



TUGAS AKHIR - EE 184801

**PERANCANGAN PENYELESAIAN KINEMATIKA BALIK
PADA POSISI DAN ORIENTASI *END EFFECTOR* ROBOT
DENSO MENGGUNAKAN *NEURO FUZZY***

Jevta Marihot Pasaribu
NRP 07111745000052

Dosen Pembimbing
Ir. Rusdhianto Effendi AK., MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - EE 184801

***DESIGN OF INVERSE KINEMATICS COMPLETION ON
POSITION AND ORIENTATION END EFFECTOR DENSO
ROBOT USING NEURO FUZZY***

Jevta Marihot Pasaribu
NRP 07111745000052

Supervisor
Ir. Rusdhianto Effendi AK., MT.

*ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2019*

[Halaman ini sengaja dikosongkan]

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Perancangan Penyelesaian Kinematika Balik Pada Posisi dan Orientasi *End Effector* Robot Denso Menggunakan *Neuro Fuzzy***" 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, 25 Juni 2019



Jevta Marihot Pasaribu
NRP 07111745000052

[Halaman ini sengaja dikosongkan]

**PERANCANGAN PENYELESAIAN KINEMATIKA BALIK
PADA POSISI DAN ORIENTASI *END EFFECTOR* ROBOT
DENSO MENGGUNAKAN *NEURO FUZZY***

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Teknik Sistem Pengaturan
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing



Ir. Rusdianto Effendi A.K., MT.
NIP. 195704241985021001



[Halaman ini sengaja dikosongkan]

PERANCANGAN PENYELESAIAN KINEMATIKA BALIK PADA POSISI DAN ORIENTASI *END EFFECTOR* ROBOT DENSO MENGGUNAKAN *NEURO FUZZY*

Nama : Jevta Marihot Pasaribu
Pembimbing : Ir. Rusdhianto Effendi AK., MT.

ABSTRAK

Robotika sering digunakan di dunia industri. Aplikasi dari bidang robotika yang sering digunakan di dunia industri adalah manipulator robot. Pada tugas akhir ini akan diselesaikan permasalahan kinematika balik pada manipulator robot denso. Pada kinematika balik diberi masukan titik target yang ingin dituju *end effector* dan masing-masing *joint* akan bergerak dengan besar sudut tertentu, sehingga *end effector* bergerak menuju titik target. Agar manipulator robot dapat bergerak ke titik target yang telah ditentukan, manipulator robot memiliki banyak kemungkinan konfigurasi yang akan terjadi. Diperlukannya metode yang digunakan sebagai solusi kinematika balik sehingga didapatkan besaran sudut tertentu dari masing-masing *joint* pada manipulator robot. Metode *neuro fuzzy* akan digunakan untuk mencari satu solusi kinematika balik sehingga dapat menentukan konfigurasi masing-masing *joint* pada manipulator robot dan mengatur pergerakan posisi dan orientasi dari manipulator robot. Setelah dilakukan perancangan dan pengujian, *neuro fuzzy* telah dapat diaplikasikan pada kinematika balik manipulator robot denso, dimana *end effector* manipulator robot telah dapat menuju posisi dan orientasi yang telah di tentukan dengan nilai laju pembelajaran (α) 0.0001 dan nilai ketelitian 0.00005 dengan rata-rata *error* posisi dibawah 0.00005 m dan *error* orientasi dibawah 0.00005 derajat.

Kata kunci : Robotika, *Neuro Fuzzy*, Manipulator Robot, Kinematika Balik.

[Halaman ini sengaja dikosongkan]

DESIGN OF INVERSE KINEMATICS COMPLETION ON POSITION AND ORIENTATION END EFFECTOR DENSO ROBOT USING NEURO FUZZY

Name : Jevta Marihot Pasaribu
Supervisor : Ir. Rusdhianto Effendi AK., MT.

ABSTRACT

Robotics is often used in the industrial world. Applications from the field of robotics that are often used in the industrial world are robotic manipulators. In this final assignment, we will solve the inverse kinematics problem on the denso robot manipulator. On the inverse kinematics the target point is to be inputted to the end effector and each joint will move with a certain angle, so that the end effector moves towards the target point. In order for the robotic manipulator to move to a predetermined target point, the robot manipulator has many possible configurations that will occur. The method needed to be used as a solution to inverse kinematics is to obtain a certain amount of angle from each joint in the robot manipulator. The neuro fuzzy method will be used to find a inverse kinematics solution so that 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, neuro fuzzy can be applied to the inverse kinematics of the Denso robot manipulator, where the robot's end effector manipulator has been able to reach the position and orientation determined by the value of the learning rate (α) 0.0001 and the accuracy value 0.00005 with the average error position below 0.00005 m and orientation error below 0.00005 degrees.

Keywords : *Robotics, Neuro fuzzy, Robot Manipulator, Inverse Kinematics*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan ke Tuhan Yesus yang telah melimpahkan segala rahmat-Nya sehingga penulis dapat menyelesaikan tugas akhir yang merupakan persyaratan untuk menyelesaikan pendidikan Sarjana pada Bidang Studi Teknik Sistem Pengaturan, Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

PERANCANGAN PENYELESAIAN KINEMATIKA BALIK PADA POSISI DAN ORIENTASI *END EFFECTOR* ROBOT DENSO MENGGUNAKAN *NEURO FUZZY*

Dengan penyusunan tugas akhir ini penyusun tidak terlepas dari bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis hendak menyampaikan rasa terima kasih kepada pihak – pihak yang telah membantu. Ucapan terima kasih ini kami sampaikan kepada:

1. Tuhan Yesus yang telah memberi berkat-Nya dalam pembuatan tugas akhir ini.
2. Kedua orang tua dan seluruh keluarga yang telah memberikan dukungan penuh dan teladan bagi penulis.
3. Bapak Ir. Rusdhianto Effendi AK., MT. selaku Dosen pembimbing yang telah meluangkan waktunya untuk membimbing penulis dalam pembuatan tugas akhir ini.
4. Teman-teman angkatan 2017 Lintas Jalur atas semangat dan kerjasamanya.

Semoga karya tulis ini dapat bermanfaat bagi penulis sendiri maupun bagi penelitian selanjutnya.

Surabaya, 25 Juni 2019

Jevta Marihot Pasaribu
NRP 07111745000052

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

HALAMAN

LAMAN JUDUL	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
LEMBAR PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xxi
1. BAB 1 PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Permasalahan	2
1.3. Tujuan.....	2
1.4. Batasan Masalah.....	3
1.5. Metodologi	3
1.6. Sistematika	4
1.7. Relevansi atau Manfaat	5
2. BAB 2 TINJAUAN PUSTAKA.....	7
2.1. Robotika	7
2.2. Konfigurasi Manipulator Robot.....	8
2.3. Jenis Sendi Manipulator Robot.....	9
2.3.1. Sendi Putar (<i>Revolute</i>).....	10
2.3.2. Sendi Geser (<i>Prismatic</i>)	10
2.4. Robot Denso	10
2.5. Transformasi Homogen	11
2.6. Kinematika Robot.....	13
2.7. Kinematika Maju	14
2.8. Kinematika Balik.....	15
2.8.1. Pendekatan Geometri	16
2.8.2. Pendekatan Numerik	18
2.9. Logika <i>Fuzzy</i>	19
2.9.1. Himpunan <i>Fuzzy</i>	19
2.9.2. Fungsi Keanggotaan.....	19
2.9.3. <i>Fuzzifikasi</i>	20
2.9.4. Sistem Inferensi <i>Fuzzy</i>	21

2.9.5.	<i>Defuzzifikasi</i>	21
2.10.	Jaringan Syaraf Tiruan	22
2.11.	<i>Neuro-Fuzzy</i>	23
2.11.1.	<i>Forward Propagation</i>	24
2.11.2.	<i>Backward Propagation</i>	25
3.	BAB 3 PERANCANGAN SISTEM	27
3.1.	Parameter <i>Denavit Hartenberg</i>	27
3.2.	Transformasi Homogen.....	31
3.3.	Persamaan Kinematika Maju.....	34
3.4.	Solusi Kinematika Balik Menggunakan <i>Neuro Fuzzy</i>	36
3.4.1.	Perancangan Struktur <i>Neuro Fuzzy</i>	37
3.4.2.	Tahap <i>Forward Propagation</i>	38
3.4.3.	Tahap <i>Backward Propagation</i>	41
3.5.	Simulasi Kinematika Maju dan Kinematika Balik	42
4.	BAB 4 PENGUJIAN DAN ANALISA	45
4.1.	Pengujian Kinematika Maju	45
4.2.	Pengujian Laju Pembelajaran (α) Kinematika Balik.....	49
4.3.	Pengujian Ketelitian Kinematika Balik	58
4.4.	Kinematika Balik Membuat Pola Segitiga	62
4.5.	Kinematika Balik Membuat Pola Lingkaran	69
5.	BAB 5 PENUTUP	77
5.1.	Kesimpulan	77
5.2.	Saran.....	78
	DAFTAR PUSTAKA	79
	LAMPIRAN	81
	RIWAYAT HIDUP	101

DAFTAR GAMBAR

HALAMAN

Gambar 2.1 (a) Konfigurasi <i>Articulated (RRR)</i> (b) Konfigurasi <i>Spherical (RRP)</i> (c) Konfigurasi <i>SCARA (RRP)</i> (d) Konfigurasi <i>Cylindrical (RPP)</i> (e) Konfigurasi <i>Cartesian (PPP)</i>	9
Gambar 2.2 Sendi Putar [4].....	10
Gambar 2.3 Sendi Geser [4].....	10
Gambar 2.4 Robot Denso VP-6242G.....	11
Gambar 2.5 Cara Kerja Kinematika Maju dan Kinematika Balik [8] ...	13
Gambar 2.6 Manipulator Robot 3 Sendi	16
Gambar 2.7 Manipulator Robot 3 Sendi dari sisi atas.....	17
Gambar 2.8 Manipulator Robot 3 Sendi dari sisi samping	17
Gambar 2.9 Struktur fungsi keanggotaan fuzzy	19
Gambar 2.10 Struktur <i>Neural Network</i>	23
Gambar 2.11 Struktur Kontroler <i>Neuro Fuzzy</i>	23
Gambar 3.1 Panjang Lengan Robot Denso VP-6242G [6]	27
Gambar 3.2 Pemberian Lembel Sumbu z [3].....	28
Gambar 3.3 Pemberian Lembel Sumbu x_0 dan y_0 [3].....	28
Gambar 3.4 Pemberian Lembel o_i [3].....	29
Gambar 3.5 Pemberian Lembel Sumbu x_i [3]	29
Gambar 3.6 Pemberian Lembel Sumbu y_i [3]	30
Gambar 3.7 Pemberian Lembel o_n , x_n , y_n dan z_n . [3]	30
Gambar 3.8 Struktur <i>Neuro Fuzzy</i>	38
Gambar 3.9 Himpunan Pendukung dx, dy, dz, dr dan dp	39
Gambar 3.10 Tampilan Simulasi Manipulator Robot Denso	43
Gambar 4.1 Bentuk Manipulator Robot Denso Percobaan 1	46
Gambar 4.2 Bentuk Manipulator Robot Denso Percobaan 2	47
Gambar 4.3 Bentuk Manipulator Robot Denso Percobaan 3	47
Gambar 4.4 Bentuk Manipulator Robot Denso Percobaan 4	48
Gambar 4.5 Bentuk Manipulator Robot Denso Percobaan 5	48
Gambar 4.6 Bentuk Manipulator Robot Denso Percobaan 6	49
Gambar 4.7 Posisi dan Orientasi Awal <i>End Effector</i> Percobaan 1.....	51
Gambar 4.8 Posisi dan Orientasi Target <i>End Effector</i> Percobaan 1	51
Gambar 4.9 Posisi dan Orientasi Awal <i>End Effector</i> Percobaan 2.....	53
Gambar 4.10 Posisi dan Orientasi Target <i>End Effector</i> Percobaan 2....	53

Gambar 4.11 Posisi dan Orientasi Awal <i>End Effector</i> Percobaan 3	55
Gambar 4.12 Posisi dan Orientasi Target <i>End Effector</i> Percobaan 3.....	55
Gambar 4.13 Titik-Titik Target Segitiga Datar	62
Gambar 4.14 Grafik Perbandingan orientasi Titik Target Dengan Keluaran Kinematika Balik Membuat Pola Segitiga Datar	63
Gambar 4.15 Grafik Perbandingan Posisi Titik Target Dengan Keluaran Kinematika Balik Membuat Pola Segitiga Datar.....	63
Gambar 4.16 Grafik <i>Error</i> Orientasi Keluaran Kinematika Balik Membuat Pola Segitiga Datar	64
Gambar 4.17 Grafik <i>Error</i> Posisi Keluaran Kinematika Balik Membuat Pola Segitiga Datar	64
Gambar 4.18 Titik-Titik Keluaran Kinematika Balik Membuat Pola Segitiga Datar	65
Gambar 4.19 Titik-Titik Target Segitiga Miring	66
Gambar 4.20 Grafik Perbandingan orientasi Titik Target Dengan Keluaran Kinematika Balik Membuat Pola Segitiga Miring	66
Gambar 4.21 Grafik Perbandingan Posisi Titik Target Dengan Keluaran Kinematika Balik Membuat Pola Segitiga Miring.....	67
Gambar 4.22 Grafik <i>Error</i> Orientasi Keluaran Kinematika Balik Membuat Pola Segitiga Miring.....	67
Gambar 4.23 Grafik <i>Error</i> Posisi Keluaran Kinematika Balik Membuat Pola Segitiga Miring	68
Gambar 4.24 Titik-Titik Keluaran Kinematika Balik Membuat Pola Segitiga Miring	68
Gambar 4.25 Titik-Titik Target Lingkaran Datar	69
Gambar 4.26 Grafik Perbandingan orientasi Titik Target Dengan Keluaran Kinematika Balik Membuat Pola Lingkaran Datar	70
Gambar 4.27 Grafik Perbandingan Posisi Titik Target Dengan Keluaran Kinematika Balik Membuat Pola Lingkaran Datar.....	70
Gambar 4.28 Grafik <i>Error</i> Orientasi Keluaran Kinematika Balik Membuat Pola Lingkaran Datar.....	71
Gambar 4.29 Grafik <i>Error</i> Posisi Keluaran Kinematika Balik Membuat Pola Lingkaran Datar	71
Gambar 4.30 Titik-Titik Keluaran Kinematika Balik Membuat Pola Lingkaran Datar.....	72
Gambar 4.31 Titik-Titik Target Lingkaran Miring	73
Gambar 4.32 Grafik Perbandingan orientasi Titik Target Dengan Keluaran Kinematika Balik Membuat Pola Lingkaran Miring	73

Gambar 4.33 Grafik Perbandingan Posisi Titik Target Dengan Keluaran Kinematika Balik Membuat Pola Lingkaran Miring.....	74
Gambar 4.34 Grafik <i>Error</i> Orientasi Keluaran Kinematika Balik Membuat Pola Lingkaran Miring.....	74
Gambar 4.35 Grafik <i>Error</i> Posisi Keluaran Kinematika Balik Membuat Pola Lingkaran Miring	75
Gambar 4.36 Titik-Titik Keluaran Kinematika Balik Membuat Pola Lingkaran Miring	75

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

	HALAMAN
Tabel 3.1 Parameter DH.....	31
Tabel 3.2 Indeks Parameter DH	31
Tabel 4.1 Masukan Kinematika Maju	45
Tabel 4.2 Keluaran Kinematika Maju	46
Tabel 4.3 Masukan Kinematika Balik Pengujian (α) Percobaan 1	50
Tabel 4.4 Keluaran Kinematika Balik Pengujian (α) Percobaan 1	50
Tabel 4.5 Iterasi Keluaran Kinematika Balik Pengujian (α) Percobaan 1	50
Tabel 4.6 Masukan Kinematika Balik Pengujian (α) Percobaan 2	52
Tabel 4.7 Keluaran Kinematika Balik Pengujian (α) Percobaan 2	52
Tabel 4.8 Iterasi Keluaran Kinematika Balik Pengujian (α) Percobaan 2	52
Tabel 4.9 Masukan Kinematika Balik Pengujian (α) Percobaan 3	54
Tabel 4.10 Keluaran Kinematika Balik Pengujian (α) Percobaan 3	54
Tabel 4.11 Iterasi Keluaran Kinematika Balik Pengujian (α) Percobaan 3	54
Tabel 4.12 Keluaran Kinematika Maju Pengujian (α) Percobaan 1.....	56
Tabel 4.13 Nilai <i>Error</i> Pengujian (α) Percobaan 1	56
Tabel 4.14 Keluaran Kinematika Maju Pengujian (α) Percobaan 2.....	57
Tabel 4.15 Nilai <i>Error</i> Pengujian (α) Percobaan 2	57
Tabel 4.16 Keluaran Kinematika Maju Pengujian (α) Percobaan 3.....	57
Tabel 4.17 Nilai <i>Error</i> Pengujian (α) Percobaan 3	57
Tabel 4.18 Keluaran Kinematika Balik Pengujian Ketelitian Percobaan 1	58
Tabel 4.19 Keluaran Kinematika Maju Pengujian Ketelitian Percobaan 1	58
Tabel 4.20 Iterasi Keluaran Kinematika Balik Pengujian Ketelitian Percobaan 1	59
Tabel 4.21 Nilai <i>Error</i> Pengujian Ketelitian Percobaan 1	59
Tabel 4.22 Keluaran Kinematika Balik Pengujian Ketelitian Percobaan 2	59
Tabel 4.23 Keluaran Kinematika Maju Pengujian Ketelitian Percobaan 2	60

Tabel 4.24 Iterasi Keluaran Kinematika Balik Pengujian Ketelitian Percobaan 2.....	60
Tabel 4.25 Nilai <i>Error</i> Pengujian Ketelitian Percobaan 2	60
Tabel 4.26 Keluaran Kinematika Balik Pengujian Ketelitian Percobaan 3	61
Tabel 4.27 Keluaran Kinematika Maju Pengujian Ketelitian Percobaan 3	61
Tabel 4.28 Iterasi Keluaran Kinematika Balik Pengujian Ketelitian Percobaan 3.....	61
Tabel 4.29 Nilai <i>Error</i> Pengujian Ketelitian Percobaan 3	61

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Robotika merupakan cabang ilmu yang memiliki hubungan dengan desain, konstruksi, pembuatan dan aplikasi sebuah robot. Robotika sangat sering digunakan di dunia industri. Aplikasi dari bidang robotika yang sering digunakan di dunia industri adalah manipulator robot. Manipulator robot adalah mekanik elektronik yang menyerupai sebuah lengan manusia, manipulator robot juga sering dikenal ataupun disebut dengan lengan robot.

Manipulator robot berguna untuk membantu melakukan pekerjaan-pekerjaan di tempat yang berbahaya bagi manusia, salah satu contohnya di industri pengecatan mobil yang mana dapat merusak kesehatan organ pernafasan dikarenakan timbal cat yang dihirup oleh manusia. Tidak hanya itu manipulator robot juga membantu melakukan pekerjaan-pekerjaan yang berat secara berulang-ulang bagi manusia, contohnya pada industri otomotif banyak sekali pekerjaan-pekerjaan berat yang dilakukan seperti memindahkan bodi mobil yang berat dan besar secara berulang-ulang. Manipulator robot juga melakukan hal yang teliti secara berulang-ulang, contohnya pada industri otomotif yaitu pengelasan bodi mobil. Sehingga penggunaan manipulator robot sangat diperlukan.

Manipulator robot bergerak menuju suatu titik koordinat yang akan dituju yang telah diprogram menggunakan komputer, untuk mengendalikan pergerakan dari manipulator robot diperlukan studi tentang kinematika. Kinematika manipulator robot merupakan studi analitis dari pergerakan manipulator robot terhadap kerangka tanpa memperhatikan gaya yang dapat mempengaruhi pergerakan dari manipulator robot. Pada tugas akhir ini kinematika yang akan digunakan yaitu kinematika balik, pada kinematika balik akan ditentukan titik koordinat tertentu yang akan dituju oleh manipulator robot, kemudian *joint* dari manipulator robot akan bergerak dengan besaran sudut tertentu, sehingga manipulator robot dapat bergerak menuju ke titik koordinat yang telah ditentukan.

Pergerakan dari manipulator robot di bagi menjadi dua pergerakan. Pertama, ialah pergerakan posisi yang mana pada pergerakan posisi terdapat tiga parameter, yaitu maju/mundur, kanan/kiri dan atas/bawah. Kedua pergerakan orientasi, yang mana pergerakan orientasi juga dibagi

menjadi tiga parameter yaitu *roll*, *pitch* dan *yaw*. Manipulator robot yang akan digunakan pada tugas akhir ini memiliki jumlah *joint* sebanyak 6, semua *joint* pada manipulator berjenis *revolute*. Manipulator dengan jenis *revolute* pada setiap *joint* nya disebut dengan manipulator robot berkonfigurasi *articulated*. Karena manipulator robot memiliki jumlah DoF sebanyak 6 membuat pergerakannya memiliki banyak kemungkinan konfigurasi untuk menuju ke titik koordinat yang telah di tentukan.

Beberapa penelitian telah dilakukan pada manipulator robot, penelitian yang pertama dilakukan oleh Rika Puspitasari Rangkuti pada tahun 2017 yang berjudul “Kinematika Balik Menggunakan *Neuro-Fuzzy* Pada Manipulator Robot Denso”, pada penelitian ini telah di dapatkan penyelesaian kinematika balik pada manipulator robot denso dengan *neuro fuzzy* namun hanya untuk posisi dari *end effector*-nya saja dan hanya mencari nilai besar sudut di tiga sendi awal [2], penelitian yang kedua dilakukan oleh Daniel Cristover Sirait pada tahun 2018 yang berjudul “Perancangan Jaringan Saraf Tiruan Untuk Menyelesaikan Kinematika Balik Manipulator Robot Denso 6-DoF”, pada penelitian ini mendapatkan solusi kinematika balik pada manipulator robot denso dengan metode jaringan saraf tiruan untuk mendapatkan besar sudut lima di sendi awal dari manipulator robot denso [3].

1.2. Permasalahan

Agar manipulator robot 6-DoF dapat bergerak ke titik koordinat yang telah ditentukan, manipulator robot memiliki banyak kemungkinan konfigurasi yang akan terjadi. Diperlukannya metode yang digunakan sebagai solusi kinematika balik sehingga didapatkan besaran sudut tertentu dari masing-masing *joint* pada manipulator robot. Rumusan masalah yang akan dibahas pada tugas akhir ini adalah bagaimana menemukan solusi kinematika balik untuk menentukan posisi dan orientasi dari *end effector* manipulator robot menuju ke titik koordinat yang telah di tentukan.

1.3. Tujuan

Tujuan yang ingin dicapai pada tugas akhir ini adalah dapat menyelesaikan kinematika balik manipulator robot denso 6-DoF menggunakan metode *neuro fuzzy*, sehingga manipulator robot denso dapat bergerak ke titik koordinat yang ditentukan. Manfaat dari tugas akhir ini agar penggunaan dari manipulator robot dapat di perbanyak.

1.4. Batasan Masalah

Dalam pembuatan Tugas Akhir ini, penulis membuat batasan dalam pengerjaan, yaitu:

1. Hanya membahas kinematika manipulator robot Denso 6-DoF tanpa pembahasan kecepatan dan dinamika.
2. Kinematika yang dibahas melingkupi posisi (X, Y dan Z) dan orientasi (*roll* dan *pitch*).
3. Kinematika balik hanya mencari besar sudut lima sendi awal.

1.5. Metodologi

Pada tugas akhir ini akan diberi masukan titik koordinat yang dituju oleh *end effector* manipulator robot dan keluarannya akan dibandingkan dengan hasil dari kinematika maju apakah titik koordinat yang dituju benar dengan memasukkan sudut *joint* yang telah didapat. Kinematika maju yang akan dibandingkan dengan kinematika balik telah dilakukan pengujian dengan membandingkan hasil keluaran dari perhitungan manual dengan hasil keluaran pada simulasi manipulator robot. Tahap-tahap yang akan dilakukan untuk menyelesaikan tugas akhir ini sebagai berikut.

1. Studi Literatur

Pada tahap ini akan dilakukan pembelajaran tentang robotika, mulai dari bagian-bagian dari manipulator robot dan cara kerja dari manipulator robot. Kemudian akan mempelajari tentang kinematika manipulator robot, kinematika yang dipelajari terlebih dahulu yaitu kinematika maju setelah mempelajari kinematika maju akan mempelajari kinematika balik. Kemudian akan mempelajari metode *Neuro Fuzzy* yang digunakan sebagai solusi penyelesaian kinematika balik.

2. Mencari Persamaan Kinematika Maju

Pada tahap ini akan dicari parameter-parameter dari manipulator robot denso 6-DoF yang akan digunakan untuk merancang kinematika maju. Kinematika maju berfungsi untuk menentukan posisi awal dan posisi yang akan dituju oleh *end effector*. Persamaan kinematika maju akan digunakan untuk mengkaji kinematika balik.

3. Menyelesaikan Kinematika Balik

Pada tahap ini akan dirancang kinematika balik yang akan digunakan untuk mendapatkan besaran sudut dari setiap *joint* sehingga manipulator robot dapat bergerak menuju titik koordinat yang akan dituju oleh manipulator robot. Di sini metode *neuro fuzzy*

digunakan untuk mendapatkan solusi kinematika balik agar manipulator robot dapat menuju titik koordinat tujuan.

4. Simulasi dan Pengujian Metode *Neuro Fuzzy*

Pada tahap ini telah mendapatkan solusi kinematika balik, maka tahap selanjutnya akan melakukan simulasi dan pengujian dari *neuro fuzzy*. Pada tahap ini melakukan simulasi menggunakan *software* Matlab. Pengujian dilakukan dengan cara menggerakkan manipulator robot dari titik koordinat awal ke titik koordinat tujuan dan memastikan manipulator robot dapat bergerak menuju titik koordinat tujuan. Perbedaan dari posisi yang diinginkan dengan posisi manipulator robot akan diukur untuk mendapatkan performa dari metode *neuro fuzzy*. Simulasi dan pengujian pada tahap ini bertujuan untuk memastikan apakah metode *neuro fuzzy* menjadi solusi kinematika balik.

5. Penyusunan Buku Tugas Akhir

Pada tahap ini akan menyusun laporan dari hasil proses yang dilakukan selama tugas akhir. Proses-proses yang dilakukan selama tugas akhir akan disusun menjadi satu buku, yang mana penyusunan buku tugas akhir dilakukan sebagai bentuk dari laporan yang tertulis dari hasil yang didapat dari proses-proses yang terkait dengan topik yang diusulkan

1.6. Sistematika

Penulisan Tugas Akhir ini disusun berdasarkan lima bab yang tiap bagiannya membahas permasalahan yang berhubungan dengan Tugas Akhir.

Bab I : Pendahuluan

Bab ini membahas tentang latar belakang, perumusan masalah, batasan masalah, maksud dan tujuan, sistematika laporan, serta relevansi tugas akhir yang dibuat.

Bab II : Tinjauan Pustaka

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

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 *Neuro fuzzy* 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 *Neuro Fuzzy* melalui *software* Matlab 2017.

Bab IV : Pengujian dan Analisa

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

Bab V : Penutup

Bab ini berisi penutup yang menjelaskan tentang kesimpulan yang didapat dari tugas akhir ini dan saran-saran dari hasil penelitian yang dilakukan.

1.7. Relevansi atau Manfaat

Hasil yang diperoleh dari tugas akhir ini diharapkan dapat memberi manfaat berikut:

1. Mendapatkan solusi kinematika balik untuk mencari besar sudut sendi dari masukan titik acuan yang telah ditentukan
2. Mengembangkan teknologi robotika di bidang sistem kontrol.

[Halaman ini sengaja dikosongkan]

BAB 2

TINJAUAN PUSTAKA

Pada bab ini membahas dasar-dasar teori yang dapat menunjang tugas akhir ini, yaitu mengenai bagian-bagian dan konfigurasi dari manipulator robot, kinematika, dan metode *neuro fuzzy* untuk menyelesaikan kinematika balik manipulator robot denso 6-DoF.

2.1. Robotika

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" [4]. Cabang ilmu yang memiliki hubungan dengan desain, konstruksi, pembuatan dan aplikasi sebuah robot merupakan robotika. Robotika sangat sering digunakan di dunia industri. Aplikasi dari bidang robotika yang sering digunakan di dunia industri adalah manipulator robot. Manipulator robot adalah mekanik elektronik yang menyerupai sebuah lengan manusia, manipulator robot juga sering dikenal ataupun disebut dengan lengan robot.

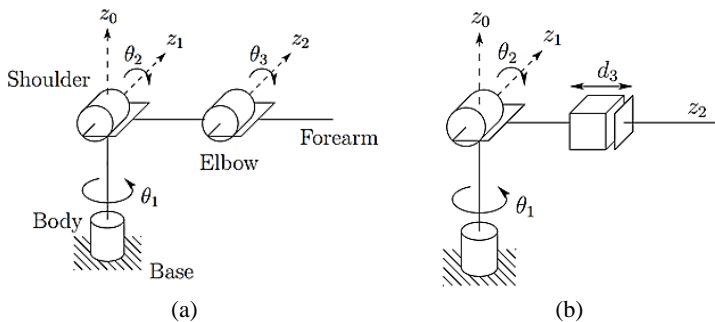
Manipulator robot berguna untuk membantu melakukan pekerjaan-pekerjaan di tempat yang berbahaya, pekerjaan-pekerjaan yang berat secara berulang-ulang, pekerjaan-pekerjaan teliti secara berulang-ulang, dan sebagainya. Sehingga penggunaan manipulator robot sangat diperlukan.

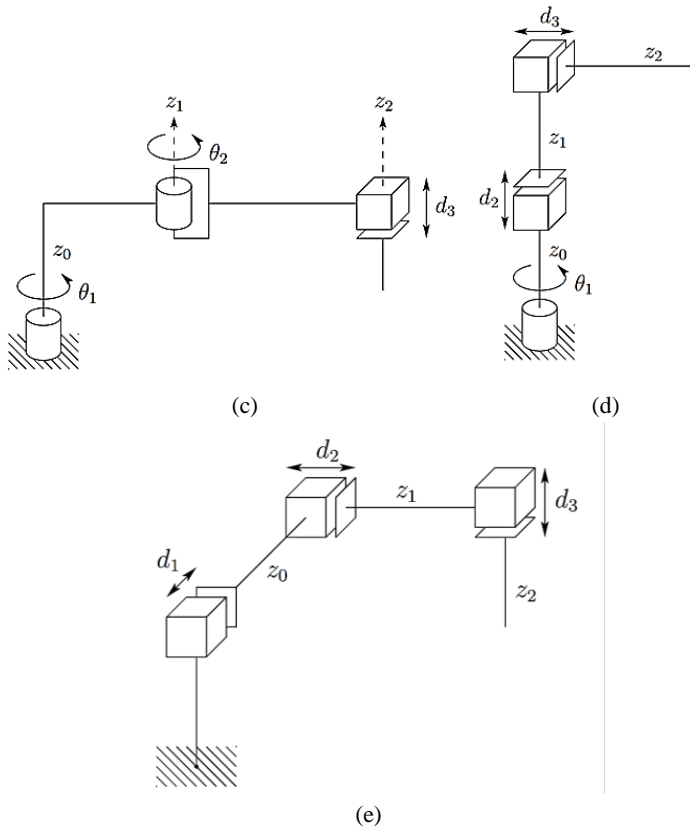
Manipulator robot bergerak menuju suatu titik koordinat yang akan dituju yang telah diprogram menggunakan komputer, untuk mengendalikan pergerakan dari manipulator robot diperlukan studi tentang kinematika. Kinematika manipulator robot merupakan studi analitis dari pergerakan manipulator robot terhadap kerangka tanpa memperhatikan gaya yang dapat mempengaruhi pergerakan dari manipulator robot. Pada tugas akhir ini kinematika yang akan digunakan yaitu kinematika balik, pada kinematika balik akan ditentukan titik koordinat tertentu yang akan dituju oleh manipulator robot, kemudian *joint* dari manipulator robot akan bergerak dengan besaran sudut tertentu, sehingga manipulator robot dapat bergerak menuju ke titik koordinat yang telah ditentukan.

Pergerakan dari manipulator robot di bagi menjadi dua pergerakan. Pertama, ialah pergerakan posisi yang mana pada pergerakan posisi terdapat tiga parameter, yaitu maju/mundur, kanan/kiri dan atas/bawah. Kedua pergerakan orientasi, yang mana pergerakan orientasi juga dibagi menjadi tiga parameter yaitu *roll*, *pitch* dan *yaw*. Manipulator robot yang akan digunakan pada tugas akhir ini memiliki jumlah *joint* sebanyak 6, semua *joint* pada manipulator berjenis *revolute*. Manipulator dengan jenis *revolute* pada setiap *joint* nya disebut dengan manipulator robot berkonfigurasi *articulated*. Karena manipulator robot memiliki jumlah DoF sebanyak 6 membuat pergerakannya memiliki banyak kemungkinan konfigurasi untuk menuju ke titik koordinat yang telah di tentukan.

2.2. Konfigurasi Manipulator Robot

Kebanyakan manipulator industri saat ini memiliki enam atau lebih sedikit derajat kebebasan. Manipulator ini biasanya diklasifikasikan secara kinematis berdasarkan tiga sendi pertama lengan, dengan pergelangan lengan yang dijelaskan secara terpisah. Mayoritas manipulator ini jatuh ke dalam salah satu dari lima jenis geometris: *articulated (Revolute Revolute Revolute)*, *Spherical (Revolute Revolute Prismatic)*, *SCARA (Revolute Revolute Prismatic)*, *Cylindrical (Revolute Prismatic Prismatic)*, dan *Cartesian (Prismatic Prismatic Prismatic)* [4]. Contoh jenis-jenis konfigurasi manipulator dapat dilihat pada gambar 2.1.





Gambar 2.1 (a) Konfigurasi *Articulated (RRR)* (b) Konfigurasi *Spherical (RRP)* (c) Konfigurasi *SCARA (RRP)* (d) Konfigurasi *Cylindrical (RPP)* (e) Konfigurasi *Cartesian (PPP)*.

2.3. Jenis Sendi Manipulator Robot

Manipulator robot denso memiliki sendi yang berfungsi untuk menghubungkan dua lengan. Terdapat dua jenis sendi pada manipulator robot, dua jenis tersebut ialah sendi putar (*revolute*) dan sendi geser (*prismatic*).

2.3.1. Sendi Putar (*Revolute*)

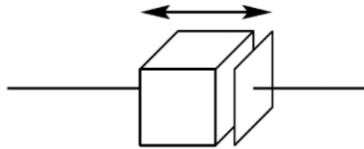
Sendi putar atau biasa di kenal *revolute* merupakan jenis sendi yang dapat bergerak secara berputar atau berotasi dengan batasan sudut tertentu. Bentuk dari sendi putar sendiri dapat dilihat pada Gambar 2.2.



Gambar 2.2 Sendi Putar [4]

2.3.2. Sendi Geser (*Prismatic*)

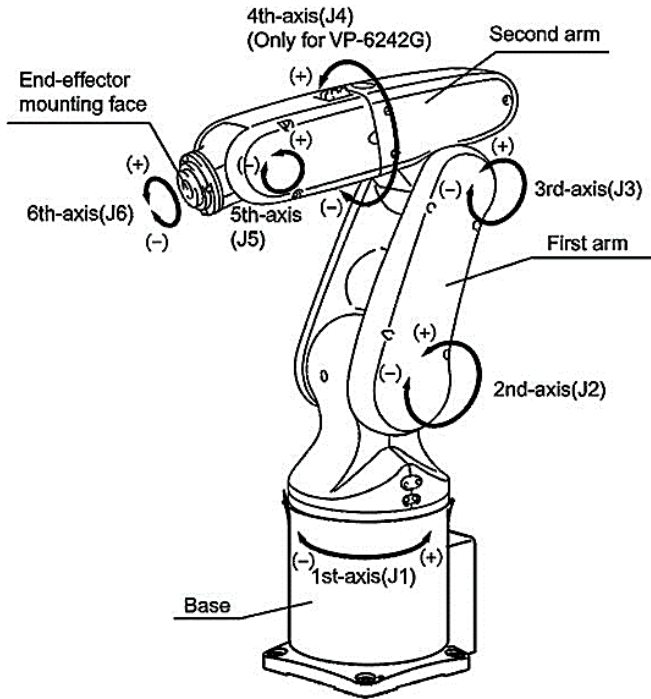
Sendi geser atau biasa di kenal *prismatic* merupakan jenis sendi yang dapat bergerak secara translasi atau bergeser dengan batasan jarak tertentu. Bentuk dari sendi geser sendiri dapat dilihat pada Gambar 2.3.



Gambar 2.3 Sendi Geser [4]

2.4. Robot Denso

Robot denso yang akan digunakan merupakan robot denso yang memiliki 6-DoF dengan tipe VP-6242G. Robot denso dengan tipe tersebut memiliki jenis sendi yang sama di keenam sendinya yaitu berjenis sendi putar (*revolute*). Pergerakan posisi dari *end effector* pada robot ini dipengaruhi oleh 3 sendi awal, sedangkan orientasi *end effector* pada robot ini dipengaruhi oleh 3 sendi selanjutnya. Bentuk robot denso 6-DoF dengan tipe VP-6242G dapat dilihat pada Gambar 2.4 [6].



Gambar 2.4 Robot Denso VP-6242G

2.5. Transformasi Homogen

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

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

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

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

$$Rot_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha & -s\alpha & 0 \\ 0 & s\alpha & c\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots (2.4)$$

$$Rot_{y,\phi} = \begin{bmatrix} c\phi & 0 & s\phi & 0 \\ 0 & 1 & 0 & 0 \\ -s\phi & 0 & c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots (2.5)$$

$$Rot_{z,\theta} = \begin{bmatrix} c\theta & -s\theta & 0 & 0 \\ s\theta & c\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots (2.6)$$

Matriks translasi dan rotasi tersebut akan membentuk sebuah matriks transformasi homogen 4x4, yang dapat di lihat pada persamaan 2.7. matriks homogen ini berfungsi untuk digunakan pada perhitungan kinematika maju manipulator robot.

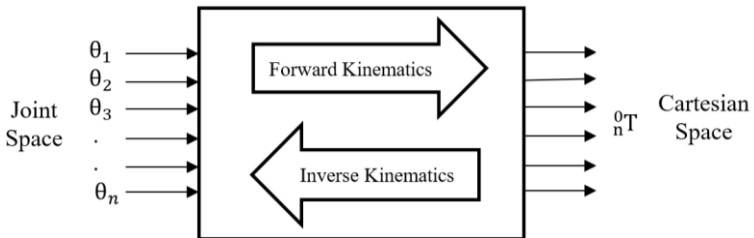
$$H_1^0 = \begin{bmatrix} R_{3x3} & P_{3x1} \\ F_{1x3} & S_{1x1} \end{bmatrix} = \begin{bmatrix} Rotation & Translation \\ Perspective & Scale \end{bmatrix} \dots\dots (2.7)$$

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

Pada persamaan 2.8 n , s , dan a merupakan vektor yang mempresentasikan dari orientasi robot manipulator sedangkan d merupakan vektor yang mempresentasikan posisi dari manipulator robot.

2.6. Kinematika Robot

Kinematika robot merupakan studi analitis dari pergerakan manipulator robot terhadap sistem kerangka yang diam ataupun bergerak tanpa memperhatikan gaya yang akan mempengaruhi pergerakannya. Kinematika dibagi menjadi dua bagian yaitu kinematika maju dan kinematika balik hubungan antara dua bagian dari kinematika tersebut dapat dilihat pada Gambar 2.5.



Gambar 2.5 Cara Kerja Kinematika Maju dan Kinematika Balik [8]

Pada kinematika maju akan diberi masukan berupa besaran sudut masing-masing joint dari manipulator robot, sehingga manipulator robot dapat bergerak menuju titik koordinat tertentu pada ruang cartesian x, y, dan z. Sedangkan pada kinematika balik diberi masukan berupa titik koordinat tertentu pada ruang cartesian x, y, dan z, dan masing-masing joint dari manipulator robot akan bergerak dengan besaran sudut tertentu, sehingga manipulator robot dapat bergerak menuju ke titik koordinat yang telah ditentukan.

2.7. Kinematika Maju

Kinematika maju berfungsi untuk menentukan posisi dan orientasi dari *end effector* manipulator robot apabila variabel sudut dari setiap *joint* telah diketahui. Variabel sudut dari *joint* yang di berikan akan diubah ke dalam posisi dan orientasi dari *end effector* yang ditujukan sebagai koordinat refrensi. Orientasi menyatakan rotasi *end effector* terhadap *base frame* sedangkan koordinat posisi menyatakan letak dari *end effector* terhadap *base frame*. Metode yang paling umum agar dapat mencari persamaan kinematika maju adalah dengan menggunakan metode *Denavit Hartenberg* (DH).

Denavit Hartenberg (DH) memiliki aturan yang mana berisi langkah-langkah yang mudah untuk mencari nilai dari kinematika maju. Pada aturan ini menyatakan bahwa setiap matriks transformasi homogen dinyatakan sebagai perkalian dari empat transformasi dasar yang melibatkan *joint* dan *link* manipulator robot. Berikut ini merupakan parameter *link* yang berfungsi dalam kaidah *Denavit Hartenberg* (DH) sebagai berikut [4] :

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 sendi *prismatic*.

α_i : Sudut antara z_{i-1} dari z_i di ukur terhadap x_i .

θ_i : Sudut antara x_{i-1} dari x_i di ukur terhadap z_i . θ_i adalah variabel jika sendi i adalah *revolute*.

Langkah-langkah untuk memperoleh variabel dari kinematika maju jika menggunakan aturan *Denavit Hartenberg* (DH) [4] adalah sebagai berikut :

Langkah 1 : Menempatkan dan memberikan label pada sendi berupa sumbu z_0, \dots, z_{n-1}

Langkah 2 : Menetapkan *base frame*. Menentukan origin pada sumbu z_0 . Sumbu x_0 dan y_0 dipilih secara acak untuk membentuk *right-hand frame*.

Langkah 3 : Menempatkan origin o_i ke z_i dan z_{i-1} memotong z_i .

Langkah 4 : Menetapkan x_i sepanjang *common normal* antara z_{i-1} dan z_i melalui o_i .

Langkah 5 : Menetapkan y_i untuk melengkapi *right-hand frame*.

Langkah 6 : Menetapkan *end effector frame* pada $o_n x_n y_n z_n$.

Langkah 7 : Membuat sebuah tabel dari parameter link $a_i, d_i, \alpha_i, \theta_i$.

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.

$$\begin{aligned}
 A_i &= Rot_{z, \theta_i} Trans_{z, d_i} Trans_{x, \alpha_i} Rot_{x, \alpha_i} \\
 &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -s_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.9}
 \end{aligned}$$

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

$$T_0^n = A_1 \dots A_n \tag{2.10}$$

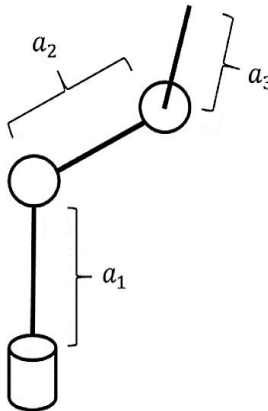
2.8. Kinematika Balik

Kinematika balik berfungsi untuk mengetahui besaran dari sudut-sudut tiap *joint* manipulator robot ketika orientasi dan posisi dari *end effector* manipulator robot terhadap sumbu *base* telah diketahui. Secara garis besar kinematika balik akan mencari nilai-nilai parameter yang harus diberikan kepada setiap aktuator untuk mencapai tujuan akhir. Pada umumnya perhitungan penyelesaian kinematika balik lebih sulit jika dibandingkan dengan perhitungan penyelesaian kinematika maju. Karena permasalahan pada kinematika balik adalah mencari besaran sudut dari setiap *joint* manipulator robot berdasarkan titik koordinat tujuan yang telah ditentukan di awal, sehingga kinematika balik memiliki hasil akhir unik. Dikatakan unik dikarenakan hasil dari setiap sudut memiliki banyak

solusi tergantung dari banyaknya joint yang terdapat pada manipulator robot. Untuk menyelesaikan kinematika balik terdapat dua metode, metode yang pertama adalah pendekatan secara geometri dan metode yang kedua adalah pendekatan secara numerik.

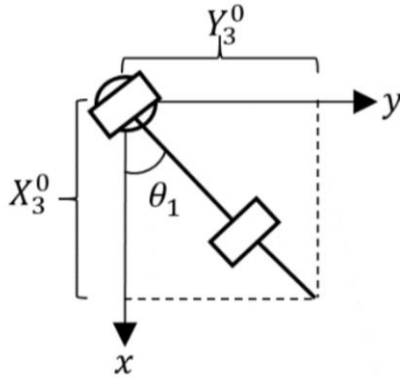
2.8.1. Pendekatan Geometri

Penyelesaian kinematika balik dengan menggunakan pendekatan geometri merupakan penyelesaian kinematika balik dengan menggunakan pengkomposisian sebuah ruang geometri dan hukum-hukum trigonometri. Contoh dari penyelesaian kinematika balik dengan digunakannya pendekatan geometri dapat dilihat pada Gambar 2.6. Dimana pada Gambar 2.6 akan terdapat manipulator robot dengan jumlah sendi 3. Dengan digunakannya kinematika balik akan diperoleh beberapa persamaan dari masing-masing sendi untuk mendapatkan besar sudut masing-masing sendi jika diketahui posisi yang diinginkan.



Gambar 2.6 Manipulator Robot 3 Sendi

Terdapat dua tahap untuk dapat mencari persamaan kinematika balik, tahap yang pertama ialah melihat manipulator dari sisi atas. Pada Gambar 2.7 dapat dilihat manipulator robot dari sisi atas.

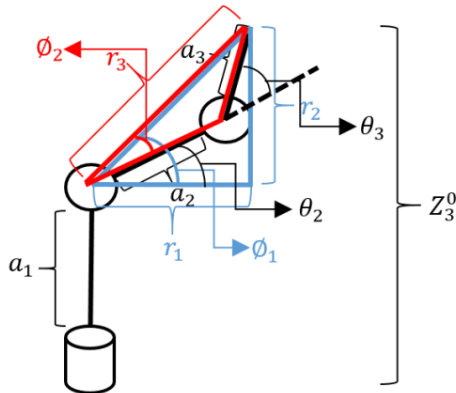


Gambar 2.7 Manipulator Robot 3 Sendi dari sisi atas

Dari Gambar 2.7 dapat didapatkan persamaan kinematika balik pada sendi 1 yang terdapat pada Persamaan 2.11.

$$\theta_1 = \tan^{-1} \left(\frac{y_3^0}{x_3^0} \right) \dots\dots\dots (2.11)$$

Pada tahap yang kedua ialah melihat manipulator dari sisi samping. Pada Gambar 2.8 dapat dilihat manipulator robot dari sisi samping.



Gambar 2.8 Manipulator Robot 3 Sendi dari sisi samping

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

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

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

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

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

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

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

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

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

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

Dari Gambar 2.8 didapatkan persamaan kinematika balik pada sendi 2 dan sendi 3 pada Persamaan 2.20 dan 2.21.

2.8.2. Pendekatan Numerik

Pada umumnya di dalam pendekatan numerik tidak mengutamakan diperoleh solusi yang tepat, tetapi mengusahakan perumusan sebuah metode yang menghasilkan solusi pendekatan yang berbeda dari solusi yang tepat sebesar suatu nilai yang dapat diterima. Pendekatan numerik biasanya memanfaatkan perangkat komputer untuk melakukan perhitungan secara berulang – ulang agar memperoleh sebuah solusi kinematika balik. Komputer akan menghitung semua kemungkinan solusi secara berulang–ulang hingga diperoleh suatu solusi yang sesuai untuk sudut–sudut pada tiap sendi yang dibutuhkan agar bisa mencapai posisi dan orientasi dari *end effector* manipulator robot yang diinginkan.

2.9. Logika Fuzzy

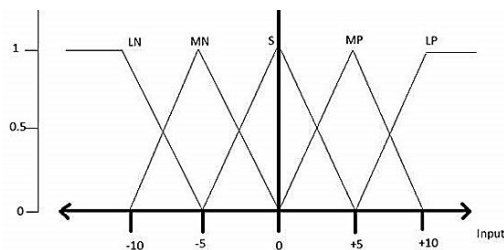
Logika *fuzzy* pertama kali diperkenalkan dan dikembangkan oleh Lotfi A. Zadeh pada tahun 1965 tentang teori himpunan *fuzzy*. Logika *fuzzy* merupakan salah satu bentuk pendekatan dimana representasi suatu kejadian di distribusikan kedalam sejumlah istilah bahasa (dinyatakan dala level kualitatif) dengan nilai kebenaran yang terletak antara 0 sampai 1. Konsep logika *fuzzy* menggantikan konsep “benar-salah” dari logika *boolean* menjadi derajat tingkat kebenaran. Oleh karena itu, konsep logika *fuzzy* sesuai dengan pola pikir manusia yang cenderung menilai. Logika fuzzy juga diterapkan pada banyak bidang mulai dari bidang kendali (*control*) sampai bidang kecerdasan buatan (*artificial intelligence*).

2.9.1. Himpunan Fuzzy

Himpunan fuzzy merupakan rentang nilai antara salah dan benar untuk setiap anggotanya atau suatu himpunan yang beranggotakan sejumlah istilah dalam pengertian bahasa yang menyatakan level kualitatif dari semesta pembicaraan. Misalnya semesta pembicaraan untuk kecepatan dapat dibuat dalam bentuk suatu himpunan *fuzzy* yaitu “Sangat Lambat”, “Lambat”, “Sedang”, “Cepat” dan “Sangat Cepat”.

2.9.2. Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik–titik input ke dalam tingkat keanggotaan dengan nilai antara 0-1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Fungsi keanggotaan suatu himpunan *fuzzy* dapat ditentukan dengan fungsi segitiga, trapesium atau fungsi Gaussian. Gambar 2.9 merupakan contoh dari fungsi keanggotaan dari *fuzzy* dengan kurva segitiga.



Gambar 2.9 Struktur fungsi keanggotaan fuzzy

2.9.3. Fuzzifikasi

Fuzzifikasi merupakan proses pengubahan nilai masukan menjadi nilai *fuzzy* yang memiliki nilai antara 0 s/d 1. Aktivasi dari nilai *fuzzy* dilakukan dengan menggunakan fungsi keanggotaan *fuzzy*. Fungsi keanggotaan mendefinisikan nilai *fuzzy* dengan dilakukannya pemetaan nilai tegas berdasarkan daerahnya untuk diasosiasikan dengan derajat keanggotaan *fuzzy*. Jumlah dari fungsi keanggotaan *fuzzy* akan berpengaruh pada keluaran *fuzzy*. Fungsi yang umum digunakan adalah fungsi segitiga, fungsi Gauss dan fungsi trapesium. Fungsi keanggotaan segitiga dituliskan pada persamaan 2.22.

$$\mu(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & c \leq x \end{cases} \dots\dots\dots (2.22)$$

Dimana x merupakan nilai dari masukan pada fungsi keanggotaan, a dan c menyatakan nilai dari batas segitiga sedangkan b menyatakan nilai puncak.

Fungsi keanggotaan Gauss adalah fungsi eksponensial dengan pangkat bernilai negatif. Persamaan fungsi keanggotaan Gauss dituliskan pada persamaan 2.23.

$$\mu(x) = e^{-\frac{(x-c)^2}{2\sigma^2}} \dots\dots\dots (2.23)$$

Dimana c merupakan titik puncak dan σ merupakan lebar fungsi Gauss. Fungsi keanggotaan trapesium dituliskan pada persamaan 2.24.

Dimana a dan b adalah batas terluar dari trapesium dengan nilai

$$\mu(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \dots\dots\dots (2.24) \\ \frac{c-x}{c-b} & c \leq x \leq d \\ 0 & d \leq x \end{cases}$$

derajat keanggotaan 0, dan b sampai c adalah batas dari trapesium dengan nilai derajat keanggotaan 1.

2.9.4. Sistem Inferensi Fuzzy

Mekanisme inferensi *fuzzy* merupakan sebuah proses pengambilan keputusan dari masing-masing premis yang sesuai dengan aturan *fuzzy*. Oleh karena itu dibutuhkan basis aturan *fuzzy* untuk memetakan masukan *fuzzy* yang diperoleh dari masukan proses *fuzzifikasi* terhadap keluaran *fuzzy*. Terdapat beberapa tipe mekanisme inferensi *fuzzy* antara lain Mamdani, Larsent dan Takagi sugeno. Perbedaan dari beberapa metode ini terletak pada pengambilan kesimpulan logika *fuzzy*. Dalam metode Mamdani maupun Larsent, penarikan kesimpulan logika fuzzy berupa derajat keanggotaan sehingga untuk menyimpulkan suatu logika *fuzzy* dibutuhkan proses *defuzzifikasi* sedangkan pada Takagi Sugeno, pengambilan kesimpulan logika fuzzy berupa persamaan sehingga tidak diperlukan sebuah proses *defuzzifikasi*.

2.9.5. Defuzzifikasi

Defuzzifikasi adalah proses mentransformasikan nilai *fuzzy* hasil dari proses inferensi *fuzzy* ke dalam suatu nilai yang tegas. Salah satu proses *defuzzifikasi* yang sering digunakan adalah model Mamdani. Beberapa metode *defuzzifikasi* dengan model Mamdani diantaranya *center of gravity* (COG) dan *center of average* (COA). Metode *center of average* (COA) dituliskan dalam persamaan 2.25.

$$u = \frac{\sum_{j=1}^n b_j \cdot y_j}{\sum_{j=1}^n y_j} \dots\dots\dots (2.25)$$

Metode *center of gravity* (COG) dituliskan dalam persamaan 2.26.

$$u = \frac{\sum_{j=1}^n b_j \int y_j}{\sum_{j=1}^n \int y_j} \dots\dots\dots (2.26)$$

Dimana *b* sebagai titik tengah dan *y* sebagai fungsi keanggotaan.

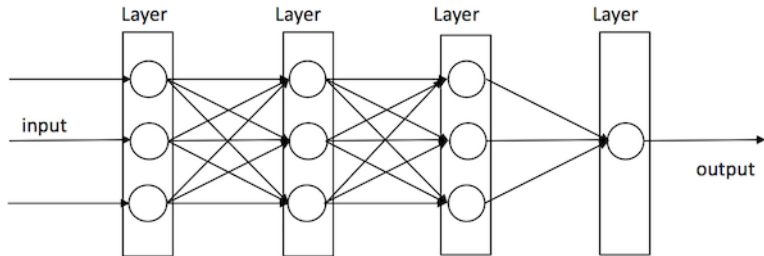
2.10. Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) adalah salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran. Istilah buatan disini digunakan karena jaringan syaraf ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran. Proses komputasi pada jaringan syaraf tiruan ini dipelajari dari struktur dan cara kerja otak manusia. Pada jaringan syaraf otak manusia, informasi disalurkan dari satu *neuron* ke *neuron* lainnya. Sementara pada jaringan syaraf tiruan proses penyaluran informasi dari satu *neuron* ke *neuron* lainnya diimplementasikan pada program komputer. Proses pelatihan jaringan syaraf tiruan, umumnya menggunakan metode pelatihan *backpropagation* yang sudah banyak diterapkan pada semua proses pelatihan yang sederhana sampai yang rumit.

Metode pelatihan *backpropagation* termasuk ke dalam metode pelatihan terawasi (*supervisory learning*). *Supervisory learning* adalah metode pelatihan yang memasukan target ke *output* dalam data untuk proses pelatihnnya. Metode *backpropagation* banyak diaplikasikan dalam berbagai proses karena metode ini didasarkan pada interkoneksi yang sederhana. Apabila *output* jaringan syaraf tiruan tidak sesuai dengan *output* yang diinginkan, maka metode *backpropagation* akan memperbaiki nilai bobot (*weight*) yang ada pada lapisan tersembunyi

(*hidden layer*) untuk mencapai ke *output* jaringan syaraf tiruan yang sesuai dengan target ke *output*.

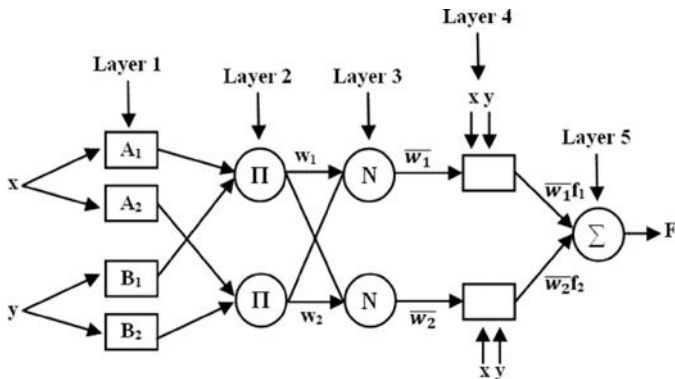
Pada dasarnya jaringan syaraf tiruan akan diberikan pola masukan sebagai pola pelatihan maka pola akan menuju ke unit-unit lapisan tersembunyi (*hidden layer*) dan akan diteruskan ke lapisan *output*. Struktur jaringan syaraf tiruan dapat dilihat pada Gambar 2.10.



Gambar 2.10 Struktur *Neural Network*

2.11. *Neuro-Fuzzy*

Neuro-fuzzy adalah sistem hibrida atau gabungan dari dua system yaitu sistem logika fuzzy dan jaringan syaraf tiruan. Pada *neuro fuzzy*, logika fuzzy akan direpresentasikan dalam jaringan syaraf tiruan yang memungkinkan terjadinya pembelajaran di dalamnya dan akan terjadi perubahan bobot untuk mengubah parameter pada logika fuzzy [1].



Gambar 2.11 Struktur Kontroler *Neuro Fuzzy*

Terdapat dua tahapan pada struktur *neuro fuzzy* tahap pertama yaitu *forward propagation* dan tahap kedua *backward propagation*.

2.11.1. Forward Propagation

Feedforward Propagation adalah tahapan yang terdiri dari 5 lapisan yang di mana setiap lapisan berisi proses-proses perhitungan *fuzzy* [2].

Lapisan 1 :

Tiap-tiap *neuron i* pada lapisan pertama adaptif terhadap parameter suatu fungsi aktivasi. *Output* dari tiap neuron berupa derajat keanggotaan yang diberikan oleh fungsi keanggotaan *input*. Fungsi simpul pada lapisan ini adalah :

$$O_{1,i} = \mu A_i(x) \text{ untuk } i = 1,2 \dots\dots\dots (2.27)$$

$$O_{1,i} = \mu B_i(y) \text{ untuk } i = 1,2 \dots\dots\dots (2.28)$$

Lapisan 2 :

Pada tahap ini berlaku operasi perkalian (AND) antara nilai fuzifikasi dari titik yang ada sebelumnya sebagai proses implikasi *fuzzy*. Persamaan keluaran pada simpul tetap ini adalah :

$$O_{2,i} = w_i = \mu A_i(x) \cdot \mu B_i(y) \text{ untuk } i = 1,2 \dots\dots\dots (2.29)$$

Lapisan 3 :

Setiap *neuron* pada lapisan ini adalah simpul tetap yang merupakan hasil perhitungan rasio dari aturan derajat keanggotaan ke-*i* dengan jumlah dari seluruh aturan derajat keanggotaan, sehingga dapat dirumuskan sebagai berikut :

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, i = 1,2 \dots\dots\dots (2.30)$$

Lapisan 4 :

Setiap *neuron* pada lapisan ini merupakan simpul adaptif terhadap suatu *output* :

$$O_{4,i} = \bar{w}_i \cdot f_i = \bar{w}_i (p_i x_1 + q_i x_2 + r_i) \dots\dots\dots (2.31)$$

Dengan \bar{w}_i adalah *normalized firing strength* pada lapisan ke-3 dan { p_i, q_i, r_i } adalah parameter- parameter pada *neuron* tersebut. Parameter-parameter pada lapisan 4 disebut dengan *consequent parameter*.

Lapisan 5 :

Lapisan 5 merupakan *output layer*. Setiap neuron pada lapisan ini merupakan simpul tunggal tetap yang menghitung keluaran dengan cara menjumlahkan semua masukan.

$$O_{5,i} = \sum_i \bar{w}_i f_i = \frac{\sum_i \bar{w}_i f_i}{\sum_i f_i} \dots\dots\dots (2.32)$$

2.11.2. Backward Propagation

Backward propagation merupakan proses *learning* terhadap kesalahan atau *error* yang muncul, proses ini berfungsi untuk meng-*update* nilai *consequent parameter* pada lapisan 4 atau titik–titik puncak *defuzzifikasi* pada lapisan 5, berdasarkan laju pembelajaran dan nilai *error* yang terjadi. Proses pembelajaran ini dapat dilakukan pada semua lapisan atau pada salah satu lapisan saja [2].

$$f_i = f(i-1) + lr * e * w_i \dots\dots\dots (2.33)$$

- f_i = *consequent parameter* baru
- $f(i-1)$ = *consequent parameter* lama
- lr = laju pembelajaran
- e = *error (reference-output defuzzifikasi)*
- w_i = *output* lapisan 3 ternormalisasi

[Halaman ini sengaja dikosongkan]

BAB 3

PERANCANGAN SISTEM

Pada Bab tiga ini akan membahas hal-hal mengenai perancangan penyelesaian kinematika balik manipulator robot denso dengan tipe VP-6242G yang meliputi menentukan parameter DH, transformasi *homogen* tiap *link* manipulator robot, persamaan kinematika maju, perancangan struktur *neuro fuzzy* dan simulasi dengan *software* Matlab 2017 yang telah ditambahkan *toolbox Peter Corke* versi 10.

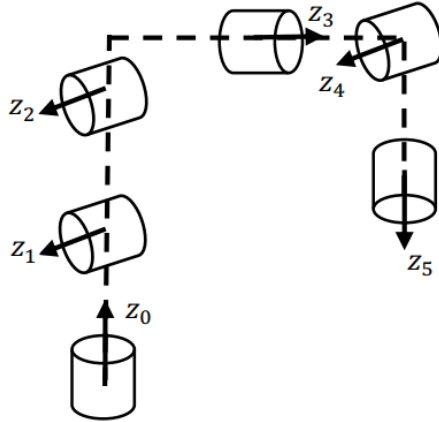
3.1. Parameter Denavit Hartenberg

Sebelum merancang penyelesaian kinematika balik hal yang terlebih dahulu dilakukan yaitu melakukan perancangan dan perhitungan dari kinematika maju. Untuk mencari persamaan kinematika maju akan dilakukan pencarian menggunakan aturan parameter DH. Letak masing-masing sendi dan panjang masing-masing lengan dapat dilihat pada Gambar 3.1.



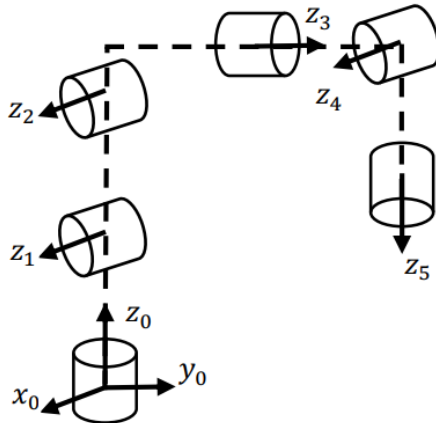
Gambar 3.1 Panjang Lengan Robot Denso VP-6242G [6]

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 [4]. Pemberian label sumbu z dapat dilihat pada Gambar 3.2.



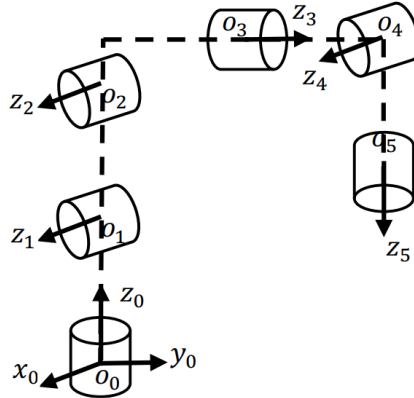
Gambar 3.2 Pemberian Lembel Sumbu z [3]

Langkah 2: menetapkan base frame. Menentukan origin pada sumbu z_0 . Sumbu x_0 dan y_0 dipilih secara acak sesuai kaidah tangan kanan [4]. Pemberian label sumbu x_0 dan y_0 dapat dilihat pada Gambar 3.3.



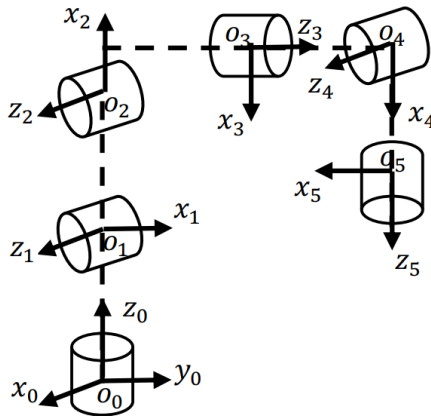
Gambar 3.3 Pemberian Lembel Sumbu x_0 dan y_0 [3]

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 [4]. Pemberian Label Sumbu o_i dapat dilihat pada Gambar 3.4.



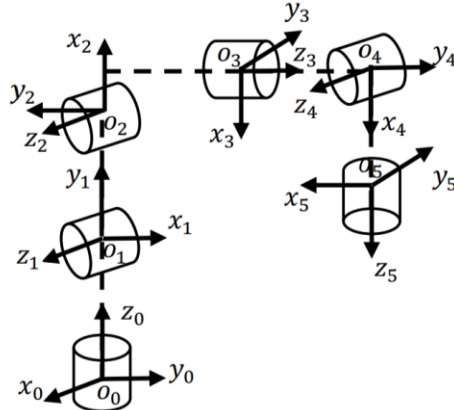
Gambar 3.4 Pemberian Lembel o_i [3]

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 [4]. Pemberian Label Sumbu x_i dapat dilihat pada Gambar 3.5.



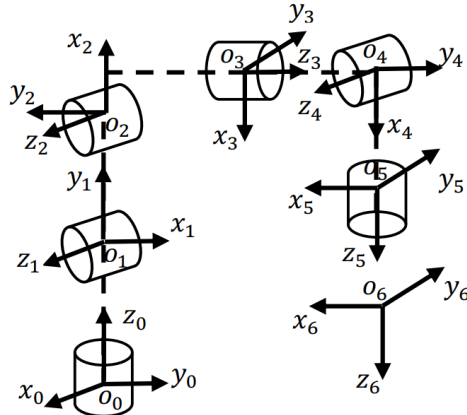
Gambar 3.5 Pemberian Lembel Sumbu x_i [3]

Langkah 5: menetapkan y_i (untuk $i=1$ sampai dengan 5) dengan aturan kaidah tangan kanan [4]. Pemberian Label Sumbu y_i dapat dilihat pada Gambar 3.6.



Gambar 3.6 Pemberian Lembel Sumbu y_i [3]

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 [4]. Pemberian Label Sumbu o_n , x_n , y_n , dan z_n dapat dilihat pada Gambar 3.7.



Gambar 3.7 Pemberian Lembel o_n , x_n , y_n dan z_n . [3]

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

Agar dapat mempermudah perhitungan, parameter d_i dan a_i yang terdapat di Tabel 3.1 (tabel parameter dh) akan di ganti dengan variabel. Sedangkan untuk variabel bertanda (*) yang terdapat 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. Transformasi Homogen

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

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

$$A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{90} & -s_{90} & 0 \\ 0 & s_{90} & c_{90} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots \quad (3.1)$$

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

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

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

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

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

$$A_3 = \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{-90} & -s_{-90} & 0 \\ 0 & s_{-90} & c_{-90} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} (3.3)$$

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

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

$$A_4 = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{90} & -s_{90} & 0 \\ 0 & s_{90} & c_{90} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots (3.4)$$

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

$$A_5 = Rot_{z,\theta_5} Rot_{x,\alpha_5}$$

$$A_5 = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{-90} & -s_{-90} & 0 \\ 0 & s_{-90} & c_{-90} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots (3.5)$$

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

$$A_6 = Rot_{z,\theta_6} Trans_{z,a_6}$$

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

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

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

3.3. Persamaan Kinematika Maju

Untuk mendapatkan persamaan kinematika maju dari manipulator robot digunakan hasil persamaan yang didapat dari tranformasi homogen. Hasil dari persamaan kinematika maju merupakan matriks yang berisi nilai posisi dan rotasi dari *end effector* manipulator robot.

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

$$T_0^6 = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 c_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_3 & 0 & -s_3 & a_3 c_3 \\ s_3 & 0 & c_3 & a_3 c_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

$$\begin{bmatrix} c_4 & 0 & s_4 & 0 \\ s_4 & 0 & -c_4 & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_5 & 0 & -s_5 & 0 \\ s_5 & 0 & c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Pada Persamaan (3.7) yang telah di dapatkan dari perkalian A_1 sanampi A_6 didapatkan besar nilai n , s , dan a yang merupakan titik orientasi dan besar nilai p yang merupakan titik posisi dari *end effector* manipulator robot denso 6-DoF.

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

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

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

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

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

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

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

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

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

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

$$\begin{aligned}
 p_y = & d_6 * ((-s_5 * (((s_1 * c_2 * c_3) - (s_1 * s_2 * s_3)) * c_4) + \\
 & (c_1 * s_4))) + (c_5 * ((-s_1 * c_2 * s_3) - (s_1 * s_2 * c_3)))) + \\
 & (((-s_1 * c_2 * s_3) - (s_1 * s_2 * c_3)) * d_4) + (s_1 * c_2 * c_3 * \\
 & a_3) - (s_1 * s_2 * s_3 * a_3) + (s_1 * c_2 * a_2)
 \end{aligned} \tag{3.18}$$

$$\begin{aligned}
 p_z = & d_6 * ((-s_5 * ((s_2 * c_3) + (c_2 * s_3)) * c_4) + (c_5 * (- \\
 & (s_2 * s_3) + (c_2 * c_3)))) + (((-s_2 * s_3) + (c_2 * c_3)) * \\
 & d_4) + (s_2 * c_3 * a_3) + (c_2 * s_3 * a_3) + (s_2 * a_2) + d_1
 \end{aligned} \tag{3.19}$$

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

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

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

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

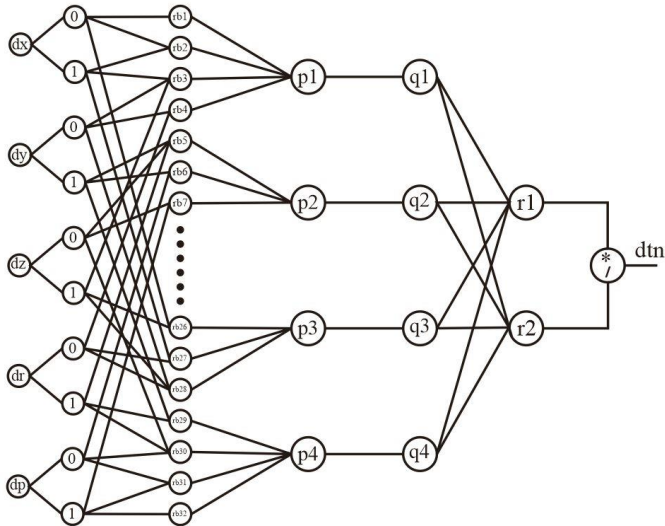
3.4. Solusi Kinematika Balik Menggunakan *Neuro Fuzzy*

Pada tugas akhir ini, akan digunakan metode *neuro fuzzy* agar dapat menyelesaikan permasalahan dari kinematika balik. *Neuro fuzzy* merupakan metode gabungan dari dua sistem, dua system tersebut yaitu system logika *fuzzy* dan jaringan syaraf tiruan. *Neuro fuzzy* memiliki beberapa tipe yaitu *neuro fuzzy* Mamdani, *neuro fuzzy* Larsent, *neuro fuzzy* Takaegi Sugeno dan tipe lainnya. Untuk mendesain struktur *neuro fuzzy* langkah awalnya adalah menentukan masukan dan keluaran yang akan dihasilkan dari *neuro fuzzy* kemudian tahap kedua yang harus dilakukan adalah tahapan menentukan basis aturan atau *rule base*. *Rule base* pada perhitungan kinematika balik didapatkan dari analisa sifat pergerakan manipulator robot denso kemudian tahap selanjutnya yaitu tahap proses normalisasi, tahap *learning* hingga tahap *defuzzyfikasi*. Tahapan-tahapan yang telah di sebutkan sebelumnya merupakan tahapan-tahapan yang masuk dalam proses *forward propagation*.

Jika keluaran dari hasil proses *forward propagation* yang dihasilkan belum sesuai dengan keluaran yang diinginkan maka akan dilakukan proses *backward propagation*. *Backward propagation* merupakan suatu proses *learning* terhadap kesalahan atau *error* yang terjadi, proses ini berfungsi untuk memperbarui nilai dari *consequent parameter*. Proses *backward propagation* dapat dilakukan pada semua lapisan atau pada salah satu lapisan saja. Saat keluaran yang dihasilkan pada proses *forward propagation* sesuai harapan maka desain struktur *neuro fuzzy* dapat digunakan untuk menyelesaikan permasalahan dari kinematika balik.

3.4.1. Perancangan Struktur *Neuro Fuzzy*

Struktur dari *neuro fuzzy* yang akan digunakan pada Tugas Akhir ini merupakan struktur *neuro fuzzy* Mamdani. Pada penelitian tugas akhir ini, metode *neuro fuzzy* digunakan untuk menghitung besar sudut masing-masing *joint* apabila diberi masukan berupa posisi dan orientasi yang dituju *end effector* manipulator robot. Hasil keluaran yang dihasilkan dari *neuro fuzzy* harus sesuai dengan posisi target dari *end effector*. Pada penelitian tugas akhir ini permasalahan kinematika balik yang dibahas hanya pergerakan 5-DoF dari pergerakan manipulator robot Denso yang mempunyai pergerakan 6-DOF. Pergerakan dari 5-DoF manipulator robot denso mempengaruhi pergerakan posisi dan orientasi dari *end effector*. Dalam mendesain struktur *neuro fuzzy* maka ditentukan masukan dan keluarannya. Masukan dari *neuro fuzzy* ini adalah posisi x , y , z dan orientasi *roll*, *pitch* dan keluaran yang dihasilkan adalah berupa besar sudut θ_1 , θ_2 , θ_3 , θ_4 dan θ_5 dimana masing – masing memiliki 2 buah himpunan pendukung. Struktur *neuro fuzzy* yang digunakan dapat dilihat pada Gambar 3.8.



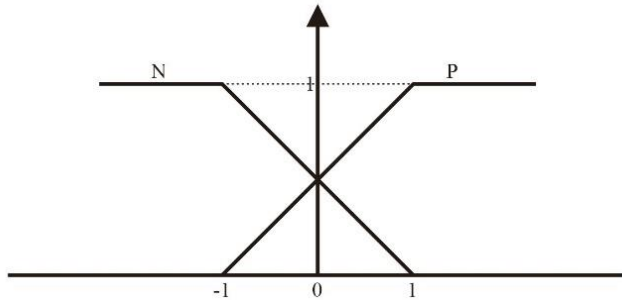
Gambar 3.8 Struktur *Neuro Fuzzy*

3.4.2. Tahap *Forward Propagation*

Pada tahap ini, akan dilakukan perhitungan *Feedforward Propagation* untuk memperoleh nilai keluaran. Tahapan *Feedforward Propagation* terdiri dari 5 lapisan yang setiap lapisan berisi proses perhitungan *fuzzy*. Berikut ini adalah tahapan struktur *neuro fuzzy* :

Lapisan 1 :

Lapisan 1 merupakan lapisan masukan awal dari struktur *neuro fuzzy*. Proses ini merupakan proses *fuzzyfikasi*, masukkan pada lapisan ini terdiri dari nilai posisi x, y, z dan nilai orientasi roll, pitch. Simbol dx merupakan nilai posisi x, simbol dy merupakan nilai posisi y, simbol dz merupakan nilai posisi z, simbol dr merupakan nilai orientasi roll dan simbol dp merupakan nilai orientasi pitch dari *end effector*. Masing-masing masukan dari nilai posisi x, y, z dan orientasi roll, pitch memiliki 2 himpunan pendukung. Fungsi keanggotaan yang digunakan yaitu fungsi segitiga yang dapat dilihat pada Gambar 3.9.



Gambar 3.9 Himpunan Pendukung dx, dy, dz, dr dan dp

Lapisan 2 :

Lapisan kedua digunakan untuk basis aturan (*rule-base*). *Rule base* adalah lapisan untuk pemberian nilai aturan (*min, max*) pada nilai keluaran dari lapisan 1 (*fuzzyfikasi*). Dengan cara ini pemetaan tingkatan menjadi lebih mudah untuk dipahami dan mudah untuk di logika. Sebagai contoh “jika dx_1, dy_1, dz_1, dr_1 dan dp_1 bernilai positif maka nilai $dt_{11} < 0$. Basis aturan merupakan kata “dan” yang artinya adalah operasi minimum [2]. Berikut ini merupakan *rule base* yang akan di gunakan pada tugas akhir ini,

- rb1=minimum(dx(n), dy(n), dz(n), dr(n), dp(n))
- rb2=minimum(dx(n), dy(n), dz(n), dr(n), dp(p))
- rb3=minimum(dx(n), dy(n), dz(n), dr(p), dp(n))
- rb4=minimum(dx(n), dy(n), dz(n), dr(p), dp(p))
- rb5=minimum(dx(n), dy(n), dz(p), dr(n), dp(n))
- rb6=minimum(dx(n), dy(n), dz(p), dr(n), dp(p))
- rb7=minimum(dx(n), dy(n), dz(p), dr(p), dp(n))
- rb8=minimum(dx(n), dy(n), dz(p), dr(p), dp(p))
- rb9=minimum(dx(n), dy(p), dz(n), dr(n), dp(n))
- rb10=minimum(dx(n), dy(p), dz(n), dr(n), dp(p))
- rb11=minimum(dx(n), dy(p), dz(n), dr(p), dp(n))
- rb12=minimum(dx(n), dy(p), dz(n), dr(p), dp(p))
- rb13=minimum(dx(n), dy(p), dz(p), dr(n), dp(n))
- rb14=minimum(dx(n), dy(p), dz(p), dr(n), dp(p))
- rb15=minimum(dx(n), dy(p), dz(p), dr(p), dp(n))
- rb16=minimum(dx(n), dy(p), dz(p), dr(p), dp(p))
- rb17=minimum(dx(p), dy(n), dz(n), dr(n), dp(n))
- rb18=minimum(dx(p), dy(n), dz(n), dr(n), dp(p))

rb19=minimum(dx(p), dy(n), dz(n), dr(p), dp(n))
 rb20=minimum(dx(p), dy(n), dz(n), dr(p), dp(p))
 rb21=minimum(dx(p), dy(n), dz(p), dr(n), dp(n))
 rb22=minimum(dx(p), dy(n), dz(p), dr(n), dp(p))
 rb23=minimum(dx(p), dy(n), dz(p), dr(p), dp(n))
 rb24=minimum(dx(p), dy(n), dz(p), dr(p), dp(p))
 rb25=minimum(dx(p), dy(p), dz(n), dr(n), dp(n))
 rb26=minimum(dx(p), dy(p), dz(n), dr(n), dp(p))
 rb27=minimum(dx(p), dy(p), dz(n), dr(p), dp(n))
 rb28=minimum(dx(p), dy(p), dz(n), dr(p), dp(p))
 rb29=minimum(dx(p), dy(p), dz(p), dr(n), dp(n))
 rb30=minimum(dx(p), dy(p), dz(p), dr(n), dp(p))
 rb31=minimum(dx(p), dy(p), dz(p), dr(p), dp(n))
 rb32=minimum(dx(p), dy(p), dz(p), dr(p), dp(p))

Setelah membuat *rule base* kemudian membuat inferensi *fuzzy*, inferensi *fuzzy* adalah proses pengambilan kesimpulan dari aturan *rule base* seperti berikut,

p1=maximum(rb1, rb2, rb3, rb4, rb5, rb6, rb7, rb8)
 p2=maximum(rb9, rb10, rb11, rb12, rb13, rb14, rb15, rb16)
 p3=maximum(rb17, rb18, rb19, rb20, rb21, rb22, rb23, rb24)
 p4=maximum(rb25, rb26, rb27, rb28, rb29, rb30, rb31, rb32)

Lapisan 3 :

Lapisan ketiga merupakan lapisan normalisasi. Normalisasi ini berfungsi agar menormalkan atau menjadikan semua keluaran pada lapisan ini memiliki nilai yang jika di jumlahkan adalah 1, hal ini dikarenakan agar nilai penjumlahan pada lapisan 3 tidak lebih dari 1 [2]. Rumus untuk lapisan normalisasi yang dapat dilihat pada persamaan 3.23.

$$\bar{w}_i = \frac{w_i}{w_1 + \dots + w_n} \dots\dots\dots (3.23)$$

\bar{w}_i = Keluaran ternormalisasi, ke-i s/d n

u = Jumlah himpunan pendukung *keluaran fuzzy*

Lapisan 4 :

Lapisan keempat merupakan proses perhitungan nilai *consequent parameter*. Nilai *consequent parameter* adalah himpunan nilai untuk melakukan proses *learning* pada lapisan 4 [2]. Algoritma perhitungan

untuk lapisan keempat dapat dilihat pada persamaan 3.24, dimana nilai f_i awal adalah nilai *random* pada proses inialisasi.

$$\bar{w}_i f_i = \bar{w}_i (p_i x_1 + q_i x_2 + r_i) \dots\dots\dots (3.24)$$

Lapisan 5 :

Lapisan kelima merupakan lapisan yang menjalankan proses *defuzzykasi* untuk menghitung keluaran dari *neuro fuzzy*. *Defuzzyfikasi* merupakan proses pengembalian dari nilai logika *fuzzy* menjadi logika sebenarnya dan digunakan dalam proses estimator parameter. Pada proses defusifikasi ini terdapat nilai bobot, dengan kondisi pertama nilai bobot tersebut diacak secara *random*. Akan tetapi pada proses *back propagation*, akan dilakukan proses revisi bobot yang bertujuan untuk mendapatkan nilai keluaran sistem sesuai dengan yang diinginkan.

3.4.3. Tahap Backward Propagation

Pada tugas akhir ini, struktur *neuro fuzzy* yang digunakan adalah struktur *neuro fuzzy* Mamdani dan proses *backward* atau revisi bobot hanya dilakukan pada lapisan kelima yaitu lapisan *defuzzifikasi*. Revisi bobot berupa revisi nilai tengah dari fungsi keanggotaan keluaran [2]. Untuk menghitung bobot atau nilai tengah baru digunakan Persamaan 3.25 dan perhitungan nilai kesalahan atau *error* pada Persamaan 3.26.

$$f_i = f(i-1) + lr * e * w_i \dots\dots\dots (3.25)$$

- f_i = consequent parameter baru
- $f(i-1)$ = consequent parameter lama
- lr = laju pembelajaran
- e = error (*reference-outputdefuzzyfikasi*)
- w_i = output lapisan 3 ternormalisasi

Dimana untuk perhitungan nilai *error* pada *joint* 1 sampai *joint* 5 :

$$e = (t_k - y_k) \dots\dots\dots (3.26)$$

Formulasi target θ_1 sampai θ_5 yang diharapkan:

$$t_1 = \tan^{-1} \left(\frac{y}{x} \right) \dots\dots\dots (3.27)$$

$$t_2 = \sqrt{x^2 + y^2} \dots\dots\dots (3.28)$$

$$t_3 = z \dots\dots\dots (3.29)$$

$$t_4 = roll \dots\dots\dots (3.30)$$

$$t_2 = pitch \dots\dots\dots (3.31)$$

Formulasi target θ_1 sampai θ_5 keluaran *neuro fuzzy*

$$t_1 = \tan^{-1} \left(\frac{y_n}{x_n} \right) \dots\dots\dots (3.32)$$

$$t_2 = \sqrt{x_n^2 + y_n^2} \dots\dots\dots (3.33)$$

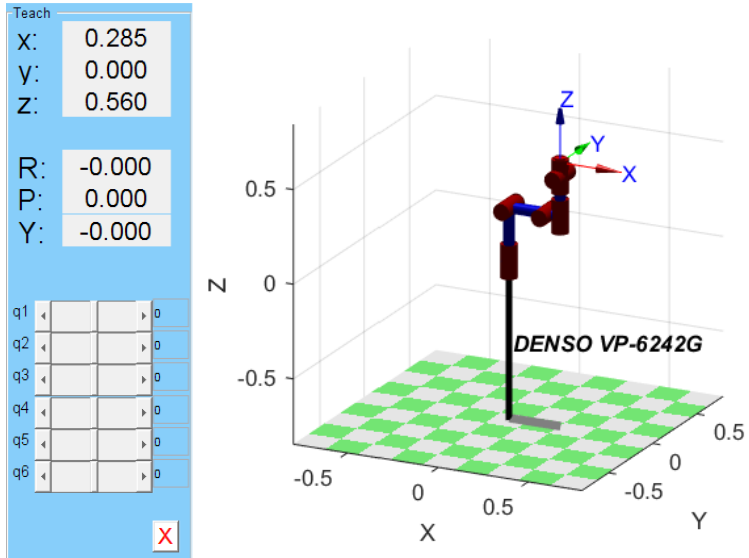
$$t_3 = z_n \dots\dots\dots (3.34)$$

$$t_4 = roll_n \dots\dots\dots (3.35)$$

$$t_2 = pitch_n \dots\dots\dots; \dots\dots\dots (3.36)$$

3.5. Simulasi Kinematika Maju dan Kinematika Balik

Dalam melakukan pengkoreksian hasil kinematika maju dan simulasi kinematika balik dengan *neuro fuzzy*, terlebih dahulu diambil data dari kinematika maju dan di bandingkan dengan hasil simulasi untuk menentukan kebenaran kinematika maju. Setelah kinematika maju telah benar, program kinematika maju akan digunakan untuk mengkoreksi hasil kinematika balik dengan cara pada program kinematika maju dimasukkan besar masing-masing sendi secara acak, sehingga didapatkan orientasi *roll* dan *pitch*, serta posisi *x,y,dan z*. Data orientasi dan posisi yang didapatkan dari program kinematika maju akan digunakan untuk masukan program kinematika balik dengan *neuro fuzzy*, 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. Tampilan simulasi dapat dilihat pada gambar 3.10.



Gambar 3.10 Tampilan Simulasi Manipulator Robot Denso

[Halaman ini sengaja dikosongkan]

BAB 4 PENGUJIAN DAN ANALISA

Pada bab ini akan menjelaskan mengenai hasil dari simulasi kinematika maju dan hasil dari penyelesaian kinematika balik menggunakan metode *neuro fuzzy*. Penyelesaian kinematika balik diselesaikan menggunakan *neuro fuzzy* dalam bentuk *m-file* pada *software* matlab dengan masukan posisi target x , y , z dan orientasi *roll*, *pitch* sehingga keluaran yang dihasilkan berupa besar sudut masing-masing *joint* manipulator robot Denso. Hasil keluaran sudut yang diperoleh dari *neuro fuzzy* akan digunakan sebagai masukan simulasi kinematika maju agar keluaran posisi dari kinematika maju sesuai dengan masukan dari kinematika balik yaitu posisi target x , y , z dan orientasi target *roll*, *pitch* dari *end effector* manipulator robot Denso.

4.1. Pengujian Kinematika Maju

Pengujian kinematika maju pada manipulator robot Denso dilakukan dengan cara membandingkan hasil keluaran dari program kinematika maju dengan keluaran hasil simulasi yang telah diberi masukan besar sudut dari masing-masing *joint* manipulator robot dalam satuan derajat. Kemudian mengkoreksi apakah hasil keluaran dari program kinematika maju sama dengan hasil keluaran pada simulasi. Keluaran dari simulasi dan kinematika maju yaitu berupa posisi dan orientasi dari *end effector* manipulator robot. Jika keluaran dari program kinematika maju dan keluaran simulasi sama maka dapat di simpulkan program kinematika maju telah benar. Masukan dan keluaran dari pengujian kinematika maju dapat dilihat pada Tabel 4.1 dan Tabel 4.2.

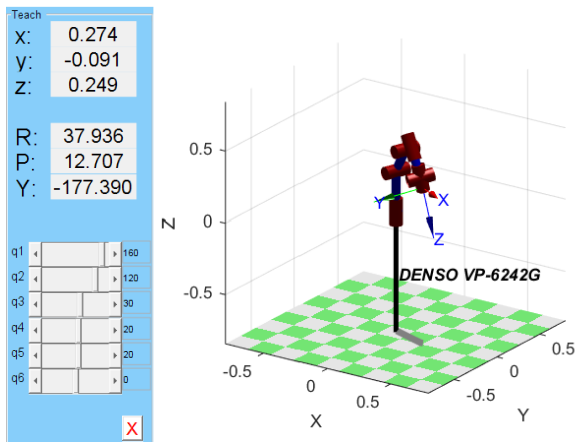
Tabel 4.1 Masukan Kinematika Maju

No.	Sudut sendi (derajat)					
	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1	160	120	30	20	20	0
2	150	100	50	10	10	0
3	120	70	30	10	15	0
4	-25	18	10	5	-12	0
5	-50	-20	50	-25	10	0
6	-75	50	10	-5	-50	0

Tabel 4.2 Keluaran Kinematika Maju

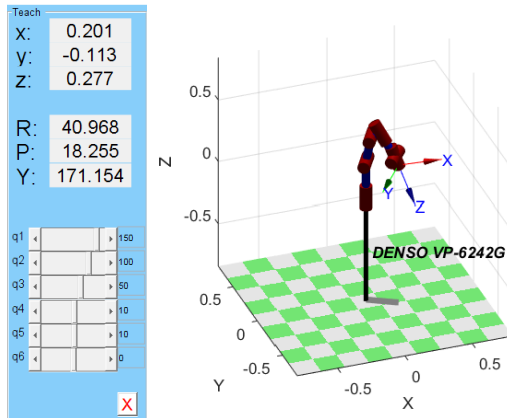
No.	Posisi (m)			Orientasi (derajat)	
	x	y	z	roll	pitch
1	0.2738	-0.0909	0.2493	37.9365	12.7073
2	0.2007	-0.1134	0.2768	40.9681	18.2554
3	0.1085	-0.1816	0.4854	85.7319	29.4996
4	0.1347	-0.0614	0.6328	-21.2292	-14.0511
5	0.0767	-0.0835	0.4817	-82.5097	-20.3590
6	-0.0101	0.0197	0.6796	-78.3650	-6.3044

Hasil yang di dapat dari pengujian kinematika maju kemudian di simulasikan dengan *software* matlab 2017 agar dapat mengetahui bentuk, posisi dan orientasi dari manipulator robot denso VP-6242G. kemudian membandingkan jika setiap *joint* dari manipulator robot diberi masukan sudut sesuai pengujian dan apakah keluaran dari program kinematika maju sesuai atau sama dengan keluaran dari simulator. bentuk, posisi dan orientasi dari manipulator robot dapat di lihat pada Gambar 4.1 sampai Gambar 4.6.



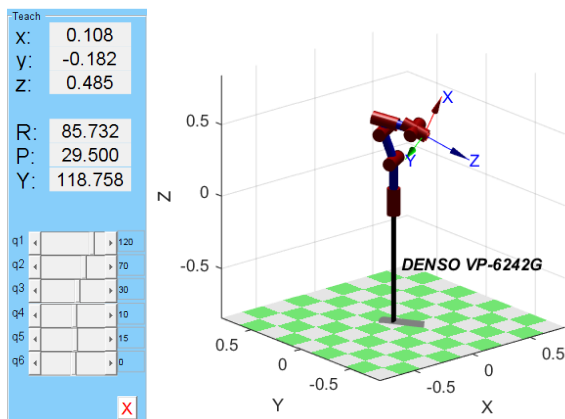
Gambar 4.1 Bentuk Manipulator Robot Denso Percobaan 1

Gambar 4.1 menunjukkan hasil simulasi dari manipulator robot denso dengan besar sudut masing-masing sendi $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6 = 160, 120, 30, 20, 20, 0$ yang menghasilkan posisi $x, y, z = 0.274, -0.091, 0.249$ dan orientasi $roll, pitch = 37.936, 12.707$.



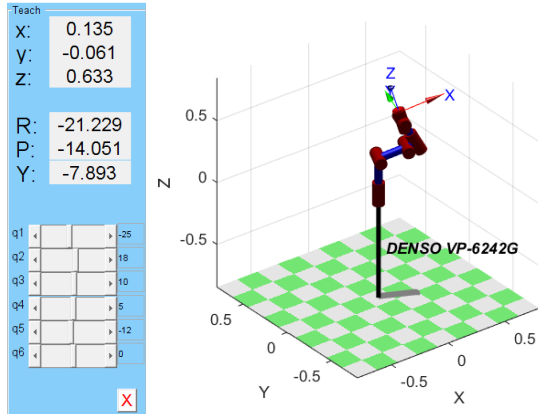
Gambar 4.2 Bentuk Manipulator Robot Denso Percobaan 2

Gambar 4.2 menunjukkan hasil simulasi dari manipulator robot denso dengan besar sudut masing-masing sendi $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6 = 150, 100, 50, 10, 10, 0$ yang menghasilkan posisi $x, y, z = 0.201, -0.113, 0.277$ dan orientasi $roll, pitch = 40.968, 18.255$.



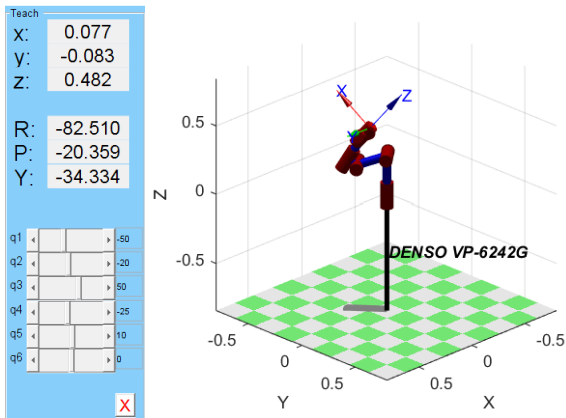
Gambar 4.3 Bentuk Manipulator Robot Denso Percobaan 3

Gambar 4.3 menunjukkan hasil simulasi dari manipulator robot denso dengan besar sudut masing-masing sendi $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6 = 120, 70, 30, 10, 15, 0$ yang menghasilkan posisi $x, y, z = 0.108, -0.182, 0.485$ dan orientasi $roll, pitch = 85.732, 29.500$.



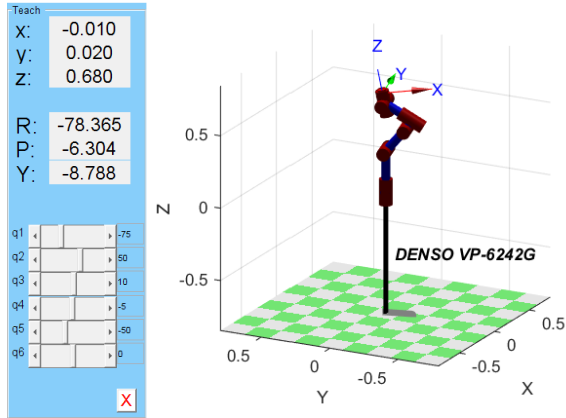
Gambar 4.4 Bentuk Manipulator Robot Denso Percobaan 4

Gambar 4.4 menunjukkan hasil simulasi dari manipulator robot denso dengan besar sudut masing-masing sendi $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6 = -25, 18, 10, 5, -12, 0$ yang menghasilkan posisi $x, y, z = 0.135, -0.061, 0.633$ dan orientasi $roll, pitch = -21.229, -14.051$.



Gambar 4.5 Bentuk Manipulator Robot Denso Percobaan 5

Gambar 4.5 menunjukkan hasil simulasi dari manipulator robot denso dengan besar sudut masing-masing sendi $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6 = -50, -20, 50, -25, 10, 0$ yang menghasilkan posisi $x, y, z = 0.77, -0.083, 0.482$ dan orientasi $roll, pitch = -82.510, -20.359$.



Gambar 4.6 Bentuk Manipulator Robot Denso Percobaan 6

Gambar 4.6 menunjukkan hasil simulasi dari manipulator robot denso dengan besar sudut masing-masing sendi $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6 = -75, 50, 10, -5, -50, 0$ yang menghasilkan posisi $x, y, z = -0.010, 0.020, 0.680$ dan orientasi *roll, pitch* $= -78.365, -6.304$.

Dari hasil dimulasi keenam percobaan dapat dilihat bahwa hasil dari program kinematika maju yang akan digunakan untuk mengoreksi hasil kinematika balik sudah sesuai dengan hasil simulasi menggunakan *software* matlab dengan tambahan *toolbox Peter Corke* versi 10.

4.2. Pengujian Laju Pembelajaran (α) Kinematika Balik

Setelah melakukan pengujian kinematika maju yang berfungsi untuk melakukan pengoreksian terhadap keluaran dari kinematika balik, selanjutnya akan melakukan pengujian terhadap keluaran dari kinematika balik dengan mengubah-ubah nilai dari nilai laju pembelajaran (α) dikarenakan Manipulator robot denso memiliki jumlah DoF sebanyak 6, yang mana pergerakan dari manipulator robot denso dibagi atas dua bagian, untuk pergerakan posisi *end effector* dipengaruhi oleh 3-DoF awal dan untuk pergerakan orientasi *end effector* dipengaruhi oleh 3-DoF yang selanjutnya, namun pada tugas akhir kali ini, pengujian yang di lakukan untuk permasalahan kinematika balik hanya membahas 5-DoF awal dari 6-Dof yang dimiliki manipulator robot denso. Pergerakan 5-DoF dari manipulator robot denso hanya melihat pergerakan posisi x, y, z dan orientasi *roll, pitch* dari *end effector*. Penyelesaian permasalahan kinematika balik ini menggunakan metode *neuro fuzzy*. Masukan pada

kinematika balik ini berupa posisi x , y , z dan orientasi $roll$, $pitch$ dari target, penyelesaian kinematika balik ini akan menghasilkan keluaran berupa besar sudut masing-masing $joint$ dalam satuan derajat, penyelesaian kinematika dengan metode *neuro fuzzy* di lakukan *software* matlab dengan format *m-file*. Pada pengujian ini besar sudut θ_6 diabaikan atau diberi nilai nol. Pada metode *neuro fuzzy* akan dilakukan beberapa percobaan yaitu akan diberikan nilai laju pembelajaran (α) yang berbeda-beda pada *range* (0.0001–0.0009) dengan ketelitian *error* ($\epsilon=0.00005$). Tabel pengujian kinematika balik menggunakan *neuro fuzzy* dapat dilihat pada Tabel 4.3 sampai Tabel 4.17.

Pada percobaan pertama setiap $joint$ diberi masukan [160 120 30 20 20 0] dan diberi posisi dan orientasi target seperti pada Table 4.3.

Tabel 4.3 Masukan Kinematika Balik Pengujian (α) Percobaan 1

Posisi Awal (m)			Orientasi Awal (derajat)	
x	y	z	roll	Pitch
0.2738	-0.0909	0.2493	37.9365	12.7073
Posisi Target (m)			Orientasi Target (derajat)	
x	y	z	roll	Pitch
0.2448	-0.1325	0.2592	53.7568	17.2788

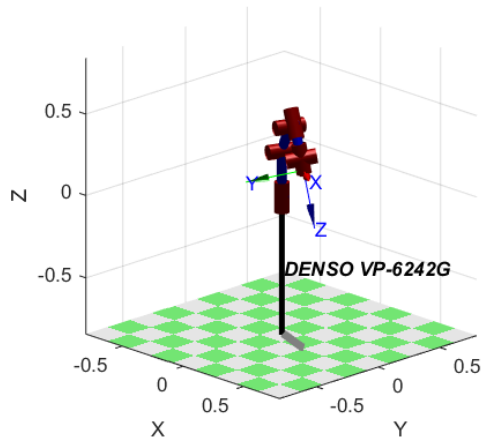
Tabel 4.4 Keluaran Kinematika Balik Pengujian (α) Percobaan 1

α	Keluaran <i>Neuro Fuzzy</i> (derajat)				
	θ_1	θ_2	θ_3	θ_4	θ_5
0.0008	149.9993	114.9907	35.0122	24.9985	14.9958
0.0007	149.9995	114.9909	35.0115	24.9984	14.9958
0.0005	149.9996	114.9909	35.0120	24.9983	14.9956
0.0003	149.9995	114.9906	35.0119	24.9983	14.9956
0.0001	149.9995	114.9907	35.0119	24.9983	14.9956

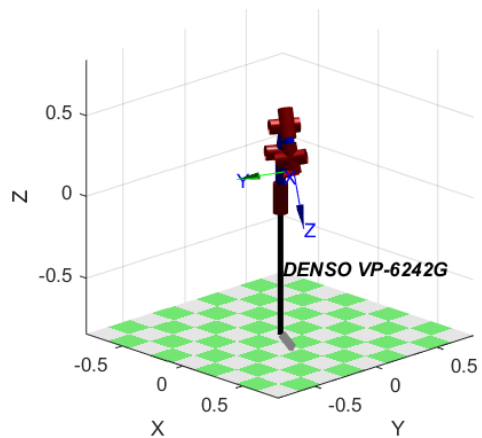
Tabel 4.5 Iterasi Keluaran Kinematika Balik Pengujian (α) Percobaan 1

α	Σ Iterasi	Keterangan
0.0008	3536569	7 dari 10 titik terkena iterasi maksimum
0.0007	3518540	7 dari 10 titik terkena iterasi maksimum
0.0005	3034814	6 dari 10 titik terkena iterasi maksimum
0.0003	3111855	6 dari 10 titik terkena iterasi maksimum
0.0001	595911	Tidak terkena iterasi maksimum

Tabel 4.4 merupakan hasil keluaran dari *neuro fuzzy* percobaan pertama dalam menentukan posisi target yang akan dituju oleh *end effector*. Posisi dan orientasi awal *end effector* manipulator robot dapat dilihat pada Gambar 4.7 sedangkan posisi dan orientasi target yang dituju *end effector* pada pengujian 1 dapat dilihat pada Gambar 4.8.



Gambar 4.7 Posisi dan Orientasi Awal *End Effector* Percobaan 1



Gambar 4.8 Posisi dan Orientasi Target *End Effector* Percobaan 1

Pada percobaan kedua setiap *joint* diberi masukan [160 120 30 20 20 0] dan diberi posisi dan orientasi target seperti pada Table 4.6.

Tabel 4.6 Masukan Kinematika Balik Pengujian (α) Percobaan 2

Posisi Awal (m)			Orientasi Awal (derajat)	
x	y	z	<i>roll</i>	<i>Pitch</i>
0.2738	-0.0909	0.2493	37.9365	12.7073
Posisi Target (m)			Orientasi Target (derajat)	
x	y	z	<i>roll</i>	<i>Pitch</i>
0.2818	-0.1262	0.3217	41.0977	29.1512

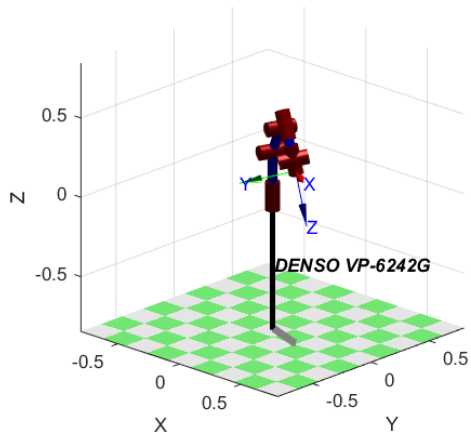
Tabel 4.7 Keluaran Kinematika Balik Pengujian (α) Percobaan 2

α	Keluaran <i>Neuro Fuzzy</i> (derajat)				
	θ_1	θ_2	θ_3	θ_4	θ_5
0.0008	155.0051	109.9860	25.0158	15.0070	15.0001
0.0007	155.0051	109.9862	25.0160	15.0072	15.0002
0.0005	155.0051	109.9860	25.0162	15.0069	15.0003
0.0003	155.0049	109.9859	25.0161	15.0070	15.0003
0.0001	155.0049	109.9860	25.0160	15.0071	15.0004

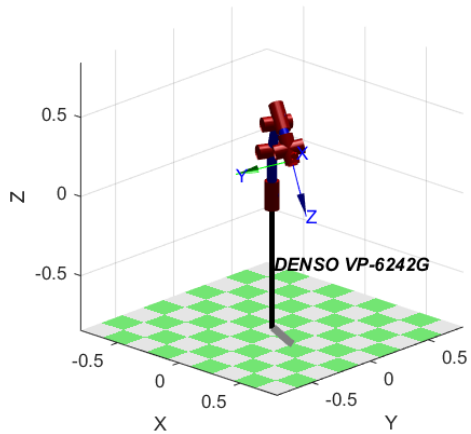
Tabel 4.8 Iterasi Keluaran Kinematika Balik Pengujian (α) Percobaan 2

α	Σ Iterasi	Keterangan
0.0008	7023691	14 dari 16 titik terkena iterasi maksimum
0.0007	7507167	15 dari 16 titik terkena iterasi maksimum
0.0005	7511473	15 dari 16 titik terkena iterasi maksimum
0.0003	3698734	7 dari 16 titik terkena iterasi maksimum
0.0001	1005509	Tidak terkena iterasi maksimum

Tabel 4.7 merupakan hasil keluaran dari *neuro fuzzy* percobaan kedua dalam menentukan posisi target yang akan dituju oleh *end effector*. Posisi dan orientasi awal *end effector* manipulator robot dapat dilihat pada Gambar 4.9 sedangkan posisi dan orientasi target yang dituju *end effector* pada pengujian 1 dapat dilihat pada Gambar 4.10.



Gambar 4.9 Posisi dan Orientasi Awal *End Effector* Percobaan 2



Gambar 4.10 Posisi dan Orientasi Target *End Effector* Percobaan 2

Pada percobaan kedua setiap *joint* diberi masukan [160 120 30 20 20 0] dan diberi posisi dan orientasi target seperti pada Table 4.6.

Tabel 4.9 Masukan Kinematika Balik Pengujian (α) Percobaan 3

Posisi Awal (m)			Orientasi Awal (derajat)	
x	y	z	<i>roll</i>	<i>Pitch</i>
0.2738	-0.0909	0.2493	37.9365	12.7073
Posisi Target (m)			Orientasi Target (derajat)	
x	y	z	<i>roll</i>	<i>Pitch</i>
0.2491	-0.1564	0.3166	55.0546	21.1216

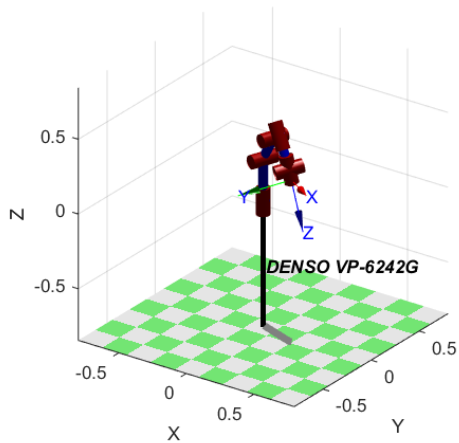
Tabel 4.10 Keluaran Kinematika Balik Pengujian (α) Percobaan 3

α	Keluaran <i>Neuro Fuzzy</i> (derajat)				
	θ_1	θ_2	θ_3	θ_4	θ_5
0.0008	144.9952	109.9898	25.0084	24.9925	29.9963
0.0007	144.9954	109.9893	25.0083	24.9924	29.9972
0.0005	144.9955	109.9896	25.0088	24.9923	29.9969
0.0003	144.9954	109.9896	25.0086	24.9922	29.9963
0.0001	144.9954	109.9896	25.0085	24.9923	29.9972

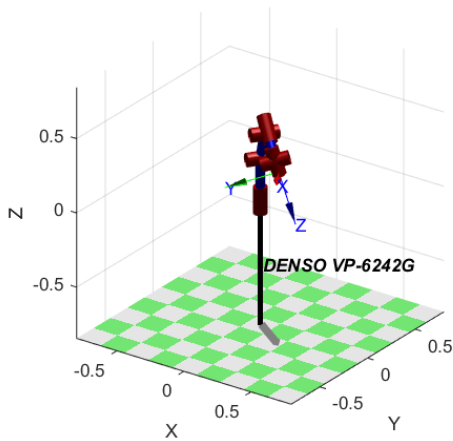
Tabel 4.11 Iterasi Keluaran Kinematika Balik Pengujian (α) Percobaan 3

α	Σ Iterasi	Keterangan
0.0008	8511116	17 dari 19 titik terkena iterasi maksimum
0.0007	8016105	16 dari 19 titik terkena iterasi maksimum
0.0005	5095705	10 dari 19 titik terkena iterasi maksimum
0.0003	4149591	8 dari 19 titik terkena iterasi maksimum
0.0001	871588	Tidak terkena iterasi maksimum

Tabel 4.10 merupakan hasil keluaran dari *neuro fuzzy* percobaan ketiga dalam menentukan posisi target yang akan dituju oleh *end effector*. Posisi dan orientasi awal *end effector* manipulator robot dapat dilihat pada Gambar 4.11 sedangkan posisi dan orientasi target yang dituju *end effector* pada pengujian 1 dapat dilihat pada Gambar 4.12.



Gambar 4.11 Posisi dan Orientasi Awal *End Effector* Percobaan 3



Gambar 4.12 Posisi dan Orientasi Target *End Effector* Percobaan 3

Dari data pengujian kinematika balik dengan nilai laju pembelajaran (α) yang berbeda-beda pada *range* (0.0001–0.0009) dengan ketelitian *error* ($\epsilon=0.00005$) dapat di lihat percobaan dengan nilai laju pembelajaran (α) 0.0008, 0.0007, 0.0005 dan 0.0003 selalu terdapat titik yang terkena

nilai iterasi maksimum, nilai iterasi maksimum di percobaan ini bernilai 500000 sedangkan nilai laju pembelajaran (α) 0.0001 disetiap titiknya mampu mendapatkan nilai θ_n tanpa menyentuh nilai iterasi maksimum, namun jika di dibandingkan dengan nilai laju pembelajaran (α) yang lain, agar mendapatkan nilai θ_n tanpa menyentuh nilai maksimal iterasi, nilai laju pembelajaran (α) 0.0001 membutuhkan waktu yang lama dapat di lihat dari nilai iterasinya.

Agar dapat mengetahui apakah keluaran *neuro fuzzy* sudah benar, maka perlu dilakukan pengoreksian dengan menggunakan nilai setiap θ_n yang telah di dapat sebagai masukan kinematika maju, kemudian membandingkan keluaran kinematika maju dengan nilai posisi dan orientasi target yang telah di gunakan untuk masukan kinematika balik, perbandingan nilai keluaran kinematika maju dengan nilai posisi dan orientasi target merupakan nilai kesalahan atau *error* dapat di lihat pada Tabel 4.13 untuk percobaan 1, Tabel 4.15 untuk percobaan 2 dan Tabel 4.15 untuk percobaan 3, dari data *error* yang di dapat digunakan untuk menyimpulkan nilai laju pembelajaran (α) yang mana yang akan di gunakan pada penelitian tugas akhir ini.

Tabel 4.12 Keluaran Kinematika Maju Pengujian (α) Percobaan 1

α	Posisi (m)			Orientasi (derajat)	
	x	Y	z	$roll$	$pitch$
0.0008	0.24479	-0.13250	0.25919	53.75711	17.27843
0.0007	0.24480	-0.13250	0.25920	53.75682	17.27885
0.0005	0.24480	-0.13249	0.25919	53.75666	17.27852
0.0003	0.24480	-0.13250	0.25920	53.75678	17.27886
0.0001	0.24480	-0.13250	0.25920	53.75677	17.27878

Tabel 4.13 Nilai *Error* Pengujian (α) Percobaan 1

α	Posisi (m)			Orientasi (derajat)	
	x	y	z	$roll$	$pitch$
0.0008	1.30e-06	2.22e-07	9.75e-07	3.08e-04	3.65e-04
0.0007	1.31e-06	6.40e-07	3.62e-07	2.82e-05	5.59e-05
0.0005	5.91e-08	4.94e-07	1.25e-06	1.40e-04	2.74e-04
0.0003	2.41e-07	1.71e-07	5.41e-07	1.78e-05	6.68e-05
0.0001	2.10e-07	1.53e-07	5.58e-08	2.21e-05	1.97e-05

Tabel 4.14 Keluaran Kinematika Maju Pengujian (α) Percobaan 2

α	Posisi (m)			Orientasi (derajat)	
	x	y	z	$roll$	$pitch$
0.0008	0.28180	-0.12619	0.32170	41.09756	29.15158
0.0007	0.28180	-0.12619	0.32169	41.09762	29.15118
0.0005	0.28180	-0.12619	0.32169	41.09733	29.15105
0.0003	0.28179	-0.12620	0.32170	41.09767	29.15121
0.0001	0.28180	-0.12620	0.32170	41.09773	29.15116

Tabel 4.15 Nilai *Error* Pengujian (α) Percobaan 2

α	Posisi (m)			Orientasi (derajat)	
	x	y	z	$Roll$	$pitch$
0.0008	1.32e-06	4.64e-07	1.22e-06	1.37e-04	3.84e-04
0.0007	9.12e-07	7.53e-07	7.28e-07	7.53e-05	1.43e-05
0.0005	1.53e-07	1.04e-06	5.53e-07	3.69e-04	1.41e-04
0.0003	8.91e-08	2.35e-09	4.05e-07	2.15e-05	1.97e-05
0.0001	1.61e-07	5.15e-08	2.28e-07	2.75e-05	3.86e-05

Tabel 4.16 Keluaran Kinematika Maju Pengujian (α) Percobaan 3

α	Posisi (m)			Orientasi (derajat)	
	x	y	z	$roll$	$pitch$
0.0008	0.24910	-0.15640	0.31659	55.05524	21.12215
0.0007	0.24910	-0.15640	0.31660	55.05467	21.12209
0.0005	0.24909	-0.15639	0.31659	55.05458	21.12156
0.0003	0.24910	-0.15640	0.31659	55.05481	21.12203
0.0001	0.24910	-0.15639	0.31659	55.05459	21.12164

Tabel 4.17 Nilai *Error* Pengujian (α) Percobaan 3

α	Posisi (m)			Orientasi (derajat)	
	x	y	z	$roll$	$pitch$
0.0008	5.11e-07	1.93e-06	7.10e-07	6.49e-04	5.59e-04
0.0007	3.91e-07	1.70e-07	2.00e-06	7.15e-05	4.93e-04
0.0005	1.79e-07	6.23e-07	1.42e-06	1.21e-05	3.24e-05
0.0003	3.53e-07	7.70e-07	5.15e-07	2.08e-04	4.35e-04
0.0001	4.16e-08	6.79e-09	3.30e-07	1.45e-06	4.21e-05

Dari nilai kesalahan atau *error* setiap percobaan yang telah di dapat di lihat bahwa nilai laju pembelajaran (α) 0.0001 memiliki nilai *error*

yang paling kecil walaupun nilai laju pembelajaran (α) 0.0001 memiliki jumlah iterasi di setiap point yang paling besar jika tidak terkena iterasi maksimum di setiap titik.

4.3. Pengujian Ketelitian Kinematika Balik

Setelah melakukan pengujian laju pembelajaran (α) dan telah mendapatkan hasil, selanjutnya akan dilakukan pengujian ketelitian pada kinematika balik dengan metode *neuro fuzzy*. Pada pengujian ini akan dilakukan dengan cara merubah-ubah nilai ketelitian dari program kinematika balik dengan metode *neuro fuzzy*. penelitian ini bertujuan untuk melihat pengaruh ketelitian terhadap hasil keluaran dari kinematika balik, ketelitian yang diuji sebesar 0.0005, 0.0001 dan 0.00005. Pada pengujian kali ini dilakukan dengan menggunakan laju pembelajaran (α) sebesar 0.0001.

Pada percobaan pertama diberi masukan yang sama dengan pengujian laju pembelajaran (α) percobaan pertama, yang mana keluaran dari percobaan pertama ini dapat di lihat pada Tabel 4.18 dan keluaran dari percobaan ini akan di gunakan sebagai masukan kinematika maju untuk mendapatkan nilai kesalahan atau *error*, keluaran dari kinematika maju percobaan yang pertama dapat di lihat pada Tabel 4.19 dan nilai *error* pada percobaan pertama dapat di lihat pada Tabel 4.21.

Tabel 4.18 Keluaran Kinematika Balik Pengujian Ketelitian Percobaan 1

Ketelitian	Keluaran <i>Neuro Fuzzy</i> (derajat)				
	θ_1	θ_2	θ_3	θ_4	θ_5
0.0005	149.9995	114.9907	35.0120	24.9989	14.9958
0.0001	149.9995	114.9907	35.0120	24.9984	14.9956
0.00005	149.9995	114.9907	35.0119	24.9983	14.9956

Tabel 4.19 Keluaran Kinematika Maju Pengujian Ketelitian Percobaan 1

Ketelitian	Posisi (m)			Orientasi (derajat)	
	x	y	z	<i>roll</i>	<i>pitch</i>
0.0005	0.24479	-0.13249	0.25919	53.75729	17.27871
0.0001	0.24479	-0.13249	0.25919	53.75686	17.27871
0.00005	0.24480	-0.13250	0.25920	53.75684	17.27878

Tabel 4.20 Iterasi Keluaran Kinematika Balik Pengujian Ketelitian Percobaan 1

ketelitian	Σ Iterasi	Keterangan
0.0005	594822	Rata-rata iterasi setiap titik 43327
0.0001	594946	Rata-rata iterasi setiap titik 43338
0.00005	595911	Rata-rata iterasi setiap titik 43353

Tabel 4.21 Nilai *Error* Pengujian Ketelitian Percobaan 1

ketelitian	Posisi (m)			Orientasi (derajat)	
	x	y	z	<i>roll</i>	<i>pitch</i>
0.0005	1.19e-07	8.87e-08	2.17e-07	4.95e-04	8.24e-05
0.0001	1.53e-07	1.54e-07	1.31e-07	6.68e-05	8.53e-05
0.00005	5.41e-08	1.74e-07	1.04e-08	4.51e-05	1.45e-05

Dari hasil yang di dapat pada percobaan yang pertama, dapat di lihat bahwa iterasi keluaran ketiga ketelitian tidak terpaut jauh, iterasi terbanyak terjadi pada ketelitian yang paling kecil namun memiliki hasil yang paling baik atau nilai *error* yang paling kecil. Nilai ketelitian pada percobaan pertama dapat di lihat pada Tabel 4.20.

Pada percobaan kedua diberi masukan yang sama dengan pengujian laju pembelajaran (α) percobaan kedua, yang mana keluaran dari percobaan pertama ini dapat di lihat pada Tabel 4.22 dan keluaran dari percobaan ini akan di gunakan sebagai masukan kinematika maju untuk mendapatkan nilai *error*, keluaran dari kinematika maju percobaan yang pertama dapat di lihat pada Tabel 4.23 dan nilai *error* pada percobaan pertama dapat di lihat pada Tabel 4.24.

Tabel 4.22 Keluaran Kinematika Balik Pengujian Ketelitian Percobaan 2

ketelitian	Keluaran <i>Neuro Fuzzy</i> (derajat)				
	θ_1	θ_2	θ_3	θ_4	θ_5
0.0005	154.9981	110.0880	24.8135	15.0292	15.1229
0.0001	155.0049	109.9860	25.0160	15.0071	15.0004
0.00005	155.0049	109.9860	25.0160	15.0071	15.0004

Tabel 4.23 Keluaran Kinematika Maju Pengujian Ketelitian Percobaan 2

Ketelitian	Posisi (m)			Orientasi (derajat)	
	<i>x</i>	<i>y</i>	<i>z</i>	<i>roll</i>	<i>pitch</i>
0.0005	0.28225	-0.12640	0.32192	41.09768	29.15118
0.0001	0.28180	-0.12620	0.32170	41.09770	29.15115
0.00005	0.28180	-0.12620	0.32170	41.09770	29.15115

Tabel 4.24 Iterasi Keluaran Kinematika Balik Pengujian Ketelitian Percobaan 2

Ketelitian	Σ Iterasi	Keterangan
0.0005	749232	Rata-rata iterasi setiap titik 45661
0.0001	1005491	Rata-rata iterasi setiap titik 58015
0.00005	1005509	Rata-rata iterasi setiap titik 58017

Tabel 4.25 Nilai *Error* Pengujian Ketelitian Percobaan 2

ketelitian	Posisi (m)			Orientasi (derajat)	
	<i>x</i>	<i>y</i>	<i>z</i>	<i>roll</i>	<i>pitch</i>
0.0005	4.56e-04	2.04e-04	2.18e-04	1.41e-05	1.86e-05
0.0001	1.20e-07	3.77e-08	2.22e-07	6.16e-07	4.90e-05
0.00005	1.20e-07	3.77e-08	2.22e-07	6.16e-07	4.90e-05

Dari hasil yang di dapat pada percobaan yang kedua, dapat di lihat bahwa iterasi keluaran dua dati ketiga ketelitian tidak terpaut jauh, ketelitian yang memiliki perbedaan lumayan jauh yaitu 0.0005, lebih cepat dari dua diantaranya, iterasi terbanyak terjadi pada ketelitian yang paling kecil namun memiliki hasil yang paling baik atau nilai kesalahan atau *error* yang paling kecil. Nilai ketelitian pada percobaan pertama dapat di lihat pada Tabel 4.24.

Pada percobaan ketiga diberi masukan yang sama dengan pengujian laju pembelajaran (α) percobaan ketiga, yang mana keluaran dari percobaan pertama ini dapat di lihat pada Tabel 4.26 dan keluaran dari percobaan ini akan di gunakan sebagai masukan kinematika maju untuk mendapatkan nilai *error*, keluaran dari kinematika maju percobaan yang pertama dapat di lihat pada Tabel 4.27 dan nilai *error* pada percobaan pertama dapat di lihat pada Tabel 4.29.

Tabel 4.26 Keluaran Kinematika Balik Pengujian Ketelitian Percobaan 3

ketelitian	Keluaran <i>Neuro Fuzzy</i> (derajat)				
	θ_1	θ_2	θ_3	θ_4	θ_5
0.0005	144.9956	109.9893	25.0089	24.9920	29.9960
0.0001	144.9954	109.9896	25.0085	24.9922	29.9970
0.00005	144.9954	109.9896	25.0085	24.9923	29.9972

Tabel 4.27 Keluaran Kinematika Maju Pengujian Ketelitian Percobaan 3

ketelitian	Posisi (m)			Orientasi (derajat)	
	x	y	z	<i>roll</i>	<i>pitch</i>
0.0005	0.24909	-0.15639	0.31659	55.05455	21.12207
0.0001	0.24910	-0.15639	0.31659	55.05454	21.12169
0.00005	0.24910	-0.15639	0.31659	55.05455	21.12163

Tabel 4.28 Iterasi Keluaran Kinematika Balik Pengujian Ketelitian Percobaan 3

ketelitian	Σ Iterasi	Keterangan
0.0005	850796	Rata-rata iterasi setiap titik 38818
0.0001	868495	Rata-rata iterasi setiap titik 39799
0.00005	871588	Rata-rata iterasi setiap titik 39971

Tabel 4.29 Nilai *Error* Pengujian Ketelitian Percobaan 3

ketelitian	Posisi (m)			Orientasi (derajat)	
	x	y	z	<i>roll</i>	<i>pitch</i>
0.0005	6.85e-08	1.94e-07	1.52e-07	4.73e-05	4.77e-04
0.0001	6. e-08	2.23e-09	3.08e-07	5.26e-05	9.33e-05
0.00005	6.99e-08	1.55e-08	1.88e-07	4.07e-05	3.63e-05

Dari hasil yang di dapat pada percobaan yang ketiga, dapat di lihat bahwa iterasi keluaran ketiga ketelitian tidak terpaut jauh, iterasi terbanyak terjadi pada ketelitian yang paling kecil namun memiliki hasil yang paling baik atau nilai kesalahan atau *error* yang paling kecil. Nilai ketelitian pada percobaan pertama dapat di lihat pada Tabel 4.29.

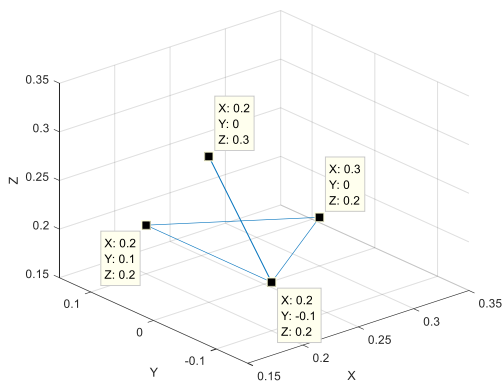
Dari hasil ketiga percobaan yang telah dilakukan, hasil yang paling bagus didapatkan pada nilai ketelitian 0.00005 walaupun jumlah iterasi yang didapat saat menggunakan nilai ketelitian 0.00005 paling besar namun nilai iterasi tidak terlalu jauh dengan nilai iterasi jika

menggunakan nilai ketelitian 0.0005 dan 0.0001 sehingga pada tugas akhir ini menggunakan nilai ketelitian sebesar 0.00005.

4.4. Kinematika Balik Membuat Pola Segitiga

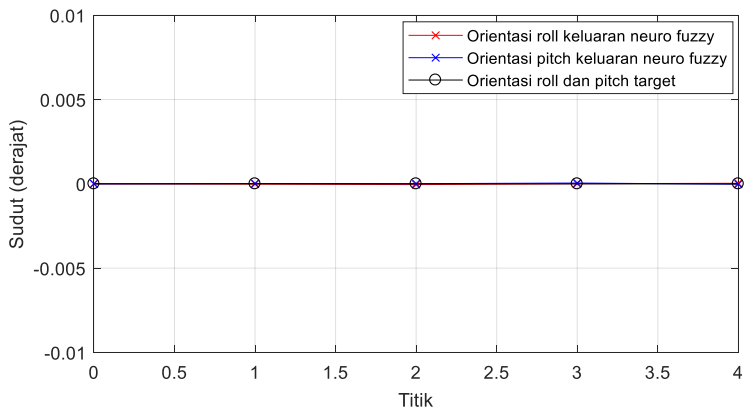
Setelah melakukan pengujian pengaruh dari nilai laju pembelajaran (α) dan pengaruh nilai ketelitian terhadap keluaran dari kinematika balik, dan telah didapatkan data yang dapat menyimpulkan nilai laju pembelajaran (α) dan nilai ketelitian yang mana yang akan digunakan pada tugas akhir ini, kemudian selanjutnya akan dilakukan pengujian manipulator robot denso untuk membuat pola segitiga yang terdiri dari beberapa titik, dimana titik-titik tersebut digunakan sebagai masukan kinematika balik dari manipulator robot denso. Titik tersebut berisikan keterangan posisi x, y dan z. Dalam membuat pola segitiga ini nilai orientasi *roll* dan *pitch* diberi nilai tetap yaitu *roll* bernilai 0 dan *pitch* juga bernilai 0, namun posisi x, y dan z berubah-ubah. *End effector* manipulator robot mula-mula berada di posisi $x = 0.2$, $y = 0$ dan $z = 0.3$ dan orientasi $roll = 0$ dan $pitch = 0$ yang terjadi pada sudut mula-mula $\theta_1 = 180$, $\theta_2 = 93.23$, $\theta_3 = 48.87$, $\theta_4 = 0$, $\theta_5 = 37.88$ dan $\theta_6 = 0$ menuju beberapa titik target yang membentuk suatu pola segitiga.

Pada percobaan yang pertama akan membuat pola sebuah segitiga dalam bidang datar. Titik-titik yang akan menjadi titik target manipulator robot dalam bentuk pola segitiga pada bidang yang datar dapat di lihat pada Gambar 4.13. pada pengujian kinematika balik dengan metode *neuro fuzzy* untuk membuat pola segitiga datar ini dilakukan dengan nilai laju pembelajaran (α) sebesar 0.0001 dan nilai ketelitian sebesar 0.00005.

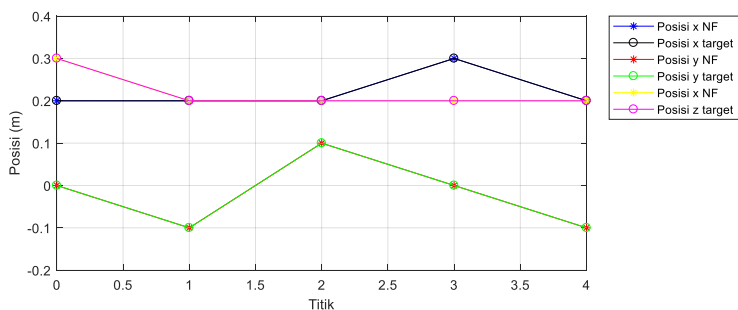


Gambar 4.13 Titik-Titik Target Segitiga Datar

Grafik dari perbandingan antara orientasi dari posisi target dengan orientasi dan posisi keluaran dari kinematika balik untuk membuat sebuah pola segitiga di bidang datar dapat dilihat pada Gambar 4.14 dan Gambar 4.15.



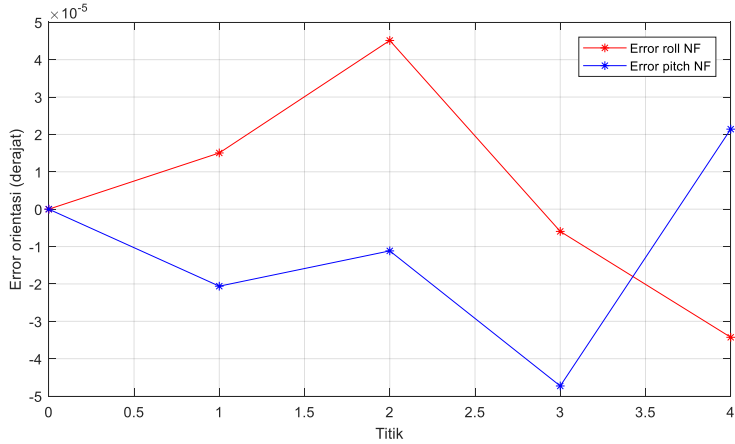
Gambar 4.14 Grafik Perbandingan orientasi Titik Target Dengan Keluaran Kinematika Balik Membuat Pola Segitiga Datar



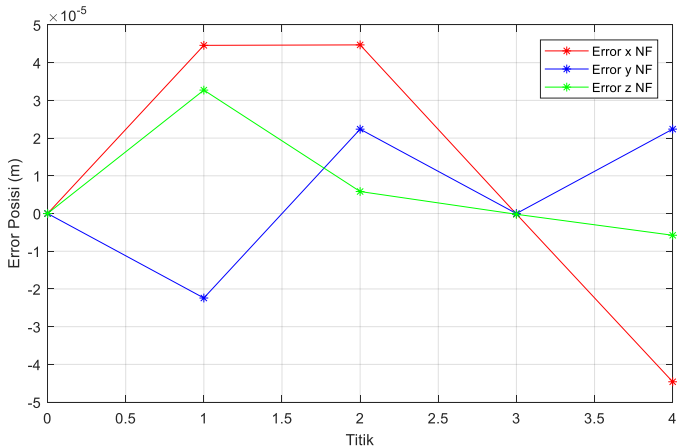
Gambar 4.15 Grafik Perbandingan Posisi Titik Target Dengan Keluaran Kinematika Balik Membuat Pola Segitiga Datar

Dari data yang didapat, dapat dilihat posisi dan orientasi dari keluaran kinematika balik dengan *neuro fuzzy* mengikuti posisi dan orientasi dari titik target untuk membuat pola segitiga datar. Grafik *error*

antara keluaran kinematika balik dengan *neuro fuzzy* terhadap titik target dapat di lihat pada Gambar 4.16 dan Gambar 4.17.

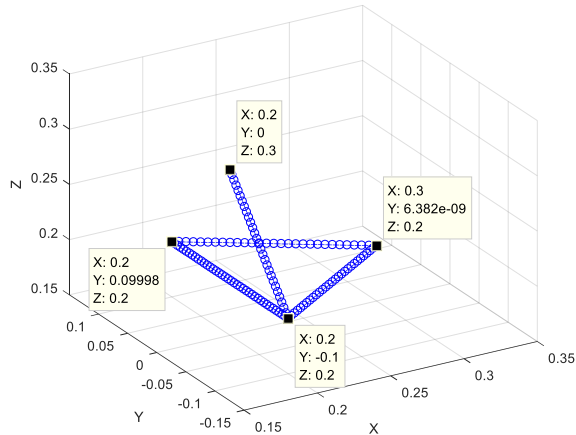


Gambar 4.16 Grafik *Error Orientasi Keluaran Kinematika Balik Membuat Pola Segitiga Datar*



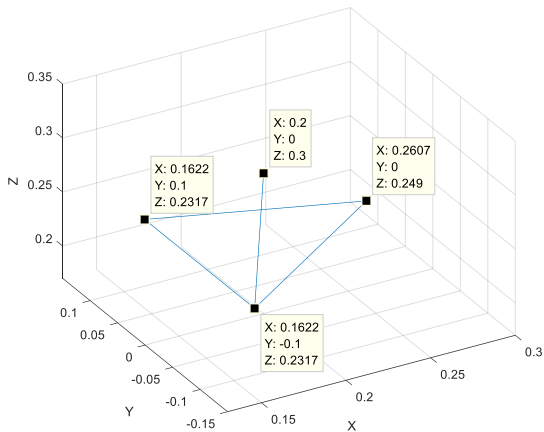
Gambar 4.17 Grafik *Error Posisi Keluaran Kinematika Balik Membuat Pola Segitiga Datar*

Hasil plot tiga dimensi keluaran hasil kinematika balik dengan *neuro fuzzy* dapat di lihat pada Gambar 4.18 dimana titik-titik tersebut merupakan titik-titik pergerakan dari *end effector* manipulator robot denso dalam membuat pola segitiga datar.



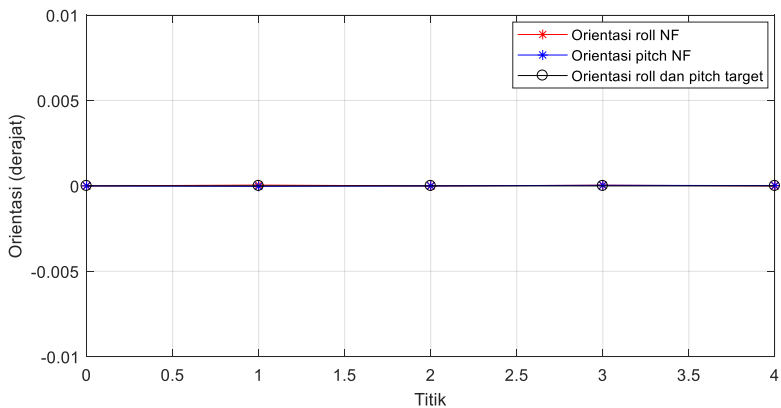
Gambar 4.18 Titik-Titik Keluaran Kinematika Balik Membuat Pola Segitiga Datar

Pada percobaan yang kedua akan membuat pola sebuah segitiga dalam bidang miring. Titik-titik yang akan menjadi titik target manipulator robot dalam bentuk pola segitiga pada bidang yang miring dapat di lihat pada Gambar 4.19. pada pengujian kinematika balik dengan metode *neuro fuzzy* untuk membuat pola segitiga miring ini dilakukan dengan nilai laju pembelajaran (α) sebesar 0.0001 dan nilai ketelitian sebesar 0.00005.

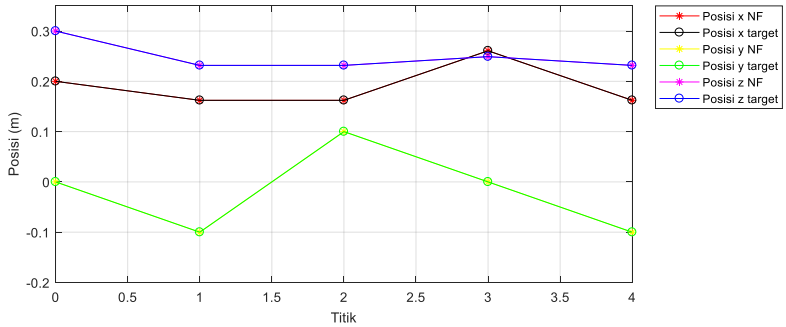


Gambar 4.19 Titik-Titik Target Segitiga Miring

Grafik dari perbandingan antara orientasi dari posisi target dengan orientasi dan posisi keluaran dari kinematika balik untuk membuat sebuah pola segitiga di bidang miring dapat dilihat pada Gambar 4.20 dan Gambar 4.21.

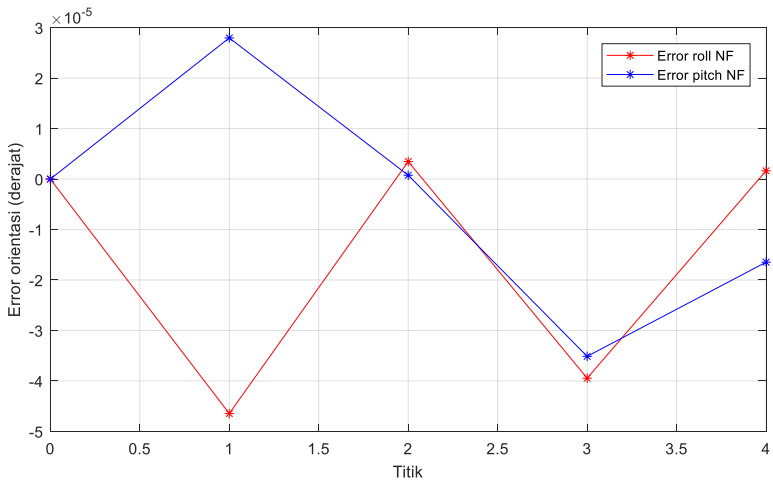


Gambar 4.20 Grafik Perbandingan orientasi Titik Target Dengan Keluaran Kinematika Balik Membuat Pola Segitiga Miring

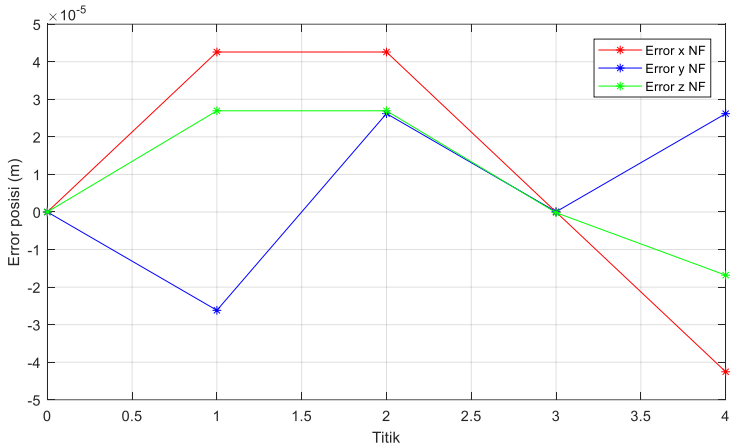


Gambar 4.21 Grafik Perbandingan Posisi Titik Target Dengan Keluaran Kinematika Balik Membuat Pola Segitiga Miring

Dari data yang didapat, dapat dilihat posisi dan orientasi dari keluaran kinematika balik dengan *neuro fuzzy* mengikuti posisi dan orientasi dari titik target untuk membuat pola segitiga miring. Grafik *error* antara keluaran kinematika balik dengan *neuro fuzzy* terhadap titik target dapat di lihat pada Gambar 4.22 dan Gambar 4.23.

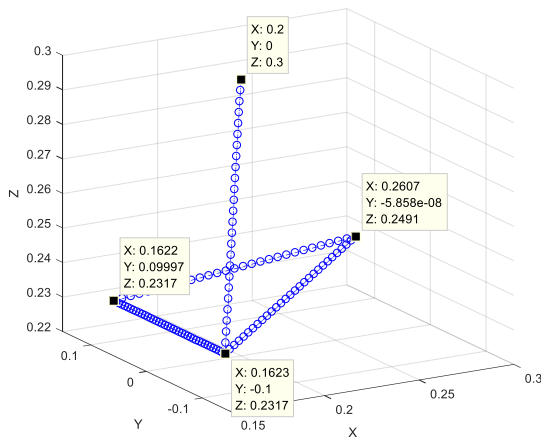


Gambar 4.22 Grafik *Error* Orientasi Keluaran Kinematika Balik Membuat Pola Segitiga Miring



Gambar 4.23 Grafik *Error Posisi Keluaran Kinematika Balik Membuat Pola Segitiga Miring*

Hasil plot tiga dimensi keluaran hasil kinematika balik dengan *neuro fuzzy* dapat di lihat pada Gambar 4.24 dimana titik-titik tersebut merupakan titik-titik pergerakan dari *end effector* manipulator robot denso dalam membuat pola segitiga miring.

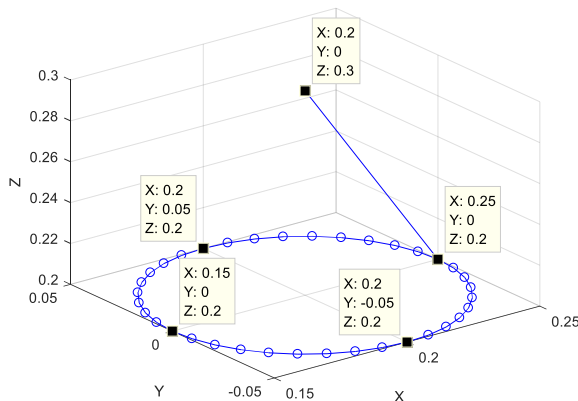


Gambar 4.24 Titik-Titik Keluaran Kinematika Balik Membuat Pola Segitiga Miring

4.5. Kinematika Balik Membuat Pola Lingkaran

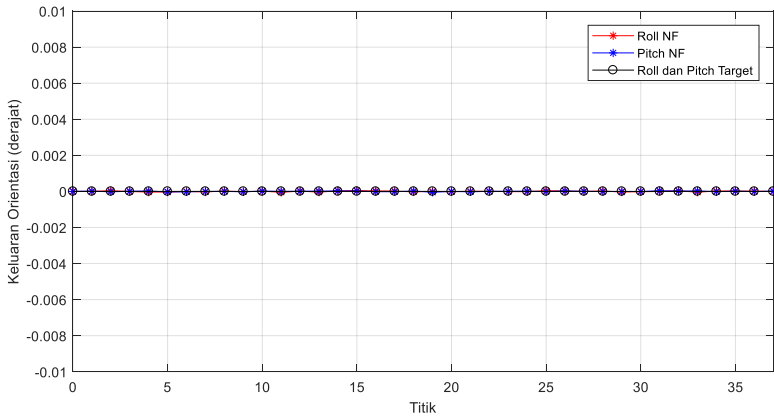
Setelah dilakukan pengujian dari manipulator robot untuk membuat pola dari gabungan garis lurus yaitu segitiga datar dan segitiga miring dan telah di dapatkan hasil pengujiannya, kemudian selanjutnya akan dilakukan pengujian manipulator robot denso untuk membuat pola dengan lengkungan yaitu lingkaran yang terdiri dari beberapa titik, dimana titik-titik tersebut digunakan sebagai masukan kinematika balik dari manipulator robot denso. Titik tersebut berisikan keterangan posisi x , y dan z . Dalam membuat pola lingkaran ini nilai orientasi $roll$ dan $pitch$ diberi nilai tetap yaitu $roll$ bernilai 0 dan $pitch$ juga bernilai 0, namun posisi x , y dan z berubah-ubah. *End effector* manipulator robot mula-mula berada di posisi $x = 0.2$, $y = 0$ dan $z = 0.3$ dan orientasi $roll = 0$ dan $pitch = 0$ yang terjadi pada sudut mula-mula $\theta_1 = 180$, $\theta_2 = 93.23$, $\theta_3 = 48.87$, $\theta_4 = 0$, $\theta_5 = 37.88$ dan $\theta_6 = 0$ menuju beberapa titik target yang membentuk suatu pola lingkaran.

Pada percobaan yang pertama akan membuat pola sebuah lingkaran dalam bidang datar. Titik-titik yang akan menjadi titik target manipulator robot dalam bentuk pola lingkaran pada bidang yang datar dapat di lihat pada Gambar 4.25. pada pengujian kinematika balik dengan metode *neuro fuzzy* untuk membuat pola lingkaran datar ini dilakukan dengan nilai laju pembelajaran (α) sebesar 0.0001 dan nilai ketelitian sebesar 0.00005.

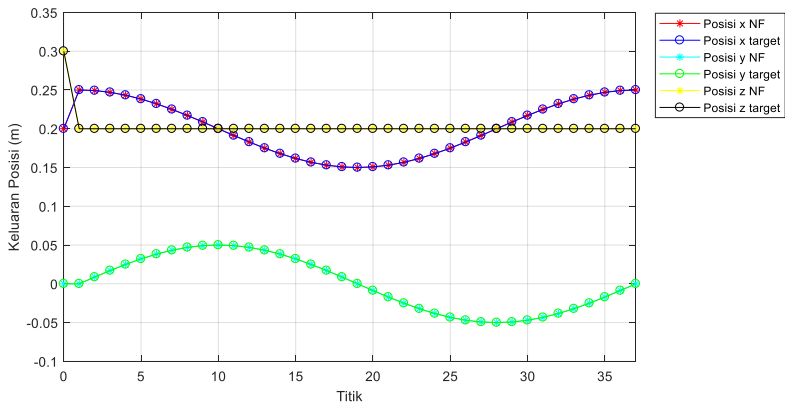


Gambar 4.25 Titik-Titik Target Lingkaran Datar

Grafik dari perbandingan antara orientasi dari posisi target dengan orientasi dan posisi keluaran dari kinematika balik untuk membuat sebuah pola lingkaran di bidang datar dapat dilihat pada Gambar 4.26 dan Gambar 4.27.

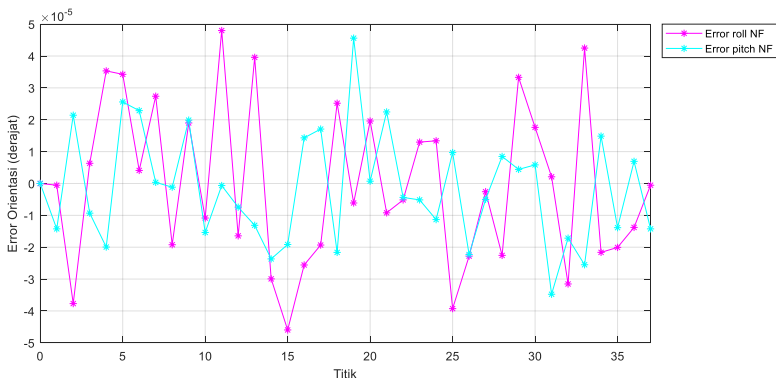


Gambar 4.26 Grafik Perbandingan orientasi Titik Target Dengan Keluaran Kinematika Balik Membuat Pola Lingkaran Datar

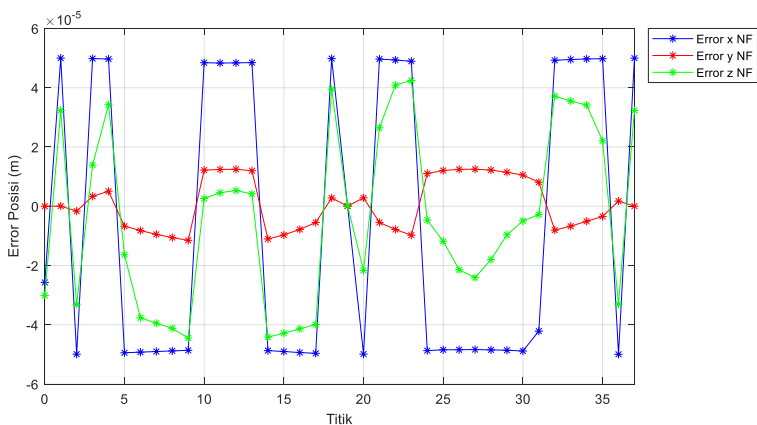


Gambar 4.27 Grafik Perbandingan Posisi Titik Target Dengan Keluaran Kinematika Balik Membuat Pola Lingkaran Datar

Dari data yang didapat, dapat dilihat posisi dan orientasi dari keluaran kinematika balik dengan *neuro fuzzy* mengikuti posisi dan orientasi dari titik target untuk membuat pola lingkaran datar. Grafik *error* antara keluaran kinematika balik dengan *neuro fuzzy* terhadap titik target dapat di lihat pada Gambar 4.28 dan Gambar 4.29.

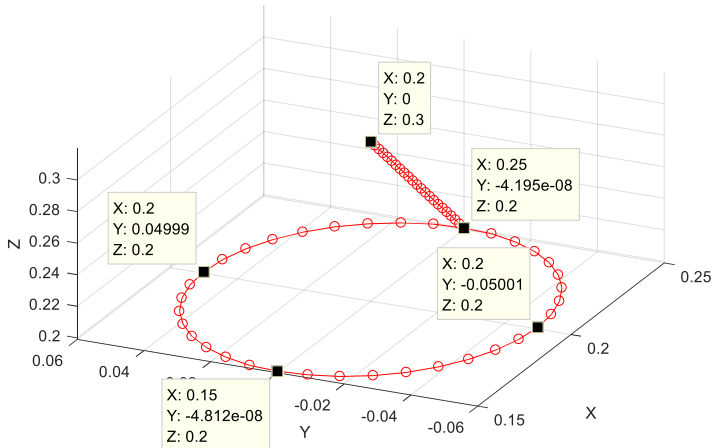


Gambar 4.28 Grafik *Error Orientasi Keluaran Kinematika Balik Membuat Pola Lingkaran Datar*



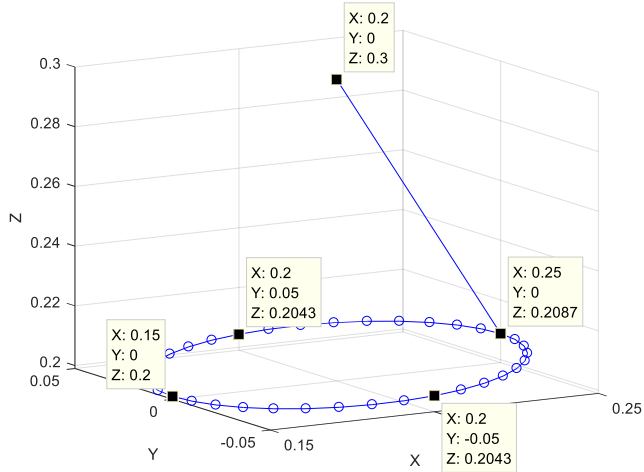
Gambar 4.29 Grafik *Error Posisi Keluaran Kinematika Balik Membuat Pola Lingkaran Datar*

Hasil plot tiga dimensi keluaran hasil kinematika balik dengan *neuro fuzzy* dapat di lihat pada Gambar 4.30 dimana titik-titik tersebut merupakan titik-titik pergerakan dari *end effector* manipulator robot denso dalam membuat pola lingkaran datar.



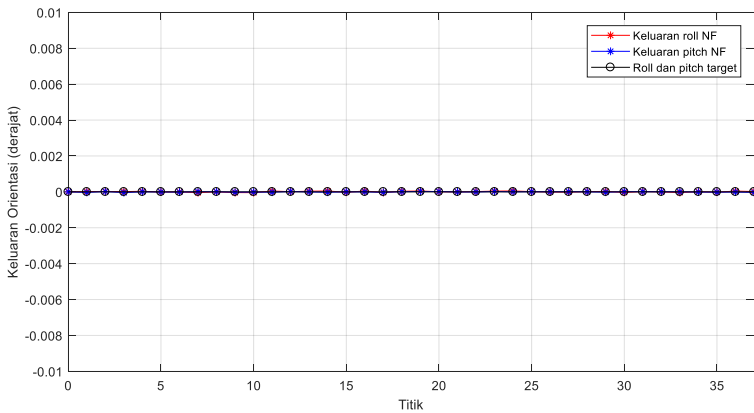
Gambar 4.30 Titik-Titik Keluaran Kinematika Balik Membuat Pola Lingkaran Datar

Pada percobaan yang kedua akan membuat pola sebuah lingkaran dalam bidang miring. Titik-titik yang akan menjadi titik target manipulator robot dalam bentuk pola lingkaran pada bidang yang miring dapat di lihat pada Gambar 4.31. pada pengujian kinematika balik dengan metode *neuro fuzzy* untuk membuat pola lingkaran miring ini dilakukan dengan nilai laju pembelajaran (α) sebesar 0.0001 dan nilai ketelitian sebesar 0.00005.

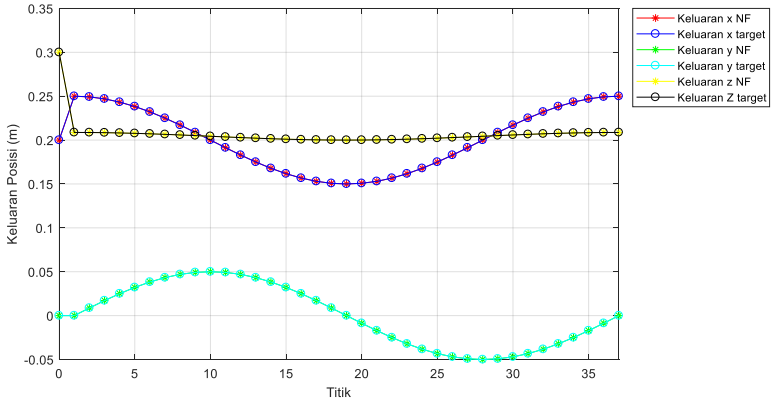


Gambar 4.31 Titik-Titik Target Lingkaran Miring

Grafik dari perbandingan antara orientasi dari posisi target dengan orientasi dan posisi keluaran dari kinematika balik untuk membuat sebuah pola lingkaran di bidang miring dapat dilihat pada Gambar 4.32 dan Gambar 4.33.

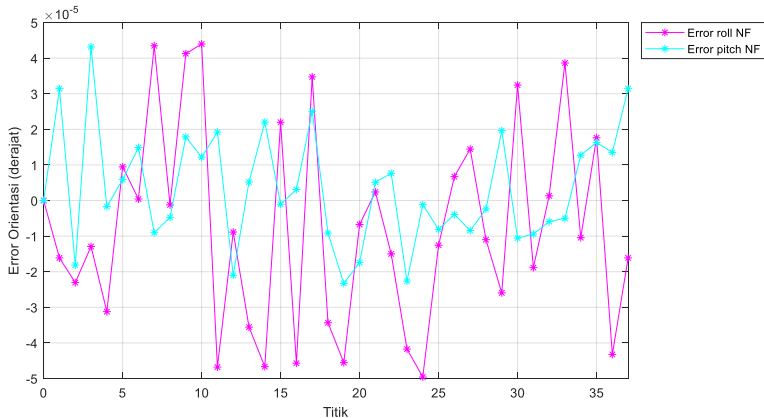


Gambar 4.32 Grafik Perbandingan orientasi Titik Target Dengan Keluaran Kinematika Balik Membuat Pola Lingkaran Miring

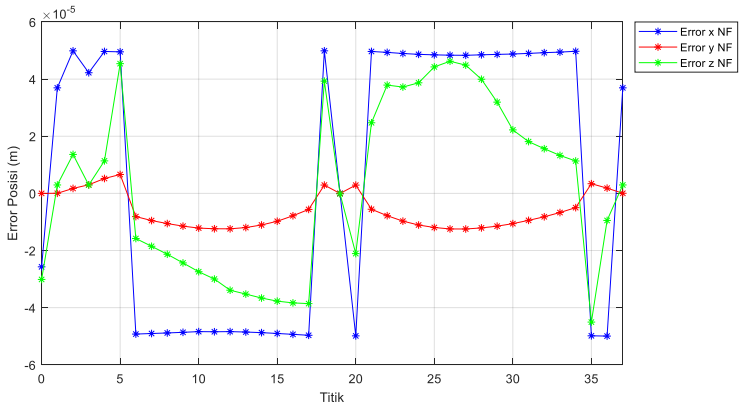


Gambar 4.33 Grafik Perbandingan Posisi Titik Target Dengan Keluaran Kinematika Balik Membuat Pola Lingkaran Miring

Dari data yang didapat, dapat dilihat posisi dan orientasi dari keluaran kinematika balik dengan *neuro fuzzy* mengikuti posisi dan orientasi dari titik target untuk membuat pola lingkaran miring. Grafik *error* antara keluaran kinematika balik dengan *neuro fuzzy* terhadap titik target dapat di lihat pada Gambar 4.22 dan Gambar 4.23.

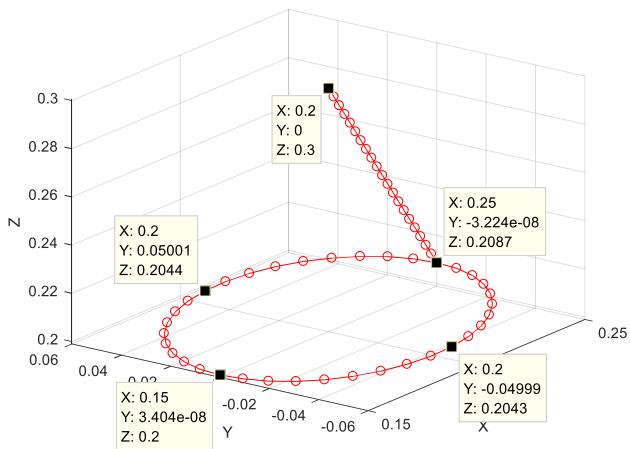


Gambar 4.34 Grafik *Error* Orientasi Keluaran Kinematika Balik Membuat Pola Lingkaran Miring



Gambar 4.35 Grafik *Error* Posisi Keluaran Kinematika Balik Membuat Pola Lingkaran Miring

Hasil plot tiga dimensi keluaran hasil kinematika balik dengan *neuro fuzzy* dapat di lihat pada Gambar 4.18 dimana titik-titik tersebut merupakan titik-titik pergerakan dari *end effector* manipulator robot denso dalam membuat pola lingkaran miring.



Gambar 4.36 Titik-Titik Keluaran Kinematika Balik Membuat Pola Lingkaran Miring

[Halaman ini sengaja dikosongkan]

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. Semakin kecil nilai laju pembelajaran (α) yang digunakan maka iterasi perhitungan agar hasil keluaran *neuro fuzzy* sesuai dengan ketelitian yang ditentukan akan semakin besar, dibuktikan dengan laju pembelajaran (α) 0.0001 percobaan 1 membutuhkan 59 ribu iterasi per-titik sedangkan laju pembelajaran (α) 0.0008 percobaan 1 membutuhkan 12 ribu iterasi per-titik.
2. Semakin kecil nilai laju pembelajaran (α) yang digunakan maka nilai *error* yang di dapat semakin kecil, dibuktikan dengan laju pembelajaran (α) 0.0001 percobaan 1 sumbu x terjadi *error* sebesar $2.10e-07$ sedangkan laju pembelajaran (α) 0.0008 percobaan 1 sumbu x terjadi *error* sebesar $1.30e-06$.
3. Semakin kecil nilai ketelitian maka iterasi perhitungan agar hasil keluaran *neuro fuzzy* sesuai dengan ketelitian yang ditentukan akan semakin besar, dibuktikan dengan. Ketelitian 0.00005 percobaan 1 sebanyak 43353 iterasi per-titik sedangkan 0.0005 percobaan 1 sebanyak 43327 iterasi per-titik.
4. Semakin kecil nilai ketelitian yang digunakan maka nilai *error* yang di dapat semakin kecil, dibuktikan dengan ketelitian 0.00005 percobaan 1 sumbu x terjadi *error* sebesar $5.41e-08$ sedangkan ketelitian 0.0005 percobaan 1 sumbu x terjadi *error* sebesar $1.19e-07$.
5. Penyelesaian kinematika balik dengan *neuro fuzzy* dapat membuat pola segitiga datar, segitiga miring, bulat datar dan bulat miring dengan *error* posisi dan orientasi < 0.00005 .

Jika menggunakan laju pembelajaran (α) diatas 0.0001 dengan ketelitian 0.00005 maka untuk mendapatkan hasil dari kinematika balik selalu terkena iterasi maksimum dan *error* yang di dapatkan > 0.00005 .

5.2. Saran

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

1. Dapat dilakukan implementasi pada manipulator robot denso VP-6242G.
2. Menambahkan perhitungan kecepatan dan dinamika pergerakan manipulator robot denso.

DAFTAR PUSTAKA

- [1] Jang Jyh-Shing Roger., Sun Chuen-Tsai., Mizutani Eiji., 1997 “*Neuro-Fuzzy and Soft Computing*”. Prentice Hall. United States of America.
- [2] Rangkuti, R. P., 2017. Kinematika Balik Menggunakan *Neuro Fuzzy* Pada Manipulator Robot Denso. Tugas Akhir. Teknik Elektro. Surabaya: Institut Teknologi Sepuluh Nopember.
- [3] Sirait, D. C., 2018. Perancangan Jaringan Saraf Tiruan Untuk Menyelesaikan Kinematika Balik Manipulator Robot Denso 6-DoF. Tugas Akhir. Teknik Elektro. Surabaya: Institut Teknologi Sepuluh Nopember.
- [4] Spong, M. W., Hutchinson, S. & Vidyasagar, M., 2004, “*Robot Dynamics and Control*”. 2nd ed. s.l.:s.n.
- [5] Craig, J. J., 2005, *Introduction to Robotics Mechanics and Control 3rd ed.* United States of America: Pearson Education, Inc.
- [6], n.d. *Quanser 6-Axis Articulated Robot*. Canada: Quanser, Inc.
- [7] Kusumadewi, Sri, Hartati, Sri., 2010, “*Neuro Fuzzy: Sistem Fuzzy & Jaringan Syaraf*”. Yogyakarta: GRAHA ILMU.
- [8] Prasetya, I. E., 2015. *Inverse Kinematics* dengan Solusi *Closed Form* Pada Robot Denso Manipulator. Tugas Akhir. Teknik Elektro. Surabaya: Institut Teknologi Sepuluh Nopember.
- [9] Corke, Peter I., *Robotics Toolbox for Matlab (Release 10)*, <http://www.petercorke.com/robot>.
- [10] Pertiwi, M. M., 2013. Perancangan Kontroler *Neuro Fuzzy* Untuk Pengaturan Kecepatan Motor Spindle Pada CNC Jenis *Milling*. Tugas Akhir. Teknik Elektro. Surabaya: Institut Teknologi Sepuluh Nopember.
- [11] Duka, A. V., 2013, *Neural Network Based Inverse Kinematics Solution for Trajectory Tracking of a Robotic Arm*. *Procedia Technology*. Romania. 12 (2014) 20 – 27.

[Halaman ini sengaja dikosongkan]

LAMPIRAN

Lampiran 1. Program Matlab Forward Kinematics

```
function [N] = fwrt(tt)
global a2 d1 d4 d6 a3
global c1 c2 c3 c4 c5 c6 s1 s2 s3 s4 s5 s6
global px py pz
a2= 0.21;d1= 0.28;d4= 0.21;d6= 0.070;a3=0.075;
t1=tt(1);
t2=tt(2);
t3=tt(3);
t4=tt(4);
t5=tt(5);
t6=tt(6);
%inisialisasi
c1=cosd(t1);c2=cosd(t2);c3=cosd(t3);c4=cosd(t4);
c5=cosd(t5);c6=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*a
2)+d1;
pitch=asind(ax);
roll=asind(sx/-cosd(pitch));
N=[px py pz roll pitch];

```

Lampiran 2. Program Matlab Inverse Kinematics

```

close all;
clear
clc;
%INPUT
t1= input('Masukkan X = ');
t2= input('Masukkan Y = ');
t3= input('Masukkan Z = ');
t4= input('Masukkan Roll = ');
t5= input('Masukkan Pitch = ');
%inisialisasi
tha=[180 140 30 20 20 0]; %Awal
xyzrpa=fk(tha);
xyzrpt=[t1 t2 t3 t4 t5];
dxyzrp=xyzrpt-xyzrpa;
dtot=sqrt(dxyzrp(1)^2+dxyzrp(2)^2+dxyzrp(3)^2);

```



```

d=0.005;
jdpoin=floor(dtot/d);
dx=dxyzrp(1)/jdpoin;
dy=dxyzrp(2)/jdpoin;
dz=dxyzrp(3)/jdpoin;
dr=dxyzrp(4)/jdpoin;
dp=dxyzrp(5)/jdpoin;
c1=rand(1);
c2=rand(1);
c3=rand(1);
c4=rand(1);
c5=rand(1);
c6=rand(1);
c7=rand(1);
c8=rand(1);
c9=rand(1);
c10=rand(1);
c11=rand(1);
c12=rand(1);
c13=rand(1);
c14=rand(1);
c15=rand(1);
c16=rand(1);
c17=rand(1);
c18=rand(1);
c19=rand(1);
c20=rand(1);

dx = zeros(2,1);
dy = zeros(2,1);
dz = zeros(2,1);
dr = zeros(2,1);
dp = zeros(2,1);
for p=1:jdpoin
    xyzrptar=zeros(p,5);
    xyzrptar(p,:)=xyzrpa+p*[dx dy dz dr dp];
    gradw1(p)=10;
    gradw2(p)=10;
    gradw3(p)=10;
    gradw4(p)=10;

```

```

gradw5(p)=10;
I(p)=0;
% Lapisan 1 : Fuzzifikasi
% fuzzifikasi delta_error x
if dx<=-1
    dxf(1) = 1;
elseif dx<1
    dxf(1) = (1-dx)/2;
    dxf(2) = (dx+1)/2;
else
    dxf(2) = 1;
end
% fuzzifikasi delta_error y
if dy<=-1
    dyf(1) = 1;
elseif dy<1
    dyf(1) = (1-dy)/2;
    dyf(2) = (dy+1)/2;
else
    dyf(2) = 1;
end
% fuzzifikasi delta_error z
if dz<=-1
    dzf(1) = 1;
elseif dz<1
    dzf(1) = (1-dz)/2;
    dzf(2) = (dz+1)/2;
else
    dzf(2) = 1;
end
% fuzzifikasi delta_error roll
if dr<=-1
    drf(1) = 1;
elseif dr<1
    drf(1) = (1-dr)/2;
    drf(2) = (dr+1)/2;
else
    drf(2) = 1;
end
% fuzzifikasi delta_error pitch

```

```

if dp<=-1
    dpf(1) = 1;
elseif dx<1
    dpf(1) = (1-dp)/2;
    dpf(2) = (dp+1)/2;
else
    dpf(2) = 1;
end
% Lapisan 2 RULEBASE
y0=min([dx(1) dy(1) dz(1) drf(1)
dpf(1)]);
y1=min([dx(1) dy(1) dz(1) drf(1)
dpf(2)]);
y2=min([dx(1) dy(1) dz(1) drf(2)
dpf(1)]);
y3=min([dx(1) dy(1) dz(1) drf(2)
dpf(2)]);
y4=min([dx(1) dy(1) dz(2) drf(1)
dpf(1)]);
y5=min([dx(1) dy(1) dz(2) drf(1)
dpf(2)]);
y6=min([dx(1) dy(1) dz(2) drf(2)
dpf(1)]);
y7=min([dx(1) dy(1) dz(2) drf(2)
dpf(2)]);
y8=min([dx(1) dy(2) dz(1) drf(1)
dpf(1)]);
y9=min([dx(1) dy(2) dz(1) drf(1)
dpf(2)]);
y10=min([dx(1) dy(2) dz(1) drf(2)
dpf(1)]);
y11=min([dx(1) dy(2) dz(1) drf(2)
dpf(2)]);
y12=min([dx(1) dy(2) dz(2) drf(1)
dpf(1)]);
y13=min([dx(1) dy(2) dz(2) drf(1)
dpf(2)]);
y14=min([dx(1) dy(2) dz(2) drf(2)
dpf(1)]);

```

```

    y15=min([dx f(1) dy f(2) dz f(2) dr f(2)
dp f(2)]);
    y16=min([dx f(2) dy f(1) dz f(1) dr f(1)
dp f(1)]);
    y17=min([dx f(2) dy f(1) dz f(1) dr f(1)
dp f(2)]);
    y18=min([dx f(2) dy f(1) dz f(1) dr f(2)
dp f(1)]);
    y19=min([dx f(2) dy f(1) dz f(1) dr f(2)
dp f(2)]);
    y20=min([dx f(2) dy f(1) dz f(2) dr f(1)
dp f(1)]);
    y21=min([dx f(2) dy f(1) dz f(2) dr f(1)
dp f(2)]);
    y22=min([dx f(2) dy f(1) dz f(2) dr f(2)
dp f(1)]);
    y23=min([dx f(2) dy f(1) dz f(2) dr f(2)
dp f(2)]);
    y24=min([dx f(2) dy f(2) dz f(1) dr f(1)
dp f(1)]);
    y25=min([dx f(2) dy f(2) dz f(1) dr f(1)
dp f(2)]);
    y26=min([dx f(2) dy f(2) dz f(1) dr f(2)
dp f(1)]);
    y27=min([dx f(2) dy f(2) dz f(1) dr f(2)
dp f(2)]);
    y28=min([dx f(2) dy f(2) dz f(2) dr f(1)
dp f(1)]);
    y29=min([dx f(2) dy f(2) dz f(2) dr f(1)
dp f(2)]);
    y30=min([dx f(2) dy f(2) dz f(2) dr f(2)
dp f(1)]);
    y31=min([dx f(2) dy f(2) dz f(2) dr f(2)
dp f(2)]);
    %inference rule
    p1=max([y0 y1 y2 y3 y4 y5 y6 y7]);
    p2=max([y8 y9 y10 y11 y12 y13 y14 y15]);
    p3=max([y16 y17 y18 y19 y20 y21 y22 y23]);
    p4=max([y24 y25 y26 y27 y28 y29 y30 y31]);
    %normalisai

```

```

q1=p1/(p1+p2+p3+p4);
q2=p2/(p1+p2+p3+p4);
q3=p3/(p1+p2+p3+p4);
q4=p4/(p1+p2+p3+p4);
while (abs(gradw1(p))>0.00005 ||
abs(gradw2(p))>0.00005 || abs(gradw3(p))>0.00005
|| abs(gradw4(p))>0.00005 ||
abs(gradw5(p))>0.00005)
%defuzyfikasi
r11=(q1*c1)+(q2*c2)+(q3*c3)+(q4*c4);
r21=q1+q2+q3+q4;
r12=(q1*c5)+(q2*c6)+(q3*c7)+(q4*c8);
r22=q1+q2+q3+q4;
r13=(q1*c9)+(q2*c10)+(q3*c11)+(q4*c12);
r23=q1+q2+q3+q4;
r14=(q1*c13)+(q2*c14)+(q3*c15)+(q4*c16);
r24=q1+q2+q3+q4;
r15=(q1*c17)+(q2*c18)+(q3*c19)+(q4*c20);
r25=q1+q2+q3+q4;
dtn1=r11/r21;
dtn2=r12/r22;
dtn3=r13/r23;
dtn4=r14/r24;
dtn5=r15/r25;
thad=tha+[dtn1 dtn2 dtn3 dtn4 dtn5 0];
if thad(1)>180
    thad=thad-[320 0 0 0 0 0];
elseif thad(1)<-180
    thad=thad+[320 0 0 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
xyzrpn=fk(thad);

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

sd2(p)=sqrt((xyzrptar(p,1)^2)+(xyzrptar(p,2)^2))
;

sd3(p)=xyzrptar(p,3);
sd4(p)=xyzrptar(p,4);
sd5(p)=xyzrptar(p,5);
sd1n(p)=atan2(xyzrpn(2),xyzrpn(1))*180/pi;
sd2n(p)=sqrt((xyzrpn(1)^2)+(xyzrpn(2)^2));
sd3n(p)=xyzrpn(3);
sd4n(p)=xyzrpn(4);
sd5n(p)=xyzrpn(5);
%Backpropagation
gradw1(p)=sd1(p)-sd1n(p);
gradw2(p)=sd2(p)-sd2n(p);
gradw3(p)=sd3(p)-sd3n(p);
gradw4(p)=sd4(p)-sd4n(p);

```

```

gradw5(p)=sd5(p)-sd5n(p);
%Inialisasi update bobot
tnd2=-1;
if sign(gradw2(p))>0
    tnd2=1;
end
tnd3=1;
if sign(gradw3(p))>0
    tnd3=-1;
end
tnd4=-1;
if sign(gradw4(p))>0
    tnd4=1;
end
tnd5=1;
if sign(gradw5(p))>0
    tnd5=-1;
end
clc;
I(p)=I(p)+1;
I(p)
I;
[sd1(p) sd1n(p) gradw1(p)
sd2(p) sd2n(p) gradw2(p)
sd3(p) sd3n(p) gradw3(p)
sd4(p) sd4n(p) gradw4(p)
sd5(p) sd5n(p) gradw5(p)]
[thad(:,1:5)
xyzrpa
xyzrptar(p,:)
xyzrpn]
[p]
alp1=0.0005;
alp2=0.0005;
alp3=0.0005;
alp4=0.0005;
alp5=0.0005;
if abs(gradw1(p))<0.0008
    alp1=0.0001;
end

```

```

if abs(gradw2(p)) < 0.0015
    alp2 = 0.0001;
end
if abs(gradw3(p)) < 0.0015
    alp3 = 0.0001;
end
if abs(gradw4(p)) < 0.0015
    alp4 = 0.0001;
end
if abs(gradw5(p)) < 0.0015
    alp5 = 0.0001;
end
if I(p) > 2000000
    break
end
% update bobot
c1 = c1 + alp1 * sign(gradw1(p)) * q1;
c2 = c2 + alp1 * sign(gradw1(p)) * q2;
c3 = c3 + alp1 * sign(gradw1(p)) * q3;
c4 = c4 + alp1 * sign(gradw1(p)) * q4;
c5 = c5 + alp2 * tnd2 * q1;
c6 = c6 + alp2 * tnd2 * q2;
c7 = c7 + alp2 * tnd2 * q3;
c8 = c8 + alp2 * tnd2 * q4;
c9 = c9 + alp3 * tnd3 * q1;
c10 = c10 + alp3 * tnd3 * q2;
c11 = c11 + alp3 * tnd3 * q3;
c12 = c12 + alp3 * tnd3 * q4;
c13 = c13 + alp4 * tnd4 * q1;
c14 = c14 + alp4 * tnd4 * q2;
c15 = c15 + alp4 * tnd4 * q3;
c16 = c16 + alp4 * tnd4 * q4;
c17 = c17 + alp5 * tnd5 * q1;
c18 = c18 + alp5 * tnd5 * q2;
c19 = c19 + alp5 * tnd5 * q3;
c20 = c20 + alp5 * tnd5 * q4;
end
hasil(p, :) = thad
end

```


Lampiran 3. Posisi dan orientasi target segitiga datar

Posisi (m)			Orientasi (derajat)	
X	Y	Z	Roll	Pitch
0.2	0	0.3	0	0
0.2	-0.1	0.2	0	0
0.2	0.1	0.2	0	0
0.3	0	0.2	0	0
0.2	-0.1	0.2	0	0

Lampiran 4. Posisi dan orientasi *neuro fuzzy* segitiga datar

Posisi (m)			Orientasi (derajat)	
X	Y	Z	Roll	Pitch
0.20004	-0.1	0.2000	3.42e-05	-2.14e-05
0.19995	-0.0999	0.1999	-1.50e-05	2.05e-05
0.19995	0.0999	0.1999	-4.51e-05	1.11e-05
0.3	6.38e-09	0.2000	5.94e-06	4.72e-05
0.2000	-0.1	0.2000	3.42e-05	-2.14e-05

Lampiran 5. Error posisi dan orientasi *neuro fuzzy* segitiga datar

Posisi (m)			Orientasi (derajat)	
X	Y	Z	Roll	Pitch
0	0	0	0	0
4.45e-05	-2.24e-05	3.27e-05	1.50e-05	-2.05e-05
4.47e-05	2.23e-05	5.83e-06	4.51e-05	-1.11e-05
-1.37e-07	-6.38e-09	-2.04e-07	-5.94e-06	-4.72e-05
-4.46e-05	2.23e-05	-5.76e-06	-3.42e-05	2.14e-05

Lampiran 6. Posisi dan orientasi target segitiga miring

Posisi (m)			Orientasi (derajat)	
X	Y	Z	Roll	Pitch
0.2	0	0.3	0	0
0.16223	-0.1	0.23169	0	0
0.16223	0.1	0.23169	0	0
0.26071	0	0.24905	0	0
0.16223	-0.1	0.23169	0	0

Lampiran 7. Posisi dan orientasi *neuro fuzzy* segitiga miring

Posisi (m)			Orientasi (derajat)	
X	Y	Z	Roll	Pitch
0.2	0	0.3	0	0
0.1621	-0.0999	0.2316	4.64e-05	-2.79e-05
0.1621	0.0999	0.2316	-3.46e-06	-7.59e-07
0.2607	-5.85e-08	0.2490	3.94e-05	3.51e-05
0.1622	-0.1	0.2317	-1.63e-06	1.64e-05

Lampiran 8. Error posisi dan orientasi *neuro fuzzy* segitiga miring

Posisi (m)			Orientasi (derajat)	
X	Y	Z	Roll	Pitch
0	0	0	0	0
4.25e-05	-2.61e-05	2.69e-05	-4.64e-05	2.79e-05
4.25e-05	2.61e-05	2.69e-05	3.46e-06	7.59e-07
-1.34e-08	5.85e-08	-1.87e-07	-3.94e-05	-3.51e-05
-4.25e-05	2.61e-05	-1.68e-05	1.63e-06	-1.64e-05

Lampiran 9. Posisi dan orientasi target lingkaran datar

Titik	Posisi (m)			Orientasi (derajat)	
	X	Y	Z	Roll	Pitch
1	0.25	0	0.2	0	0
2	0.24924	0.008682	0.2	0	0
3	0.246985	0.017101	0.2	0	0
4	0.243301	0.025	0.2	0	0
5	0.238302	0.032139	0.2	0	0
6	0.232139	0.038302	0.2	0	0
7	0.225	0.043301	0.2	0	0
8	0.217101	0.046985	0.2	0	0
9	0.208682	0.04924	0.2	0	0
10	0.2	0.05	0.2	0	0
11	0.191318	0.04924	0.2	0	0
12	0.182899	0.046985	0.2	0	0

Titik	Posisi (m)			Orientasi (derajat)	
	X	Y	Z	Roll	Pitch
13	0.175	0.043301	0.2	0	0
14	0.167861	0.038302	0.2	0	0
15	0.161698	0.032139	0.2	0	0
16	0.156699	0.025	0.2	0	0
17	0.153015	0.017101	0.2	0	0
18	0.15076	0.008682	0.2	0	0
19	0.15	0	0.2	0	0
20	0.15076	-0.00868	0.2	0	0
21	0.153015	-0.0171	0.2	0	0
22	0.156699	-0.025	0.2	0	0
23	0.161698	-0.03214	0.2	0	0
24	0.167861	-0.0383	0.2	0	0
25	0.175	-0.0433	0.2	0	0
26	0.182899	-0.04698	0.2	0	0
27	0.191318	-0.04924	0.2	0	0
28	0.2	-0.05	0.2	0	0
29	0.208682	-0.04924	0.2	0	0
30	0.217101	-0.04698	0.2	0	0
31	0.225	-0.0433	0.2	0	0
32	0.232139	-0.0383	0.2	0	0
33	0.238302	-0.03214	0.2	0	0
34	0.243301	-0.025	0.2	0	0
35	0.246985	-0.0171	0.2	0	0
36	0.24924	-0.00868	0.2	0	0
37	0.25	0	0.2	0	0

Lampiran 10. Posisi dan orientasi *neuro fuzzy* lingkaran datar

Titik	Posisi (m)			Orientasi (derajat)	
	X	Y	Z	Roll	Pitch
1	0.24995	-4.19e-08	0.19996	5.50e-07	1.42e-05
2	0.24929	0.00868	0.20003	3.76e-05	-2.14e-05
3	0.24693	0.01710	0.19998	-6.32e-06	9.31e-06
4	0.24325	0.02499	0.19996	-3.53e-05	1.99e-05
5	0.23835	0.03215	0.20001	-3.42e-05	-2.55e-05
6	0.23219	0.03831	0.20003	-4.08e-06	-2.28e-05
7	0.22505	0.04331	0.20003	-2.73e-05	-3.51e-07
8	0.21715	0.04699	0.20004	1.92e-05	1.16e-06
9	0.20873	0.04925	0.20004	-1.89e-05	-1.98e-05
10	0.19995	0.04998	0.19999	1.08e-05	1.53e-05
11	0.19126	0.04922	0.19999	-4.79e-05	6.53e-07
12	0.18285	0.04697	0.19999	1.64e-05	7.43e-06
13	0.17495	0.04328	0.19999	-3.95e-05	1.31e-05
14	0.16791	0.03831	0.20004	2.99e-05	2.36e-05
15	0.16175	0.03214	0.20004	4.59e-05	1.91e-05
16	0.15675	0.02501	0.20004	2.55e-05	-1.43e-05
17	0.15306	0.01711	0.20003	1.93e-05	-1.70e-05
18	0.15071	0.00868	0.19996	-2.51e-05	2.16e-05
19	0.14999	-4.81e-08	0.19999	6.12e-06	-4.56e-05
20	0.15081	-0.00868	0.20002	-1.96e-05	-7.12e-07
21	0.15296	-0.01709	0.19997	9.20e-06	-2.24e-05
22	0.15665	-0.024992	0.19995	5.15e-06	4.37e-06
23	0.16165	-0.03213	0.19995	-1.29e-05	5.15e-06
24	0.16791	-0.03831	0.20000	-1.33e-05	1.13e-05
25	0.17505	-0.04331	0.20001	3.92e-05	-9.69e-06
26	0.18295	-0.04699	0.20002	2.28e-05	2.23e-05
27	0.19137	-0.04925	0.20002	2.58e-06	5.00e-06
28	0.20005	-0.05001	0.20001	2.25e-05	-8.48e-06
29	0.20873	-0.04925	0.20000	-3.33e-05	-4.41e-06
30	0.21715	-0.04699	0.20000	-1.76e-05	-5.83e-06

Titik	Posisi (m)			Orientasi (derajat)	
	X	Y	Z	Roll	Pitch
31	0.22504	-0.04331	0.20000	-2.09e-06	3.47e-05
32	0.23209	-0.03829	0.19996	3.15e-05	1.71e-05
33	0.23825	-0.03213	0.19996	-4.24e-05	2.55e-05
34	0.24325	-0.02499	0.19996	2.16e-05	-1.48e-05
35	0.24693	-0.01709	0.19997	2.00e-05	1.38e-05
36	0.24929	-0.00868	0.20003	1.37e-05	-6.94e-06
37	0.24995	-4.19e-08	0.19996	5.50e-07	1.42e-05

Lampiran 11. Error posisi dan orientasi *neuro fuzzy* lingkaran datar

Titik	Posisi (m)			Orientasi (derajat)	
	X	y	z	Roll	Pitch
1	4.99e-05	4.19e-08	3.23e-05	-5.50e-07	-1.42e-05
2	-4.99e-05	-1.64e-06	-3.31e-05	-3.76e-05	2.14e-05
3	4.98e-05	3.34e-06	1.39e-05	6.32e-06	-9.31e-06
4	4.96e-05	5.05e-06	3.42e-05	3.53e-05	-1.99e-05
5	-4.94e-05	-6.72e-06	-1.63e-05	3.42e-05	2.55e-05
6	-4.92e-05	-8.20e-06	-3.76e-05	4.08e-06	2.28e-05
7	-4.90e-05	-9.48e-06	-3.94e-05	2.73e-05	3.51e-07
8	-4.88e-05	-1.05e-05	-4.11e-05	-1.92e-05	-1.16e-06
9	-4.86e-05	-1.150e-05	-4.44e-05	1.89e-05	1.98e-05
10	4.84e-05	1.21e-05	2.60e-06	-1.08e-05	-1.53e-05
11	4.83e-05	1.24e-05	4.54e-06	4.79e-05	-6.53e-07
12	4.83e-05	1.24e-05	5.31e-06	-1.64e-05	-7.43e-06
13	4.84e-05	1.19e-05	4.14e-06	3.95e-05	-1.31e-05
14	-4.87e-05	-1.10e-05	-4.41e-05	-2.99e-05	-2.36e-05
15	-4.90e-05	-9.68e-06	-4.28e-05	-4.59e-05	-1.91e-05
16	-4.93e-05	-7.82e-06	-4.13e-05	-2.55e-05	1.43e-05
17	-4.96e-05	-5.49e-06	-3.99e-05	-1.93e-05	1.70e-05
18	4.98e-05	2.81e-06	3.95e-05	2.51e-05	-2.16e-05
19	9.66e-08	4.81e-08	1.91e-07	-6.12e-06	4.56e-05
20	-4.98e-05	2.84e-06	-2.16e-05	1.96e-05	7.12e-07
21	4.96e-05	-5.48e-06	2.64e-05	-9.20e-06	2.24e-05
22	4.93e-05	-7.84e-06	4.08e-05	-5.15e-06	-4.37e-06
23	4.89e-05	-9.76e-06	4.24e-05	1.29e-05	-5.15e-06

Titik	Posisi (m)			Orientasi (derajat)	
	X	Y	Z	Roll	Pitch
24	-4.87e-05	1.10e-05	-4.69e-06	1.33e-05	-1.13e-05
25	-4.84e-05	1.20e-05	-1.18e-05	-3.92e-05	9.69e-06
26	-4.84e-05	1.24e-05	-2.14e-05	-2.28e-05	-2.23e-05
27	-4.83e-05	1.24e-05	-2.41e-05	-2.58e-06	-5.01e-06
28	-4.84e-05	1.21e-05	-1.79e-05	-2.25e-05	8.48e-06
29	-4.85e-05	1.14e-05	-9.62e-06	3.33e-05	4.41e-06
30	-4.88e-05	1.05e-05	-4.95e-06	1.76e-05	5.83e-06
31	-4.21e-05	8.10e-06	-2.82e-06	2.09e-06	-3.47e-05
32	4.92e-05	-8.06e-06	3.71e-05	-3.15e-05	-1.71e-05
33	4.95e-05	-6.73e-06	3.54e-05	4.24e-05	-2.55e-05
34	4.97e-05	-5.07e-06	3.41e-05	-2.16e-05	1.48e-05
35	4.97e-05	-3.44e-06	2.20e-05	-2.01e-05	-1.38e-05
36	-4.99e-05	1.74e-06	-3.30e-05	-1.37e-05	6.94e-06
37	4.99e-05	4.19e-08	3.23e-05	-5.50e-07	-1.42e-05

Lampiran 12. Posisi dan orientasi target lingkaran miring

Titik	Posisi (m)			Orientasi (derajat)	
	X	y	z	Roll	Pitch
1	0.25	0	0.208704	0	0
2	0.24924	0.008682	0.208638	0	0
3	0.246985	0.017101	0.208441	0	0
4	0.243301	0.025	0.20812	0	0
5	0.238302	0.032139	0.207684	0	0
6	0.232139	0.038302	0.207147	0	0
7	0.225	0.043301	0.206525	0	0
8	0.217101	0.046985	0.205837	0	0
9	0.208682	0.04924	0.205103	0	0
10	0.2	0.05	0.204346	0	0
11	0.191318	0.04924	0.203589	0	0
12	0.182899	0.046985	0.202856	0	0
13	0.175	0.043301	0.202167	0	0
14	0.167861	0.038302	0.201545	0	0
15	0.161698	0.032139	0.201008	0	0

Titik	Posisi (m)			Orientasi	
	X	Y	Z	Roll	Pitch
16	0.156699	0.025	0.200572	0	0
17	0.153015	0.017101	0.200251	0	0
18	0.15076	0.008682	0.200055	0	0
19	0.15	0	0.199988	0	0
20	0.15076	-0.00868	0.200055	0	0
21	0.153015	-0.0171	0.200251	0	0
22	0.156699	-0.025	0.200572	0	0
23	0.161698	-0.03214	0.201008	0	0
24	0.167861	-0.0383	0.201545	0	0
25	0.175	-0.0433	0.202167	0	0
26	0.182899	-0.04698	0.202856	0	0
27	0.191318	-0.04924	0.203589	0	0
28	0.2	-0.05	0.204346	0	0
29	0.208682	-0.04924	0.205103	0	0
30	0.217101	-0.04698	0.205837	0	0
31	0.225	-0.0433	0.206525	0	0
32	0.232139	-0.0383	0.207147	0	0
33	0.238302	-0.03214	0.207684	0	0
34	0.243301	-0.025	0.20812	0	0
35	0.246985	-0.0171	0.208441	0	0
36	0.24924	-0.00868	0.208638	0	0
37	0.25	0	0.208704	0	0

Lampiran 13. Posisi dan orientasi *neuro fuzzy* lingkaran miring

Titik	Posisi (m)			Orientasi (derajat)	
	X	y	Z	Roll	Pitch
1	0.20002	0	0.30003	0	0
2	0.24996	-3.22e-08	0.20870	1.61e-05	-3.14e-05
3	0.24919	0.00868	0.20862	2.30e-05	1.81e-05
4	0.24694	0.01709	0.20843	1.29e-05	-4.31e-05
5	0.24325	0.02499	0.20810	3.12e-05	1.70e-06
6	0.23825	0.03213	0.20763	-9.45e-06	-5.88e-06
7	0.23218	0.03831	0.20716	-4.18e-07	-1.48e-05

Titik	Posisi (m)			Orientasi (derajat)	
	X	Y	Z	Roll	Pitch
8	0.22504	0.04331	0.20654	-4.34e-05	9.04e-06
9	0.21714	0.04699	0.20585	1.22e-06	4.71e-06
10	0.20873	0.04925	0.20512	-4.12e-05	-1.78e-05
11	0.20004	0.05001	0.20437	-4.39e-05	-1.21e-05
12	0.19136	0.04925	0.20361	4.67e-05	-1.92e-05
13	0.18294	0.04699	0.20288	8.86e-06	2.10e-05
14	0.17504	0.04331	0.20220	3.55e-05	-5.21e-06
15	0.16790	0.03831	0.20158	4.66e-05	-2.20e-05
16	0.16174	0.03214	0.20104	-2.19e-05	1.04e-06
17	0.15674	0.02500	0.20061	4.56e-05	-3.11e-06
18	0.15306	0.01710	0.20028	-3.47e-05	-2.50e-05
19	0.15071	0.00867	0.20001	3.43e-05	9.12e-06
20	0.15000	3.40e-08	0.19998	4.55e-05	2.32e-05
21	0.15080	-0.00868	0.20007	6.73e-06	1.74e-05
22	0.15296	-0.01709	0.20022	-2.40e-06	-5.12e-06
23	0.15664	-0.02499	0.20053	1.49e-05	-7.65e-06
24	0.16164	-0.03213	0.20097	4.17e-05	2.26e-05
25	0.16781	-0.03828	0.20150	4.95e-05	1.17e-06
26	0.17495	-0.04328	0.20212	1.25e-05	8.12e-06
27	0.18285	-0.04696	0.20280	-6.72e-06	3.93e-06
28	0.19126	-0.04922	0.20354	-1.44e-05	8.44e-06
29	0.19995	-0.04998	0.20430	1.09e-05	2.38e-06
30	0.20863	-0.04922	0.20507	2.59e-05	-1.96e-05
31	0.21705	-0.04696	0.20581	-3.24e-05	1.05e-05
32	0.22495	-0.04329	0.20650	1.88e-05	9.41e-06
33	0.23208	-0.03829	0.20713	-1.33e-06	5.90e-06
34	0.23825	-0.03213	0.20767	-3.86e-05	4.99e-06
35	0.24325	-0.02499	0.20810	1.04e-05	-1.27e-05
36	0.24703	-0.01710	0.20848	-1.76e-05	-1.63e-05
37	0.24928	-0.00868	0.20864	4.32e-05	-1.35e-05

Lampiran 14. *Error* posisi dan orientasi *neuro fuzzy* lingkaran miring

Titik	Posisi (m)			Orientasi (derajat)	
	X	y	z	Roll	Pitch
1	3.69e-05	3.22e-08	2.94e-06	-1.61e-05	3.14e-05
2	4.98e-05	1.73e-06	1.36e-05	-2.30e-05	-1.81e-05
3	4.22e-05	3.00e-06	2.93e-06	-1.29e-05	4.31e-05
4	4.96e-05	5.20e-06	1.13e-05	-3.12e-05	-1.70e-06
5	4.95e-05	6.60e-06	4.54e-05	9.45e-06	5.8e-06
6	-4.92e-05	-8.12e-06	-1.57e-05	4.18e-07	1.48e-05
7	-4.90e-05	-9.51e-06	-1.85e-05	4.34e-05	-9.04e-06
8	-4.88e-05	-1.05e-05	-2.13e-05	-1.22e-06	-4.71e-06
9	-4.86e-05	-1.14e-05	-2.43e-05	4.12e-05	1.78e-05
10	-4.83e-05	-1.21e-05	-2.74e-05	4.39e-05	1.21e-05
11	-4.84e-05	-1.23e-05	-2.99e-05	-4.67e-05	1.92e-05
12	-4.83e-05	-1.24e-05	-3.38e-05	-8.86e-06	-2.10e-05
13	-4.85e-05	-1.19e-05	-3.52e-05	-3.55e-05	5.21e-06
14	-4.87e-05	-1.10e-05	-3.66e-05	-4.66e-05	2.20e-05
15	-4.90e-05	-9.75e-06	-3.77e-05	2.19e-05	-1.04e-06
16	-4.93e-05	-7.83e-06	-3.83e-05	-4.56e-05	3.11e-06
17	-4.96e-05	-5.61e-06	-3.85e-05	3.47e-05	2.50e-05
18	4.99e-05	2.90e-06	3.92e-05	-3.43e-05	-9.12e-06
19	-5.04e-09	-3.40e-08	-1.72e-07	-4.55e-05	-2.32e-05
20	-4.98e-05	2.89e-06	-2.10e-05	-6.73e-06	-1.74e-05
21	4.96e-05	-5.55e-06	2.47e-05	2.40e-06	5.12e-06
22	4.93e-05	-7.81e-06	3.79e-05	-1.49e-05	7.65e-06
23	4.89e-05	-9.71e-06	3.72e-05	-4.17e-05	-2.26e-05
24	4.86e-05	-1.10e-05	3.87e-05	-4.95e-05	-1.17e-06
25	4.84e-05	-1.19e-05	4.42e-05	-1.25e-05	-8.12e-06
26	4.83e-05	-1.24e-05	4.62e-05	6.72e-06	-3.93e-06
27	4.83e-05	-1.24e-05	4.48e-05	1.44e-05	-8.44e-06
28	4.85e-05	-1.20e-05	3.99e-05	-1.09e-05	-2.38e-06
29	4.86e-05	-1.14e-05	3.19e-05	-2.59e-05	1.96e-05
30	4.87e-05	-1.05e-05	2.22e-05	3.24e-05	-1.05e-05
31	4.89e-05	-9.44e-06	1.81e-05	-1.88e-05	-9.41e-06
32	4.92e-05	-8.19e-06	1.56e-05	1.33e-06	-5.90e-06
33	4.94e-05	-6.70e-06	1.32e-05	3.86e-05	-4.99e-06
34	4.97e-05	-4.99e-06	1.13e-05	-1.04e-05	1.27e-05

Titik	Posisi (m)			Orientasi (derajat)	
	X	Y	Z	Roll	Pitch
35	-4.98e-05	3.38e-06	-4.50e-05	1.76e-05	1.63e-05
36	-4.99e-05	1.83e-06	-9.47e-06	-4.32e-05	1.35e-05
37	3.69e-05	3.22e-08	2.94e-06	-1.61e-05	3.14e-05

RIWAYAT HIDUP



Jevta Marihot Pasaribu, lahir di Surabaya pada tanggal 18 Maret 1996. Anak kedua dari pasangan Repelita Pasaribu dan Agustina Napitupulu. Setelah menempuh pendidikan formal di SDN Tanah Kali Kedinding IV Surabaya, SMP YP 17 Surabaya dan SMAN 19 Surabaya, penulis melanjutkan studi Diploma 3 jurusan Teknik Elektro Otomasi di Institut Teknologi Sepuluh Nopember Surabaya dan lulus tahun 2017. Kemudian melanjutkan studi Lintas Jalur Sarjana di Institut Teknologi Sepuluh Nopember Surabaya dengan mengambil Jurusan Teknik Elektro, Bidang Studi Teknik Sistem Pengaturan.

Kontak: marihot1996@gmail.com

[Halaman ini sengaja dikosongkan]