orientation and position obtained by transformasi matrix. Then inverse kinematics obtained by performing decrease in transformasi matrix so that is obtained the angle at each joint.

Keywords: Robot Manipulator, Simulator Arm Robot, Forward and Inverse Kinematics.

1. PENDAHULUAN

Perkembangan ilmu pengetahuan dan teknologi terutama dalam bidang elektronika berkembang demikian pesatnya. Perkembangan ini seiring dengan naiknya tuntutan masyarakat akan barang yang berkualitas tinggi yang dihasilkan industri. Untuk menunjang kualitas yang tinggi banyak proses industri beralih dari sistem manual ke sistem otomatis, dengan peran manusia yang semakin kecil, maka diperlukan suatu alat pendukung kinerja di industri. Alat tersebut salah satunya adalah robot yang dapat menggantikan tugas manusia yang berhubungan dengan kegiatan yang berulang-ulang, yang membutuhkan daya tahan serta konsentrasi tinggi, terutama untuk melakukan pekerjaan fisik yang berat, memindahkan barang, memposisikan benda, dan proses-proses lainnya.

Salah satu jenis robot yang digunakan di dunia industri adalah *arm robot*. Dengan semakin pentingnya peran *arm robot* dalam dunia industri, banyak mahasiswa tertarik mempelajari *arm robot*. Sedangkan untuk membuat *arm robot* skala dunia industri tidaklah murah. Untuk itu maka diperlukan simulator *arm robot* yang dapat mempresentasikan *arm robot* skala industri dengan biayanya relatif murah. Dari simulator *arm robot* dapat digunakan untuk menentukan arah dan pergerakan. Untuk mengetahui arah dan pergerakan dari *arm robot* diperlukan analisa *forward* dan *inverse kinematics*. *Forward kinematics* merupakan proses menghitung orientasi dan posisi dari *end-effector* berdasarkan

sudut-sudut pada *joint*. Sedangkan *inverse kinematics* adalah sebaliknya, diberikan posisi *end-effector*, maka yang akan dicari adalah berapa besar sudut yang harus diubah untuk tiap *joint* untuk dapat mencapai posisi *end-effector* tersebut.

Jadi dalam Tugas Akhir ini, penulis akan mendesain dan membuat simulator *arm robot 5 dof* yang merupakan representasi *arm robot* untuk industri yang nantinya digunakan untuk bidang pendidikan. Simulator *arm robot* ini dibuat dengan biaya relatif murah. Selanjutnya menganalisa *forward* dan *inverse kinematics* akan dilakukan dengan menggunakan MATLAB sebagai alat bantu perhitungannya. Untuk memverifikasi hasil analisa akan digunakan *RoboAnalyzer*.

2. DASAR TEORI

2.1 Definisi Robot

Robot berasal dari kata "robota" yang dalam bahasa Ceko berarti budak, pekerja, atau kuli. Pertama kali kata "robota" diperkenalkan oleh Karel Caper dalam sebuah pentas sandiwara pada tahun 1921 yang berjudul RUR (Rossum's Universal Robots) pentas ini mengisahkan mesin yang menyerupai manusia yang dapat bekerja tanpa lelah yang kemudian memberontak dan menguasai manusia[1].

Istilah robot kemudian mulai terkenal dan digunakan untuk menggantikan istilah yang dikenal saat ini yaitu *automation*. Selanjutnya banyak definisi menjelaskan pengertian robot. Dari berbagai definisi

yang ada, definisi paling dapat diterima adalah dari “Robot Institute of America”, yang mendefinisikan robot sebagai manipulator yang berfungsi jamak yang dapat diprogram ulang dan dirancang untuk memindahkan benda kerja, komponen-komponen, peralatan atau perangkat khusus dengan berbagai macam tugas[2].

2.2 Klasifikasi Umum Robot

2.2.1 Klasifikasi Robot Berdasarkan Penggunaan Aktuator

a) Robot Manipulator

Robot ini hanya memiliki satu tangan seperti tangan manusia yang fungsinya untuk memegang atau memindahkan barang. Robot manipulator merupakan sebuah rangkaian benda kaku (*rigid bodies*) terbuka yang terdiri atas sendi dan terhubung dengan *link* dimana setiap posisi sendi ditentukan dengan variabel tunggal sehingga jumlah sendi sama dengan nilai derajat kebebasan. Untuk lengkapnya, gambar robot manipulator ditunjukkan pada Gambar 1.



Gambar 1. Robot manipulator[1]

b) Mobile Robot

Mobile robot adalah konstruksi robot yang ciri khasnya adalah mempunyai aktuator untuk menggerakkan keseluruhan badan robot tersebut, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik yang lain. Secara umum robot *mobile* dibagi menjadi dua yaitu robot beroda dan robot berkaki. Gambar 2 menunjukkan robot beroda dan Gambar 3 menunjukkan robot berkaki.



Gambar 2. Robot beroda[2]



Gambar 3. Robot berkaki[3]

2.2.2 Klasifikasi Robot Berdasarkan Kebutuhan akan Operator Robot

a) *Autonomous Robot*

Robot *autonomous* adalah robot yang dapat melakukan tugas-tugas yang diinginkan dalam lingkungan yang tidak terstruktur tanpa bimbingan manusia secara terus menerus. Gambar 4 menunjukkan *autonomous robot*.



Gambar 4. *Autonomous robot*[4]

b) *Teleoperated Robot*

Robot ini dalam pengoperasian mesinnya dikendalikan dari kejauhan. Hal ini mirip dalam arti untuk frase "*remote control*", dikendalikan oleh operator (manusia) menggunakan *remote control*. Gambar 5 menunjukkan *teleoperated robot*.



Gambar 5. *Teleoperated robot*[5]

c) *Semi Autonomous Robot*

Robot semi *autonomous* adalah robot yang pengendaliannya secara otomatis untuk pengerjaan tertentu dan pengendalian jarak jauh dengan menggunakan *remote control*. Gambar 6 menunjukkan semi *autonomous robot*.



Gambar 6. Semi *autonomous robot*[6]

2.2.3 Klasifikasi Robot Berdasarkan Kegunaan

Industrial robot adalah robot yang digunakan di dalam industri. Biasanya robot ini berupa robot manipulator yang bisa dikendalikan secara otomatis maupun *manual*. Gambar 7 menunjukkan robot manipulator di dunia industri.



Gambar 7. Robot manipulator di dunia industri[7]

a) *Service Robot*

Service robot dipergunakan untuk melayani kebutuhan manusia sehari – hari. Robot ini digunakan untuk membantu pekerjaan yang kotor, berbahaya, berulang-ulang dan termasuk pekerjaan rumah tangga. Gambar 8 menunjukkan salah satu jenis *service robot*.

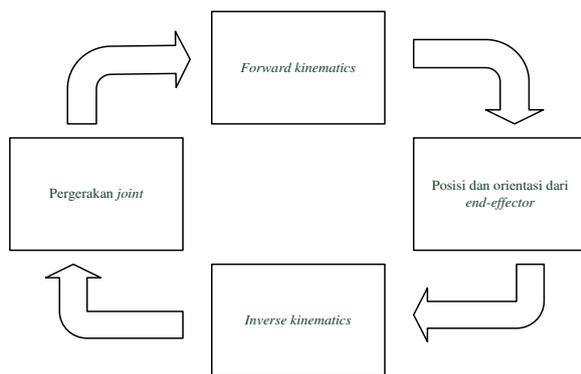


Gambar 8. *Service robot*[8]

2.3 Permodelan Kinematik

Kinematika adalah cabang dari mekanika yang membahas gerakan benda tanpa mempersoalkan gaya penyebab gerakan. Sedangkan dinamika atau sering disebut dengan kinetika adalah ilmu mekanik yang membahas gaya yang menyebabkan gerakan.

Terdapat dua topik pembahasan mendasar pada kinematika robot. Yang pertama adalah *forward kinematics* merupakan proses menghitung orientasi dan posisi dari *end-effector* berdasarkan sudut-sudut *joint*, sedangkan *inverse kinematics* sebaliknya diberikan posisi *end-effector*, dimana yang akan dicari adalah besaran sudut yang harus diubah untuk tiap *joint* dalam mencapai posisi *end-effector* tersebut. Gambar 9 menunjukkan *block diagram* sederhana dari permodelan kinematika.



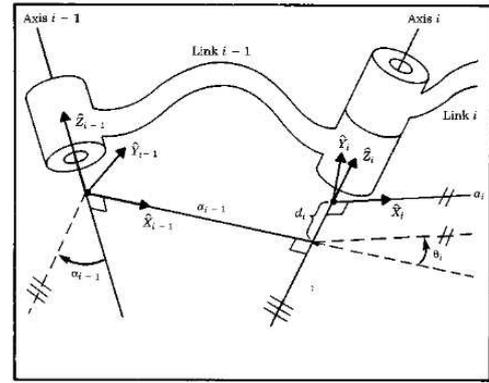
Gambar 9. *Block diagram* kinematika

2.3.1 *Forward Kinematics*

a) *Denavit-Hertenberg Parameter*

Denavit-Hertenberg merupakan tahap berikutnya dalam memodelkan *arm robot* secara kinematika yaitu berupa penentuan nilai parameter dan variabel yang terdapat pada masing-masing *link*.

Penjelasan mengenai parameter dan variabel pada *arm robot* dapat dilihat pada Gambar 10.



Gambar 10. Notasi *Denavit-Hartenberg*[9]

Masing-masing notasi memiliki ketetapan diantaranya adalah:

- a. i adalah link ke i
- b. $i-1$ adalah link ke $i-1$
- c. α_i adalah sudut antara z_i dan z_{i+1} pada sumbu x_i ;
- d. a_i adalah jarak antara z_i menuju z_{i+1} diukur sepanjang x_i ;
- e. θ_i adalah sudut antara x_{i-1} dan x_i pada sumbu z_i ; dan
- f. d_i adalah jarak antara x_{i-1} menuju x_i diukur sepanjang z_i .

b) *Matriks Transformasi*

Untuk menentukan posisi dan orientasi maka perlu dilakukan perhitungan transformasi. Matriks transformasi adalah matriks yang memetakan sebuah vektor atau posisi pada satu sistem koordinat ke sistem koordinat yang lain dengan memperhatikan rotasi, translasi, penskalaan dan perspektif / sudut pandang. Perhitungan matriks transformasi dapat dilakukan setelah parameter dan variabel pada *link arm robot* telah ditentukan. sehingga kita dapat menentukan posisi T *end-effector* terhadap *base* (0T_n).

$$T = {}^0T_n = {}^0T_1 {}^1T_2 \dots {}^{n-1}T_n \dots \dots \dots (1)$$

Untuk mencari sistem koordinat berdekatan ${}^{i-1}A_i$ yaitu sistem koordinat i dan sistem koordinat $i-1$ sebagai berikut:

$${}^{i-1}A_i = Trans(0, 0, d_i)Rot(z, \theta_i)Trans(a_i, 0, 0)Rot(x, \alpha_i)$$

$${}^{i-1}A_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.3.2 *Inverse Kinematics*

Inverse kinematics adalah kebalikan *forward kinematics*, dimana diberikan posisi *end-effector*, dan yang akan dicari adalah berapa besar sudut yang harus diubah untuk tiap *joint* untuk dapat mencapai posisi *end-effector* tersebut. Pada Tugas Akhir ini kami

menggunakan pendekatan *algebraic approach* untuk solusi *inverse kinematics arm robot 5 dof*. Hal ini dikarenakan sifat penurunan matriks transformasi diantara pendekatan yang lain, pendekatan *algebraic* yang paling efisiensi waktu.

a) Algebraic Approach

Pendekatan aljabar, solusi dicari dengan cara melakukan penurunan persamaan-persamaan kinematik dari matriks transformasi *homogeneous* sehingga didapatkan sudut-sudut *joint* yang dibutuhkan. Sebagai contoh, lokasi dan *frame* {3} relatif terhadap *base frame* {0} adalah:

$${}^0T_3 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \dots \dots \dots (2)$$

dimana:

$${}^0T_1 = {}^0A_1$$

$${}^1T_2 = {}^1A_2$$

$${}^2T_3 = {}^2A_3$$

Untuk mencari besarnya persamaan sudut pada setiap *joint*, bisa dilakukan dengan memindahkan elemen dari bagian kanan ke bagian kiri pada persamaan 2, sehingga menjadi:

$$({}^0T_1)^{-1} \cdot {}^0T_3 = {}^1T_2 \cdot {}^2T_3 \dots \dots \dots (3)$$

$$({}^0T_1)^{-1} \cdot {}^0T_3 = {}^1T_3 \dots \dots \dots (4)$$

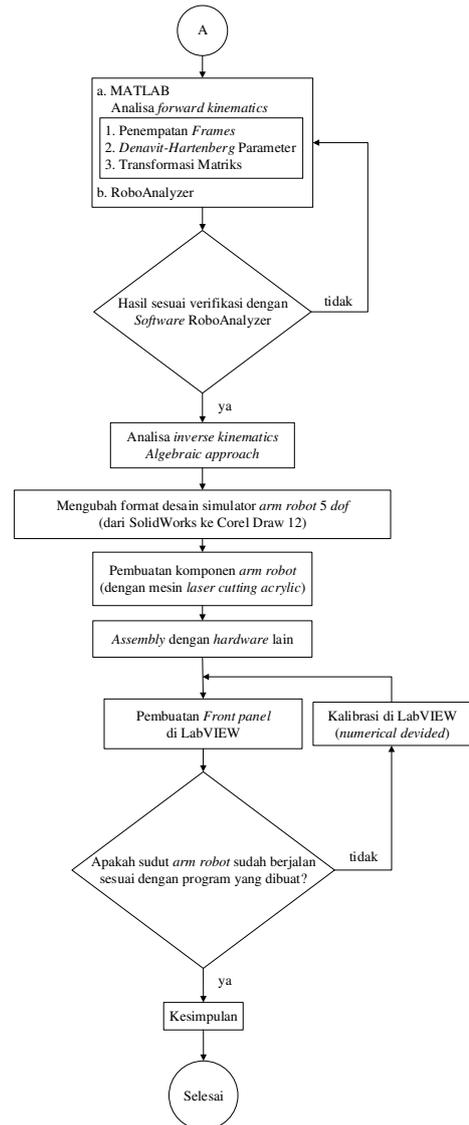
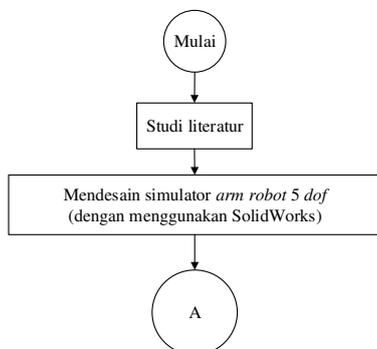
Dengan menggunakan persamaan diatas, maka bisa dicari besarnya sudut-sudut *joint* yang dibutuhkan untuk mencapai suatu posisi. Apabila masih ada sudut *joint* lain yang belum didapatkan, maka pindahkan lagi elemen dibagian kanan ke kiri, sehingga menjadi:

$$({}^0T_1)^{-1} \cdot ({}^1T_2)^{-1} \cdot {}^0T_3 = {}^2T_3 \dots \dots \dots (5)$$

$$({}^0T_2)^{-1} \cdot {}^0T_3 = {}^2T_3 \dots \dots \dots (6)$$

Dengan bantuan persamaan diatas, maka solusi untuk *inverse kinematics* bisa didapatkan.

3. METODOLOGI



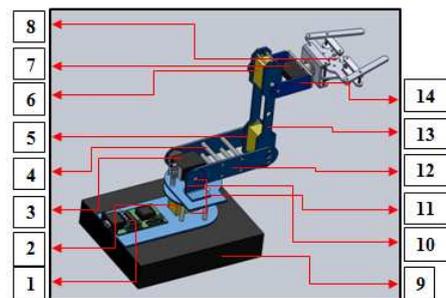
Gambar 11. Diagram alir dalam Tugas Akhir

4. DESAIN DAN PEMBUATAN SIMULATOR ARM ROBOT

4.1 Desain Simulator Arm Robot 5 DOF

4.1.1 Desain Hardware

Desain menggunakan alat bantu *software* Solidworks. Kemudian hasil desain dirubah formatnya ke CorelDraw dengan tujuan agar hasil desain yang dibuat bisa dibaca pada mesin potong akrilik. Gambar 12 menunjukkan hasil desain di SolidWorks.



Gambar 12. Desain arm robot 5 dof di SolidWorks

Keterangan Gambar 12 :

- | | |
|---------------------|-----------------------|
| 1 : Mikrokontroler | 8 : Gripper |
| 2 : Joint 1 | 9 : Base Utama |
| 3 : Joint 2 | 10 : Penyangga Link 2 |
| 4 : Baut penghubung | 11 : Link 1 |
| 5 : Joint 3 | 12 : Link 2 |
| 6 : Joint 4 | 13 : Link 3 |
| 7 : Joint 5 | 14 : Link 4 |

Untuk spesifikasi *servomotor* tiap *joint* dapat dilihat pada Tabel 1.

Tabel 1. Spesifikasi *servomotor* tiap *joint*

No	Joint	Servomotor	Spesifikasi <i>servomotor</i>
1	1, 3 dan 4	Hobbyking HK-15288A	Tegangan : 4,8v-6v Torsi : 8kg.cm@4,8v 9kg.cm @6v Berat : 51 gram Kecepatan : 0,21 s / 60deg @4,8v 0,20 s / 60 deg @6v
2	2	Turnigy Metal Gear	Tegangan : 4,8v-6v Torsi : 15,5kg.cm @4,8v 17kg.cm @6v Berat : 60 gram Kecepatan : 0,16 s / 60deg @ 4,8v 0,14s / 60 deg @ 6v
3	5 dan gripper	Hitec HS-81	Tegangan : 4,8v-6v Torsi : 2.6kg.cm @4,8v 3kg.cm @6v Berat : 51 gram Kecepatan : 0,11 s / 60deg @ 4,8v 0,09 s / 60 deg @ 6v

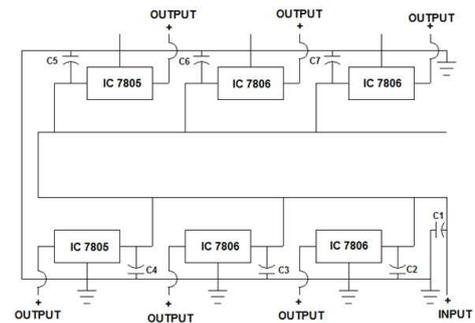
Power Supply yang digunakan yaitu baterai DC merek *Turnigy* 11,1 volt dengan kapasitas 1.300 mAh. Gambar 13 menunjukkan baterai *turnigy*.



Gambar 13. Baterai *Turnigy*

Arm robot ini memerlukan regulator. Regulator adalah rangkaian regulasi atau pengatur tegangan keluaran dari sebuah catu daya agar efek dari naik atau turunnya tegangan tidak mempengaruhi tegangan catu daya sehingga menjadi stabil. Diperlukan penurunan tegangan sebesar 6volt untuk *servomotor* HK-15288A dan *servomotor* merek *Turnigy*, 5 volt untuk *servomotor* HS-81.

Untuk menurunkan tegangan ke 6 volt diperlukan perangkat elektronik IC-7806 sedangkan untuk menurunkan tegangan ke 5 volt diperlukan IC-7806. Gambar 14 menunjukkan rangkaian regulator.



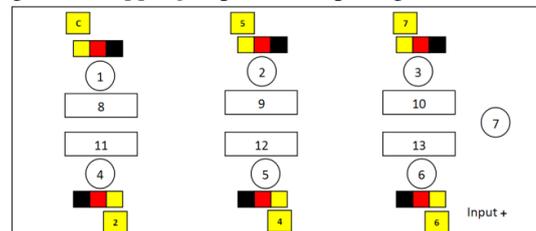
Gambar 14. Rangkaian regulator

Keterangan pada Gambar 12 :

C1 : Kapasitor 1000 μ F, 16 volt

C2-C7 : Kapasitor 100 μ F, 16 volt

Regulator *mapping* dapat dilihat pada gambar 15.



Gambar 15. Regulator *mapping*

Keterangan pada Gambar 13 :

1-7 : Kapasitor

8 dan 11 : IC 7805

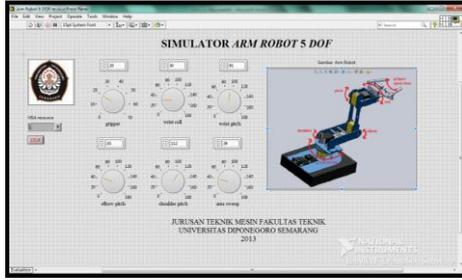
9,10, 12, 13 : IC 7806

 : Pin pada *servomotor*, terdiri dari 3 yaitu *ground* (warna hitam), *positive* (warna merah) dan *signal* (warna kuning).

 : Pin *signal servomotor* yang terhubung pada pin *Arduino*. Angka yang tertera di dalamnya merupakan pin *Arduino*.

4.1.2 Desain Software

Menggunakan *LabVIEW* yang dapat terhubung dengan *Arduino* yang nantinya bisa menggerakkan *servomotor* secara *manual* di *front panel* *LabVIEW*. Untuk menghubungkan *Arduino* dengan *LabVIEW* dibutuhkan *Firmata Arduino*. Gambar 16 menunjukkan blok diagram.

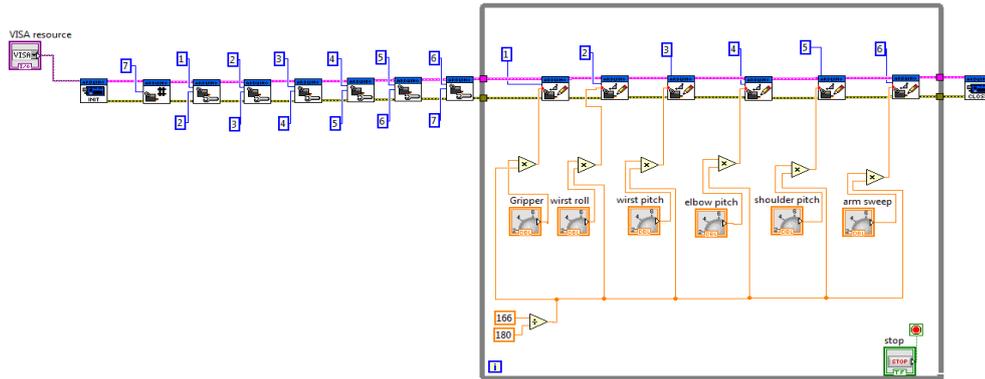


Gambar 17. Front panel LabVIEW



Gambar 18. Simulator arm robot 5 dof

Hasil pembuatan simulator *arm robot 5 dof* dapat dilihat pada gambar 18.

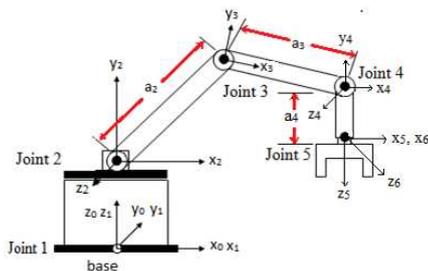


Gambar 16. Blok diagram LabVIEW

4.2 Analisa Forward dan Inverse Kinematics pada Simulator Arm Robot 5 DOF

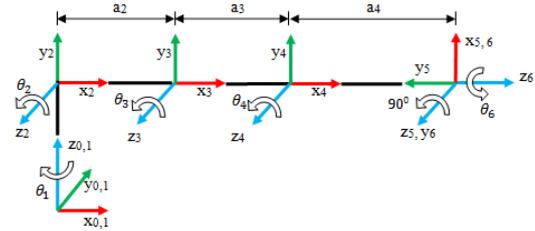
4.2.1 Penempatan Frames

Simulator *Arm robot 5 dof* memiliki 6 buah *servomotor* dan sebuah *gripper* sebagai *end-effector*. *Gripper* tersebut mempunyai jari yang berfungsi untuk memegang benda kerja. *Arm robot* mirip dengan manipulator pada industri yang mengambil ide berdasarkan prinsip kerja lengan manusia. *Joints* tersebut terdiri dari beberapa bagian gerakan pinggang, gerakan bahu, gerakan siku, gerakan pergelangan tangan, dan gerakan *gripper*. Tampilan *frame* dari semua *joints* ditampilkan dalam Gambar 19



Gambar 19. Penempatan Frames dari Arm Robot 5 dof.

Agar lebih mudah dalam memahami *frame* dimasing-masing *joint* maka dibuatlah Gambar 20 yaitu koordinat *frames*. Koordinat *frames* untuk *arm robot* mempunyai 5 *Joint* putar dan pergerakan mencengkeram.



Gambar 20. Koordinat Frames untuk Arm Robot 5 dof.

4.2.2 Devanit-Hartenberg Parameter

Devanit-Hartenberg merupakan representasi yang digunakan sebagai model *joint arm robot 5 dof*. Semua *joint* ditetapkan menggunakan prinsip kaidah *Devanit-Hartenberg*. Lebih jelasnya dapat dilihat Tabel 2.

Tabel 2. D-H parameter untuk *arm robot*

Link	α_{i-1}	a_{i-1}	d_i	θ_i
1	0^0	0	0	θ_1
2	90^0	0	0	θ_2
3	0^0	a_2	0	θ_3
4	0^0	a_3	0	θ_4
5	0^0	a_4	0	90^0
6	90^0	0	0	θ_6

4.2.3 Transformasi Link

Setelah D-H parameter diperoleh, maka tahap berikutnya adalah melakukan transformasi matrik. Dengan melakukan transformasi matrik antar *link*, maka akan diperoleh hasil akhir berupa kedudukan *link-i* bergerak relatif terhadap *link* yang menjadi

referensi. Dengan kata lain, matrik transformasi antar *link* menunjukkan posisi dan orientasi posisi ujung *arm robot (end-effector)* bergerak relatif terhadap *base*. Dalam melakukan perhitungan matrik transformasi di bawah ini, parameter yang digunakan masih berupa simbol, dan bukan merupakan angka. Hal ini bertujuan untuk memudahkan dalam memasukkan hasil perhitungan ke dalam program pengendalian *arm robot*.

$${}^{i-1}T_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots(7)$$

Transformasi matrik *link* n terhadap *link* n-1

$${}^0T_1 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; {}^1T_2 = \begin{bmatrix} \cos \theta_2 & 0 & \sin \theta_2 & 0 \\ \sin \theta_2 & 0 & -\cos \theta_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2T_3 = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & a_3 \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & a_3 \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; {}^3T_4 = \begin{bmatrix} \cos \theta_4 & -\sin \theta_4 & 0 & a_4 \cos \theta_4 \\ \sin \theta_4 & \cos \theta_4 & 0 & a_4 \sin \theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^4T_5 = \begin{bmatrix} \cos \theta_5 & -\sin \theta_5 & 0 & a_5 \cos \theta_5 \\ \sin \theta_5 & \cos \theta_5 & 0 & a_5 \sin \theta_5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; {}^5T_6 = \begin{bmatrix} \cos \theta_6 & 0 & \sin \theta_6 & 0 \\ \sin \theta_6 & 0 & -\cos \theta_6 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Untuk memperoleh transformasi matrik *link* 6 terhadap *link* 0 yang akan menentukan posisi dan orientasi *end-effector* terhadap *base*, maka dilakukan dengan mengalikan matrik transformasi antar *link* dan untuk mempermudah pemahaman simbol sebagai berikut:

$$s_i = \sin(\theta_i)$$

$$c_i = \cos(\theta_i)$$

$$s_{23} = \sin(\theta_2 + \theta_3) = (s_2 c_3 + c_2 s_3)$$

$$c_{23} = \cos(\theta_2 + \theta_3) = (c_2 c_3 - s_2 s_3)$$

$$s_{345} = s_3(c_4 c_5 - s_4 s_5) + c_3(s_4 c_5 + c_4 s_5)$$

$$c_{345} = c_3(c_4 c_5 - s_4 s_5) - s_3(s_4 c_5 + c_4 s_5)$$

$$s_{3456} = c_6(c_34 c_5 - s_34 s_5) - s_6(c_34 s_5 + s_34 c_5)$$

$$s_{3456} = c_6(s_{34} c_5 + c_{34} s_5) - s_6(s_{34} s_5 + c_{34} c_5)$$

Hasil transformasi link 6 terhadap *link* 0 sebagai berikut:

$${}^0T_6 = {}^0T_5 \cdot {}^5T_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots(8)$$

Dimana (p_x, p_y, p_z) menunjukkan posisi dan $(\{n_x, n_y, n_z\}, \{o_x, o_y, o_z\}, \{a_x, a_y, a_z\})$ menunjukkan orientasi pada *end-effector*. Orientasi dan posisi di *end-effector* dapat dihitung sebagai berikut.

$$n_x = c_6(c_{12}(c_{34}c_5 - s_{34}s_5)) - s_6(c_{12}(c_{34}c_5 + s_{34}s_5))$$

$$n_y = c_6(s_{12}(c_{34}c_5 - s_{34}s_5)) - s_6(s_{12}(c_{34}c_5 + s_{34}s_5))$$

$$n_z = c_6(s_{34}c_5 + c_{34}s_5) - s_6(s_{34}s_5 + c_{34}c_5)$$

$$o_x = s_{12}$$

$$o_y = -c_{12}$$

$$o_z = 0$$

$$a_x = s_6(c_{12}(c_{34}c_5 - s_{34}s_5)) + c_6(c_{12}(c_{34}c_5 + s_{34}s_5))$$

$$a_y = s_6(s_{12}(c_{34}c_5 - s_{34}s_5)) + c_6(s_{12}(c_{34}c_5 + s_{34}s_5))$$

$$a_z = s_6(s_{34}c_5 + c_{34}s_5) + c_6(s_{34}s_5 + c_{34}c_5)$$

$$p_x = c_{12}(a_4(c_{34}c_5 - s_{34}s_5)) + a_3 c_{34} + a_2 c_3$$

$$p_y = s_{12}(a_4(c_{34}c_5 - s_{34}s_5)) + a_3 c_{34} + a_2 c_3$$

$$p_z = (a_4(c_{34}s_5 + s_{34}c_5)) + a_3 s_{34} + a_2 s_3$$

4.2.4 Forward Kinematics untuk Arm Robot 5 DOF

Berdasarkan pengukuran data untuk masing-masing *link*, didapat panjang efektif dari pusat rotasi *joint* (d dan a) yang kemudian hasil pengukuran tersebut ditabelkan pada Tabel D-H parameter seperti pada Tabel 3.

Tabel 3. D-H parameter setelah nilai dimasukkan.

Link	α_{i-1}	a_{i-1}	d_i	θ_i
1	0^0	0	0	60^0
2	90^0	0	0	60^0
3	0^0	119	0	60^0
4	0^0	140	0	60^0
5	0^0	53	0	90^0
6	90^0	0	0	60^0

Untuk mengetahui transformasi matrik 0T_6 , yang menggambarkan transformasi matriks dari *base* di *arm robot* terhadap *end-effector* diperlukan persamaan 8 yang secara detail ditunjukkan hasilnya sebagai berikut:

$${}^0T_6 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 \cdot {}^5T_6$$

$${}^0T_6 = \begin{bmatrix} -0,0000 & 0,8660 & 0,5 & 28,1997 \\ 0,0000 & 0,5000 & -0,8660 & -48,8433 \\ -1,0000 & 0 & -0,0000 & 197,8006 \\ 0 & 0 & 0 & 1,0000 \end{bmatrix} \dots\dots\dots(9)$$

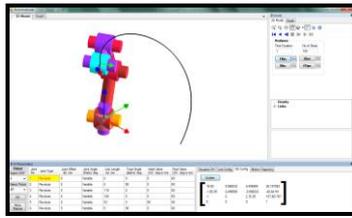
Transformasi 0T_6 dihitung dengan menggunakan bantuan *software* MATLAB yang lengkap pada Lampiran. Berdasarkan matriks transformasi 0T_6 pada persamaan 9, maka dapat diperoleh transformasi orientasi dan posisi *end-effector*. Untuk orientasi dapat diperoleh dengan matriks 3x3. Sedangkan untuk posisi ditunjukkan pada Tabel 4.

Tabel 4. Posisi dan orientasi *arm robot*

Nilai posisi	Nilai 0T_6 (mm)
p_x	28
p_y	-48
p_z	197

4.2.5 Verifikasi Arm Robot 5 DOF menggunakan RoboAnalyzer

RoboAnalyzer ini akan digunakan sebagai verifikasi dalam perhitungan yang telah dilakukan dengan MATLAB. Hasil dari matriks transformasi ditunjukkan Gambar 21.



Gambar 21. Hasil matriks transformasi pada D-H parameter

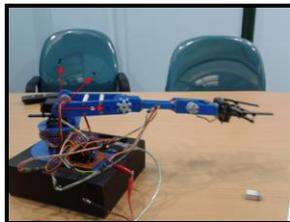
Kemudian kita bandingkan hasil perhitungan *forward kinematics* antara MATLAB dan *RoboAnalyzer* :

$${}^0T_6 = \begin{bmatrix} -0,0000 & 0,8660 & 0,5000 & 28,1997 \\ 0,0000 & 0,5000 & -0,8660 & -48,8433 \\ -1,0000 & 0 & -0,0000 & 197,8006 \\ 0 & 0 & 0 & 1,0000 \\ 1E-05 & 0,8660 & 0,4999 & 28,1979 \\ -1,8E-05 & 0,4999 & -0,8660 & -48,8419 \\ -1,0000 & 0 & 2,1E-05 & 197,8018 \\ 0 & 0 & 0 & 1,0000 \end{bmatrix}$$

Terlihat dari hasil perhitungan menunjukkan bahwa nilai matriks transformasi *forward kinematics* antara MATLAB dan *RoboAnalyzer* bernilai sama sehingga bisa dinyatakan benar.

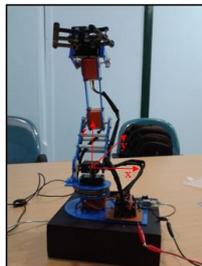
4.2.6 Validasi dengan melakukan percobaan *Forward Kinematics* pada Simulator *Arm Robot 5 DOF*.

Untuk mengetahui apakah benar atau tidak hasil analisa *forward kinematics* maka diperlukan percobaan atau pengujian pada simulator *arm robot*. Tahap awal adalah mengetahui posisi awal *arm robot* pada *end-effector*, setelah dilakukan pengukuran didapat data sebagai berikut : $p_x = 310$ mm, $p_y = 0$ mm dan $p_z = 0$ mm sebagai posisi awal yang ditunjukkan pada Gambar 22.



Gambar 22. Posisi awal *forward kinematics* pada simulator *arm robot*

Kemudian dengan memasukan sudut θ pada masing-masing *joint* pada *front panel* di LabVIEW, maka: $p_x = 28$ mm, $p_y = -50$ mm dan $p_z = 220$ mm yang ditunjukkan pada Gambar 23.



Gambar 23. Posisi akhir *forward kinematics* pada simulator *arm robot*

Selanjutnya berdasarkan hasil perhitungan dan pengujian diperoleh data posisi yang ditabelkan pada Tabel 5.

Tabel 5. Hasil pengujian

Posisi	Hasil			Error
	Perhitungan menggunakan MATLAB (mm)	Perhitungan menggunakan <i>RoboAnalyzer</i> (mm)	Pengujian menggunakan simulator <i>arm robot</i> (mm)	
P_x	28	28	28	0%
P_y	-48	-48	-50	4,2%
P_z	197	197	210	6,6%

Error diperoleh berdasarkan perbandingan antara pengujian dan hasil perhitungan MATLAB atau *software RoboAnalyzer*, dimana masih terjadi selisih untuk sumbu y sebesar 4,2% dan sumbu z sebesar 6,6% selisih tersebut disebabkan *servomotor* memiliki error 1,67% tiap *joint*. Hasil ini masuk dalam toleransi bila simulator digunakan untuk bidang pendidikan. Akan tetapi jika dipakai untuk industri, maka harus diteliti lebih lanjut.

4.2.7 *Inverse Kinematics* untuk *Arm Robot 5 DOF*

Solusi *inverse kinematics* untuk mendapatkan nilai sudut *joint* yang diperlukan dalam mencapai posisi yang diinginkan dengan menggunakan pendekatan *algebraic*. Pendekatan *algebraic* adalah dengan melakukan penurunan persamaan-persamaan kinematika yang direpresentasikan oleh matriks transformasi yang ditunjukkan pada persamaan 10.

$$T = {}^0T_1 * {}^1T_2 * {}^2T_3 * {}^3T_4 * {}^4T_5 * {}^5T_6 = G = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots(10)$$

Untuk menemukan solusi *inverse kinematics* pada *joint* pertama θ_1 , transformasi *inverse link* dikalikan kembali seperti ditunjukkan persamaan (11) dengan memindahkan dari ruas kanan ke kiri.

$$[{}^0T_1]^{-1} * G = ({}^0T_1)^{-1} * {}^0T_1 * {}^1T_2 * {}^2T_3 * {}^3T_4 * {}^4T_5 * {}^5T_6 \dots\dots\dots(11)$$

Sehingga dihasilkan matriks transformasi sebagai berikut persamaan (12).

$$\begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 & 0 \\ -\sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} c_1 n_x + s_1 n_y & c_1 o_x + s_1 o_y & c_1 a_x + s_1 a_y & c_1 p_x + s_1 p_y \\ -s_1 n_x + c_1 n_y & -s_1 o_x + c_1 o_y & -s_1 a_x + c_1 a_y & -s_1 p_x + c_1 p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots(12)$$

dimana:

$$\begin{aligned} \alpha_{11} &= c_{3456} c_{12} c_1 + c_{3456} s_{12} s_1 \\ \alpha_{12} &= s_{12} c_1 - c_{12} s_1 \\ \alpha_{13} &= s_{3456} c_{12} c_1 + s_{3456} s_{12} s_1 \\ \alpha_{14} &= c_{12} c_1 (a_3 c_{34} + a_2 c_3 + a_4 c_{345}) + s_{12} s_1 (a_3 c_{34} + a_2 c_3 + a_4 c_{345}) \\ \alpha_{21} &= c_{3456} s_{12} c_1 - c_{3456} c_{12} s_1 \\ \alpha_{22} &= -s_{12} s_1 - c_{12} c_1 \\ \alpha_{23} &= s_{3456} s_{12} c_1 - s_{3456} c_{12} s_1 \\ \alpha_{24} &= s_{12} c_1 (a_3 c_{34} + a_2 c_3 + a_4 c_{345}) - c_{12} s_1 (a_3 c_{34} + a_2 c_3 + a_4 c_{345}) \\ \alpha_{31} &= s_{3456} \\ \alpha_{32} &= 0 \\ \alpha_{33} &= -c_{3456} \\ \alpha_{34} &= a_3 s_{34} + a_2 s_3 + a_4 s_{345} \end{aligned}$$

Kemudian berdasarkan persamaan (12), maka dapat digunakan untuk mencari θ_1 (13) yaitu dengan menggunakan baris ke-2 dengan kolom ke-4 [2, 4].

$$s_1 = \frac{c_1 p_y}{(s_2 c_1 (a_3 c_{34} + a_2 c_3 + a_4 c_{345}) - c_{12} (a_3 c_{34} + a_2 c_3 + a_4 c_{345})) + p_x - (s_1 p_x)}$$

$$c_1 = \frac{(s_{12} (a_3 c_{34} + a_2 c_3 + a_4 c_{345}) - c_{21} (a_3 c_{34} + a_2 c_3 + a_4 c_{345})) - p_y}{(s_2 c_1 (a_3 c_{34} + a_2 c_3 + a_4 c_{345}) - c_{12} (a_3 c_{34} + a_2 c_3 + a_4 c_{345})) + p_x + c_1 p_y}$$

Sehingga nilai $\theta_1 = \arcsin(s_1, c_1)$(13)

Selanjutnya menghitung θ_1 dengan menggunakan persamaan (12) yaitu:

Menggunakan baris ke-1 dengan kolom ke-4 [1, 4]

$$c_2 = \frac{c_1 p_x + s_1 p_y}{(c_1 c_2) (a_3 c_{34} + a_2 c_3 + a_4 c_{345}) + s_{12} s_1 (a_3 c_{34} + a_2 c_3 + a_4 c_{345}) - s_1 p_x + c_1 p_y}$$

$$s_2 = \frac{((s_2 c_1 (a_3 c_{34} + a_2 c_3 + a_4 c_{345}) - c_{12} (a_3 c_{34} + a_2 c_3 + a_4 c_{345})) - p_y)}{(s_2 c_1 (a_3 c_{34} + a_2 c_3 + a_4 c_{345}) - c_{12} (a_3 c_{34} + a_2 c_3 + a_4 c_{345})) + p_x + c_1 p_y}$$

Sehingga nilai $\theta_2 = \arcsin(s_2, c_2)$(14)

θ_3 dapat ditemukan dengan cara yang sama menggunakan persamaan (12):

Menggunakan baris ke-1 dengan kolom ke-4 [1, 4]

$$c_3 = \frac{s_1 s_{12} a_4 s_{34} s_5 + c_1 c_{12} a_4 s_{34} s_5}{a}$$

Setelah diperoleh s_3 dilanjutkan dengan mencari s_3 , menggunakan baris ke-3 dan kolom ke-4 [3, 4].

$$s_3 = \frac{a_4 c_{34} s_5}{(a_3 s_4 + a_2 + a_4 s_{45}) - (a_4 s_4 c_5 + a_3 s_4 + a_2)}$$

sehingga nilai $\theta_3 = \arcsin(s_3, c_3)$(15)

θ_4 didapatkan sebagai berikut:
 Menggunakan baris ke-3 dengan kolom ke-4 [3, 4]

$$s_4 = \frac{a_4 c_{34} s_5 + a_2 s_3 - a_2 s_3}{(a_3 s_3 + a_4 s_{35}) - (a_4 s_3 c_5 + a_3 s_3)}$$

Menggunakan baris ke-3 dengan kolom ke-4 [3, 4]

$$c_4 = \frac{a_3 s_{34} + a_2 s_3 + a_4 s_{345} - a_4 s_{34} c_5 - a_3 s_{34} - a_2 s_3}{a_4 c_3 s_5}$$

sehingga nilai $\theta_4 = \arcsin(s_4, c_4)$(16)

Untuk θ_5 sudah ditetapkan sebesar 90^0 dan merupakan *frame* tambahan pada *arm robot 5 dof*.
 sehingga nilai $\theta_5 = 90^0$(17)

Mendapatkan θ_6 sebagai berikut:
 Menggunakan baris ke-3 dengan kolom ke-3 [3, 3]

$$c_6 = \frac{s_6 (c_{34} s_5 + s_{34} c_5) - s_6 (s_{34} c_5 + c_{34} s_5)}{(s_{34} s_5 + c_{34} c_5) + (c_{34} c_5 - s_{34} s_5)}$$

Menggunakan baris ke-3 dengan kolom ke-3 [3, 3]

$$s_6 = \frac{c_6 (s_{34} s_5 + c_{34} c_5) + c_6 (c_{34} c_5 - s_{34} s_5)}{(c_{34} s_5 + s_{34} c_5) - (s_{34} c_5 + c_{34} s_5)}$$

sehingga nilai $\theta_6 = \arcsin(s_6, c_6)$(18)

Tahap selanjutnya adalah melakukan analisa *inverse kinematics* dengan menentukan posisi awal $P_x, P_y, P_z = [310 \text{ mm}, 0 \text{ mm}, 0 \text{ mm}]$ seperti pada Gambar 22 dan posisi akhir yang ingin dicapai $P_x, P_y, P_z = [28 \text{ mm}, -50 \text{ mm}, 210 \text{ mm}]$ seperti pada Gambar 23 dengan menggunakan persamaan 13, 14, 15, 16, 17 dan 18 maka diperoleh masing-masing sudutnya adalah:

$$\theta_1 = \arcsin(28,1959;48,1959) = 60,00^0 \cong 60^0$$

$$\theta_2 = \arcsin(24,4234;14,0902) = 60,01^0 \cong 60^0$$

$$\theta_3 = \arcsin(240,3393;128,1820) = 61,92^0 \cong 62^0$$

$$\theta_4 = \arcsin(0,5556;1,0000) = 60,94^0 \cong 61^0$$

$$\theta_5 (\text{frame tambahan}) = 90^0 \cong 90^0$$

$$\theta_6 = \arcsin(0,5000;0,8660) = 59,99^0 \cong 60^0$$

Kemudian kita uji lagi dengan menjadikan posisi akhir $P_x, P_y, P_z = [28 \text{ mm}, -50 \text{ mm}, 210 \text{ mm}]$ sebagai posisi awal untuk pengujian kedua, dimana posisi akhir yang kita kehendaki adalah $P_x, P_y, P_z = [89 \text{ mm}, 41 \text{ mm}, 118 \text{ mm}]$ maka diperoleh nilai θ sebagai berikut : $\theta_1 = 65^0, \theta_2 = 20^0, \theta_3 = 5^0, \theta_4 = 30^0, \theta_5 = 90^0$ dan $\theta_6 = 60^0$. Lengkapnya ditunjukkan pada Tabel 6.

Tabel 6. Perhitungan *inverse kinematics*

Pengujian	Posisi	Posisi awal	Posisi akhir	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
I	P_x	310	28	60^0	60^0	62^0	61^0	90^0	60^0
	P_y	0	-48						
	P_z	0	197						
II	P_x	28	89	65^0	20^0	5^0	30^0	90^0	60^0
	P_y	-48	41						
	P_z	197	118						

4.2.8 Verifikasi dengan melakukan percobaan *Inverse Kinematics* pada Simulator *Arm Robot 5 DOF*

Verifikasi seharusnya dilakukan dengan memasukkan posisi awal dan posisi akhir pada *front panel* LABVIEW. Akan tetapi pada penelitian ini belum berhasil dibuat *software* untuk *inverse kinematic*, masih sebatas *forward kinematic* yaitu dilakukan dengan cara memasukkan masing-masing sudut θ ke tiap *joint* di *front panel* LABVIEW. Verifikasi pertama dengan menentukan posisi awal $P_x, P_y, P_z = [28 \text{ mm}, -50 \text{ mm}, 210 \text{ mm}]$ dan memasukkan sudut sebesar $\theta_1 = 65^0, \theta_2 = 20^0, \theta_3 = 5^0, \theta_4 = 30^0, \theta_5 = 90^0$ dan $\theta_6 = 60^0$ berdasarkan nilai sudut yang dihasilkan bila menggunakan perhitungan analisa *inverse kinematic* dengan MATLAB (Tabel 6) pada *front panel*. Dan hasilnya diperoleh posisi akhir $P_x, P_y, P_z = [28 \text{ mm}, -50 \text{ mm}, 210 \text{ mm}]$ yang diukur menggunakan penggaris sebagai alat bantu pengukuran posisi yang dihasilkan. Hasil posisi akhir ini tampaknya berbeda dengan posisi akhir pada Tabel 7.



Gambar 23. Posisi akhir *inverse kinematics* pada simulator *arm robot*

Verifikasi kedua dilakukan dengan menentukan posisi awal $P_x, P_y, P_z = [28 \text{ mm}, -50 \text{ mm}, 210 \text{ mm}]$ dan

nilai θ secara berturut-turut sebagai berikut : $\theta_1 = 65^0$, $\theta_2 = 20^0$, $\theta_3 = 5^0$, $\theta_4 = 30^0$, $\theta_5 = 90^0$ dan $\theta_6 = 60^0$. Hasil posisi akhir adalah $P_x, P_y, P_z = [89 \text{ mm}, 44 \text{ mm}, 126 \text{ mm}]$ yang secara lengkap terdapat pada Tabel 7. Hasil dari verifikasi.

Antara perhitungan dengan MATLAB jika dibandingkan dengan hasil pada simulator tampaknya terdapat kesalahan. Hal ini dikarenakan belum bisanya kami dalam membuat dan merakit khususnya pada tahap *assembly* benar-benar antara *link* dan *joint* benar-benar *center* antar sumbu.

Tabel 7. Pengujian dengan simulator *arm robot 5 dof*

Pengujian	Posisi	Posisi awal	Posisi akhir	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
I	P_x	310	28	60^0	60^0	62^0	61^0	90^0	60^0
	P_y	0	-50						
	P_z	0	210						
II	P_x	28	89	65^0	20^0	5^0	30^0	90^0	60^0
	P_y	-50	44						
	P_z	210	126						

5. KESIMPULAN

Berkenaan dengan rancang bangun simulator *arm robot 5 dof*, penulis berhasil merancang sebuah simulator *arm robot 5 dof* dibuat yang dibuat dari bahan akrilik dengan bantuan *software* SolidWorks dan CorelDraw 12, *front Panel* berhasil dibuat yang menghubungkan LabVIEW dengan mikrokontroler Arduino dimana *servomotor* dapat digerakkan secara *manual*, masih terdapat kesalahan posisi sudut pada *front panel* dengan sudut *servomotor* sebesar 1,67 %.

Pada pengujian dengan memasukkan nilai $\theta_1 = 60^0$, $\theta_2 = 60^0$, $\theta_3 = 60^0$, $\theta_4 = 60^0$, $\theta_5 = 90^0$, dan $\theta_6 = 60^0$, dengan *software* MATLAB maka diperoleh posisi koordinat (x, y, z) *end-effector* adalah [28 mm, -48 mm, 197 mm]. Kemudian dilanjutkan dengan verifikasi *software RoboAnalyzer* dan hasilnya sama. Sedangkan untuk verifikasi dengan simulator *arm robot* terdapat error pada sumbu y sebesar 4,2% dan sumbu z sebesar 6,6 % selisih tersebut disebabkan *servomotor* memiliki error 1,67% tiap *joint*. Hasil ini masuk dalam toleransi bila simulator digunakan untuk pendidikan. Tetapi jika dipakai untuk industri, maka harus diteliti lebih lanjut.

Selanjutnya analisa *inverse kinematics* merupakan analisa mencari besaran sudut yang harus diubah untuk tiap *joint* dalam mencapai posisi *end-effector*. Analisa ini dilakukan dengan bantuan *software* MATLAB dan verifikasi dengan simulator *arm robot*. Pada pengujian untuk posisi awal $P_x, P_y, P_z = [310 \text{ mm}, 0 \text{ mm}, 0 \text{ mm}]$ dan posisi akhir yang ingin dicapai $P_x, P_y, P_z = [28 \text{ mm}, -50 \text{ mm}, 210 \text{ mm}]$ menghasilkan sudut di tiap *joint* : $\theta_1 = 60^0$, $\theta_2 = 60^0$, $\theta_3 = 62^0$, $\theta_4 = 61^0$, $\theta_5 = 90^0$, dan $\theta_6 = 60^0$. Kemudian pengujian kedua dengan posisi awal $P_x, P_y, P_z = [28 \text{ mm}, -50 \text{ mm}, 210 \text{ mm}]$ dan posisi akhir $P_x, P_y, P_z = [89 \text{ mm}, 44 \text{ mm}, 126 \text{ mm}]$ maka diperoleh nilai θ sebagai berikut : $\theta_1 = 65^0$, $\theta_2 = 20^0$, $\theta_3 = 5^0$, $\theta_4 = 30^0$, $\theta_5 = 90^0$ dan $\theta_6 = 60^0$. Sedangkan proses verifikasi menggunakan simulator *arm robot* terjadi perbedaan. Hal ini kemungkinan besar disebabkan oleh ketidak

centeran antar *link* terhadap *joint* pada proses *assembly*.

6. DAFTAR PUSTAKA

- http://www.cisco-eagle.com/material-handling-system/palletizers, (diakses: 19 November 2012).
- http://andras.tantoonline.com/robot.htm, (diakses: 24 November 2012).
- http://machinesthatgobing.com/.htm, (diakses: 24 November 2012).
- http://www2.ece.ohio-state.edu/~passino/projects.html, (diakses: 24 November 2012).
- Transcend. 11 Februari 2011. "Robots Become More Autonomous, Lines Blur Between Military & Civilian Applications". (Online). (<http://www.transcend.ws/?p=2110>, diakses: 18 november 2012).
- http://apod.nasa.gov/apod/ap051214.html, (diakses: 24 November 2012).
- http://www.emeraldinsight.com, (diakses: 24 November 2012).
- http://www.plasticpals.com, (diakses: 24 November 2012).
- Craig, John J. 1989. "Introduction To Robotics Mechanics And Control 2nd Edition," Addison-Wesley Publishing Company. USA.
- Artanto, Dian. (2012). "Interaksi Arduino dan LabView," PT Elexmedia Computindo. Jakarta.
- Koyuncu,B., Guzel,M. 2008. *Software Development for Kinematic Analysis of a Lynx 6 Robot Arm*. Internasional Journal of Engineering and Applied Sciences 4:4