



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE 141599

PERANCANGAN JARINGAN SARAF TIRUAN UNTUK MENYELESAIKAN KINEMATIKA BALIK MANIPULATOR ROBOT DENSO 6-DOF

Daniel Cristover Sirait
NRP 2216105069

Dosen Pembimbing
Ir. Rusdhianto Effendi AK., MT.
Zulkifli Hidayat, ST., M.Sc.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - TE 141599

***NEURAL NETWORK PLANNING TO SOLVE INVERSE
KINEMATICS OF 6-DOF DENSO ROBOT MANIPULATOR***

Daniel Cristover Sirait
NRP 2216105069

Supervisor
Ir. Rusdhianto Effendi AK., MT.
Zulkifli Hidayat, ST., M.Sc.

ELECTRICAL ENGINEERING DEPARTEMENT
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Perancangan Jaringan Saraf Tiruan Untuk Menyelesaikan Kinematika Balik Manipulator Robot Denso 6-DoF**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, (5 Juli 2018)



Daniel Cristover Sirait
NRP 2216105069

-----*Halaman ini sengaja dikosongkan*-----

**PERANCANGAN JARINGAN SARAF TIRUAN UNTUK
MENYELESAIKAN KINEMATIKA BALIK MANIPULATOR
ROBOT DENSO 6-DOF**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik**

**Pada
Bidang Studi Teknik Sistem Pengaturan
Departemen Teknik Elektro
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing I

Dosen Pembimbing II


Ir. Rusdhianto Effendi A.K., MT.
Nip : 195704241985021001


Zulkifli Hidayat, ST., M.Sc.
Nip : 197012251999031002



-----*Halaman ini sengaja dikosongkan*-----

PERANCANGAN JARINGAN SARAF TIRUAN UNTUK MENYELESAIKAN KINEMATIKA BALIK MANIPULATOR ROBOT DENSO 6-DOF

Nama : Daniel Cristover Sirait
Pembimbing : 1. Ir. Rusdhianto Effendi A.K., MT.
2. Zulkifli Hidayat, ST., M.Sc.

ABSTRAK

Robotika sangat berguna di dunia industri, salah satu aplikasi robotika di dunia industri adalah manipulator robot. Manipulator robot adalah mekanik elektronik yang menyerupai lengan manusia, sehingga lebih sering disebut dengan lengan robot. Lengan robot terdiri dari lengan (*link*) dan sendi (*joint*). Pada tugas akhir ini akan diselesaikan permasalahan kinematika balik pada manipulator robot denso 6-DoF. Pada kinematika balik ditentukan titik acuan yang ingin dituju oleh manipulator robot dan masing-masing sendi akan bergerak dengan besar sudut tertentu, sehingga *end-effector* manipulator robot dapat bergerak menuju titik acuan yang telah ditentukan. Karena manipulator robot berjenis 6-DoF maka manipulator robot dapat bergerak menuju suatu titik acuan yang telah di tentukan dengan banyak kemungkinan konfigurasi. Metode Jaringan Saraf Tiruan (JST) atau *neural network* digunakan untuk mencari satu solusi kinematika balik sehingga dapat menentukan konfigurasi masing-masing sendi pada manipulator robot dan mengatur pegrakan posisi dan orientasi dari manipulator robot. Setelah dilakukan perancangan dan pengujian, JST telah dapat diaplikasikan pada kinematika balik manipulator robot denso 6-DoF, dimana *end-effector* manipulator robot telah dapat menuju suatu titik acuan, membentuk pola segitiga dan lingkaran baik dalam bidang datar maupun bidang miring dengan masukan *roll*, *pitch*, x,y,z serta rata-rata *error* posisi dibawah 0.00005 meter dan *error* orientasi dibawah 0.00005 derajat.

Kata Kunci : Manipulator Robot Denso, Kinematika Balik, Jaringan Saraf Tiruan, Posisi, Orientasi

-----*Halaman ini sengaja dikosongkan*-----

NEURAL NETWORK PLANNING TO SOLVE INVERSE KINEMATICS OF 6-DOF DENSO MANIPULATOR ROBOT

Name : Daniel Cristover Sirait
Advisor : 1. Ir. Rusdhianto Effendi A.K., MT.
2. Zulkifli Hidayat, ST., M.Sc.

ABSTRACT

Robotics is very useful in the industrial world, one of the robotics applications in the industrial world is a robot manipulator. Robot manipulator is an electronic mechanic that resembles the human arm, so it is more commonly referred as a robotic arm. Robotic arm consists of arm and joint. In this final project will be resolve inverse kinematic of denso robot manipulator 6-DoF. In inverse kinematics a reference point is determined and each joint moves with a certain angle, so that the robot's end-effector can move toward a reference point. Because of the robot manipulator is a 6-DoF type, the robot manipulator can move toward a specified reference point with many possible configurations. Neural network method is used to find inverse kinematic solution so it can determine the configuration of each joint on the robot manipulator and adjust the position and orientation of the robot manipulator. After design and testing, neural network has been applied to inverse kinematics of 6-DoF denso robot manipulator, where the robot end-effector manipulator has been able to reach a point of reference, forming triangle and circle patterns in both flat and inclined plane with roll, pitch, x, y, z input and the average of position error below 0.00005 meters and orientation error below 0.00005 degrees.

Keywords: *Denso Robot Manipulator, Inverse Kinematics, Neural Network, Position, Orientation*

-----*Halaman ini sengaja dikosongkan*-----

KATA PENGANTAR

Puji syukur pada Tuhan Yang Maha Esa yang telah memberikan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul **“Perancangan Jaringan Saraf Tiruan Untuk Menyelesaikan Kinematika Balik Manipulator Robot Denso 6-DoF”**. Tugas Akhir ini disusun guna memenuhi sebagian persyaratan menyelesaikan pendidikan S1 pada Bidang Studi Teknik Sistem Pengaturan, Jurusan Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya.

Pelaksanaan serta penyelesaian laporan Tugas Akhir tidak terlepas dari bimbingan, motivasi, masukan dan bantuan dari berbagai pihak. Untuk itu pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada:

1. Segenap keluarga, terutama Bapak dan Ibu tercinta serta adik penulis yang selalu memberikan dukungan, semangat dan doa sehingga penulis dapat menyelesaikan Tugas Akhir saat ini.
2. Bapak Ir.Rusdhianto Effendie AK, MT. dan bapak Zulkifli Hidayat, ST., M.Sc. sebagai dosen pembimbing penulis yang telah memberikan pengetahuan, arahan dan bantuan dalam menyelesaikan Tugas Akhir ini.
3. Teman-Teman di bidang studi sistem pengaturan lintas jalur 2016 yang selalu memberikan dukungan dan arahan dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa dalam penyusunan laporan Tugas Akhir ini masih terdapat kekurangan dan jauh dari kata sempurna. Untuk itu penulis mengharapkan saran dan kritik yang sifatnya konstruktif dalam penyempurnaan laporan ini. Akhir kata penulis berharap semoga laporan Tugas Akhir ini dapat bermanfaat bagi semua pihak, baik bagi diri penulis pribadi maupun pembaca.

Surabaya, Juni 2018

Penulis

-----*Halaman ini sengaja dikosongkan*-----

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN JUDUL	ii
PERNYATAAN KEASLIAN TUGAS AKHIR	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xxi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Metodologi Penelitian	3
1.6 Sistematika Laporan	4
1.7 Relevansi	5
BAB 2 TEORI DASAR	7
2.1 Tinjauan Pustaka	7
2.2 Manipulator Robot Denso	8
2.3 Jenis Sendi	8
2.3.1 Sendi Putar	9
2.3.2 Sendi Geser	9
2.4 Konfigurasi Manipulator Robot	9
2.5 Transformasi Homogen	10
2.6 Kinematika Robot	12
2.7 Kinematika Maju	12
2.8 Kinematika Balik	14
2.8.1 Pendekatan Geometri	14
2.8.2 Pendekatan Numerik	17
2.9 Jaringan Saraf Tiruan (JST)	17
2.9.1 Fungsi Aktivasi JST	17
2.9.2 Arsitektur JST	19
2.9.3 Metode Pembelajaran JST	21
2.9.4 JST <i>Bacpropagation</i>	22

BAB 3 PERANCANGAN SISTEM	27
3.1 Menentukan Parameter <i>Denavit Hartenberg</i>	27
3.2 Membentuk Transformasi Homogen	31
3.3 Mencari Persamaan Kinematika Maju	33
3.4 Perancangan Kinematika Balik Dengan JST	35
3.4.1 Perancangan Struktur JST	36
3.4.2 Perhitungan <i>Feedforward</i> dan <i>Backward</i>	37
3.5 Simulasi Kinematika Maju dan Kinematika Balik	40
BAB 4 SIMULASI DAN ANALISA DATA	41
4.1 Pengujian Kinematika Maju	41
4.2 Pengujian Kinematika Balik dengan JST	45
4.2.1 Pengujian Jumlah <i>Neuron</i> Lapisan <i>Hidden</i>	45
4.2.2 Pengujian Ketelitian Kinematika Balik dengan JST	48
4.3 Plot Hasil Data Pengujian	51
4.3.1 Plot Hasil Data Pengujian 1	51
4.3.2 Plot Hasil Data Pengujian 2	52
4.3.3 Plot Hasil Data Pengujian 3	53
4.3.4 Plot Hasil Data Pengujian 4	54
4.3.5 Plot Hasil Data Pengujian 5	55
4.4 Kinematika Balik dengan JST untuk Pola Segitiga	56
4.4.1 Pola Segitiga Datar	57
4.4.2 Pola Segitiga Miring	60
4.5 Kinematika Balik dengan JST untuk Pola Lingkaran	63
4.5.1 Pola Lingkaran Datar	63
4.5.2 Pola Lingkaran Miring	67
BAB 5 PENUTUP	71
5.1 Kesimpulan	71
5.2 Sarann	71
DAFTAR PUSTAKA	73
LAMPIRAN	75
RIWAYAT PENULIS	91

DAFTAR GAMBAR

Gambar 2.1 Robot Denso VP-6242G [2]	8
Gambar 2.2 Sendi Putar [1]	9
Gambar 2.3 Sendi Geser [1]	9
Gambar 2.4 Struktur Konfigurasi RRR dan <i>The Motoman SK16</i> [1].....	10
Gambar 2.5 Daerah Kerja Konfigurasi RRR [1]	10
Gambar 2.6 Cara Kerja Kinematika Maju dan Kinematika Balik [7].....	12
Gambar 2.7 Manipulator Robot 3 Sendi.....	15
Gambar 2.8 Manipulator Robot 3 Sendi Dari Sisi Atas.....	15
Gambar 2.9 Manipulator Robot 3 Sendi Dari Sisi Samping.....	16
Gambar 2.10 Fungsi Aktivasi Linear [4].....	18
Gambar 2.11 Fungsi Aktivasi <i>Hard Limit</i> Biner [4].....	18
Gambar 2.12 Fungsi Aktivasi <i>Hard Limit</i> Bipolar [4].....	18
Gambar 2.13 Fungsi Aktivasi <i>Sigmoid</i> [4]	19
Gambar 2.14 Fungsi Aktivasi <i>Sigmoid Bipolar</i> [4].....	19
Gambar 2.15 Arsitektur Jaringan Satu Lapis [4]	20
Gambar 2.16 Arsitektur Jaringan Multilapis [4].....	20
Gambar 2.17 Arsitektur Jaringan Kompetitif [4].....	21
Gambar 2.18 Arsitektur JST <i>Backpropagation</i> [4].....	22
Gambar 3.1 Letak Sendi dan Panjang Lengan.....	27
Gambar 3.2 Pemberian Label Sumbu z	28
Gambar 3.3 Pemberian Label Sumbu x_0 dan y_0	28
Gambar 3.4 Pemberian Label Sumbu o_i	29
Gambar 3.5 Pemberian Label Sumbu x_i	29
Gambar 3.6 Pemberian Label Sumbu y_i	30
Gambar 3.7 Pemberian Label Sumbu o_n, x_n, y_n , dan z_n	30
Gambar 3.8 Struktur Kinematika Balik Dengan JST.	36
Gambar 4.1 Bentuk Manipulator Robot Denso Percobaan 1.....	42
Gambar 4.2 Bentuk Manipulator Robot Denso Percobaan 2.....	42
Gambar 4.3 Bentuk Manipulator Robot Denso Percobaan 3.....	43
Gambar 4.4 Bentuk Manipulator Robot Denso Percobaan 4.....	43
Gambar 4.5 Bentuk Manipulator Robot Denso Percobaan 5.....	44
Gambar 4.6 Bentuk Manipulator Robot Denso Percobaan 6.....	44
Gambar 4.7 Bentuk Manipulator Robot Denso Pengujian 1	52
Gambar 4.8 Titik Perpindahan dari Posisi Awal ke Titik Tujuan 1 ...	52
Gambar 4.9 Bentuk Manipulator Robot Denso Pengujian 2	53

Gambar 4.10	Titik Perpindahan dari Posisi Awal ke Titik Tujuan 2	.53
Gambar 4.11	Bentuk Manipulator Robot Denso Pengujian 354
Gambar 4.12	Titik Perpindahan dari Posisi Awal ke Titik Tujuan 3	.54
Gambar 4.13	Bentuk Manipulator Robot Denso Pengujian 455
Gambar 4.14	Titik Perpindahan dari Posisi Awal ke Titik Tujuan 4	.55
Gambar 4.15	Bentuk Manipulator Robot Denso Pengujian 556
Gambar 4.16	Titik Perpindahan dari Posisi Awal ke Titik Tujuan 5	.56
Gambar 4.17	Titik –Titik Pola Segitiga Bidang Datar57
Gambar 4.18	Grafik Perbandingan Orientasi Tujuan dan JST Membentuk Pola Segitiga Bidang Datar58
Gambar 4.19	Grafik Perbandingan Posisi Tujuan dan JST Membentuk Pola Segitiga Bidang Datar58
Gambar 4.20	Grafik <i>Error</i> Orientasi Tujuan dan JST Membentuk Pola Segitiga Bidang Datar59
Gambar 4.21	Grafik <i>Error</i> Posisi Tujuan dan JST Membentuk Pola Segitiga Bidang Datar59
Gambar 4.22	Titik –Titik Pola Segitiga Keluaran JST60
Gambar 4.23	Titik –Titik Pola Segitiga Bidang Miring60
Gambar 4.24	Grafik Perbandingan Orientasi Tujuan dan JST Membentuk Pola Segitiga Bidang Miring61
Gambar 4.25	Grafik Perbandingan Posisi Tujuan dan JST Membentuk Pola Segitiga Bidang Miring61
Gambar 4.26	Grafik <i>Error</i> Orientasi Tujuan dan JST Membentuk Pola Segitiga Bidang Miring62
Gambar 4.27	Grafik <i>Error</i> Perbandingan Orientasi Tujuan dan JST Membentuk Pola Segitiga Bidang Miring62
Gambar 4.28	Titik Pola Segitiga Miring Keluaran JST63
Gambar 4.29	Titik –Titik Pola Lingkaran Bidang Datar64
Gambar 4.30	Grafik Perbandingan Orientasi Tujuan dan JST Membentuk Pola Lingkaran Bidang Datar64
Gambar 4.31	Grafik Perbandingan Posisi Tujuan dan JST Membentuk Pola Lingkaran Bidang Datar65
Gambar 4.32	Grafik <i>Error</i> Orientasi Tujuan dan JST Membentuk Pola Lingkaran Bidang Datar65
Gambar 4.33	Grafik <i>Error</i> Posisi Tujuan dan JST Membentuk Pola Lingkaran Bidang Datar66
Gambar 4.34	Titik –Titik Pola Lingkaran Keluaran JST66
Gambar 4.35	Titik –Titik Pola Lingkaran Bidang Miring67

Gambar 4.36	Grafik Perbandingan Orientasi Tujuan dan JST Membentuk Pola Lingkaran Bidang Miring	68
Gambar 4.37	Grafik Perbandingan Posisi Tujuan dan JST Membentuk Pola Lingkaran Bidang Miring	68
Gambar 4.38	Grafik <i>Error</i> Orientasi Tujuan dan JST Membentuk Pola Lingkaran Bidang Datar	69
Gambar 4.39	Grafik <i>Error</i> Posisi Tujuan dan JST Membentuk Pola Lingkaran Bidang Datar	69
Gambar 4.40	Titik –Titik Pola Lingkaran Keluaran JST	70

-----*Halaman ini sengaja dikosongkan*-----

DAFTAR TABEL

Tabel 3.1 Parameter DH.....	31
Tabel 3.2 Indeks Parameter DH	31
Tabel 4.1 Masukan Kinematika Maju	41
Tabel 4.2 Keluaran Kinematika Maju	41
Tabel 4.3 Posisi dan Orientasi Keluaran 5 <i>Neuron</i> Lapisan <i>Hidden</i> ...	45
Tabel 4.4 <i>Error</i> Posisi dan Orientasi 5 <i>Neuron</i> Lapisan <i>Hidden</i>	46
Tabel 4.5 Posisi dan Orientasi Keluaran 10 <i>Neuron</i> Lapisan <i>Hidden</i> .	46
Tabel 4.6 <i>Error</i> Posisi dan Orientasi 10 <i>Neuron</i> Lapisan <i>Hidden</i>	46
Tabel 4.7 Posisi dan Orientasi Keluaran 25 <i>Neuron</i> Lapisan <i>Hidden</i> .	47
Tabel 4.8 <i>Error</i> Posisi dan Orientasi 25 <i>Neuron</i> Lapisan <i>Hidden</i>	47
Tabel 4.9 Posisi dan Orientasi Keluaran 50 <i>Neuron</i> Lapisan <i>Hidden</i> .	48
Tabel 4.10 <i>Error</i> Posisi dan Orientasi 50 <i>Neuron</i> Lapisan <i>Hidden</i>	48
Tabel 4.11 Posisi dan Orientasi Keluaran dengan Ketelitian 0.0005 ..	49
Tabel 4.12 <i>Error</i> Posisi dan Orientasi dengan Ketelitian 0.0005	49
Tabel 4.13 Posisi dan Orientasi Keluaran dengan Ketelitian 0.0001 ..	50
Tabel 4.14 <i>Error</i> Posisi dan Orientasi dengan Ketelitian 0.0001	50
Tabel 4.15 Posisi dan Orientasi Keluaran dengan Ketelitian 0.00005	50
Tabel 4.16 <i>Error</i> Posisi dan Orientasi dengan Ketelitian 0.00005	51

-----*Halaman ini sengaja dikosongkan*-----

BAB 1

PENDAHULUAN

Bab 1 ini berisi pendahuluan yang terdiri atas latar belakang, rumusan masalah, tujuan, dan metodologi pada tugas akhir ini yang berjudul perancangan jaringan saraf tiruan untuk menyelesaikan kinematika balik manipulator robot denso 6-DoF.

1.9 Latar Belakang

Robotika adalah salah satu cabang teknologi yang berhubungan dengan desain, konstruksi, dan operasi dengan menggunakan robot. Robotika sangat berguna di dunia industri, salah satu aplikasi robotika di dunia industri adalah manipulator robot. Manipulator robot adalah mekanik elektronik yang menyerupai lengan manusia, sehingga lebih sering disebut dengan lengan robot. Lengan robot terdiri dari lengan (*link*) dan sendi (*joint*) serta dapat dikendalikan melalui perangkat komputer.

Lengan robot digunakan untuk melakukan pekerjaan-pekerjaan di tempat yang berbahaya, contohnya adalah penggunaan lengan robot pada laboratorium riset kimia, dimana pada tempat ini dipenuhi dengan kandungan gas beracun dan radiasi radioaktif yang sangat berbahaya bagi manusia, sehingga penggunaan lengan robot diperlukan. Lengan robot juga digunakan untuk melakukan pekerjaan-pekerjaan yang memerlukan keakuratan yang tinggi secara berulang-ulang. Contohnya adalah penggunaan lengan robot pada dunia industri yang digunakan untuk memindahkan suatu objek dari satu posisi ke posisi lainya secara berulang-ulang dengan keakuratan yang tinggi, dimana hal ini sulit dilakukan oleh manusia.

Pergerakan lengan robot dapat dikendalikan menuju suatu titik acuan yang telah ditentukan dengan menggunakan perangkat komputer. Untuk mengatur pergerakan lengan robot sesuai perintah, maka diperlukan studi mengenai kinematika. Kinematika robot adalah studi analitis pergerakan lengan robot terhadap sistem kerangka yang diam atau bergerak tanpa memperhatikan gaya yang mempengaruhi pergerakannya. Kinematika yang digunakan pada tugas akhir ini adalah kinematika balik. Pada kinematika balik ditentukan koordinat tertentu yang ingin dituju oleh lengan robot dan masing-masing sendi akan bergerak dengan besar sudut tertentu, sehingga lengan robot bergerak menuju koordinat yang telah ditentukan.

Pergrekan lengan robot dapat dibagi menjadi dua, yaitu pergerakan posisi dan pergerakan orientasi. Dalam pergerakan posisi, terdapat tiga parameter yaitu maju atau mundur, kanan atau kiri, dan atas atau bawah, sedangkan pada pergerakan orientasi, terdapat tiga parameter lengan robot yaitu *roll*, *pitch*, dan *yaw*. Lengan robot yang digunakan memiliki 6 sendi dengan konfigurasi *articulated*, dimana semua sendi yang terdapat pada lengan robot bejenis sendi putar (*revolute*). Gerakan yang dapat dilakukan oleh sendi putar adalah berotasi atau berputar. Karena lengan robot berjenis 6-DoF maka lengan robot dapat bergerak menuju suatu titik acuan yang telah di tentukan dengan banyak kemungkinan konfigurasi.

1.10 Permasalahan

Untuk sampai ke titik acuan yang ditentukan maka lengan robot memiliki banyak sekali kemungkinan konfigurasi. Metode Jaringan Saraf Tiruan (JST) atau *neural network* digunakan untuk mencari satu solusi kinematika balik sehingga dapat menentukan konfigurasi masing-masing sendi pada lengan robot. Rumusan masalah yang dibahas pada Tugas Akhir ini adalah bagaimana menemukan solusi kinematika balik untuk menentukan posisi dan orientasi *end-effector* lengan robot Denso 6-DoF menuju titik acuan yang telah ditentukan dengan menggunakan metode JST.

1.11 Batasan Masalah

Ada beberapa batasan yang terdapat dalam pembahasan tugas akhir ini, yaitu sebagai berikut:

1. Tugas akhir ini membahas kinematika manipulator robot Denso 6-DoF yang memiliki konfigurasi RRR.
2. Hanya membahas kinematika tanpa pembahasan kecepatan dan dinamika.
3. Kinematika yang dibahas melingkupi posisi x,y,z serta orientasi *roll* dan *pitch*.
4. Kinematika balik hanya mencari besar sudut lima sendi awal.

1.12 Tujuan

Tujuan yang ingin dicapai dari tugas akhir ini adalah untuk dapat mengaplikasikan JST dengan metode *backpropagation* dalam menyelesaikan kinematika balik lengan robot denso 6-DoF, sehingga lengan robot dapat bergerak menuju titik acuan yang telah ditentukan.

1.13 Metodologi Penelitian

Metodologi yang digunakan dalam penyusunan tugas akhir ini antara lain:

1. Studi Literatur

Pada tahap ini akan dilakukan pembelajaran terhadap cara kerja dan bagian-bagian dari lengan robot. Selanjutnya dipelajari mengenai kinematika robot, yaitu bagaimana cara menentukan sendi dan koordinat *end-effector* lengan robot. Setelah mempelajari kinematika robot dilanjutkan dengan mempelajari metode JST yang akan digunakan sebagai penyelesaian kinematika balik untuk menemukan titik – titik koordinat yang ingin dicapai oleh robot.

2. Mencari Persamaan Kinematika Maju

Parameter-parameter dari robot akan digunakan untuk merancang kinematika maju (*forward kinematics*) yang berfungsi untuk menentukan posisi dan orientasi awal serta posisi dan orientasi tujuan dari *end-effector* manipulator robot. Hasil dari persamaan kinematika maju digunakan untuk mendapatkan data uji untuk menguji kinematika balik (*inverse kinematics*).

3. Menyelesaikan Kinematika Balik

Pada tahap ini hasil perhitungan kinematika maju akan digunakan untuk memnentukan koordinat awal dan koordinat tujuan dari *end-effector* manipulator robot. Kinematika balik digunakan untuk mendapatkan besar sudut masing-masing sendi sehingga lengan robot dapat bergerak menuju koordinat tujuan yang telah ditentukan. JST digunakan untuk menemukan solusi kinematika balik sehingga lengan robot bergerak menuju koordinat yang telah ditentukan

4. Simulasi dan Pengujian Metode JST

Setelah solusi kinematika balik telah didapatkan, maka tahap selanjutnya adalah melakukan simulasi dan pengujian pada metode JST dengan menggunakan data kinematika maju. Simulasi akan dilakukan dengan menggunakan *software* Matlab 2017. Pengujian dilakukan dengan cara menentukan posisi dan orientasi yang diinginkan, dan memastikan lengan robot dapat bergerak menuju posisi dan orientasi tersebut, perbedaan antara

posisi dan orientasi yang diinginkan dengan posisi dan orientasi keluran JST akan diukur untuk menghitung performa dari metode JST. Pengujian dan simulasi dilakukan untuk memastikan metode JST dapat menemukan satu solusi kinematika balik.

5. Penyusunan Buku Tugas Akhir

Pada tahap ini akan dilakukan penyusunan laporan terkait hasil proses Tugas Akhir yang telah dilakukan. Penyusunan buku tugas akhir dilakukan sebagai bentuk laporan tertulis dari proses dan hasil kerja terkait topik yang diusulkan.

1.14 Sistematika Laporan

Pembahasan pada tugas akhir ini dibagi menjadi lima bab dengan sistematika penulisan sebagai berikut :

Bab I Pendahuluan

Pada bab ini menjelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi penelitian, sistematika penulisan, serta relevansi.

Bab II Teori Dasar

Bab ini menjelaskan mengenai tinjauan pustaka, konsep dasar dari robot, kinematika maju, kinematika balik dan JST.

Bab III Perancangan Sistem

Bagian ini berisi pembahasan tentang cara menentukan parameter Denavit-Hartenberg (DH) manipulator robot Denso. Mencari persamaan kinematika maju untuk mendapatkan posisi dan orientasi yang diinginkan. Pembentukan struktur JST pada solusi kinematika balik untuk mencari besar sudut sendi yang diperlukan manipulator robot bergerak ke titik acuan. Dari kedua persamaan tersebut dibentuk desain simulasi kinematika maju dan kinematika balik dengan menggunakan JST melalui *software* Matlab 2017.

Bab IV Simulasi, Implementasi dan Analisis Sistem

Bab ini memuat hasil simulasi beserta analisa data pada setiap pengujian kinematika maju dan kinematika balik dengan menggunakan JST.

Bab V Penutup

Analisis yang dilakukan pada Bab IV akan diambil suatu kesimpulan. Saran diberikan sebagai bahan evaluasi penelitian kedepannya.

1.15 Relevansi

Hasil yang diperoleh dari tugas akhir ini diharapkan menjadi referensi dalam pengembangan sistem kontrol di bidang robotika pada manipulator robot. Selain itu, mengetahui fungsi dari metode JST yang digunakan sebagai solusi kinematika balik untuk mencari besar sudut sendi dari masukan titik acuan yang ditentukan.

-----*Halaman ini sengaja dikosongkan*-----

BAB 2

TEORI DASAR

Bab 2 ini berisi teori dasar yang menunjang tugas akhir ini, yaitu teori dasar mengenai bagian-bagian dan konfigurasi dari manipulator robot, lalu teori dasar mengenai kinematika maju dan kinematika balik, dan metode JST untuk menyelesaikan kinematika balik pada manipulator robot denso 6-DoF.

2.1 Tinjauan Pustaka

Robot adalah alat mekanik yang dapat diprogram untuk melakukan berbagai macam aplikasi. Istilah robot pertama kali diperkenalkan kedalam kosa kata oleh dramawan Ceko Karel Capek pada tahun 1920 berupa kata “*robota*”, yang berarti “bekerja” [1]. Penggunaan robot di dunia industri mulai terlihat pada tahun 1960, bersamaan dengan sistem *Computer-Aided Design* (CAD) dan sistem *Computer-Aided Manufacturing* (CAM), yang merupakan tren terbaru dalam proses otomasi manufaktur [2].

Salah satu kegunaan robot di dunia industri adalah manipulator robot. Manipulator robot adalah mekanik elektronik yang menyerupai lengan manusia, sehingga lebih sering disebut dengan lengan robot. Lengan robot terdiri dari lengan dan sendi. Terdapat dua jenis sendi yaitu sendi putar dan sendi geser (*prismatic*). Pada ujung dari gabungan lengan pada lengan robot terdapat *end effector*.

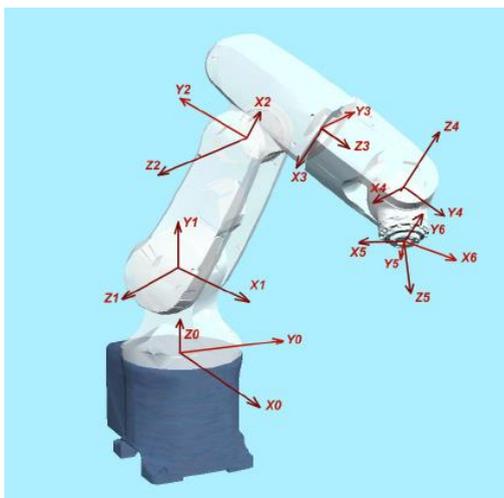
Dalam mengatur pergerakan dari lengan robot perlu dipelajari terlebih dahulu mengenai kinematika. kinematika robot adalah studi analitis pergerakan lengan robot terhadap sistem kerangka yang diam atau bergerak tanpa memperhatikan gaya yang mempengaruhi pergerakannya. Kinematika dibagi menjadi dua bagian yaitu kinematika maju dan kinematika balik. Pada kinematika maju ditentukan besar sudut masing-masing sendi dari manipulator robot, sehingga manipulator robot akan bergerak menuju suatu titik. Pada kinematika balik ditentukan suatu titik acuan masing-masing sendi akan bergerak dengan besar sudut tertentu, sehingga manipulator robot bergerak menuju koordinat yang telah ditentukan.

Pada umumnya, perhitungan untuk kinematika balik lebih sulit bila dibandingkan dengan kinematika maju. Karena permasalahan pada kinematika balik adalah mencari sudut dari setiap sendi berdasar pada

posisi dan orientasi yang diinginkan sebuah robot, sehingga kinematika balik memiliki solusi atau hasil akhir yang unik. Dikatakan unik karena hasil dari sudut untuk tiap sendi memiliki banyak konfigurasi tergantung dari banyaknya sendi yang digunakan. Karena banyaknya konfigurasi tersebut maka digunakan metode JST untuk menentukan konfigurasi yang tepat dari masing-masing sendi. Desain untuk simulasi Kinematika maju dan Kinematika balik akan menggunakan batuan dari software Matlab 2017.

2.2 Manipulator Robot Denso

Robot denso yang digunakan merupakan robot denso 6-DoF VP-6242G. Semua sendi dari robot ini merupakan sendi putar. Pergerakan posisi robot ditentukan dari pergerakan 3 sendi awal robot, sedangkan pergerakan orientasi *end effector* ditentukan dari pergerakan 3 sendi selanjutnya. Bentuk robot denso 6-DoF VP-6242G dapat dilihat pada Gambar 2.1.



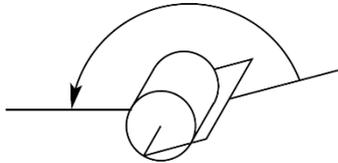
Gambar 2.1 Robot Denso VP-6242G [2]

2.3 Jenis Sendi

Sendi pada manipulator robot berfungsi untuk menghubungkan dua lengan. Terdapat dua jenis sendi pada robotika, yaitu sendi putar dan sendi geser.

2.3.1 Sendi Putar

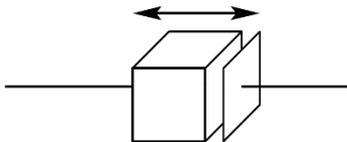
Sendi putar adalah sendi yang dapat bergerak secara berputar atau berotasi dengan batasan sudut tertentu. Bentuk dari sendi putar dapat dilihat pada Gambar 2.2.



Gambar 2.2 Sendi Putar [1]

2.3.2 Sendi Geser

Sendi geser adalah sendi yang dapat bergerak secara translasi atau bergeser dengan batasan tertentu. Bentuk dari sendi geser dapat dilihat pada Gambar 2.3.

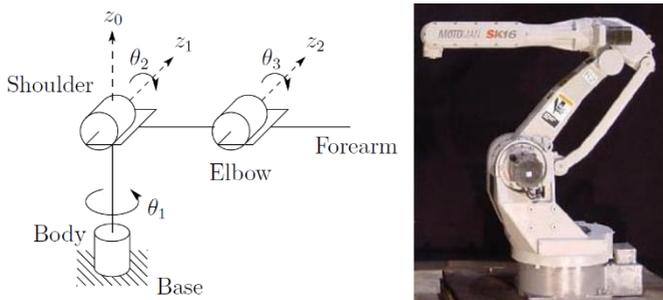


Gambar 2.3 Sendi Geser [1]

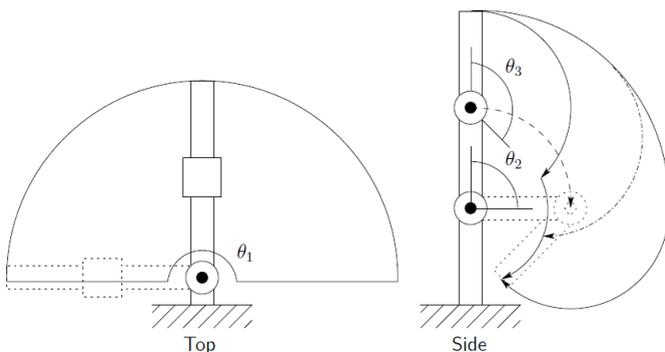
2.4 Konfigurasi Manipulator Robot

Terdapat lima jenis konfigurasi manipulator robot, yaitu *articulate* (RRR), *spherical* (RRP), *SCARA* (RRP), *cylindrical* (RPP), dan *cartesian* (PPP) [1]. Manipulator robot denso 6-DoF memiliki konfigurasi *articulated* (RRR) dimana *artiuculated* atau lebih sering disebut *revolute*, adalah konfigurasi manipulator robot dimana semua sendi dari manipulator tersebut adalah berjenis *revolute*, yang dapat bergerak secara berputar atau berotasi. Manipulator *articulated* memiliki pergerakan yang sama dengan pergerakan lengan manusia sehingga lebih sering disebut *elbow manipulator*. Dapat dilihat sturktur dari manipulator *articulated* dan salah satu contoh manipulator *articulated*, yaitu manipulator *the*

motoman SK16 pada Gambar 2.4. Daerah kerja dari manipulator robot dengan konfigurasi *articulated* dapat dilihat pada Gambar 2.5.



Gambar 2.4 Struktur Konfigurasi RRR dan *The Motoman SK16* [1]



Gambar 2.5 Daerah Kerja Konfigurasi RRR [1]

2.5 Transformasi Homogen

Pergerakan pada robot terdiri dari pergerakan posisi dan orientasi. Transformasi homogen digunakan untuk menggabungkan dua konsep pergerakan tersebut. Matriks gerakan translasi dapat dilihat pada Persamaan 2.1 hingga Persamaan 2.3. Matriks gerakan rotasi dapat dilihat pada Persamaan 2.4 hingga Persamaan 2.6.

$$Trans_{x,a} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

$$Trans_{y,b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

$$Trans_{z,c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

$$Rot_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha & 0 \\ 0 & s_\alpha & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

$$Rot_{y,\phi} = \begin{bmatrix} c_\phi & 0 & s_\phi & 0 \\ 0 & 1 & 0 & 0 \\ -s_\phi & 0 & c_\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

$$Rot_{z,\theta} = \begin{bmatrix} c_\theta & -s_\theta & 0 & 0 \\ s_\theta & c_\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

Matriks translasi dan rotasi tersebut akan membentuk matriks transformasi homogen 4x4, yang dapat dilihat pada Persamaan 2.7. Matriks transformasi homogen ini akan digunakan pada perhitungan kinematika maju.

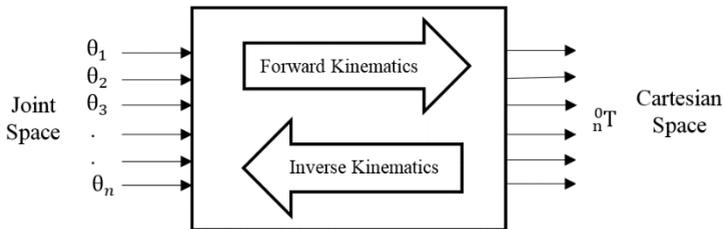
$$H_1^0 = \left[\begin{array}{c|c} R_{3x3} & P_{3x1} \\ \hline f_{1x3} & S_{1x1} \end{array} \right] = \left[\begin{array}{c|c} \text{Rotation} & \text{Translation} \\ \hline \text{Perspective} & \text{Scale} \end{array} \right] \quad (2.7)$$

$$H_1^0 = \begin{bmatrix} n_x & s_x & a_x & d_x \\ n_y & s_y & a_y & d_y \\ n_z & s_z & a_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

Pada Persamaan 2.7 n , s , dan a merupakan vektor yang merepresntasikan orientasi dan d adalah vektor yang merepresntasikan posisi.

2.6 Kinematika Robot

kinematika robot adalah studi analitis pergerakan lengan robot terhadap sistem kerangka yang diam atau bergerak tanpa memperhatikan gaya yang mempengaruhi pergerakannya. Kinematika dibagi menjadi dua bagian yaitu kinematika maju dan kinematika balik Hubungan antara kinematika maju dan kinematika balik ditunjukkan pada Gambar 2.6.



Gambar 2.6 Cara Kerja Kinematika Maju dan Kinematika Balik [7]

Pada kinematika maju ditentukan besar sudut masing-masing sendi dari manipulator robot, sehingga manipulator robot akan bergerak menuju suatu titik. Pada kinematika balik ditentukan titik acuan dan masing-masing sendi akan bergerak dengan besar sudut tertentu, sehingga manipulator robot bergerak menuju titik acuan yang telah ditentukan.

2.7 Kinematika Maju

Pada kinematika maju ditentukan besar pergerakan masing-masing sendi, dan robot akan bergerak, sehingga posisi dan orientasi robot tersebut menuju suatu titik tertentu. Dalam menemukan persamaan kinematika maju maka terlebih dahulu harus menemukan parameter DH

dari manipulator robot denso 6-DoF. Berikut ini langkah- langkah untuk menentukan parameter DH.

Langkah 1: Menempatkan dan melabeli sumbu z_0, \dots, z_{n-1} pada masing-masing sendi.

Langkah 2: Menetapkan *base frame*. Menentukan *origin* pada sumbu z_0 . Sumbu x_0 dan y_0 dipilih secara acak sesuai kaidah tangan kanan.

Langkah 3: Menempatkan *origin* o_i , dimana *common normal* z_i dan z_{i-1} memotong z_i . Jika z_i memotong z_{i-1} tempatkan o_i pada perpotongan tersebut. Jika z_i dan z_{i-1} paralel, tempatkan o_i pada sendi i .

Langkah 4: Menetapkan x_i (untuk $i=1$ sampai dengan 5) sepanjang *common normal* antara z_{i-1} dan z_i melalui o_i , atau arah normal dari z_{i-1} - z_i jika z_{i-1} dan z_i berpotongan.

Langkah 5: Menetapkan y_i (untuk $i=1$ sampai dengan 5) dengan aturan kaidah tangan kanan.

Langkah 6: Menetapkan *frame end-effector* o_n, x_n, y_n , dan z_n .

Langkah 7: Membuat tabel parameter lengan $a_i, d_i, \alpha_i, \theta_i$.

a_i = jarak sepanjang x_i dari o_i ke perpotongan dari sumbu x_i dan z_{i-1} .

d_i = jarak sepanjang z_{i-1} dari o_{i-1} ke perpotongan dari sumbu x_i dan z_{i-1} . d_i adalah variabel jika sendi I adalah prismatic.

α_i = sudut antara z_{i-1} dan z_i diukur terhadap x_i .

θ_i = sudut antara x_{i-1} dan x_i dikur terhadap z_i . θ_i adalah variabel jika sendi adalah *revolute*.

Langkah 8: Membentuk matriks transformasi homogen dengan melakukan substitusi parameter $a_i, d_i, \alpha_i, \theta_i$.

Langkah 9: Membentuk matriks kinematika maju $T_0^n = A_1, \dots, A_n$. Matriks ini memberikan posisi dan orientasi dari *tool frame* yang diekspresikan dalam koordinat dasar.

Setiap transformasi homogen A_i direpresentasikan dengan perkalian dari empat transformasi dasar, dapat dilihat pada Persamaan 2.9.

$$A_i = Rot_{z, \theta_i} Trans_{z, d_i} Trans_{x, a_i} Rot_{x, \alpha_i} \quad (2.9)$$

$$= \begin{bmatrix} C_{\theta_i} & -S_{\theta_i} & 0 & 0 \\ S_{\theta_i} & C_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_{\alpha_i} & -S_{\alpha_i} & 0 \\ 0 & S_{\alpha_i} & C_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_{\theta_i} & -S_{\theta_i}C_{\alpha_i} & S_{\theta_i}S_{\alpha_i} & a_iC_{\theta_i} \\ S_{\theta_i} & C_{\theta_i}C_{\alpha_i} & -S_{\theta_i}S_{\alpha_i} & a_iS_{\theta_i} \\ 0 & S_{\alpha_i} & C_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Setelah setiap transformasi homogen A_i ditemukan, maka pada langkah selanjutnya dilakukan pembentukan matriks kinematika maju dengan menggunakan Persamaan 2.10.

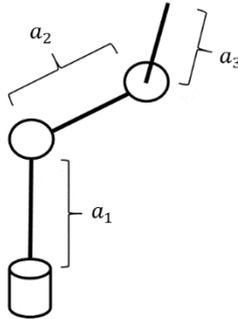
$$T_0^n = A_1 \dots A_n \quad (2.10)$$

2.8 Kinematika Balik

Pada kinematika balik ditentukan titik acuan yang ingin dituju, dan masing-masing sendi akan bergerak dengan besar sudut tertentu, sehingga manipulator robot bergerak menuju titik acuan yang telah ditentukan. Pada umumnya, perhitungan untuk kinematika balik lebih sulit bila dibandingkan dengan kinematika maju. Karena permasalahan pada kinematika balik adalah mencari sudut dari setiap sendi berdasar pada posisi akhir yang diinginkan sebuah robot, sehingga kinematika balik memiliki solusi atau hasil akhir yang unik. Dikatakan unik karena hasil dari sudut untuk tiap sendi memiliki banyak solusi tergantung dari banyaknya sendi yang digunakan. Terdapat dua metode dalam menyelesaikan kinematika balik, yang pertama adalah pendekatan secara geometri dan yang kedua adalah pendekatan secara numerik.

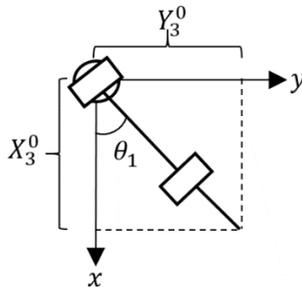
2.8.1 Pendekatan Geometri

Penyelesaian kinematika balik dengan menggunakan pendekatan geometri adalah penyelesaian kinematika balik dengan menggunakan pengkomposisian ruang geometri dan hukum-hukum trigonometri. Contoh dari penyelesaian kinematika balik menggunakan pendekatan geometri dapat dilihat pada Gambar 2.7. Dimana pada Gambar 2.7 akan terdapat manipulator robot dengan 3 sendi. Dengan menggunakan kinematika balik akan diperoleh persamaan-persamaan dari masing-masing sendi untuk mendapatkan besar sudut masing-masing sendi jika diketahui posisi yang diinginkan.



Gambar 2.7 Manipulator Robot 3 Sendi

Terdapat dua tahap dalam mencari persamaan kinematika balik, tahap yang pertama adalah melihat manipulator dari sisi atas. Pada Gambar 2.8 dapat dilihat manipulator robot dari sisi atas.

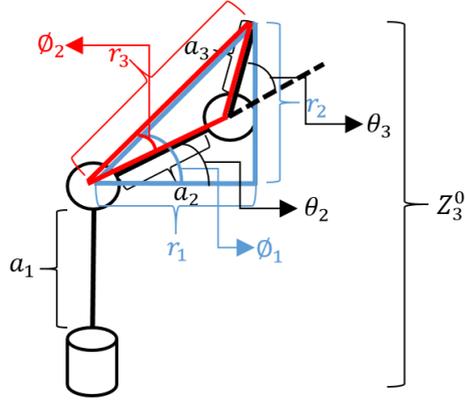


Gambar 2.8 Manipulator Robot 3 Sendi Dari Sisi Atas

Dari Gambar 2.8 didapatkan persamaan kinematika balik pada sendi 1 yang terdapat pada Persamaan 2.11.

$$\theta_1 = \tan^{-1} \left(\frac{Y_3^0}{X_3^0} \right) \quad (2.11)$$

Tahap yang kedua adalah melihat manipulator dari sisi samping. Pada Gambar 2.9 dapat dilihat manipulator robot dari sisi samping.



Gambar 2.9 Manipulator Robot 3 Sendi Dari Sisi Samping

$$r_1 = \sqrt{(X_3^0)^2 + (Y_3^0)^2} \quad (2.12)$$

$$r_2 = Z_3^0 - a_1 \quad (2.13)$$

$$r_3 = \sqrt{r_1^2 + r_2^2} \quad (2.14)$$

$$a_3^2 = a_2^2 + r_3^2 - 2a_2r_3\cos\phi_1 \quad (2.15)$$

$$r_3^2 = a_2^2 + a_3^2 - 2a_2a_3\cos\phi_3 \quad (2.16)$$

$$\phi_1 = \cos^{-1} \left(\frac{a_3^2 - a_2^2 - r_3^2}{-2a_2r_3} \right) \quad (2.17)$$

$$\phi_2 = \tan^{-1} \left(\frac{r_2}{r_1} \right) \quad (2.18)$$

$$\phi_3 = \cos^{-1} \left(\frac{r_3^2 - a_2^2 - a_3^2}{-2a_2a_3} \right) \quad (2.19)$$

$$\theta_2 = \phi_2 - \phi_1 \quad (2.20)$$

$$\theta_3 = 180 - \phi_3 \quad (2.21)$$

Dari Gambar 2.14 didapatkan persamaan kinematika balik pada sendi 2 dan sendi 3 pada Persamaan 2.20 dan 2.21.

2.8.2 Pendekatan Numerik

Pada umumnya pendekatan numerik tidak mengutamakan diperoleh solusi yang tepat, tetapi mengusahakan perumusan metode yang menghasilkan solusi pendekatan yang berbeda dari solusi yang tepat sebesar suatu nilai yang dapat diterima. Pendekatan numerik memanfaatkan perangkat komputer untuk melakukan perhitungan secara berulang – ulang agar memperoleh solusi kinematika balik. Komputer akan menghitung semua kemungkinan solusi secara berulang – ulang sampai diperoleh suatu solusi yang sesuai untuk sudut – sudut setiap sendi yang dibutuhkan agar bisa mencapai posisi dan orientasi yang diinginkan.

2.9 Jaringan Saraf Tiruan (JST)

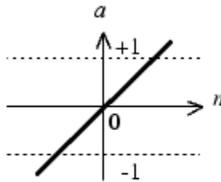
JST adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan sistem saraf manusia. JST menghubungkan sejumlah masukan dan keluaran suatu sistem yang diorganisasikan dalam lapisan pemroses. JST terdiri atas elemen pemroses bernama *neuron*, yang dihubungkan dengan elemen pemroses lainnya. Setiap *neuron* dihubungkan ke *neuron* lainnya oleh penghubung terarah dengan bobot tertentu. Bobot tersebut merepresentasikan informasi yang akan digunakan oleh jaringan untuk menyelesaikan suatu permasalahan.

2.9.1 Fungsi Aktivasi JST

Setiap *neuron* mempunyai keadaan internal yang disebut aktivasi yang merupakan fungsi dari masukan yang diterimanya. Kemudian *neuron* akan mengirim aktivasinya ke beberapa neuron lainnya. Berikut ini adalah fungsi aktivasi yang sering digunakan dalam JST.

- a. Fungsi identitas atau fungsi linear adalah fungsi aktivasi dimana nilai masukan dan nilai keluaran sama. Fungsi linear dapat dinyatakan dengan Persamaan 2.22, dimana a adalah kemiringan dari fungsi. Jika $a=1$ maka fungsi aktivasi tersebut adalah fungsi identitas. Fungsi aktivasi ini dapat dilihat pada Gambar 2.10.

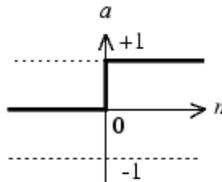
$$y = f(x) = ax \tag{2.22}$$



Gambar 2.10 Fungsi Aktivasi Linear [4]

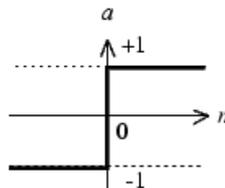
- b. Fungsi *hard limit* adalah fungsi aktivasi yang memiliki nilai antara satu dan nol. Terdapat dua jenis fungsi *hard limit*, yaitu fungsi *hard limit* biner dan fungsi *hard limit* bipolar. Fungsi *hard limit* biner ditunjukkan pada Persamaan 2.23 dan Gambar 2.11, dan fungsi *hard limit* bipolar ditunjukkan pada Persamaan 2.24 dan Gambar 2.12.

$$y = f(x) = \begin{cases} 0 & \text{jika } x < 0 \\ 1 & \text{jika } x \geq 0 \end{cases} \tag{2.23}$$



Gambar 2.11 Fungsi Aktivasi *Hard Limit* Biner [4]

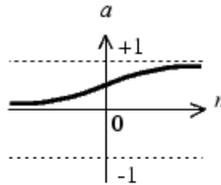
$$y = f(x) = \begin{cases} -1 & \text{jika } x < 0 \\ 1 & \text{jika } x \geq 0 \end{cases} \tag{2.24}$$



Gambar 2.12 Fungsi Aktivasi *Hard Limit* Bipolar [4]

- c. Fungsi *sigmoid* adalah fungsi yang sering digunakan dalam JST. Fungsi aktivasi ini membawa masukan ke keluaran dengan menggunakan rumus *log-sigmoid*. Fungsi *sigmoid* ini memiliki nilai antara 0 sampai dengan 1. Fungsi *sigmoid* ditunjukkan pada Persamaan 2.25 dan Gambar 2.13.

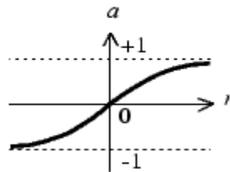
$$a = \frac{1}{1+e^{-n}} \quad (2.25)$$



Gambar 2.13 Fungsi Aktivasi *Sigmoid* [4]

- d. Fungsi *sigmoid bipolar* adalah fungsi aktivasi yang membawa nilai masukan pada keluaran dengan menggunakan rumus *hyperbolic tangen sigmoid*. Fungsi *sigmoid bipolar* memiliki nilai maksimal adalah 1 dan nilai minimal adalah -1. Fungsi *sigmoid bipolar* ditunjukkan pada Persamaan 2.26 dan Gambar 2.14.

$$a = \frac{1-e^n}{1+e^{-n}} \quad (2.26)$$



$$a = \text{tansig}(n)$$

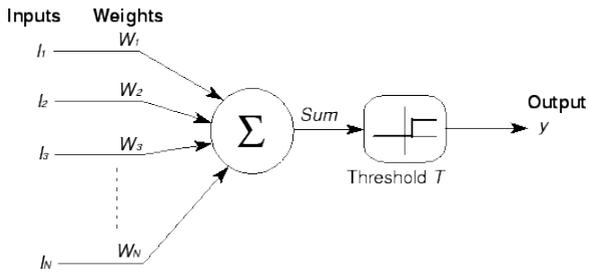
Gambar 2.14 Fungsi Aktivasi *Sigmoid Bipolar* [4]

2.9.2 Arsitektur JST

JST digambarkan ke dalam lapisan-lapisan neuron. Umumnya neuron dalam satu lapis mempunyai perilaku yang sama. Kunci utama yang menentukan perilaku neuron adalah fungsi aktivasi dan pola hubungan terbobot pengiriman dan penerimaan sinyal. Susunan neuron dalam bentuk lapisan-lapisan dan pola hubungan di dalam dan antar

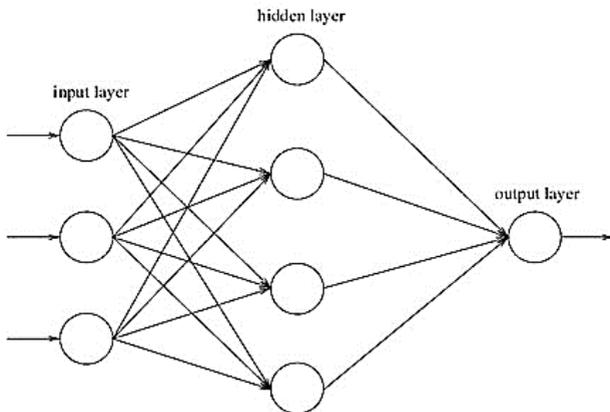
lapisan disebut arsitektur jaringan . Berdasarkan arsitekturnya kita mengenal beberapa jaringan sebagai berikut.

- a. Jaringan satu lapis, jaringan ini mempunyai satu lapis hubungan terbobot dengan neuron masukan yang menerima sinyal dari luar dan neuron keluaran yang menghasilkan respon jaringan. Untuk jaringan satu lapis ini bobot untuk suatu neuron keluaran tidak akan mempengaruhi bobot untuk keluaran neuron yang lainnya. Arsitektur jaringan satu lapis dapat dilihat pada Gambar 2.15.



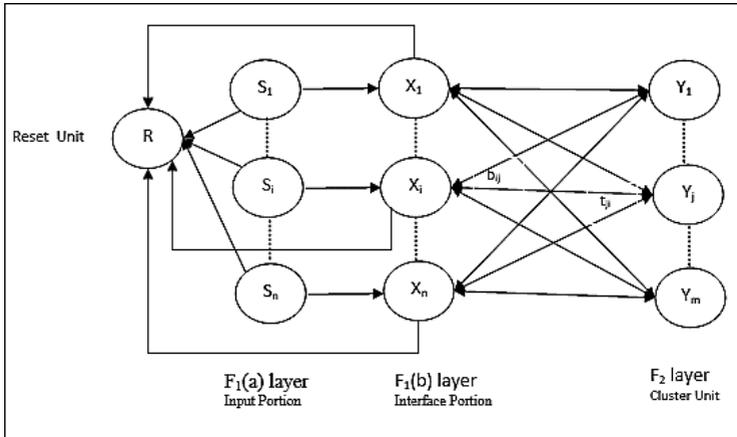
Gambar 2.15 Arsitektur Jaringan Satu Lapis [4]

- b. Jaringan Multilapis, jaringan dengan satu atau lebih lapisan tersembunyi yang terdapat diantara neuron masukan dan neuron keluaran. Arsitektur jaringan satu lapis dapat dilihat pada Gambar 2.16.



Gambar 2.16 Arsitektur Jaringan Multilapis [4]

- c. Jaringan Kompetitif, Dalam lapisan kompetitif ini hanya ada satu neuron dengan aktivasi terbesar yang akan keluar sebagai pemenang. Lapisan kompetitif ini merupakan bagian terbesar dari jaringan saraf. Contoh jaringan kompetitif adalah ART (*Adaptive Resonance Theory*) yang terdapat pada Gambar 2.17.



Gambar 2.17 Arsitektur Jaringan Kompetitif [4]

2.9.3 Metode Pembelajaran JST

Belajar atau *training* bagi JST merupakan proses penentuan nilai bobot jaringan. Ada dua macam metode belajar, yaitu metode terbimbing (*supervised training*) dan metode tak terbimbing (*unsupervised training*).

- Metode terbimbing, pada metode ini, *training* dilakukan dengan memberikan sederetan vektor *training* atau pola-pola sesuai dengan vektor target keluaran. Aplikasi dari metode belajar ini misalnya pada masalah klasifikasi pola, penggabungan pola yang dikenal sebagai *associative memory*, serta pemetaan *nonlinear* dari n dimensi vektor masukan ke m dimensi vektor keluaran.
- Metode tak terbimbing, pada metode ini hanya diperlukan sederetan vektor masukan tanpa vektor targetnya. Jaringan akan memodifikasi bobotnya sehingga vektor masukan yang mempunyai kesamaan paling banyak akan dikelompokkan ke dalam neuron keluaran yang

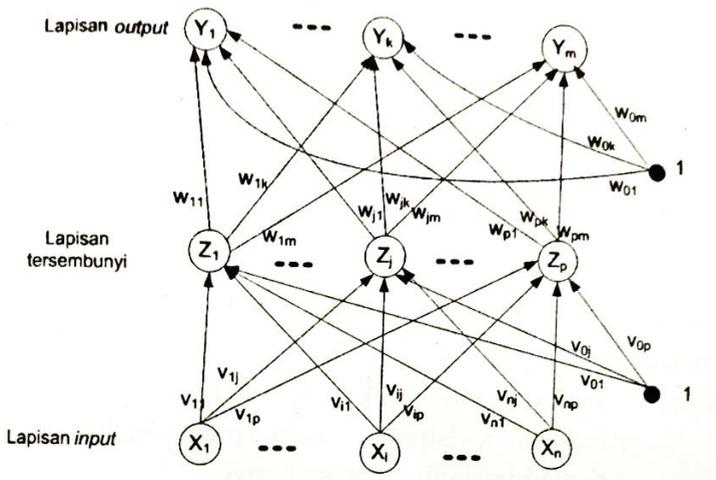
sama (*cluster*). Jaringan ini sering disebut sebagai *self-organizing neural nets*.

2.9.4 JST *Bacpropagation*

Metode pembelajaran JST *backpropagation* menggunakan ide perambatan balik nilai *error* atau lebih dikenal sebagai *generalized delta rule*. *Delta rule* merupakan metode *gradient descent* untuk meminimalkan jumlah kuadrat *error* keluaran yang dihasilkan oleh jaringan. Pembelajaran jaringan menggunakan metode *backpropagation* meliputi tiga langkah yaitu.

1. *feedforward*
2. Perhitungan dan *backpropagation* nilai *error*
3. Penyesuaian nilai bobot jaringan berdasarkan besarnya *error* tersebut.

Arsitektur JST *beckpropagation* dengan satu lapisan *hidden* ditunjukkan pada Gambar 2.18.



Gambar 2.18 Arsitektur JST *Backpropagation* [4]

Selama perhitungan arah maju, setiap neuron masukan (x_i) menerima sinyal masukan dan memancarkannya ke setiap neuron tersembunyi z_1, \dots, z_n . Setiap neuron tersembunyi kemudian menghitung aktivasi-nya dan mengirim sinyal aktivasi tersebut ke setiap neuron

keluaran. Setiap neuron keluaran y_k menghitung aktivasinya untuk menyatakan respon jaringan terhadap pola masukan yang diberikan.

Selama proses pembelajaran, setiap neuron keluaran dibandingkan aktivasinya y_k dengan nilai target t_k untuk menentukan besarnya *error*. Berdasarkan besarnya *error* ini, kemudian dihitung faktor δ_k ($k = 1, \dots, m$). δ_k digunakan untuk mendistribusikan nilai *error* pada neuron keluaran y_k kembali ke semua neuron pada lapisan sebelumnya. Nilai ini juga digunakan untuk memperbarui nilai bobot antara neuron keluaran dan neuron-neuron tersembunyi. Dengan cara yang sama, faktor δ_j ($j = 1, \dots, p$) dihitung untuk setiap neuron tersembunyi z_j . Nilai δ_j tidak dirambatkan kembali ke neuron masukan, namun digunakan untuk memperbarui nilai bobot antara neuron-neuron tersembunyi dengan neuron masukan. Setelah semua faktor δ dihitung, bobot untuk semua lapisan diperbarui secara simultan.

Algoritma yang digunakan dalam proses pembelajaran JST *backpropagation* adalah sebagai berikut.

Langkah 0 : Pemberian inisialisasi bobot

Langkah 1 : Untuk setiap pasangan data pelatihan, lakukan langkah 2 hingga langkah 7 (*feedforward*).

Langkah 2 : Setiap neuron masukan ($x_i, i = 1, \dots, n$) menerima sinyal masukan x_i dan meneruskan sinyal tersebut ke lapisan di atasnya (lapisan tersembunyi).

Langkah 3 : Setiap neuron tersembunyi dikalikan dengan faktor pembobot dan ditambahkan dengan nilai pembobot pada bias.

$$z_in_j = v_{oj} + \sum_{i=1}^n x_i v_{ij} \quad (2.27)$$

Kemudian menerapkan fungsi aktivasinya untuk menghitung sinyal keluaran:

$$z_j = f(z_in_j) \quad (2.28)$$

Langkah 4 : Setiap neuron keluaran ($y_k, k = 1, 2, 3, \dots, m$) dikalikan dengan faktor pembobot dan dijumlahkan:

$$y_in_k = w_{ok} + \sum_{j=1}^p z_j w_{jk} \quad (2.29)$$

dan menerapkan fungsi aktivasinya untuk menghitung sinyal keluaran

$$y_k = f(y_{in_k}) \quad (2.30)$$

Backpropagation nilai *error*:

Langkah 5: Setiap neuron keluaran ($y_k, k = 1, \dots, m$) menerima pola target sesuai dengan pola masukan, dan dihitung nilai *error*-nya

$$\delta_k = (t_k - y_k)f'(y_{in_k}) \quad (2.31)$$

Menghitung nilai koreksi bobot yang digunakan untuk memperbarui nilai bobot w_{jk} :

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (2.32)$$

Menghitung nilai koreksi bias yang digunakan untuk memperbarui nilai w_{ok} :

$$\Delta w_{ok} = \alpha \delta_k \quad (2.33)$$

Langkah 6 : Setiap neuron tersembunyi ($z_j, j = 1, \dots, m$) dikalikan dengan *delta* dan dijumlahkan sebagai masukan ke unit-unit lapisan berikutnya.

$$\delta_{in_j} = \sum_{k=1}^m \delta_j w_{jk} \quad (2.34)$$

Selanjutnya dikalikan dengan turunan dari fungsi aktivasinya untuk menghitung *error*

$$\delta_j = \delta_{in_j} f'(y_{in_j}) \quad (2.35)$$

Selanjutnya menghitung nilai koreksi bobot yang digunakan untuk memperbarui nilai bobot v_{jk} :

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.36)$$

Selanjutnya menghitung nilai koreksi bias yang digunakan untuk memperbarui nilai v_{oj} :

$$\Delta w_{oj} = \alpha \delta_j \quad (2.37)$$

Langkah 7 : Setiap neuron keluaran ($y_k, k = 1, \dots, m$) memperbarui nilai bias dan bobot ($j, k = 1, \dots, p$):

$$w_{jk}(new) = w_{oj}(old) + \Delta w_{jk} \quad (2.38)$$

Setiap neuron tersembunyi ($z_j, j = 1, \dots, m$) memperbarui nilai bias dan bobot ($k = 0, \dots, n$)

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij} \quad (2.39)$$

Langkah 8 : Menguji kondisi pemberhentian.

Langkah 9 : Ulangi langkah 1 hingga 8 sampai kondisi akhir iterasi terpenuhi.

Daftar notasi dari algoritma yang digunakan dalam pembelajaran JST *backpropagation* adalah sebagai berikut:

x_i = masukan ke- i pada lapisan masukan

z_in_j = net masukan untuk z_j

v_{oj} = nilai bobot pada bias untuk unit z_j

v_{ij} = nilai bobot dari unit x_i ke unit z_j

z_j = nilai aktivasi dari z_in_j

y_{in_k} = net masukan untuk y_k

w_{ok} = nilai bobot pada bias untuk keluaran unit y_k

w_{jk} = nilai bobot dari z_j ke unit y_k

y_k = keluaran unit ke- k pada lapisan keluaran

δ_k = faktor pengaturan nilai bobot sambungan pada lapisan keluaran

t_k = target keluaran dari JST

Δw_{jk} = selisih antara $w_{jk}(t)$ dengan $w_{jk}(t + 1)$

α = *learning rate* (konstanta laju pembelajaran)

Δw_{ok} = selisih antara $w_{ok}(t)$ dengan $w_{ok}(t + 1)$

δ_j = faktor pengaturan nilai bobot sambungan pada lapisan *hidden*

Δv_{ij} = selisih antara $v_{ij}(t)$ dengan $v_{ij}(t + 1)$

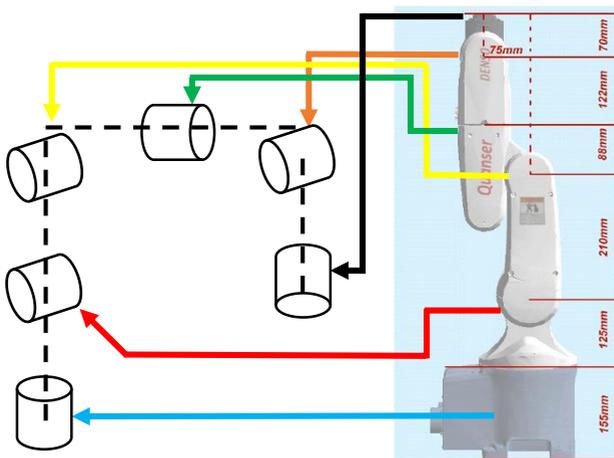
-----*Halaman ini sengaja dikosongkan*-----

BAB 3 PERANCANGAN SISTEM

Pada bab ini akan dibahas mengenai perancangan JST untuk menyelesaikan kinematika balik manipulator robot denso 6-DoF. Dalam tahap ini hal yang pertama kali dilakukan adalah menentukan parameter DH dari manipulator robot denso 6-DoF. Setelah parameter DH didapatkan maka dilakukan perancangan dan perhitungan kinematika maju, dimana hasil dari perhitungan kinematika maju akan digunakan sebagai data uji kinematika balik. Setelah itu akan dilakukan perancangan struktur dari JST untuk menyelesaikan permasalahan pada kinematika balik. Ketika semua telah diselesaikan maka akan dilakukan simulasi pada matlab.

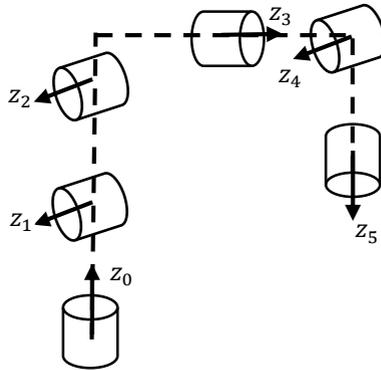
3.1 Menentukan Parameter *Denavit Hartenberg*

Sebelum mencari kinematika balik, terlebih dahulu dilakukan perancangan dan perhitungan kinematika maju. Persamaan kinematika maju akan dicari menggunakan standarisasi aturan parameter DH. Letak masing-masing sendi dan panjang masing-masing lengan dapat dilihat pada Gambar 3.1.



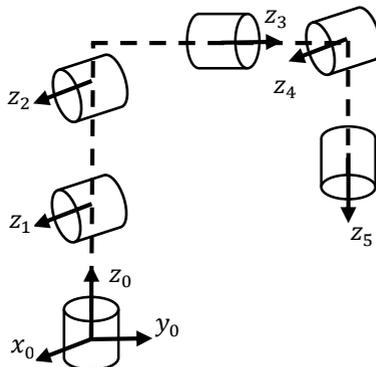
Gambar 3.1 Letak Sendi dan Panjang Lengan

Langkah 1: Menempatkan dan melabeli sumbu z_0, \dots, z_{n-1} pada masing-masing sendi. Karena terdapat 6 sendi maka akan dilakukan pelabelan z_0 hingga z_5 . Sumbu putar diambil sebagai sumbu z . Pemberian label sumbu z dapat dilihat pada Gambar 3.2.



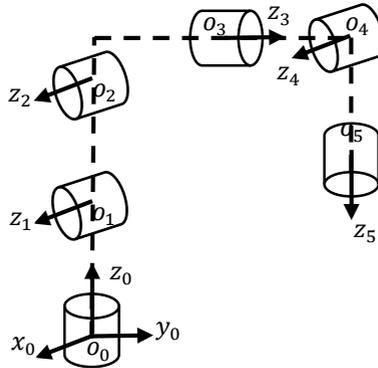
Gambar 3.2 Pemberian Label Sumbu z

Langkah 2: menetapkan *base frame*. Menentukan *origin* pada sumbu z_0 . Sumbu x_0 dan y_0 dipilih secara acak sesuai kaidah tangan kanan. Pemberian label sumbu x_0 dan y_0 dapat dilihat pada Gambar 3.3.



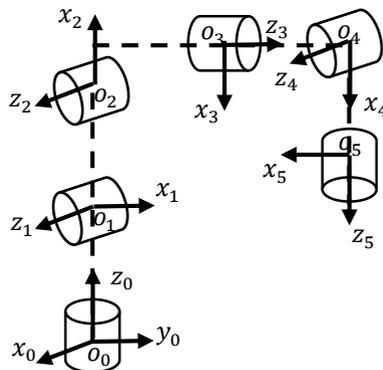
Gambar 3.3 Pemberian Label Sumbu x_0 dan y_0

Langkah 3: menempatkan *origin* o_i , dimana *common normal* z_i dan z_{i-1} memotong z_i . Jika z_i memotong z_{i-1} tempatkan o_i pada perpotongan tersebut. Jika z_i dan z_{i-1} paralel, tempatkan o_i pada sendi i . Pemberian Label Sumbu o_i dapat dilihat pada Gambar 3.4.



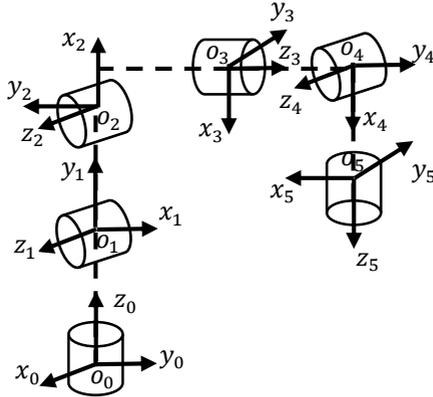
Gambar 3.4 Pemberian Label Sumbu o_i

Langkah 4: menetapkan x_i (untuk $i=1$ sampai dengan 5) sepanjang *common normal* antara z_{i-1} dan z_i melalui o_i , atau arah normal dari z_{i-1} - z_i jika z_{i-1} dan z_i berpotongan. Pemberian Label Sumbu x_i dapat dilihat pada Gambar 3.5.



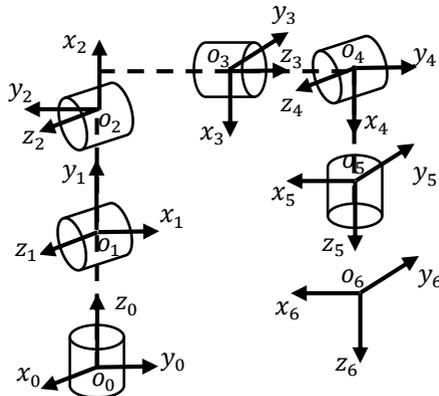
Gambar 3.5 Pemberian Label Sumbu x_i

Langkah 5: menetapkan y_i (untuk $i=1$ sampai dengan 5) dengan aturan kaidah tangan kanan. Pemberian Label Sumbu y_i dapat dilihat pada Gambar 3.6



Gambar 3.6 Pemberian Label Sumbu y_i

Langkah 6 : menetapkan *frame end-effector* o_n, x_n, y_n , dan z_n , dengan mengasumsikan bahwa sendi ke- n adalah sendi putar. Arah sumbu z_6 mengikuti sumbu z terakhir, yaitu z_5 , selanjutnya melengkapi o_6, x_6 , dan y_6 dengan menggunakan kaidah tangan kanan. Pemberian Label Sumbu o_n, x_n, y_n , dan z_n dapat dilihat pada Gambar 3.7.



Gambar 3.7 Pemberian Label Sumbu o_n, x_n, y_n , dan z_n .

Langkah 7 : Membuat tabel parameter lengan $a_i, d_i, \alpha_i, \theta_i$.

Tabel 3.1 Parameter DH

Sendi	a_i	d_i	α_i	θ_i	Range (derajat)
1	0	280	90	θ_1^*	-160 s/d 160
2	210	0	0	θ_2^*	-120 s/d 120
3	75	0	-90	θ_3^*	20 s/d 160
4	0	210	90	θ_4^*	-160 s/d 160
5	0	0	-90	θ_5^*	-120 s/d 120
6	0	70	0	θ_6^*	-360 s/d 360

Parameter d_i dan a_i pada tabel parameter akan diganti dengan variabel untuk mempermudah perhitungan. Sedangkan untuk variabel yang memiliki tanda (*) pada θ_1 hingga θ_6 menunjukkan bahwa variabel tersebut dapat berubah-ubah.

Tabel 3.2 Indeks Parameter DH

Sendi	a_i	d_i	α_i	θ_i	Range (derajat)
1	0	d_1	90	θ_1^*	-160 s/d 160
2	a_2	0	0	θ_2^*	-120 s/d 120
3	a_3	0	-90	θ_3^*	20 s/d 160
4	0	d_4	90	θ_4^*	-160 s/d 160
5	0	0	-90	θ_5^*	-120 s/d 120
6	0	d_6	0	θ_6^*	-360 s/d 360

3.2 Membentuk Transformasi Homogen

Transformasi Homogen adalah matriks yang merepresentasikan nilai translasi dan rotasi dari sebuah sendi. Setelah menemukan parameter DH maka akan dibentuk transformasi homogen pada masing-masing A_i (untuk $i=1$ sampai dengan 6).

$$A_1 = Rot_{z,\theta_1} Trans_{z,d_1} Rot_{x,\alpha_1}$$

$$A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{90} & -s_{90} & 0 \\ 0 & s_{90} & c_{90} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

$$A_2 = Rot_{z,\theta_2} Trans_{x,a_2}$$

$$A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

$$A_3 = Rot_{z,\theta_3} Rot_{x,\alpha_3} Trans_{x,a_3}$$

$$A_3 = \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{-90} & -s_{-90} & 0 \\ 0 & s_{-90} & c_{-90} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} c_3 & 0 & -s_3 & a_3 c_3 \\ s_3 & 0 & c_3 & a_3 s_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

$$A_4 = Rot_{z,\theta_4} Trans_{z,d_4} Rot_{x,\alpha_4}$$

$$A_4 = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{90} & -s_{90} & 0 \\ 0 & s_{90} & c_{90} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} c_4 & 0 & s_4 & 0 \\ s_4 & 0 & -c_4 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$$A_5 = Rot_{z,\theta_5} Trans_{z,d_5}$$

$$A_5 = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{-90} & -s_{-90} & 0 \\ 0 & s_{-90} & c_{-90} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} c_5 & 0 & -s_5 & 0 \\ s_5 & 0 & c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$$A_6 = Rot_{z,\theta_6} Trans_{z,d_6}$$

$$A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

Pada Persamaan 3.1 sampai dengan Persamaan 3.6 adalah persamaan transformasi homogen dari manipulator robot denso 6-DoF yang dihitung dari lengan satu hingga lengan enam, selanjutnya transformasi homogen ini digunakan untuk menghitung persamaan kinematika maju. Pada Persamaan 3.1 sampai dengan Persamaan 3.6 “s” adalah sin dan “c” adalah cos.

3.3 Mencari Persamaan Kinematika Maju

Setelah transformasi homogen didapatkan, maka langkah selanjutnya adalah mencari persamaan kinematika maju, dimana hasil dari persamaan kinematika maju adalah matriks yang berisi nilai rotasi dan posisi dari *end-effector*.

$$T_0^6 = A_1 A_2 A_3 A_4 A_5 A_6 = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

$$T_0^6 = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_3 & 0 & -s_3 & a_3 c_3 \\ s_3 & 0 & c_3 & a_3 s_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} c_4 & 0 & s_4 & 0 \\ s_4 & 0 & -c_4 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_5 & 0 & -s_5 & 0 \\ s_5 & 0 & c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Pada Persamaan 3.7 terdapat nilai $n, s,$ dan a yang merupakan titik orientasi yang dan nilai p adalah nilai dari posisi dari *end-effector* dari manipulator robot denso 6-DoF. Besar nilai n, s, a, p dapat dicari dengan perhitungan perkalian matriks A_1 hingga matriks A_6 .

$$n_x = c_6 * ((c_5 * (((c_1 * c_2 * c_3) - (c_1 * s_2 * s_3)) * c_4) - (s_1 * s_4))) + (s_5 * (-c_1 * c_2 * c_3) - (c_1 * s_2 * s_3))) + s_6 * (-(((c_1 * c_2 * c_3) - (c_1 * s_2 * s_3)) * s_4) + (s_1 * c_4)) \quad (3.8)$$

$$n_y = c_6 * ((c_5 * (((s_1 * c_2 * c_3) - (s_1 * s_2 * s_3)) * c_4) + (c_1 * s_4))) + (s_5 * (-s_1 * c_2 * s_3) - (s_1 * s_2 * c_3))) + s_6 * (-(((s_1 * c_2 * c_3) - (s_1 * s_2 * s_3)) * s_4) - (c_1 * c_4)) \quad (3.9)$$

$$n_z = c_6 * ((c_5 * ((s_2 * c_3) + (c_2 * s_3)) * c_4) + (s_5 * (-s_2 * s_3) + (c_2 * c_3))) + s_6 * (-(((s_2 * c_3) + (c_2 * s_3)) * s_4)) \quad (3.10)$$

$$s_x = -s_6 * ((c_5 * (((c_1 * c_2 * c_3) - (c_1 * s_2 * s_3)) * c_4) - (s_1 * s_4))) + (s_5 * (-c_1 * c_2 * s_3) - (c_1 * s_2 * c_3))) + c_6 * (-(((c_1 * c_2 * c_3) - (c_1 * s_2 * s_3)) * s_4) + (s_1 * c_4)) \quad (3.11)$$

$$s_y = -s_6 * ((c_5 * (((s_1 * c_2 * c_3) - (s_1 * s_2 * s_3)) * c_4) + (c_1 * s_4))) + (s_5 * (-s_1 * c_2 * s_3) - (s_1 * s_2 * c_3))) + c_6 * (-(((s_1 * c_2 * c_3) - (s_1 * s_2 * s_3)) * s_4) - (c_1 * c_4)) \quad (3.12)$$

$$s_z = -s_6 * ((c_5 * ((s_2 * c_3) + (c_2 * s_3)) * c_4) + (s_5 * (-s_2 * s_3) + (c_2 * c_3))) + c_6 * (-(((s_2 * c_3) + (c_2 * s_3)) * s_4)) \quad (3.13)$$

$$a_x = (-s_5 * (((c_1 * c_2 * c_3) - (c_1 * s_2 * s_3)) * c_4) - (s_1 * s_4)) + (c_5 * (-c_1 * c_2 * s_3) - (c_1 * s_2 * c_3)) \quad (3.14)$$

$$a_y = (-s_5 * (((s_1 * c_2 * c_3) - (s_1 * s_2 * s_3)) * c_4) + (c_1 * s_4)) + (c_5 * (-s_1 * c_2 * s_3) - (s_1 * s_2 * c_3)) \quad (3.15)$$

$$a_z = (-s_5 * ((s_2 * c_3) + (c_2 * s_3)) * c_4) + (c_5 * (-s_2 * s_3) + (c_2 * c_3))) \quad (3.16)$$

$$p_x = d_6 * (((-s_5 * (((c_1 * c_2 * c_3) - (c_1 * s_2 * s_3)) * c_4) - (s_1 * s_4))) + (c_5 * (-(c_1 * c_2 * s_3) - (c_1 * s_2 * c_3)))) + ((-(c_1 * c_2 * s_3) - (c_1 * s_2 * c_3)) * d_4) + (c_1 * c_2 * c_3 * a_3) - (c_1 * s_2 * s_3 * a_3) + (c_1 * c_2 * a_2)) \quad (3.17)$$

$$p_y = d_6 * (((-s_5 * (((s_1 * c_2 * c_3) - (s_1 * s_2 * s_3)) * c_4) + (c_1 * s_4))) + (c_5 * (-(s_1 * c_2 * s_3) - (s_1 * s_2 * c_3)))) + ((-(s_1 * c_2 * s_3) - (s_1 * s_2 * c_3)) * d_4) + (s_1 * c_2 * c_3 * a_3) - (s_1 * s_2 * s_3 * a_3) + (s_1 * c_2 * a_2)) \quad (3.18)$$

$$p_z = d_6 * ((-s_5 * ((s_2 * c_3) + (c_2 * s_3)) * c_4) + (c_5 * (-s_2 * s_3) + (c_2 * c_3))) + (((-s_2 * s_3) + (c_2 * c_3)) * d_4) + (s_2 * c_3 * a_3) + (c_2 * s_3 * a_3) + (s_2 * a_2) + d_1 \quad (3.19)$$

Untuk mendapatkan nilai *roll*, *pitch*, dan *yaw* dari matriks orientasi maka dapat menggunakan rumus berikut,

$$pitch = \sin^{-1}(a_x) \quad (3.20)$$

$$roll = \sin^{-1}\left(\frac{s_x}{-\cos(pitch)}\right) \quad (3.21)$$

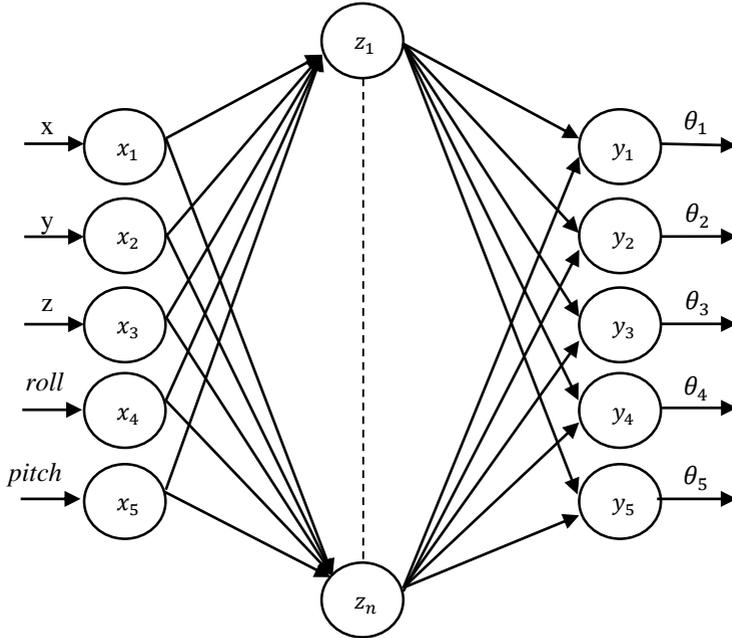
$$yaw = -\sin^{-1}\left(\frac{a_y}{\cos(pitch)}\right) \quad (3.22)$$

3.4 Perancangan Kinematika Balik Dengan JST

Terdapat beberapa tahap dalam perancangan kinematika balik dengan menggunakan JST. tahap yang pertama dilakukan adalah mengatur struktur dari JST yang akan digunakan, Struktur tersebut terdiri dari jumlah masukan dan keluaran yang digunakan, jumlah lapisan dan *neuron* yang digunakan, dan fungsi aktivasi yang diterapkan pada JST. Setelah JST telah dibuat, maka akan dilakukan dua tahap perhitungan yaitu perhitungan *feedforward* dan *backpropagation*.

3.4.1 Perancangan Struktur JST

Struktur JST yang digunakan adalah *multilayer perceptron*, yang terdiri dari lapisan masukan, lapisan *hidden*, dan lapisan keluaran.



Gambar 3.8 Struktur Kinematika Balik Dengan JST.

Pada lapisan masukan terdapat 5 masukan (x_1, x_2, x_3, x_4, x_5), yang digunakan untuk masukan posisi x, posisi y, posisi z, orientasi *roll*, dan orientasi *pitch* dari manipulator robot. Pada lapisan *hidden* terdapat beberapa *neuron* (z_1, \dots, z_n) dimana pada pengujian jumlah *neuron* di bab 4, z_n akan diubah-ubah menjadi 5, 10, 25, dan 50, sedangkan pada pengujian lainnya z_n dibuat tetap yaitu 50. Fungsi aktivasi yang digunakan pada lapisan *hidden* adalah fungsi aktivasi *sigmoid biner*. Pada lapisan keluaran terdapat 5 keluaran (z_1, z_2, z_3, z_4, z_5), yang menunjukkan besar sudut untuk sendi 1 hingga sendi 5 pada manipulator robot, dan fungsi aktivasi yang digunakan pada lapisan ini adalah fungsi aktivasi *linier*. Struktur dari kinematika balik yang dengan menggunakan

kinematika balik ditunjukkan pada Gambar 3.8. dimana pada Gambar 3.8 ditunjukkan masing-masing lapisan yang digunakan pada JST.

3.4.2 Perhitungan *Feedforward* dan *Backward*

Dalam proses jalannya data dalam JST untuk menyelesaikan kinematika balik hal pertama yang dilakukan adalah perhitungan *feedforward*. Dalam perhitungan *feedforward* ini data masukan yang masuk melalui lapisan masukan berupa posisi x,y,z dan orientasi *roll* dan *pitch* akan diteruskan ke lapisan *hidden*, dimana pada lapisan *hidden*, data masukan tersebut akan dikalikan dengan fungsi aktivasi pada lapisan *hidden*. Setelah itu data hasil aktivasi pada lapisan *hidden* akan disalurkan pada lapisan keluaran, dimana pada lapisan keluaran, data tersebut akan dikalikan kembali dengan fungsi aktivasi pada lapisan keluaran, maka didapatkan data besar sudut sendi manipulator robot. Berikut ini adalah langkah-langkah dalam melakukan perhitungan *feedforward*:

1. Pemberian nilai awal 0 untuk masing-masing bobot
2. Masing-masing masukan ($x_i, i = 1, \dots, 5$) menerima masukan berupa orientasi *roll* dan *pitch*, serta posisi x,y,z . dan masukan tersebut disebar ke setiap *neuron* lapisan *hidden*
3. Masing-masing unit pada lapisan *hidden* di kalikan dengan bobot dan dijumlahkan

$$z_in_j = \sum_{i=1}^n x_i v_{ij} \quad (3.23)$$

Dimana ($j = 1, \dots, 50$) karena terdapat 50 *neuron* pada lapisan *hidden* dan ($i = 1, \dots, 5$) karena terdapat 5 masukan. Kemudian hasilnya dihitung dengan fungsi aktivasi *sigmoid*.

$$z_j = f(z_in_j)$$

$$z_j = \frac{1}{1+e^{-(z_in_j)}} \quad (3.24)$$

4. Masing-masing unit pada lapisan keluaran di kalikan dengan bobot dan dijumlahkan

$$y_in_k = \sum_{j=1}^p z_j w_{jk} \quad (3.25)$$

Dimana ($k = 1, \dots, 5$) karena terdapat 50 *neuron* pada lapisan *hidden* dan ($j = 1, \dots, 50$) karena terdapat 5 keluaran. Kemudian hasilnya dihitung dengan fungsi aktivasi *linier*.

$$y_k = f(y_in_k)$$

$$y_k = y_in_k \quad (3.26)$$

Setelah melakukan perhitungan *feedforward* maka didapatkan keluaran data berupa besar sudut masing-masing sendi manipulator robot, tetapi karena keluaran data tersebut belum sesuai dengan tujuan yang ingin dicapai, maka dilakukan perhitungan *backpropagation*. Dalam perhitungan ini akan dicari besar *error* antara tujuan dan hasil perhitungan JST, dan akan dilakukan revisi bobot dalam JST, sehingga besar *error* antara tujuan dan hasil perhitungan JST dapat diperkecil sesuai dengan ketelitian yang telah kita tentukan. Berikut ini adalah langkah-langkah dalam melakukan perhitungan *backpropagation*:

5. Masing-masing keluaran ($y_k, k = 1, \dots, 5$) yang merupakan besar sudut $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$ dan nilai θ_6 yang di nolkan dimasukkan kedalam persamaan kinematika maju, dan didapatkan keluaran dari JST berupa $roll_n, pitch_n, x_n, y_n, z_n$.
6. Mencari *gradient error* sendi 1 sampai sendi 5

$$\delta_k = (t_k - y_k) \quad (3.27)$$

Formulasi target θ_1 sampai θ_5 yang diharapkan:

$$t_1 = \tan^{-1} \left(\frac{y}{x} \right) \quad (3.28)$$

$$t_2 = \sqrt{x^2 + y^2} \quad (3.29)$$

$$t_3 = z \quad (3.30)$$

$$t_4 = roll \quad (3.31)$$

$$t_5 = pitch \quad (3.32)$$

Formulasi target θ_1 sampai θ_5 keluaran JST

$$t_1 = \tan^{-1}\left(\frac{y_n}{x_n}\right) \quad (3.33)$$

$$t_2 = \sqrt{x_n^2 + y_n^2} \quad (3.34)$$

$$t_3 = z_n \quad (3.35)$$

$$t_4 = roll_n \quad (3.36)$$

$$t_5 = pitch_n \quad (3.37)$$

Selanjutnya menghitung nilai koreksi bobot yang digunakan untuk memperbarui nilai bobot w_{jk} :

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (3.38)$$

Dimana *learning rate* ($\alpha = 0.00025$), lalu ($j = 1, \dots, 50$) karena terdapat 50 *neuron* pada lapisan *hidden* dan ($k = 1, \dots, 5$) karena terdapat 5 keluaran

7. Masing-masing pembobot yang menghubungkan unit pada lapisan keluaran dengan unit pada lapisan *hidden* dikalikan dengan delta dan dijumlahkan, maka hasilnya digunakan sebagai masukan.

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (3.39)$$

Dimana ($j = 1, \dots, 50$) karena terdapat 50 *neuron* pada lapisan *hidden* dan ($k = 1, \dots, 5$) karena terdapat 5 keluaran. Selanjutnya dikalikan dengan turunan dari fungsi aktivasinya untuk menghitung *error*

$$\begin{aligned} \delta_j &= \delta_{in_j} f'(y_{in_j}) \\ \delta_j &= \delta_{in_j} z_j (1 - z_j) \end{aligned} \quad (3.40)$$

Selanjutnya menghitung nilai koreksi bobot yang digunakan untuk memperbarui nilai bobot v_{ij} :

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (3.41)$$

Dimana *learning rate* ($\alpha = 0.0001$) , lalu ($j = 1, \dots, 50$) karena terdapat 50 *neuron* pada lapisan *hidden* dan ($i = 1, \dots, 5$) karena terdapat 5 masukan

8. Setiap neuron keluaran ($y_k, k = 1, \dots, 5$) memperbarui nilai bias dan bobot ($j = 1, \dots, 50$):

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk} \quad (2.42)$$

Setiap neuron tersembunyi ($z_j, j = 1, \dots, 50$) memperbarui nilai bias dan bobot ($i = 1, \dots, 5$)

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij} \quad (2.43)$$

9. Ulangi langkah 3 sampai 8 hingga keluaran sesuai dengan ketelitian yang diinginkan.

3.5 Simulasi Kinematika Maju dan Kinematika Balik

Dalam melakukan simulasi kinematika balik dengan JST, terlebih dahulu diambil data dari kinematika maju. Pada program kinematika maju dimasukkan besar masing-masing sendi secara acak, sehingga didapatkan orientasi *roll*, *pitch*, dan *yaw*, serta posisi x, y , dan z . Data orientasi dan posisi yang didapatkan dari program kinematika maju akan digunakan untuk masukan program kinematika balik dengan JST, dimana keluaran program ini adalah besar sudut masing-masing sendi dari manipulator robot. Dalam pengujian kinematika balik juga terdapat pengujian manipulator robot denso dalam membentuk pola segitiga dan lingkaran baik dalam bidang datar maupun dalam bidang miring. Untuk simulasi manipulator robot denso digunakan *toolbox Peter Corke* versi 10 untuk animasi dari manipulator robot.

BAB 4 SIMULASI DAN ANALISA DATA

Pada bab 4 ini akan dibahas mengenai simulasi dari program kinematika maju dan kinematika balik dengan menggunakan JST. Simulasi dilakukan menggunakan *software* Matlab versi 2017 dan *toolbox peter corke* versi 10.

4.1 Pengujian Kinematika Maju

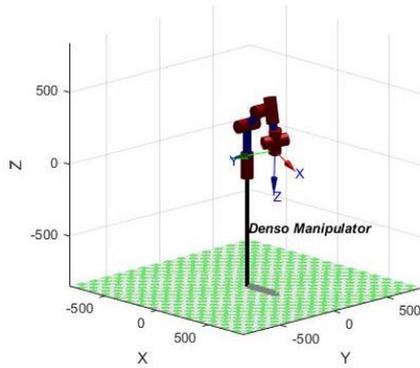
Pengujian kinematika maju manipulator robot denso dilakukan dengan cara memasukan besar sudut masing-masing sendi dari manipulator robot denso yang dapat dilihat pada Tabel 4.1 pada program kinematika maju. Dari program kinematika maju diperoleh keluaran posisi dan orientasi dari manipulator robot tersebut, yang dapat dilihat pada Tabel 4.2. keluaran posisi dan orientasi tersebut akan digunakan sebagai data uji untuk pengujian kinematika balik dengan JST.

Tabel 4.1 Masukan Kinematika Maju

No.	Besarnya sudut sendi (derajat)					
	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1	180	140	30	20	20	0
2	185	145	25	10	30	0
3	175	135	15	10	15	0
4	160	120	10	10	5	0
5	170	130	10	15	10	0
6	165	145	15	15	25	0

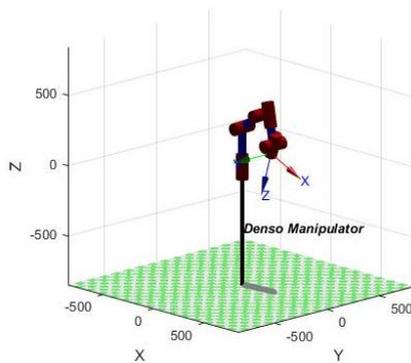
Tabel 4.2 Keluaran Kinematika Maju

No.	Orientasi (derajat)			Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>yaw</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	19.928	-8.820	-173.201	0.260	0.008	0.152
2	5.158	-19.927	-176.503	0.257	0.028	0.140
3	14.145	15.375	-178.697	0.335	-0.026	0.216
4	36.535	42.100	162.110	0.342	-0.123	0.334
5	24.863	30.309	177.121	0.357	-0.059	0.267
6	29.036	-2.453	-172.829	0.300	-0.072	0.159



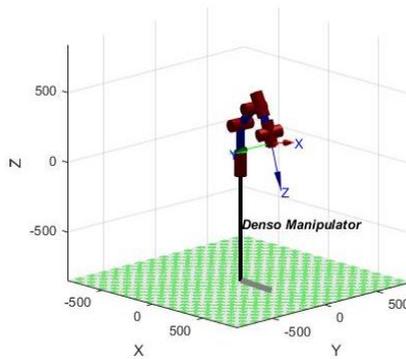
Gambar 4.1 Bentuk Manipulator Robot Denso Percobaan 1

Gambar 4.1 menunjukkan bentuk dari manipulator robot denso dengan besar sudut masing-masing sendi $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6] = [180 \ 140 \ 30 \ 20 \ 20 \ 0]$ yang menghasilkan orientasi $[roll, pitch, yaw] = [19.928 \ -8.820 \ -173.201]$ dan posisi $[x,y,z] = [0.260 \ 0.008 \ 0.152]$.



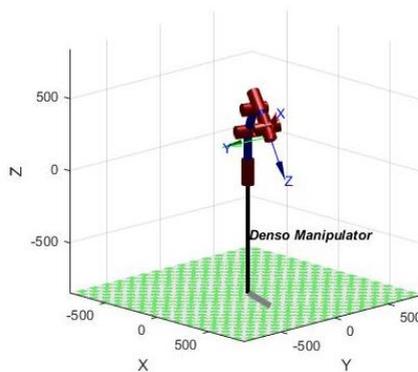
Gambar 4.2 Bentuk Manipulator Robot Denso Percobaan 2

Gambar 4.2 menunjukkan bentuk dari manipulator robot denso dengan besar sudut masing-masing sendi $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6] = [185 \ 145 \ 25 \ 10 \ 30 \ 0]$ yang menghasilkan orientasi $[roll, pitch, yaw] = [5.158 \ -19.927 \ -176.503]$ dan posisi $[x,y,z] = [0.257 \ 0.028 \ 0.140]$.



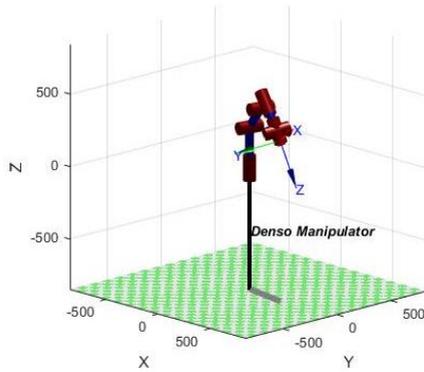
Gambar 4.3 Bentuk Manipulator Robot Denso Percobaan 3

Gambar 4.3 menunjukkan bentuk dari manipulator robot denso dengan besar sudut masing-masing sendi $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6] = [175 \ 135 \ 15 \ 10 \ 15 \ 0]$ yang menghasilkan orientasi $[roll, pitch, yaw] = [14.145 \ 15.375 \ -178.697]$ dan posisi $[x, y, z] = [0.335 \ -0.026 \ 0.216]$.



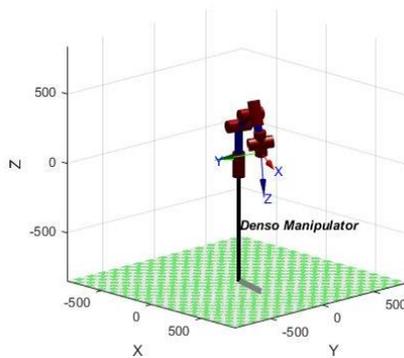
Gambar 4.4 Bentuk Manipulator Robot Denso Percobaan 4

Gambar 4.4 menunjukkan bentuk dari manipulator robot denso dengan besar sudut masing-masing sendi $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6] = [160 \ 120 \ 10 \ 10 \ 5 \ 0]$ yang menghasilkan orientasi $[roll, pitch, yaw] = [36.535 \ 42.100 \ 162.110]$ dan posisi $[x, y, z] = [0.342 \ -0.123 \ 0.334]$.



Gambar 4.5 Bentuk Manipulator Robot Denso Percobaan 5

Gambar 4.5 menunjukkan bentuk dari manipulator robot denso dengan besar sudut masing-masing sendi $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6] = [170, 130, 10, 15, 10, 0]$ yang menghasilkan orientasi $[roll, pitch, yaw] = [24.863, 30.309, 177.121]$ dan posisi $[x, y, z] = [0.357, -0.059, 0.267]$.



Gambar 4.6 Bentuk Manipulator Robot Denso Percobaan 6

Gambar 4.6 menunjukkan bentuk dari manipulator robot denso dengan besar sudut masing-masing sendi $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6] = [165, 145, 15, 15, 25, 0]$ yang menghasilkan orientasi $[roll, pitch, yaw] = [29.036, -2.453, -172.829]$ dan posisi $[x, y, z] = [0.300, -0.072, 0.159]$.

4.2 Pengujian Kinematika Balik dengan JST

Pengujian kinematika balik dengan JST dilakukan menjadi beberapa bagian, yang pertama adalah pengujian dengan mengganti jumlah *neuron* lapisan *hidden* yang digunakan dalam JST dan pengujian yang kedua adalah pengujian dengan mengganti ketelitian yang digunakan dalam JST.

4.2.1 Pengujian Jumlah *Neuron* Lapisan *Hidden*

Pengujian jumlah *neuron* lapisan *hidden* dilakukan dengan cara merubah-ubah jumlah *neuron* lapisan *hidden* yang digunakan. Masukan kinematika balik yang telah kita dapatkan dari keluaran kinematika maju sebelumnya akan diproses pada *neuron* lapisan *hidden* tersebut dan akan disalurkan pada lapisan keluaran. Pengujian ini dilakukan untuk mengetahui pengaruh dari jumlah *neuron* lapisan *hidden* terhadap jalannya program. Pengujian dilakukan dengan menggunakan 5, 10, 25, dan 50 *neuron* lapisan *hidden*. Titik berangkat manipulator robot adalah dari orientasi [*roll*, *pitch*, *yaw*] = [19.928 -8.820 -173.201] dan posisi [*x*,*y*,*z*] = [0.260 0.008 0.152], dengan besar masing-masing sendi [θ_1 , θ_2 , θ_3 , θ_4 , θ_5 , θ_6] = [180 140 30 20 20 0].

Pengujian yang pertama dilakukan adalah pengujian dengan 5 *neuron* lapisan *hidden*. Pengujian ini dilakukan dengan menggunakan *learning rate* 0.00025 dan ketelitian sebesar 0.00005. Data posisi dan orientasi keluaran dari JST dapat dilihat pada Tabel 4.3 dan data *error* antara orientasi dan posisi yang diinginkan dengan orientasi dan posisi keluaran JST dapat dilihat pada Tabel 4.8.

Tabel 4.3 Posisi dan Orientasi Keluaran 5 *Neuron* Lapisan *Hidden*

No.	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	5.158	-19.927	0.257	0.028	0.140
2	14.145	15.375	0.335	-0.026	0.216
3	36.535	42.100	0.342	-0.123	0.334
4	24.863	30.309	0.357	-0.059	0.267
5	29.036	-2.453	0.300	-0.072	0.159

Tabel 4.4 Error Posisi dan Orientasi 5 Neuron Lapisan Hidden

No.	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	2.97e-05	-4.47e-05	4.96e-05	5.42e-06	4.06e-05
2	9.30e-06	-4.38e-05	2.43e-05	-1.75e-06	8.66e-06
3	-7.40e-07	4.05e-05	-1.43e-07	9.74e-08	3.65e-07
4	4.99e-05	-1.38e-05	3.77e-05	-6.63e-06	8.05e-06
5	-5.08e-06	3.93e-05	-4.71e-05	1.12e-05	-5.72e-06
RMSE	2.64e-05	1.99e-09	3.82e-05	2.45e-09	3.66e-05

Iterasi rata-rata yang diperlukan dalam memproses data adalah 166084 dengan waktu selama 5475 detik. Pengujian yang kedua dilakukan menggunakan 10 *neuron* lapisan *hidden*. Pengujian ini dilakukan dengan menggunakan *learning rate* 0.00025 dan ketelitian sebesar 0.00005. Data posisi dan orientasi keluaran dari JST dapat dilihat pada Tabel 4.5 dan data *error* dapat dilihat pada Tabel 4.6.

Tabel 4.5 Posisi dan Orientasi Keluaran 10 Neuron Lapisan Hidden

No.	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	5.158	-19.927	0.257	0.028	0.140
2	14.145	15.375	0.335	-0.026	0.216
3	36.535	42.100	0.342	-0.123	0.334
4	24.863	30.309	0.357	-0.059	0.267
5	29.036	-2.453	0.300	-0.072	0.159

Tabel 4.6 Error Posisi dan Orientasi 10 Neuron Lapisan Hidden

No.	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	-3.59e-05	-2.59e-05	4.64e-05	5.27e-06	4.45e-05
2	-2.23e-05	-2.92e-06	5.90e-06	-2.01e-07	3.24e-05
3	1.50e-05	-2.87e-05	2.89e-07	-2.72e-07	1.33e-06
4	-3.94e-05	6.95e-06	2.84e-05	-5.03e-06	1.21e-05
5	-3.86e-05	-6.73e-06	3.98e-05	-9.63e-06	3.30e-05
RMSE	3.18e-05	1.79e-05	3.03e-05	5.40e-06	2.92e-05

Iterasi rata-rata yang diperlukan dalam memproses data adalah 90710 dengan waktu selama 3279 detik. Pengujian yang ketiga adalah pengujian dengan 25 *neuron* lapisan *hidden*. Pengujian ini dilakukan dengan menggunakan *learning rate* 0.00025 dan ketelitian sebesar 0.00005. Data posisi dan orientasi keluaran dari JST dapat dilihat pada Tabel 4.7 dan data *error* antara orientasi dan posisi yang diinginkan dengan orientasi dan posisi keluaran JST dapat dilihat pada Tabel 4.8.

Tabel 4.7 Posisi dan Orientasi Keluaran 25 *Neuron* Lapisan *Hidden*

No.	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	5.158	-19.927	0.257	0.028	0.140
2	14.145	15.375	0.335	-0.026	0.216
3	36.535	42.101	0.342	-0.123	0.334
4	24.863	30.309	0.357	-0.059	0.267
5	29.036	-2.453	0.300	-0.072	0.159

Tabel 4.8 *Error* Posisi dan Orientasi 25 *Neuron* Lapisan *Hidden*

No.	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	1.00e-06	-1.61e-05	9.28e-06	8.76e-07	4.21e-05
2	-6.77e-06	-4.84e-06	1.72e-05	-1.10e-06	1.39e-05
3	-2.56e-06	-4.96e-05	-8.32e-07	6.30e-07	3.28e-06
4	-4.14e-05	-4.84e-05	-5.56e-07	3.80e-08	1.96e-06
5	-3.07e-05	-3.29e-07	8.39e-07	-2.58e-07	2.36e-05
RMSE	2.33e-05	3.19e-05	8.79e-06	7.01e-07	2.25e-05

Iterasi rata-rata yang diperlukan dalam memproses data adalah 68873 dengan waktu selama 2506 detik. Pengujian yang keempat adalah pengujian dengan 50 *neuron* lapisan *hidden*. Pengujian ini dilakukan dengan menggunakan *learning rate* 0.00025 dan ketelitian sebesar 0.00005. Data posisi dan orientasi keluaran dari JST dapat dilihat pada Tabel 4.9 dan data *error* antara orientasi dan posisi yang diinginkan dengan orientasi dan posisi keluaran JST dapat dilihat pada Tabel 4.10.

Tabel 4.9 Posisi dan Orientasi Keluaran 50 *Neuron* Lapisan *Hidden*

No.	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	5.158	-19.927	0.257	0.028	0.140
2	14.145	15.375	0.335	-0.026	0.216
3	36.535	42.100	0.342	-0.123	0.334
4	24.863	30.309	0.357	-0.059	0.267
5	29.036	-2.453	0.300	-0.072	0.159

Tabel 4.10 *Error* Posisi dan Orientasi 50 *Neuron* Lapisan *Hidden*

No	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	2.55e-05	2.94e-05	2.35e-06	1.84e-07	4.82e-05
2	-3.13e-05	-2.73e-05	7.38e-07	2.09e-07	-1.06e-05
3	2.12e-05	1.42e-05	1.14e-06	-2.82e-07	-3.28e-06
4	-2.70e-05	-4.99e-05	1.81e-06	-4.99e-07	3.34e-06
5	1.05e-05	2.68e-06	1.32e-06	-4.97e-07	2.99e-05
RMSE	2.42e-05	2.93e-05	1.57e-06	3.61e-07	2.59e-05

Iterasi rata-rata yang diperlukan dalam memproses data adalah 43162 dengan waktu selama 2227 detik. Berdasarkan pengujian jumlah *neuron* lapisan *hidden* yang telah dilakukan, dapat dilihat bahwa jumlah *neuron* lapisan *hidden* berpengaruh pada jalannya program kinematika balik dengan JST. Jumlah *neuron* lapisan *hidden* berpengaruh pada jumlah iterasi dan lamanya iterasi berlangsung, dimana semakin banyak jumlah *neuron* pada lapisan *hidden* maka rata-rata iterasi akan menjadi lebih sedikit dan waktu yang diperlukan untuk melakukan iterasi tersebut juga akan semakin singkat.

4.2.2 Pengujian Ketelitian Kinematika Balik dengan JST

Setelah dilakukan pengujian pada jumlah *neuron* lapisan *hidden* pengujian yang dilakukan selanjutnya adalah pengujian pada ketelitian kinematika balik dengan JST. Pengujian ketelitian dilakukan dengan cara merubah-ubah ketelitian dari program kinematika balik dengan JST. Pengujian dilakukan untuk mengetahui pengaruh besarnya nilai ketelitian dalam jalannya program. Dalam pengujian ini dilakukan menggunakan tiga nilai ketelitian sebesar 0.0005, 0.0001, dan 0.00005. Titik berangkat

manipulator robot adalah dari orientasi $[roll, pitch, yaw] = [19.9289 - 8.8202 -173.201]$ dan posisi $[x,y,z] = [0.2604 0.0081 0.1525]$, dengan besar masing-masing sendi $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6] = [180 140 30 20 20 0]$. Pengujian yang pertama dilakukan adalah pengujian dengan ketelitian 0.0005. Pengujian ini dilakukan dengan menggunakan *learning rate* 0.00025 dan 50 *neuron* lapisan *hidden*. Data posisi dan orientasi keluaran dari JST dapat dilihat pada Tabel 4.11 dan data *error* antara orientasi dan posisi yang diinginkan dengan orientasi dan posisi keluaran JST dapat dilihat pada Tabel 4.12.

Tabel 4.11 Posisi dan Orientasi Keluaran dengan Ketelitian 0.0005

No.	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	5.158	-19.927	0.256	0.028	0.140
2	14.145	15.375	0.335	-0.026	0.216
3	36.536	42.101	0.342	-0.123	0.334
4	24.863	30.309	0.358	-0.059	0.267
5	29.037	-2.453	0.301	-0.072	0.159

Tabel 4.12 *Error* Posisi dan Orientasi dengan Ketelitian 0.0005

No.	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	0.0001	-6.50e-07	0.0004	5.46e-05	0.0004
2	-0.0001	0.0002	0.0003	-2.76e-05	7.26e-06
3	-8.63e-05	-0.0001	-2.64e-07	-5.72e-07	5.15e-06
4	0.0004	0.0004	-0.0004	7.86e-05	-8.88e-06
5	-0.0004	0.0001	-0.0004	0.0001	-0.0001
RMSE	0.0003	0.0002	0.0004	6.97e-05	0.0001

Iterasi rata-rata yang diperlukan dalam memproses data adalah 14310 dengan waktu selama 527 detik. Pengujian yang kedua adalah pengujian dengan ketelitian 0.0001. Pengujian ini dilakukan dengan menggunakan *learning rate* 0.00025 dan 50 *neuron* lapisan *hidden*. Data posisi dan orientasi keluaran dari JST dapat dilihat pada Tabel 4.13 dan data *error* antara orientasi dan posisi yang diinginkan dengan orientasi dan posisi keluaran JST dapat dilihat pada Tabel 4.14.

Tabel 4.13 Posisi dan Orientasi Keluaran dengan Ketelitian 0.0001

No.	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	5.158	-19.927	0.257	0.028	0.140
2	14.145	15.375	0.335	-0.026	0.216
3	36.535	42.100	0.342	-0.123	0.334
4	24.863	30.309	0.357	-0.059	0.267
5	29.036	-2.453	0.300	-0.072	0.159

Tabel 4.14 Error Posisi dan Orientasi dengan Ketelitian 0.0001

No.	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	8.21e-05	-2.65e-05	6.76e-05	7.60e-06	6.31e-05
2	-9.64e-05	-5.05e-05	-2.11e-06	4.52e-07	8.06e-06
3	4.84e-05	4.76e-05	2.28e-06	-6.35e-07	-5.62e-06
4	-9.04e-06	7.84e-05	-7.20e-07	2.03e-08	3.95e-06
5	3.15e-05	-2.70e-05	6.66e-06	-1.48e-06	1.91e-05
RMSE	6.24e-05	4.98e-05	3.04e05	3.48e-06	2.99e-05

Iterasi rata-rata yang diperlukan dalam memproses data adalah 28036 dengan waktu selama 868 detik. Pengujian yang ketiga adalah pengujian dengan ketelitian 0.00005. Pengujian ini dilakukan dengan menggunakan *learning rate* 0.00025 dan 50 *neuron* lapisan *hidden*. Data posisi dan orientasi keluaran dari JST dapat dilihat pada Tabel 4.15 dan data *error* antara orientasi dan posisi yang diinginkan dengan orientasi dan posisi keluaran JST dapat dilihat pada Tabel 4.16.

Tabel 4.15 Posisi dan Orientasi Keluaran dengan Ketelitian 0.00005

No.	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	5.158	-19.927	0.257	0.028	0.140
2	14.145	15.375	0.335	-0.026	0.216
3	36.535	42.100	0.342	-0.123	0.334
4	24.863	30.309	0.357	-0.059	0.267
5	29.036	-2.453	0.300	-0.072	0.159

Tabel 4.16 Error Posisi dan Orientasi dengan Ketelitian 0.00005

No.	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	2.55e-05	2.94e-05	2.35e-06	1.84e-07	4.82e-05
2	-3.13e-05	-2.73e-05	7.38e-07	2.09e-07	-1.06e-05
3	2.12e-05	1.42e-05	1.14e-06	-2.82e-07	-3.28e-06
4	-2.70e-05	-4.99e-05	1.81e-06	-4.99e-07	3.34e-06
5	1.05e-05	2.68e-06	1.32e-06	-4.97e-07	2.99e-05
RMSE	2.42e-05	2.93e-05	1.57e-06	3.61e-07	2.59e-05

Iterasi rata-rata yang diperlukan dalam memproses data adalah 43162 dengan waktu selama 2227 detik. Berdasarkan pengujian ketelitian yang telah dilakukan, dapat dilihat bahwa besarnya nilai ketelitian berpengaruh pada jalannya program kinematika balik dengan JST. Besarnya nilai ketelitian berpengaruh pada jumlah iterasi dan lamanya iterasi berlangsung, dimana semakin banyak besar nilai ketelitian maka rata-rata iterasi akan menjadi lebih sedikit dan waktu yang diperlukan untuk melakukan iterasi tersebut juga akan semakin singkat.

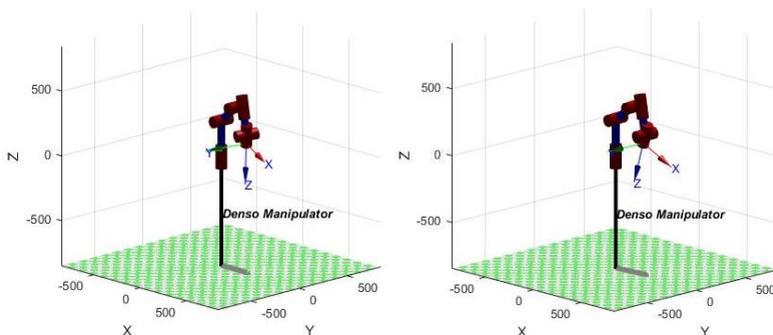
4.3 Plot Hasil Data Pengujian

Setelah dilakukan pengujian pada kinematika balik dengan JST dengan menggunakan 5 data uji, maka dilakukan plot masing-masing data pengujian tersebut, dimana pada plot hasil data uji ini akan diberikan gambar bentuk manipulator robot denso dari titik berangkat dan gambar bentuk manipulator robot pada masing-masing titik uji, serta titik-titik berpindahan manipulator robot dari titik berangkat menuju masing-masing titik uji. Titik berangkat manipulator robot adalah dari orientasi [*roll*, *pitch*, *yaw*] = [19.928 -8.820 -173.201] dan posisi [*x,y,z*] = [0.260 0.008 0.152], dengan besar masing-masing sendi adalah [$\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$] = [180 140 30 20 20 0].

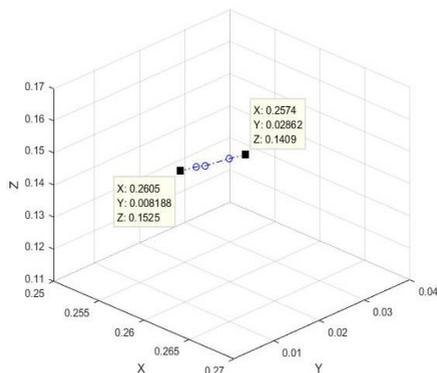
4.3.1 Plot Hasil Data Pengujian 1

Pada data pengujian pertama manipulator robot denso bergerak dari titik berangkat yaitu dengan orientasi [*roll*, *pitch*, *yaw*] = [19.928 -8.820 -173.201] dan posisi [*x,y,z*] = [0.260 0.008 0.152], dengan besar masing-masing sendi [$\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$] = [180 140 30 20 20 0], menuju orientasi [*roll* *pitch*] = [5.185 -19.927] dan posisi [*x,y,z*] = [0.257 0.028 0.140]. Gambar 4.7 sebelah kiri menunjukkan posisi dan orientasi awal

robot, dan Gambar 4.7 sebelah kanan menunjukkan posisi dan orientasi tujuan robot. Gambar 4.8 menunjukkan titik-titik perpindahan robot dari posisi mula-mula ke posisi tujuan.



Gambar 4.7 Bentuk Manipulator Robot Denso Pengujian 1

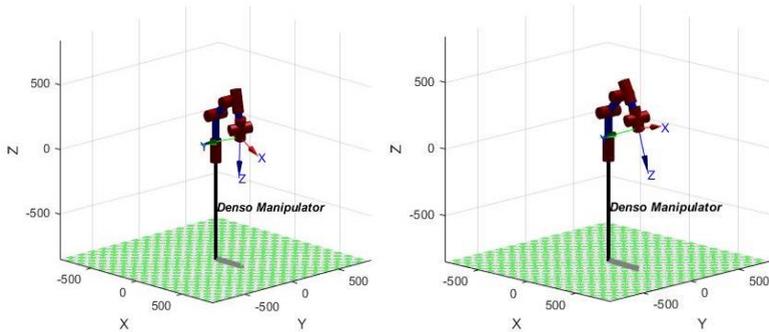


Gambar 4.8 Titik Perpindahan dari Posisi Awal ke Titik Tujuan 1

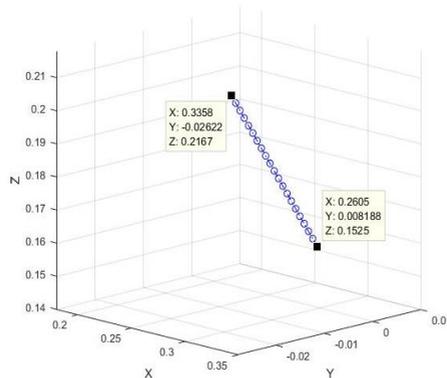
4.3.2 Plot Hasil Data Pengujian 2

Data pengujian kedua adalah berawal dari titik berangkat manipulator robot dengan orientasi $[roll, pitch, yaw] = [19.928 -8.820 -173.201]$ dan posisi $[x,y,z] = [0.260 0.008 0.152]$, dengan besar masing-masing sendi $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6] = [180 140 30 20 20 0]$, menuju orientasi $[roll pitch] = [14.145 15.375]$ dan posisi $[x,y,z] = [0.335 -0.026$

0.216]. Gambar 4.9 sebelah kiri menunjukkan posisi dan orientasi awal robot, dan Gambar 4.9 sebelah kanan menunjukkan posisi dan orientasi tujuan robot. Gambar 4.10 menunjukkan titik-titik perpindahan robot dari posisi mula-mula ke posisi tujuan.



Gambar 4.9 Bentuk Manipulator Robot Denso Pengujian 2

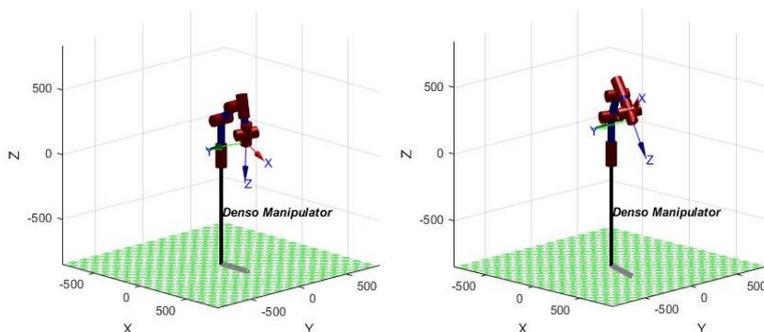


Gambar 4.10 Titik Perpindahan dari Posisi Awal ke Titik Tujuan 2

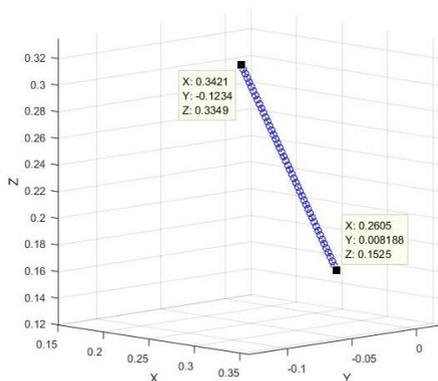
4.3.3 Plot Hasil Data Pengujian 3

Data pengujian ketiga adalah berawal dari titik berangkat manipulator robot dengan orientasi $[roll, pitch, yaw] = [19.928 \ -8.820 \ -173.201]$ dan posisi $[x,y,z] = [0.260 \ 0.008 \ 0.152]$, dengan besar masing-masing sendi $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6] = [180 \ 140 \ 30 \ 20 \ 20 \ 0]$, menuju orientasi $[roll \ pitch] = [36.535 \ 42.100]$ dan posisi $[x,y,z] = [0.342 \ -0.123$

0.334]. Gambar 4.11 sebelah kiri menunjukkan posisi dan orientasi awal robot, dan Gambar 4.11 sebelah kanan menunjukkan posisi dan orientasi tujuan robot. Gambar 4.12 menunjukkan titik-titik perpindahan robot dari posisi mula-mula ke posisi tujuan.



Gambar 4.11 Bentuk Manipulator Robot Denso Pengujian 3

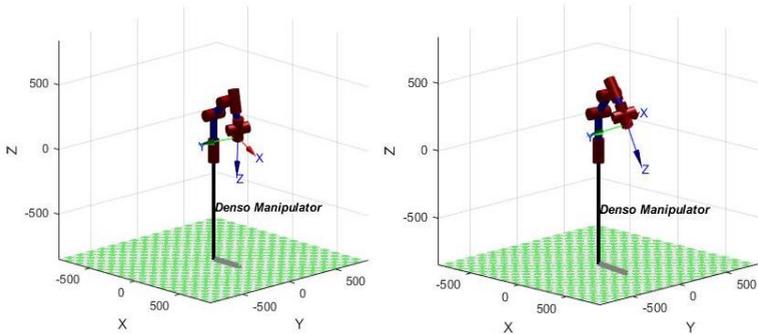


Gambar 4.12 Titik Perpindahan dari Posisi Awal ke Titik Tujuan 3

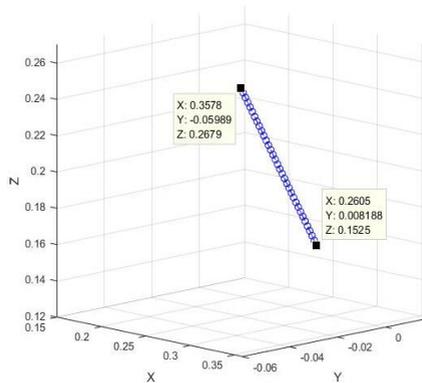
4.3.4 Plot Hasil Data Pengujian 4

Data pengujian keempat adalah berawal dari titik berangkat manipulator robot dengan orientasi $[roll, pitch, yaw] = [19.928 -8.820 -173.201]$ dan posisi $[x,y,z] = [0.260 0.008 0.152]$, dengan besar masing-masing sendi $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6] = [180 140 30 20 20 0]$, menuju orientasi $[roll pitch] = [24.863 30.309]$ dan posisi $[x,y,z] = [0.357 -0.059$

0.267]. Gambar 4.13 sebelah kiri menunjukkan posisi dan orientasi awal robot, dan Gambar 4.13 sebelah kanan menunjukkan posisi dan orientasi tujuan robot. Gambar 4.14 menunjukkan titik-titik perpindahan robot dari posisi mula-mula ke posisi tujuan.



Gambar 4.13 Bentuk Manipulator Robot Denso Pengujian 4

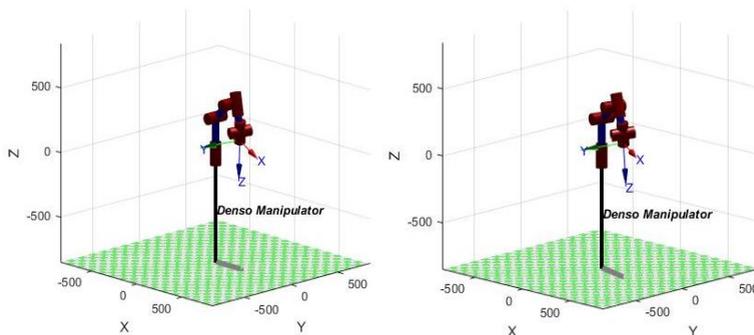


Gambar 4.14 Titik Perpindahan dari Posisi Awal ke Titik Tujuan 4

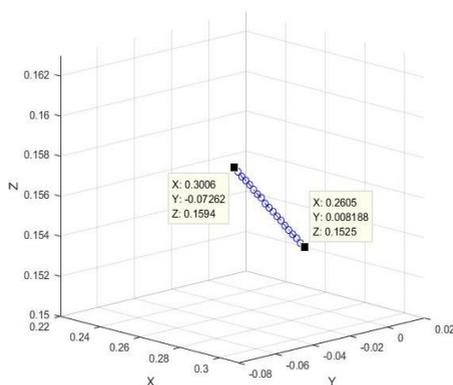
4.3.5 Plot Hasil Data Pengujian 5

Data pengujian kelima adalah berawal dari titik berangkat manipulator robot dengan orientasi $[roll, pitch, yaw] = [19.928 -8.820 -173.201]$ dan posisi $[x,y,z] = [0.260 0.008 0.152]$, dengan besar masing-masing sendi $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6] = [180 140 30 20 20 0]$, menuju orientasi $[roll pitch] = [29.036 -2.453]$ dan posisi $[x,y,z] = [0.300 -0.072$

0.159]. Gambar 4.15 sebelah kiri menunjukkan posisi dan orientasi awal robot, dan Gambar 4.15 sebelah kanan menunjukkan posisi dan orientasi tujuan robot. Gambar 4.16 menunjukkan titik-titik perpindahan robot dari posisi mula-mula ke posisi tujuan.



Gambar 4.15 Bentuk Manipulator Robot Denso Pengujian 5



Gambar 4.16 Titik Perpindahan dari Posisi Awal ke Titik Tujuan 5

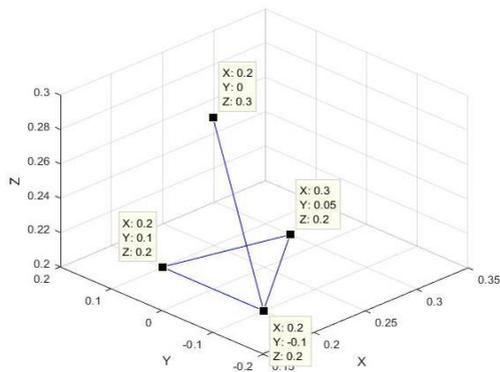
4.4 Kinematika Balik dengan JST untuk Pola Segitiga

Setelah dilakukan pengujian pada pengaruh jumlah *neuron* lapisan *hidden* dan besarnya nilai ketelitian terhadap masing-masing data uji yang telah didapat melalui persamaan kinematika maju, maka selanjutnya akan dilakukan pengujian agar *end-effector* dari manipulator robot denso membuat suatu pola segitiga dengan masukan orientasi *roll* dan *pitch*

serta posisi x,y, dan z. Dalam membuat pola segitiga ini orientasi *roll* dan *pitch* dibuat tetap yaitu [0 0], tetapi titik x,y,dan z nya yang berubah-ubah. Manipulator robot denso akan bergerak dari orientasi [*roll pitch*] = [0 0], dan posisi [x,y,z] = [0.2 0 0.3], dengan sudut [$\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$] = [180 93.23 48.87 0 37.88 0] menuju titik-titik tujuan yang membentuk pola segitiga.

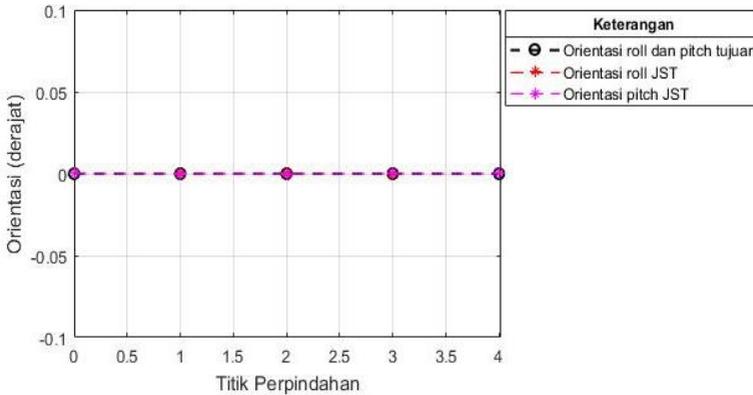
4.4.1 Pola Segitiga Datar

Pada pengujian yang pertama, akan dibuat pola segitiga dalam bidang datar. Titik-titik yang akan menjadi tujuan manipulator robot denso dalam membentuk pola segitiga pada bidang datar dapat dilihat pada Gambar 4.17. Pengujian kinematika balik dengan JST dalam membentuk pola segitiga datar ini dilakukan dengan menggunakan ketelitian sebesar 0.00005, *learning rate* sebesar 0.00025, dan 50 *neuron hidden*.

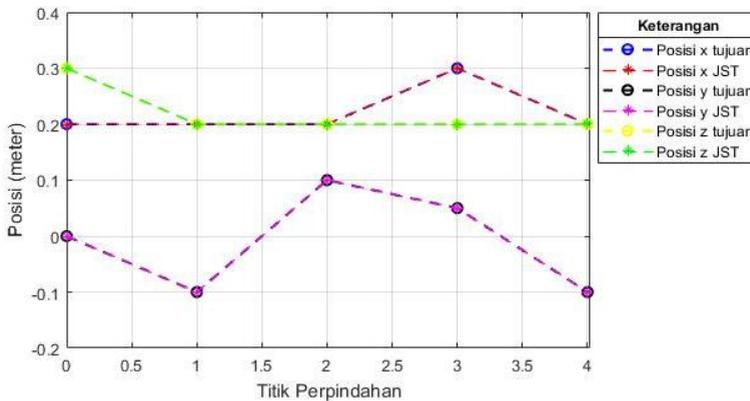


Gambar 4.17 Titik –Titik Pola Segitiga Bidang Datar

Grafik perbandingan antara orientasi dan posisi tujuan yang ditentukan untuk membentuk pola segitiga dalam bidang datar dengan orientasi dan posisi keluaran JST dapat dilihat pada Gambar 4.18 dan Gambar 4.19.

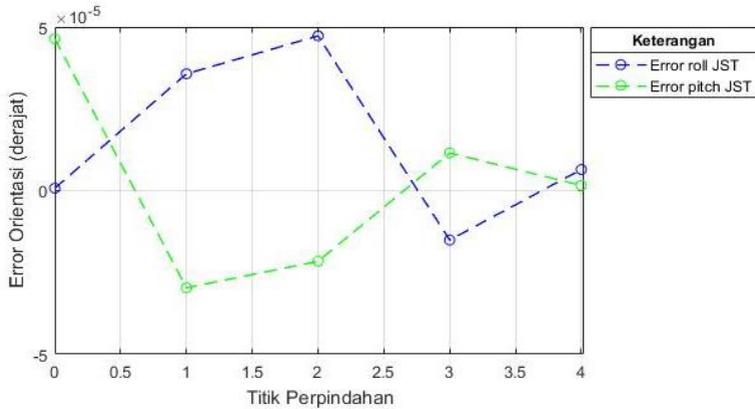


Gambar 4.18 Grafik Perbandingan Orientasi Tujuan dan JST Membentuk Pola Segitiga Bidang Datar

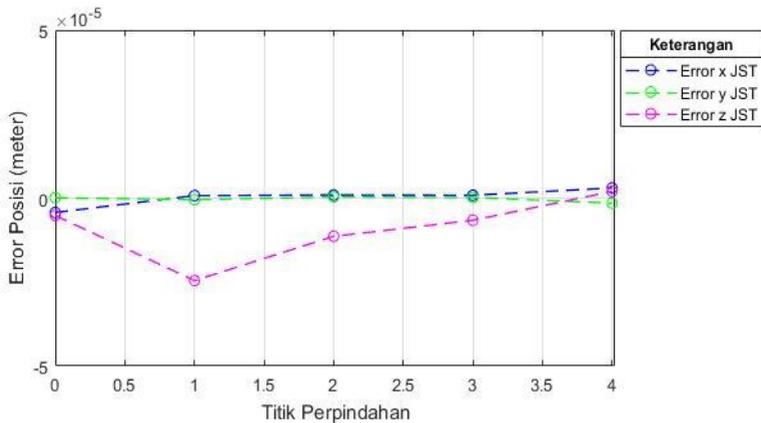


Gambar 4.19 Grafik Perbandingan Posisi Tujuan dan JST Membentuk Pola Segitiga Bidang Datar

Dapat dilihat orientasi dan posisi keluaran JST mengikuti orientasi dan posisi tujuan untuk membentuk pola segitiga dalam bidang datar. Grafik *error* antara posisi dan orientasi keluaran JST dengan posisi dan orientasi tujuan dapat dilihat pada Gambar 4.20 dan Gambar 4.21.

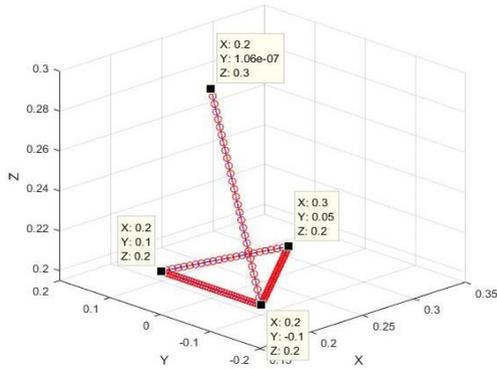


Gambar 4.20 Grafik *Error Orientasi Tujuan dan JST Membentuk Pola Segitiga Bidang Datar*



Gambar 4.21 Grafik *Error Posisi Tujuan dan JST Membentuk Pola Segitiga Bidang Datar*

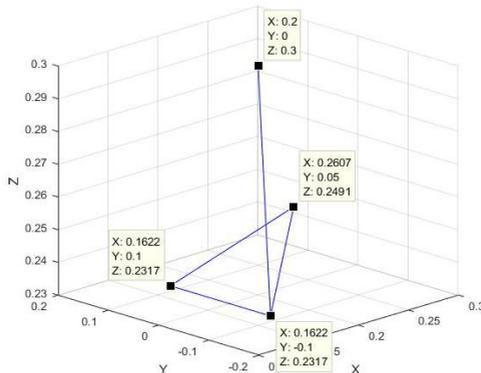
Hasil plot segitiga dalam bidang datar keluaran dari program kinematika balik dengan JST dapat dilihat pada Gambar 4.22 dimana titik-titik tersebut merupakan titik-titik pergerakan dari manipulator robot denso untuk membentuk pola segitiga pada bidang datar.



Gambar 4.22 Titik –Titik Pola Segitiga Keluaran JST

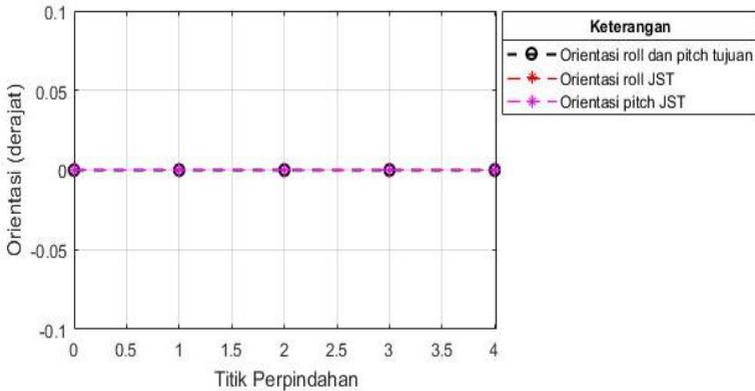
4.4.2 Pola Segitiga Miring

Pada pengujian yang kedua, akan dibuat pola segitiga dalam bidang miring ,dengan kemiringan 10° . Titik yang akan menjadi tujuan manipulator robot denso untuk membentuk pola segitiga dalam bidang miring dapat dilihat pada Gambar 4.23. Pengujian kinematika balik dengan JST dilakukan dengan menggunakan ketelitian sebesar 0.00005, *learning rate* sebesar 0.00025, dan 50 *neuron* lapisan *hidden*.

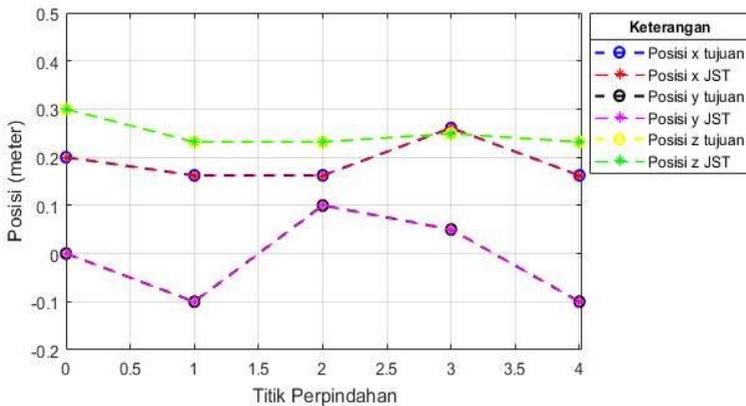


Gambar 4.23 Titik –Titik Pola Segitiga Bidang Miring

Grafik perbandingan antara orientasi dan posisi tujuan yang ditentukan untuk membentuk pola segitiga dalam bidang miring dengan orientasi dan posisi keluaran JST dapat dilihat pada Gambar 4.24 dan Gambar 4.25.

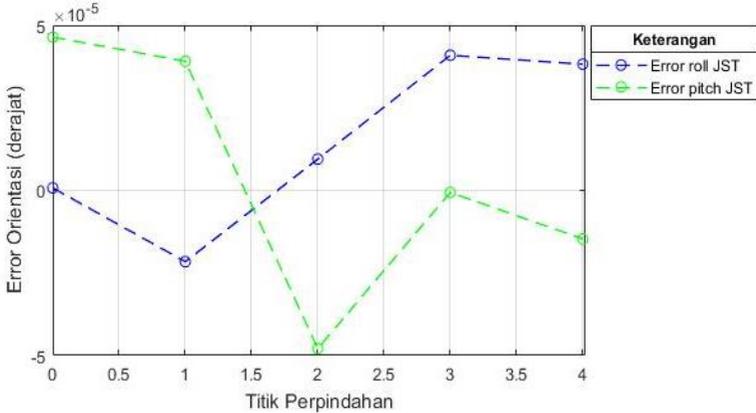


Gambar 4.24 Grafik Perbandingan Orientasi Tujuan dan JST Membentuk Pola Segitiga Bidang Miring

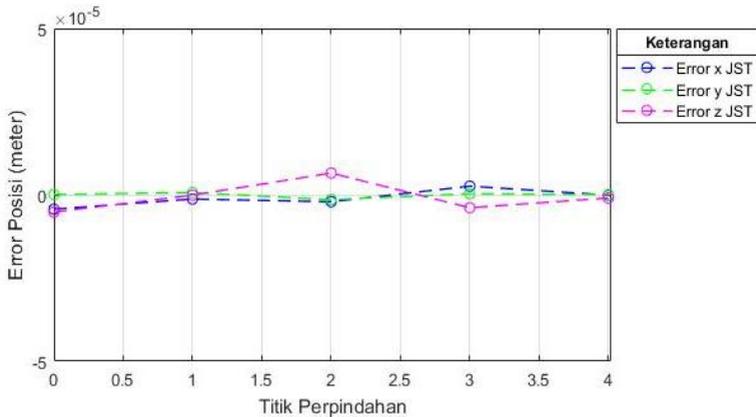


Gambar 4.25 Grafik Perbandingan Posisi Tujuan dan JST Membentuk Pola Segitiga Bidang Miring

Dapat dilihat orientasi dan posisi keluaran JST mengikuti orientasi dan posisi tujuan untuk membentuk pola segitiga dalam bidang miring. Grafik *error* antara posisi dan orientasi keluaran JST dengan posisi dan orientasi tujuan dapat dilihat pada Gambar 4.26 dan Gambar 4.27.

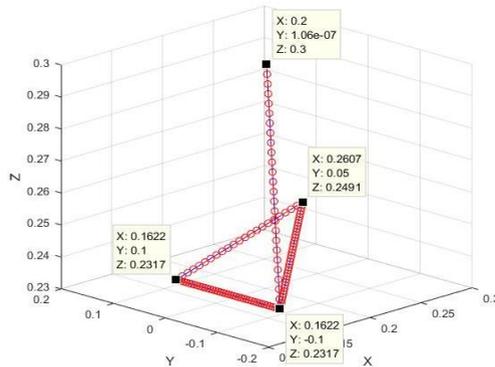


Gambar 4.26 Grafik *Error Orientasi Tujuan dan JST Membentuk Pola Segitiga Bidang Miring*



Gambar 4.27 Grafik *Error Perbandingan Orientasi Tujuan dan JST Membentuk Pola Segitiga Bidang Miring*

Hasil plot segitiga pada bidang miring keluaran dari program kinematika balik dengan JST dapat dilihat pada Gambar 4.28 dimana titik-titik tersebut merupakan titik-titik pergerakan dari manipulator robot denso untuk membentuk pola segitiga pada bidang miring.



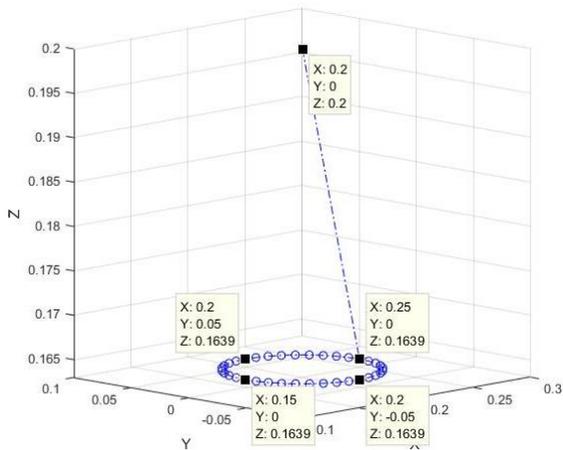
Gambar 4.28 Titik Pola Segitiga Miring Keluaran JST

4.5 Kinematika Balik dengan JST untuk Pola Lingkaran

Setelah dilakukan pengujian terhadap masing-masing data uji dan pola segitiga dalam bidang datar dan dalam bidang miring, maka dilakukan pengujian untuk membuat suatu pola lingkaran dengan masukan orientasi *roll* dan *pitch* serta posisi x, y , dan z . Dalam membuat pola lingkaran ini orientasi *roll* dan *pitch* dibuat tetap yaitu $[0 \ 0]$, tetapi titik x, y , dan z nya yang berubah-ubah. Pola lingkaran yang dibuat akan ada dua jenis, yaitu lingkaran dalam bidang datar dan juga lingkaran dalam bidang miring, dengan sudut kemiringan tertentu.

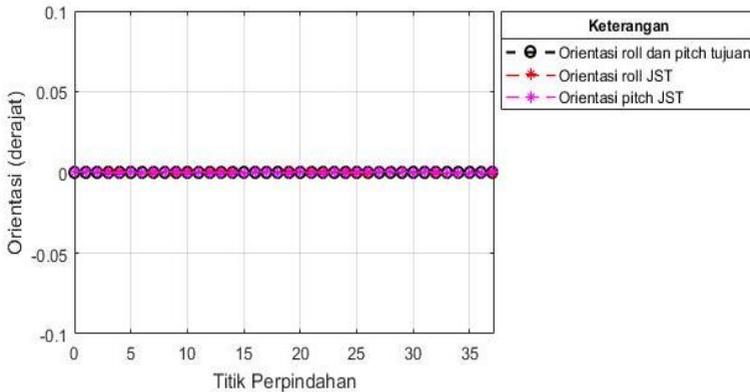
4.5.1 Pola Lingkaran Datar

Dalam membentuk pola lingkaran ini manipulator robot denso akan bergerak dari orientasi $[roll \ pitch] = [0 \ 0]$, dan posisi $[x, y, z] = [0.2 \ 0 \ 0.2]$, dengan sudut $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6] = [180 \ 117.06 \ 54.65 \ 0 \ 8.28]$. Pada pengujian ini, akan dibuat lingkaran dalam bidang datar dengan 36 titik yang dapat dilihat pada Gambar 4.29. Pengujian kinematika balik dengan JST dilakukan dengan menggunakan ketelitian sebesar 0.00005, *learning rate* sebesar 0.00025, dan 50 *neuron* lapisan *hidden*.

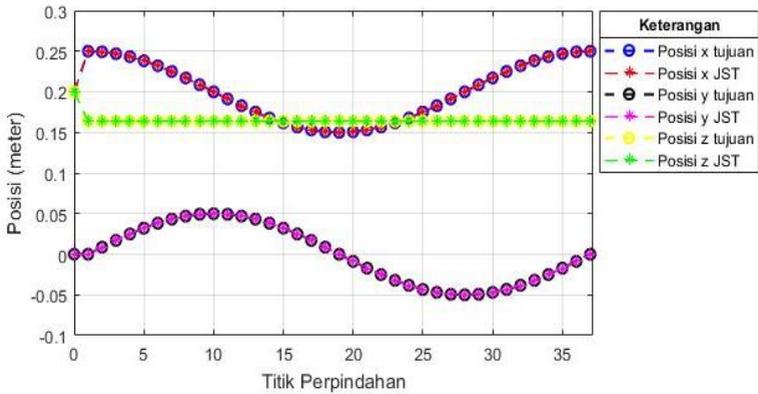


Gambar 4.29 Titik –Titik Pola Lingkaran Bidang Datar

Grafik perbandingan antara orientasi dan posisi tujuan yang ditentukan untuk membentuk pola segitiga dalam bidang datar dengan orientasi dan posisi keluaran JST dapat dilihat pada Gambar 4.30 dan Gambar 4.31.

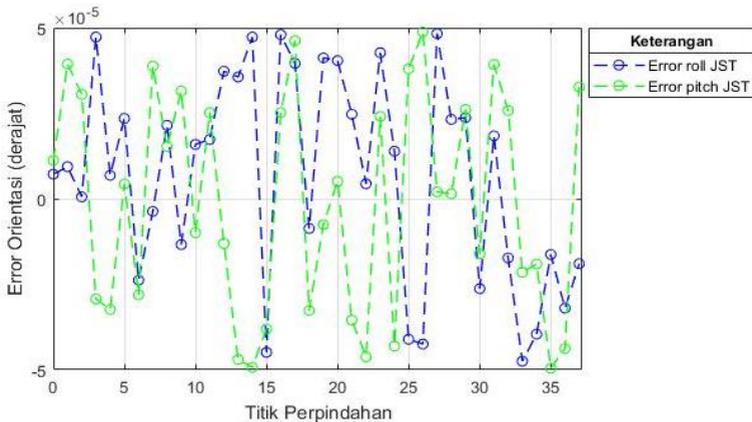


Gambar 4.30 Grafik Perbandingan Orientasi Tujuan dan JST Membentuk Pola Lingkaran Bidang Datar

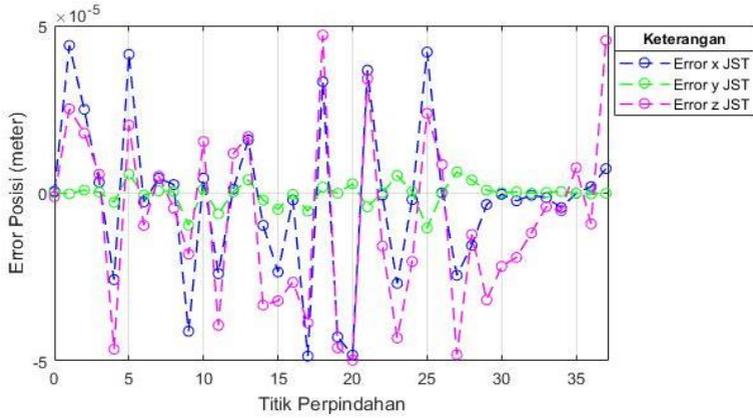


Gambar 4.31 Grafik Perbandingan Posisi Tujuan dan JST Membentuk Pola Lingkaran Bidang Datar

Dapat dilihat orientasi dan posisi keluaran JST mengikuti orientasi dan posisi tujuan untuk membentuk pola lingkaran dalam bidang datar. Grafik *error* antara posisi dan orientasi keluaran JST dengan posisi dan orientasi tujuan dapat dilihat pada Gambar 4.32 dan Gambar 4.33.

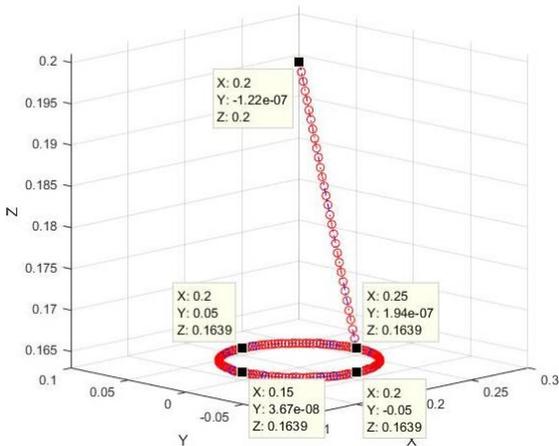


Gambar 4.32 Grafik *Error* Orientasi Tujuan dan JST Membentuk Pola Lingkaran Bidang Datar



Gambar 4.33 Grafik *Error* Posisi Tujuan dan JST Membentuk Pola Lingkaran Bidang Datar

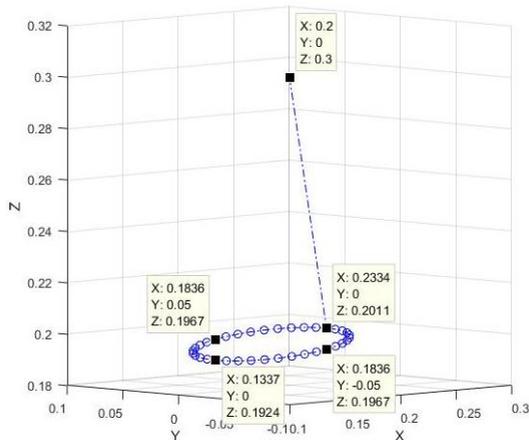
Hasil plot lingkaran pada bidang datar keluaran dari program kinematika balik dengan JST dapat dilihat pada Gambar 4.34, dimana titik-titik tersebut merupakan titik-titik pergerakan dari manipulator robot denso untuk membentuk pola lingkaran pada bidang datar.



Gambar 4.34 Titik –Titik Pola Lingkaran Keluaran JST

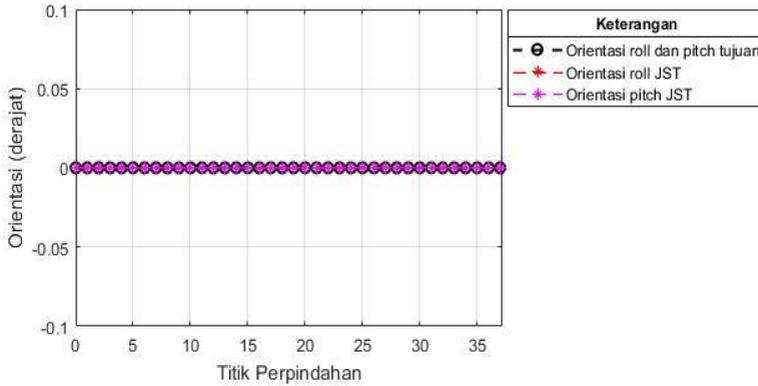
4.5.2 Pola Lingkaran Miring

Pada pengujian ini, akan dibuat lingkaran dalam bidang miring dengan kemiringan sebesar 5° . Dalam membentuk pola lingkaran ini manipulator robot denso akan bergerak dari orientasi $[\text{roll pitch}] = [0\ 0]$, dan posisi $[x,y,z] = [0.2\ 0\ 0.3]$, dengan sudut $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6] = [180\ 93.23\ 48.87\ 0\ 37.89\ 0]$. Pola lingkaran dalam bidang miring disusun dari 36 titik yang dapat dilihat pada Gambar 4.35, dimana titik-titik tersebut akan menjadi tujuan dari manipulator robot denso. Pengujian kinematika balik dengan JST dilakukan dengan menggunakan ketelitian sebesar 0.00005, *learning rate* sebesar 0.00025, dan 50 *neuron* lapisan *hidden*.

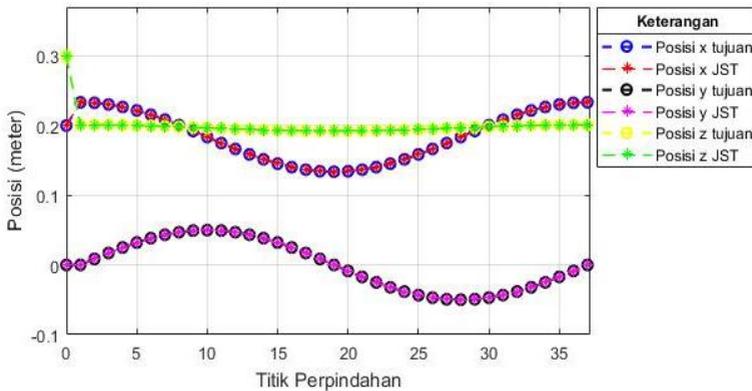


Gambar 4.35 Titik –Titik Pola Lingkaran Bidang Miring

Grafik perbandingan antara orientasi dan posisi tujuan yang ditentukan untuk membentuk pola segitiga dalam bidang datar dengan orientasi dan posisi keluaran JST dapat dilihat pada Gambar 4.36 dan Gambar 4.37. Dapat dilihat pada Gambar 4.36 dan Gambar 4.37 orientasi dan posisi keluaran JST mengikuti orientasi dan posisi tujuan untuk membentuk pola lingkaran dalam bidang datar.

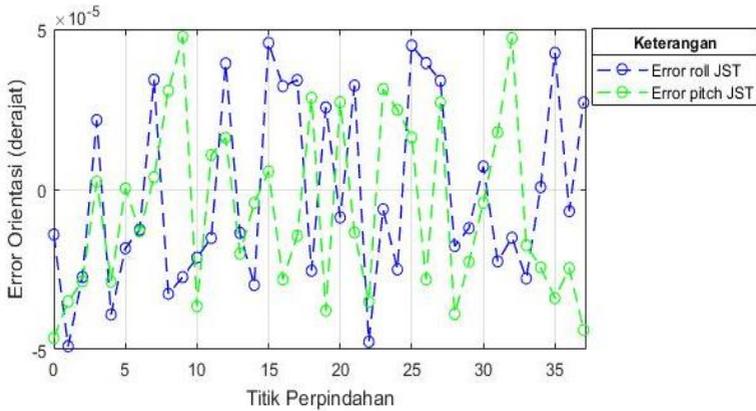


Gambar 4.36 Grafik Perbandingan Orientasi Tujuan dan JST Membentuk Pola Lingkaran Bidang Miring

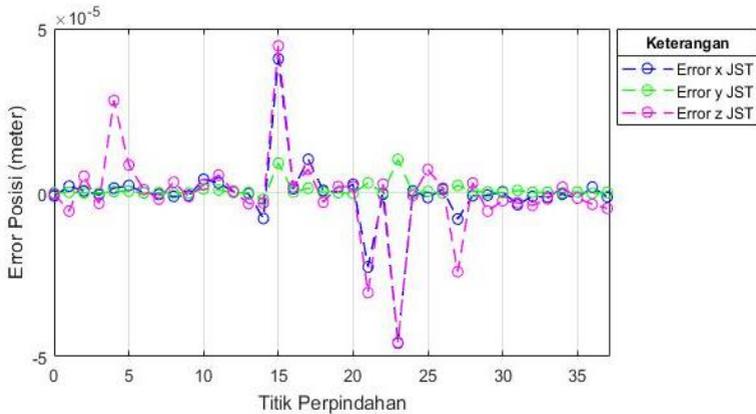


Gambar 4.37 Grafik Perbandingan Posisi Tujuan dan JST Membentuk Pola Lingkaran Bidang Miring

Grafik *error* antara posisi dan orientasi keluaran JST dengan posisi dan orientasi tujuan dapat dilihat pada Gambar 4.38 dan Gambar 4.39.

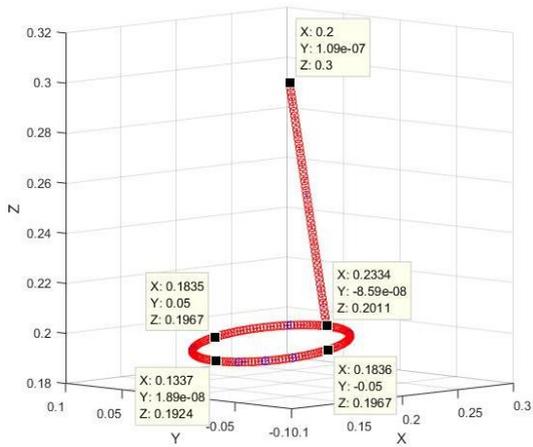


Gambar 4.38 Grafik *Error Orientasi Tujuan dan JST Membentuk Pola Lingkaran Bidang Datar*



Gambar 4.39 Grafik *Error Posisi Tujuan dan JST Membentuk Pola Lingkaran Bidang Datar*

Hasil plot lingkaran dalam bidang miring keluaran dari program kinematika balik dengan JST dapat dilihat pada Gambar 4.40 dimana titik-titik tersebut merupakan titik-titik pergerakan dari manipulator robot denso untuk membentuk pola segitiga pada bidang datar.



Gambar 4.40 Titik –Titik Pola Lingkaran Keluaran JST

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan simulasi dan analisa data pada bab sebelumnya, terdapat beberapa hal yang dapat disimpulkan pada penelitian Tugas Akhir ini, yaitu:

1. JST telah dapat di aplikasikan untuk menyelesaikan kinematika balik manipulator robot denso 6-DoF.
2. Melalui simulasi manipulator robot denso telah dapat menemukan posisi titik untuk pola segitiga dan lingkaran baik pada bidang datar maupun pada bidang miring menggunakan kinematika balik dengan JST, dengan rata-rata *error* posisi dibawah 0.00005 meter dan *error* orientasi dibawah 0.00005 derajat.
3. Semakin besar jumlah *neuron* lapisan *hidden* yang digunakan maka iterasi perhitungan agar hasil keluaran JST sesuai dengan ketelitian yang ditentukan akan semakin cepat, dibuktikan dengan 5 *neuron*, program memiliki iterasi rata-rata sebesar 166084 dengan waktu selama 5475 detik, sedangkan dengan 50 *neuron*, program memiliki iterasi rata-rata sebesar 43162 dengan waktu selama 2227 detik.
4. Semakin tinggi ketelitian maka iterasi perhitungan agar hasil keluaran JST sesuai dengan ketelitian yang ditentukan akan semakin lama, dibuktikan dengan ketelitian sebesar 0.0005, program memiliki iterasi rata-rata sebesar 14310 dengan waktu selama 527 detik, sedangkan dengan ketelitian sebesar 0.00005, program memiliki iterasi rata-rata sebesar 43162 dengan waktu selama 2227 detik.

5.2 Saran

Sebagai pengembangan penelitian Tugas Akhir ini, ada beberapa hal yang dapat digunakan untuk perkembangan selanjutnya, yaitu:

1. Dapat dilakukan implementasi secara langsung pada manipulator robot denso 6 DoF.
2. Disarankan untuk menambahkan perhitungan kecepatan pada pergerakan robot.

-----*Halaman ini sengaja dikosongkan*-----

DAFTAR PUSTAKA

- [1] Spong, M. W. & Vidyasagar, M., 1989, *Robot Dynamics and control*. 1st ed. Canada: John Wiley & Sons, Inc.
- [2] Craig, J. J., 2005, *Introduction to Robotics Mechanics and Control* 3rd ed. United States of America: Pearson Education, Inc.
- [3], n.d. *Quanser 6-Axis Articulated Robot*. Canada: Quanser, Inc.
- [4] Ratna Wati, D. A., 2011, *Sistem Kendali Cerdas*. Yogyakarta: GRAHA ILMU.
- [5] Arlean, T. W., 2017. Kinematika Balik Manipulator Robot Denso Dengan Metode *Neural Network*. *Tugas Akhir*. Teknik Elektro. Surabaya: Institut Teknologi Sepuluh Nopember.
- [6] Rangkuti, R. P., 2017. Kinematika Balik Menggunakan *Neuro-Fuzzy* Pada Manipulator Robot Denso. *Tugas Akhir*. Teknik Elektro. Surabaya: Institut Teknologi Sepuluh Nopember.
- [7] Duka, A. V., 2013, Neural Network Based Inverse Kinematics Solution for Trajectory Tracking of a Robotic Arm. *Procedia Technology*. Romania. 12 (2014) 20 – 27.
- [8] Prasetya, I. E., 2015. Inverse Kinematics dengan Solusi Closed Form Pada Robot Denso Manipulator. *Tugas Akhir*. Teknik Elektro. Surabaya: Institut Teknologi Sepuluh Nopember.

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN

1. Program *Forward Kinematics*

```
function [N] = porward11(mas)
global a2 d1 d4 d6 a3
global c1 c2 c3 c4 c5 c6 s1 s2 s3 s4 s5 s6
a2= 0.21;d1= 0.28;d4= 0.21;d6= 0.070;a3=0.075;
t1=mas(1);
t2=mas(2);
t3=mas(3);
t4=mas(4);
t5=mas(5);
t6=mas(6);
%inisialisasi
c1=cosd(t1);c2=cosd(t2);c3=cosd(t3);c4=cosd(t4);c5=cosd(t5);c
6=cosd(t6);
s1=sind(t1);s2=sind(t2);s3=sind(t3);s4=sind(t4);s5=sind(t5);s6
=sind(t6);
%inisialisasi matrix
nx=c6*(c5*((c1*c2*c3-c1*s2*s3)*c4-s1*s4)+s5*(-c1*c2*s3-
c1*s2*c3))+s6*(-(c1*c2*c3-c1*s2*s3)*s4+s1*c4);
ny=c6*((c5*(((s1*c2*c3)-(s1*s2*s3))*c4)+(c1*s4)))+(s5*(-
(s1*c2*s3)-(s1*s2*c3)))+s6*(-(((s1*c2*c3)-(s1*s2*s3))*s4)-
(c1*c4));
nz=c6*((c5*((s2*c3)+(c2*s3))*c4)+(s5*(-
(s2*s3)+(c2*c3)))+s6*(-(((s2*c3)+(c2*s3))*s4));
sx=-s6*((c5*(((c1*c2*c3)-(c1*s2*s3))*c4)-(s1*s4)))+(s5*(-
(c1*c2*s3)-(c1*s2*c3)))+c6*(-(((c1*c2*c3)-
(c1*s2*s3))*s4)+(s1*c4));
sy=-s6*((c5*(((s1*c2*c3)-(s1*s2*s3))*c4)+(c1*s4)))+(s5*(-
(s1*c2*s3)-(s1*s2*c3)))+c6*(-(((s1*c2*c3)-(s1*s2*s3))*s4)-
(c1*c4));
sz=-s6*((c5*((s2*c3)+(c2*s3))*c4)+(s5*(-
(s2*s3)+(c2*c3)))+c6*(-(((s2*c3)+(c2*s3))*s4));
ax=(-s5*(((c1*c2*c3)-(c1*s2*s3))*c4)-(s1*s4)))+(c5*(-
(c1*c2*s3)-(c1*s2*c3));
ay=(-s5*(((s1*c2*c3)-(s1*s2*s3))*c4)+(c1*s4)))+(c5*(-
(s1*c2*s3)-(s1*s2*c3));
```

```

az=(-s5*((s2*c3)+(c2*s3))*c4)+(c5*(-(s2*s3)+(c2*c3)));
px=d6*(-s5*(((c1*c2*c3)-(c1*s2*s3))*c4)-(s1*s4)))+(c5*(-
(c1*c2*s3)-(c1*s2*c3)))+( -(c1*c2*s3)-
(c1*s2*c3))*d4)+(c1*c2*c3*a3)-(c1*s2*s3*a3)+(c1*c2*a2);
py=d6*(-s5*(((s1*c2*c3)-(s1*s2*s3))*c4)+(c1*s4)))+(c5*(-
(s1*c2*s3)-(s1*s2*c3)))+( -(s1*c2*s3)-
(s1*s2*c3))*d4)+(s1*c2*c3*a3)-(s1*s2*s3*a3)+(s1*c2*a2);
pz=d6*(-s5*((s2*c3)+(c2*s3))*c4)+(c5*(-
(s2*s3)+(c2*c3)))+( ((s2*s3)+(c2*c3))*d4)+(s2*c3*a3)+(c2*
s3*a3)+(s2*a2)+d1;
h=[nx sx ax px
ny sy ay py
nz sz az pz
0 0 0 1];
pitch=asind(h(1,3));
roll=asind(h(1,2)/-cosd(pitch));
px=h(1,4);
py=h(2,4);
pz=h(3,4);
N=[px py pz roll pitch];

```

2. Program Inverse Kinematics

```

close all;
clear
clc;
%INIALISASI
tha=[180 140 30 20 20 0];
t1= input('Masukkan Roll = ');
t2= input('Masukkan Pitch = ');
t4= input('Masukkan X = ');
t5= input('Masukkan Y = ');
t6= input('Masukkan Z = ');
%inisialisasi
Mth=[porward11(tha
t4 t5 t6 t1 t2)];
NH = 5; %jumlah neuron hidden
NO = 5; %jumlah neuron output
a=0;
for ktk=1:length(Mth(:,1))-1

```

```

dxyz=Mth(ktk+1,:) -Mth(ktk, :);
dtot=sqrt(dxyz(1)^2+dxyz(2)^2+dxyz(3)^2);
d=0.005;
jdpoin=floor(dtot/d);
dp(1)=dxyz(1)/jdpoin; %dx
dp(2)=dxyz(2)/jdpoin; %dy
dp(3)=dxyz(3)/jdpoin; %dz
dp(4)=dxyz(4)/jdpoin; %dr
dp(5)=dxyz(5)/jdpoin; %dp
% Perhitungan bobot random
bv = 0*rand(5,NH); %bobot layer hidden
bw = 0*rand(NH,5); %bobot layer output
for p=1:jdpoin
    xyztar=zeros(p,5);
    xyztar(p,:) = Mth(ktk, :)+ p*[dp(1) dp(2)
        dp(3) dp(4) dp(5)];
    gradw(1)=10;
    gradw(2)=10;
    gradw(3)=10;
    gradw(4)=10;
    gradw(5)=10;
    I(p)=0;
    Y = zeros(size(xyztar(p,:)));
    while (abs(gradw(1))>0.00005 ||
        abs(gradw(2))>0.00005 ||
        abs(gradw(3))>0.00005 ||
        abs(gradw(4))>0.00005 ||
        abs(gradw(5))>0.00005)
        %Feedforward
        for j=1:NH
            sumxv(j)=0;
            for i=1:5 %jumlah neuron
                hidden
                sumxv(j) =
                    sumxv(j) + (dp(i) *bv(i, j));
            end
            z_in(j) = sumxv(j);
            Z(j)=1/(1+exp(-z_in(j)));
        end
    end
for k=1:NO

```

```

sumvw(k)=0;
for j=1:NH
    sumvw(k)= sumvw(k) +
                (Z(j)*bw(j,k));
end
y_in(k)=sumvw(k);
Y(p,k)=y_in(k);
end
thad= tha + [Y(p,1) Y(p,2) Y(p,3)
             Y(p,4) Y(p,5) 0];
if thad(1)>160
    thad=thad-[320 0 0 0];
elseif thad(1)<-160
    thad=thad+[320 0 0 0];
else
    thad=thad;
end
if thad(2)>180
    thad=thad-[0 180 0 0 0 0];
elseif thad(2)<0
    thad=thad+[0 180 0 0 0 0];
else
    thad=thad;
end
if thad(3)>40
    thad=thad-[0 0 40 0 0 0];
elseif thad(3)<0
    thad=thad+[0 0 40 0 0 0];
else
    thad=thad;
end
if thad(4)>40
    thad=thad-[0 0 0 40 0 0];
elseif thad(4)<0
    thad=thad+[0 0 0 40 0 0];
else
    thad=thad;
end
if thad(5)>40
    thad=thad-[0 0 0 0 40 0];

```

```

elseif thad(5)<0
    thad=thad+[0 0 0 0 40 0];
else
    thad=thad;
end
xyzn=porward11(thad);

sd1(p)=atan2(xyztar(p,2),xyztar(p,1))*180/pi;

sd2(p)=sqrt((xyztar(p,1)^2)+(xyztar(p,2)^2));
sd3(p)=xyztar(p,3);
sd4(p)=xyztar(p,4);
sd5(p)=xyztar(p,5);

sd1n(p)=atan2(xyzn(2),xyzn(1))*180/pi;

sd2n(p)=sqrt((xyzn(1)^2)+(xyzn(2)^2));
sd3n(p)=xyzn(3);

sd4n(p)=xyzn(4);
sd5n(p)=xyzn(5);
%Backpropagation
gradw(1)=sd1(p)-sd1n(p);
gradw(2)=sd2(p)-sd2n(p);
gradw(3)=sd3(p)-sd3n(p);
gradw(4)=sd4(p)-sd4n(p);
gradw(5)=sd5(p)-sd5n(p);
%Inialisasi update bobot
tnd2=-1;
if sign(gradw(2))>0
    tnd2=1;
end
tnd3=1;
if sign(gradw(3))>0
    tnd3=-1;

```

```

end
tnd4=-1;
if sign(gradw(4))>0
    tnd4=1;
end
tnd5=1;
if sign(gradw(5))>0
    tnd5=-1;
end
clc;
I(p)=I(p)+1;
I(p)
I;
[ thad(:,1:6) ]
[jdpoin p ktk]
ljpemw(1)=0.00025;
ljpemw(2)=0.00025;
ljpemw(3)=0.00025;
ljpemw(4)=0.00025;
ljpemw(5)=0.00025;
ljpemv=0.1;
if abs(gradw(1))<0.005

    ljpemw(1)=0.0000001+rand(1,1)*(0
        .0001-0.0000001);
end
if abs(gradw(2))<0.005

    ljpemw(2)=0.0000001+rand(1,1)*(0
        .0001-0.0000001);
end
if abs(gradw(3))<0.005

    ljpemw(3)=0.0000001+rand(1,1)*(0
        .0001-0.0000001);
end
if abs(gradw(4))<0.005

    ljpemw(4)=0.0000001+rand(1,1)*(0
        .0001-0.0000001);

```

```

end
if abs(gradw(5)) < 0.005

    ljpemw(5) = 0.0000001 + rand(1,1) *
        (0.0001 - 0.0000001);
end
if I(p) > 5000
    ljpemv = 0.000001;
end
if I(p) > 5000000
    break
end
%Hitung delta bobot w
for j = 1:NH

    delw(j,1) = ljpemw(1) * sign(gradw(1))
        * Z(j);
        delw(j,2) = ljpemw(2) * tnd2 * Z(j);
        delw(j,3) = ljpemw(3) * tnd3 * Z(j);
        delw(j,4) = ljpemw(4) * tnd4 * Z(j);
        delw(j,5) = ljpemw(5) * tnd5 * Z(j);
end
for j = 1:NH
    grad_in(j) = 0;
    for k = 1:NO

        grad_in(j) = grad_in(j) + (gradw(k) * bw
            (j,k));
        end
    end
    for j = 1:NH
        gradh(j) = grad_in(j) * Z(j) * (1 -
            Z(j));
    end
    %Hitung delta bobot v
    for i = 1:5
        for j = 1:NH

            delv(i,j) = ljpemv * gradh(j) * dp(i);
            end
        end
    end
end

```

```

end
%Bobot w
for j=1:NH
    for k=1:NO
        bw(j,k)=bw(j,k)+delw(j,k);
    end
end
%Bobot v
for i=1:5
    for j=1:NH
        bv(i,j)=bv(i,j)+delv(i,j);
    end
end
end
hasil(p,:)=thad
end
hasilakh(a+1:a+p,:)=hasil(1:p,:);
a=a+p;
end

```

3. Posisi dan Orientasi Target Segitiga

Titik	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	0	0	0.2	-0.1	0.2
2	0	0	0.2	0.1	0.2
3	0	0	0.3	0.05	0.2
4	0	0	0.2	-0.1	0.2

4. Posisi dan Orientasi Segitiga JST

Titik	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	3.5e-05	-2.9e-05	0.200001	-0.1	0.199975
2	4.3e-05	-2.1e-05	0.200001	0.1	0.199989
3	-1.5e-05	1.1e-05	0.300001	0.05	0.199993
4	6.4e-06	1.5e-06	0.200003	-0.1	0.200002

5. Error Posisi dan Orientasi Segitiga JST

Titik	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	3.5e-05	-2.9e-05	7.31e-07	-4.99e-07	-2.4e-05
2	4.7e-05	-2.1e-05	9.13e-07	3.75e-07	-1.1e-05
3	-1.5e-05	1.1e-05	8.21e-07	1.91e-07	-6.6e-06
4	6.4e-06	1.5e-06	3.03e-06	-1.52e-06	1.99e-06
RMSE	2.75e-05	1.7e-05	1.48e-06	7.42e-07	1.24e-05

6. Posisi dan Orientasi Target Segitiga Bidang Miring

Titik	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	0	0	0.16223	-0.1	0.23169
2	0	0	0.16223	0.1	0.23169
3	0	0	0.26071	0.05	0.24905
4	0	0	0.16223	-0.1	0.23169

7. Posisi dan Orientasi Segitiga Miring JST

Titik	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	-2.1e-05	3.9e-05	0.16223	-0.1	0.23169
2	9.4e-06	-4.7e-05	0.16223	0.100001	0.23168
3	4.1e-05	-6.2e-07	0.26071	0.05	0.24906
4	3.8e-05	-1.4e-05	0.16223	-0.1	0.23169

8. Error Posisi dan Orientasi Segitiga Miring

Titik	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	-2.1e-05	3.9e-05	1.26e-06	-6.83e-07	8.77e-08
2	9.4e-06	-4.7e-05	2.07e-06	1.44e-06	-6.5e-06
3	4.1e-05	-6.2e-07	-2.60e-06	-3.46e-07	3.87e-06
4	3.8e-05	-1.4e-05	9.23e-08	-6.61e-08	9.57e-07
RMSE	2.72e-05	2.8e-05	1.59e-06	7.32e-07	3.42e-06

9. Posisi dan Orientasi Target Lingkaran

Titik	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	0	0	0.25	0	0.1639
2	0	0	0.24924	0.008682	0.1639
3	0	0	0.246985	0.017101	0.1639
4	0	0	0.243301	0.025	0.1639
5	0	0	0.238302	0.032139	0.1639
6	0	0	0.232139	0.038302	0.1639
7	0	0	0.225	0.043301	0.1639
8	0	0	0.217101	0.046985	0.1639
9	0	0	0.208682	0.04924	0.1639
10	0	0	0.2	0.05	0.1639
11	0	0	0.191318	0.04924	0.1639
12	0	0	0.182899	0.046985	0.1639
13	0	0	0.175	0.043301	0.1639
14	0	0	0.167861	0.038302	0.1639
15	0	0	0.161698	0.032139	0.1639
16	0	0	0.156699	0.025	0.1639
17	0	0	0.153015	0.017101	0.1639
18	0	0	0.15076	0.008682	0.1639
19	0	0	0.15	0	0.1639
20	0	0	0.15076	-0.00868	0.1639
21	0	0	0.153015	-0.0171	0.1639
22	0	0	0.156699	-0.025	0.1639
23	0	0	0.161698	-0.03214	0.1639
24	0	0	0.167861	-0.0383	0.1639
25	0	0	0.175	-0.0433	0.1639
26	0	0	0.182899	-0.04698	0.1639
27	0	0	0.191318	-0.04924	0.1639
28	0	0	0.2	-0.05	0.1639
29	0	0	0.208682	-0.04924	0.1639
30	0	0	0.217101	-0.04698	0.1639
31	0	0	0.225	-0.0433	0.1639
32	0	0	0.232139	-0.0383	0.1639
33	0	0	0.238302	-0.03214	0.1639
34	0	0	0.243301	-0.025	0.1639

Titik	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
35	0	0	0.246985	-0.0171	0.1639
36	0	0	0.24924	-0.00868	0.1639

10. Posisi dan Orientasi Lingkaran JST

Titik	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	9.3e-06	3.9e-05	0.249956	1.94E-07	0.163875
2	5.5e-07	3.0e-05	0.249215	0.008681	0.163882
3	4.7e-05	-2.9e-05	0.246981	0.017101	0.163894
4	6.8e-06	-3.2e-05	0.243327	0.025003	0.163947
5	2.3e-05	4.3e-06	0.238261	0.032134	0.16388
6	-2.3e-05	-2.8e-05	0.232142	0.038303	0.16391
7	-3.7e-06	3.8e-05	0.224996	0.043301	0.163895
8	2.1e-05	1.5e-05	0.217098	0.046984	0.163905
9	-1.3e-05	3.1e-05	0.208724	0.04925	0.163918
10	1.5e-05	-9.9e-06	0.199996	0.049999	0.163885
11	1.7e-05	2.5e-05	0.191342	0.049247	0.163939
12	3.7e-05	-1.3e-05	0.182898	0.046984	0.163888
13	3.5e-05	-4.7e-05	0.174984	0.043297	0.163883
14	4.7e-05	-4.9e-05	0.16787	0.038304	0.163933
15	-4.4e-05	-3.8e-05	0.161721	0.032144	0.163932
16	4.8e-05	2.5e-05	0.156701	0.025	0.163927
17	3.9e-05	4.6e-05	0.153064	0.017106	0.163939
18	-8.6e-06	-3.2e-05	0.150726	0.008681	0.163853
19	4.1e-05	-7.5e-06	0.150043	3.67E-08	0.163946
20	4.0e-05	5.0e-06	0.150808	-0.00869	0.16395
21	2.4e-05	-3.5e-05	0.152979	-0.0171	0.163866
22	4.3e-06	-4.6e-05	0.156699	-0.025	0.163916
23	4.2e-05	2.4e-05	0.161725	-0.03214	0.163943
24	1.3e-05	-4.3e-05	0.167863	-0.0383	0.16392
25	-4.1e-05	3.8e-05	0.174958	-0.04329	0.163876
26	-4.2e-05	4.8e-05	0.182899	-0.04698	0.163891
27	4.8e-05	2.0e-06	0.191342	-0.04925	0.163948
28	2.3e-05	1.4e-06	0.200016	-0.05	0.163912
29	2.3e-05	2.6e-05	0.208686	-0.04924	0.163932

Titik	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
30	-2.6e-05	-1.6e-05	0.217101	-0.04698	0.163922
31	1.8e-05	3.9e-05	0.225002	-0.0433	0.163919
32	-1.7e-05	2.5e-05	0.23214	-0.0383	0.163912
33	-4.7e-05	-2.1e-05	0.238304	-0.03214	0.163904
34	-3.9e-05	-1.9e-05	0.243306	-0.025	0.163905
35	-1.6e-05	-4.9e-05	0.246985	-0.0171	0.163892
36	-3.2e-05	-4.3e-05	0.249238	-0.00868	0.163909

11. Error Posisi dan Orientasi Lingkaran JST

Titik	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	9.3e-06	3.9e-05	4.41e-05	-1.94e-07	2.51e-05
2	5.5e-07	3.0e-05	2.50e-05	9.35e-07	1.78e-05
3	4.7e-05	-2.9e-05	3.29e-06	3.05e-07	5.65e-06
4	6.8e-06	-3.2e-05	-2.58e-05	-2.69e-06	-4.65e-05
5	2.3e-05	4.3e-06	4.14e-05	5.73e-06	2.03e-05
6	-2.3e-05	-2.8e-05	-2.58e-06	-5.77e-07	-9.58e-06
7	-3.7e-06	3.8e-05	4.45e-06	7.19e-07	5.08e-06
8	2.1e-05	1.5e-05	2.56e-06	4.49e-07	-4.54e-06
9	-1.3e-05	3.1e-05	-4.12e-05	-9.55e-06	-1.81e-05
10	1.5e-05	-9.9e-06	4.45e-06	1.22e-06	1.54e-05
11	1.7e-05	2.5e-05	-2.40e-05	-6.12e-06	-3.94e-05
12	3.7e-05	-1.3e-05	1.25e-06	4.21e-07	1.18e-05
13	3.5e-05	-4.7e-05	1.59e-05	4.00e-06	1.68e-05
14	4.7e-05	-4.9e-05	-9.59e-06	-2.05e-06	-3.33e-05
15	-4.4e-05	-3.8e-05	-2.35e-05	-4.81e-06	-3.21e-05
16	4.8e-05	2.5e-05	-2.00e-06	-3.30e-07	-2.65e-05
17	3.9e-05	4.6e-05	-4.86e-05	-5.36e-06	-3.87e-05
18	-8.6e-06	-3.2e-05	3.33e-05	1.83e-06	4.71e-05
19	4.1e-05	-7.5e-06	-4.29e-05	-3.67e-08	-4.61e-05
20	4.0e-05	5.0e-06	-4.83e-05	2.80e-06	-4.98e-05
21	2.4e-05	-3.5e-05	3.67e-05	-4.10e-06	3.41e-05
22	4.3e-06	-4.6e-05	-5.85e-07	1.63e-07	-1.58e-05
23	4.2e-05	2.4e-05	-2.69e-05	5.22e-06	-4.32e-05
24	1.3e-05	-4.3e-05	-1.93e-06	5.46e-07	-2.03e-05

Titik	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
25	-4.1e-05	3.8e-05	4.21e-05	-1.04e-05	2.37e-05
26	-4.2e-05	4.8e-05	1.47e-07	-8.07e-08	8.53e-06
27	4.8e-05	2.0e-06	-2.45e-05	6.36e-06	-4.81e-05
28	2.3e-05	1.4e-06	-1.56e-05	3.94e-06	-1.23e-05
29	2.3e-05	2.6e-05	-3.37e-06	8.88e-07	-3.18e-05
30	-2.6e-05	-1.6e-05	-3.22e-07	1.57e-07	-2.18e-05
31	1.8e-05	3.9e-05	-2.27e-06	4.24e-07	-1.91e-05
32	-1.7e-05	2.5e-05	-5.89e-07	-3.00e-08	-1.18e-05
33	-4.7e-05	-2.1e-05	-1.34e-06	2.21e-07	-4.02e-06
34	-3.9e-05	-1.9e-05	-4.32e-06	5.02e-07	-5.18e-06
35	-1.6e-05	-4.9e-05	-2.24e-08	1.55e-09	7.61e-06
36	-3.2e-05	-4.3e-05	1.95e-06	-2.45e-07	-9.15e-06
RMSE	3.13e-05	3.2e-05	2.39e-05	3.60e-06	2.81e-05

12. Posisi dan Orientasi Target Lingkaran Miring

Titik	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	0	0	0.233361	0	0.201104
2	0	0	0.232604	0.008682	0.201038
3	0	0	0.230357	0.017101	0.200841
4	0	0	0.226687	0.025	0.20052
5	0	0	0.221707	0.032139	0.200084
6	0	0	0.215568	0.038302	0.199547
7	0	0	0.208456	0.043301	0.198925
8	0	0	0.200587	0.046985	0.198237
9	0	0	0.1922	0.04924	0.197503
10	0	0	0.183551	0.05	0.196746
11	0	0	0.174902	0.04924	0.195989
12	0	0	0.166515	0.046985	0.195256
13	0	0	0.158646	0.043301	0.194567
14	0	0	0.151534	0.038302	0.193945
15	0	0	0.145394	0.032139	0.193408
16	0	0	0.140414	0.025	0.192972
17	0	0	0.136745	0.017101	0.192651
18	0	0	0.134498	0.008682	0.192455

Titik	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
19	0	0	0.133741	0	0.192388
20	0	0	0.134498	-0.00868	0.192455
21	0	0	0.136745	-0.0171	0.192651
22	0	0	0.140414	-0.025	0.192972
23	0	0	0.145394	-0.03214	0.193408
24	0	0	0.151534	-0.0383	0.193945
25	0	0	0.158646	-0.0433	0.194567
26	0	0	0.166515	-0.04698	0.195256
27	0	0	0.174902	-0.04924	0.195989
28	0	0	0.183551	-0.05	0.196746
29	0	0	0.1922	-0.04924	0.197503
30	0	0	0.200587	-0.04698	0.198237
31	0	0	0.208456	-0.0433	0.198925
32	0	0	0.215568	-0.0383	0.199547
33	0	0	0.221707	-0.03214	0.200084
34	0	0	0.226687	-0.025	0.20052
35	0	0	0.230357	-0.0171	0.200841
36	0	0	0.232604	-0.00868	0.201038

13. Posisi dan Orientasi Lingkaran Miring JST

Titik	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	-4.9e-05	-3.5e-05	0.233359	-8.59E-08	0.20111
2	-2.7e-05	-2.8e-05	0.232603	0.008682	0.201033
3	2.1e-05	2.3e-06	0.230357	0.017101	0.200844
4	-3.9e-05	-2.8e-05	0.226686	0.025	0.200492
5	-1.8e-05	2.8e-07	0.221705	0.032139	0.200076
6	-1.2e-05	-1.3e-05	0.215568	0.038302	0.199546
7	3.4e-05	3.7e-06	0.208456	0.043301	0.198927
8	-3.2e-05	3.0e-05	0.200588	0.046985	0.198233
9	-2.7e-05	4.7e-05	0.192201	0.049241	0.197504
10	-2.1e-05	-3.6e-05	0.183547	0.049999	0.196744
11	-1.5e-05	1.0e-05	0.174899	0.04924	0.195984
12	3.9e-05	1.6e-05	0.166515	0.046985	0.195255
13	-1.3e-05	-2.0e-05	0.158646	0.043301	0.194571

Titik	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
14	-2.9e-05	-4.2e-06	0.151542	0.038304	0.193948
15	4.5e-05	5.5e-06	0.145354	0.03213	0.193363
16	3.2e-05	-2.8e-05	0.140413	0.025	0.19297
17	3.4e-05	-1.4e-05	0.136735	0.0171	0.192644
18	-2.5e-05	2.8e-05	0.134497	0.008682	0.192458
19	2.5e-05	-3.7e-05	0.133741	1.89E-08	0.192387
20	-8.7e-06	2.7e-05	0.134495	-0.00868	0.192453
21	3.2e-05	-1.3e-05	0.136768	-0.0171	0.192682
22	-4.7e-05	-3.5e-05	0.140415	-0.025	0.19297
23	-6.2e-06	3.1e-05	0.14544	-0.03215	0.193454
24	-2.5e-05	2.4e-05	0.151533	-0.0383	0.193946
25	4.4e-05	1.6e-05	0.158648	-0.0433	0.19456
26	3.9e-05	-2.8e-05	0.166514	-0.04698	0.195254
27	3.3e-05	2.7e-05	0.17491	-0.04924	0.196014
28	-1.7e-05	-3.8e-05	0.183552	-0.05	0.196743
29	-1.2e-05	-2.2e-05	0.192201	-0.04924	0.197509
30	7.1e-06	-4.2e-06	0.200587	-0.04698	0.198239
31	-2.2e-05	1.7e-05	0.208459	-0.0433	0.198928
32	-1.5e-05	4.7e-05	0.215569	-0.0383	0.199551
33	-2.7e-05	-1.7e-05	0.221709	-0.03214	0.200086
34	6.8e-07	-2.4e-05	0.226688	-0.025	0.200519
35	4.2e-05	-3.4e-05	0.230358	-0.0171	0.200843
36	-6.8e-06	-2.4e-05	0.232602	-0.00868	0.201041

14. Error Posisi dan Orientasi Lingkaran Miring JST

Titik	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	-4.9e-05	-3.5e-05	1.95e-06	8.59e-08	-5.65e-06
2	-2.7e-05	-2.8e-05	5.05e-07	-8.29e-08	5.00e-06
3	2.1e-05	2.3e-06	-7.10e-07	-1.71e-07	-3.28e-06
4	-3.9e-05	-2.8e-05	1.35e-06	3.45e-07	2.80e-05
5	-1.8e-05	2.8e-07	2.17e-06	4.42e-07	8.38e-06
6	-1.2e-05	-1.3e-05	-4.89e-08	-1.24e-07	9.03e-07
7	3.4e-05	3.7e-06	-5.60e-07	-6.02e-08	-2.02e-06
8	-3.2e-05	3.0e-05	-1.18e-06	-2.14e-07	3.17e-06

Titik	Orientasi (derajat)		Posisi (m)		
	<i>roll</i>	<i>pitch</i>	<i>x</i>	<i>y</i>	<i>z</i>
9	-2.7e-05	4.7e-05	-8.84e-07	-1.38e-07	-1.19e-06
10	-2.1e-05	-3.6e-05	4.03e-06	1.13e-06	2.34e-06
11	-1.5e-05	1.0e-05	2.95e-06	8.28e-07	5.31e-06
12	3.9e-05	1.6e-05	2.05e-07	-9.70e-09	4.13e-07
13	-1.3e-05	-2.0e-05	-3.37e-07	-9.89e-09	-3.37e-06
14	-2.9e-05	-4.2e-06	-7.90e-06	-2.11e-06	-3.23e-06
15	4.5e-05	5.5e-06	4.08e-05	8.94e-06	4.47e-05
16	3.2e-05	-2.8e-05	1.13e-06	1.57e-07	1.74e-06
17	3.4e-05	-1.4e-05	1.01e-05	1.32e-06	7.11e-06
18	-2.5e-05	2.8e-05	6.34e-07	1.35e-07	-2.95e-06
19	2.5e-05	-3.7e-05	1.46e-08	-1.89e-08	1.83e-06
20	-8.7e-06	2.7e-05	2.52e-06	-2.43e-07	1.79e-06
21	3.2e-05	-1.3e-05	-2.27e-05	2.95e-06	-3.05e-05
22	-4.7e-05	-3.5e-05	-4.87e-07	-3.66e-08	2.58e-06
23	-6.2e-06	3.1e-05	-4.59e-05	1.01e-05	-4.56e-05
24	-2.5e-05	2.4e-05	4.88e-07	-1.79e-07	-8.62e-07
25	4.4e-05	1.6e-05	-1.50e-06	3.63e-07	7.01e-06
26	3.9e-05	-2.8e-05	9.47e-07	-1.36e-07	1.33e-06
27	3.3e-05	2.7e-05	-8.07e-06	2.14e-06	-2.41e-05
28	-1.7e-05	-3.8e-05	-9.64e-07	3.95e-07	2.89e-06
29	-1.2e-05	-2.2e-05	-7.77e-07	1.38e-07	-5.64e-06
30	7.1e-06	-4.2e-06	2.36e-07	1.33e-08	-2.49e-06
31	-2.2e-05	1.7e-05	-3.70e-06	6.39e-07	-3.10e-06
32	-1.5e-05	4.7e-05	-1.19e-06	1.13e-07	-3.91e-06
33	-2.7e-05	-1.7e-05	-1.34e-06	8.62e-08	-1.89e-06
34	6.8e-07	-2.4e-05	-4.97e-07	-5.53e-08	1.62e-06
35	4.2e-05	-3.4e-05	-1.68e-06	1.85e-07	-1.69e-06
36	-6.8e-06	-2.4e-05	1.66e-06	-2.12e-07	-3.56e-06
RMSE	2.93e-05	2.7e-05	1.13e-05	2.39e-06	1.38e-05

RIWAYAT PENULIS



Daniel Cristover Sirait, dilahirkan di Surabaya pada tanggal 27 Desember 1995, penulis merupakan anak terakhir dari empat bersaudara. Penulis telah menempuh pendidikan formal di SDN Dukuh Kupang 6 (2001-2007), SMPN 10 Surabaya (2007-2010), SMAN 21 Surabaya (2010-2013), dan D3 Teknik Elektro bidang studi *Computer Control* ITS (2013-2016). Selanjutnya terdapat di Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember Surabaya. Di Jurusan Teknik Elektro penulis mengambil bidang studi Teknik Sistem Pengaturan dengan judul Tugas Akhir “PERANCANGAN JARINGAN SARAF TIRUAN UNTUK MENYELESAIKAN KINEMATIKA BALIK MANIPULATOR ROBOT DENSO 6 DOF”.

Contact Person :

Email : danielsirait11@gmail.com

No. HP : 089682871463

Line : daniel_sirait

-----Halaman ini sengaja dikosongkan-----