INVERSE KINEMATICS MANIPULATOR 3D BERBASIS LOGIKA FUZZY

TUGAS AKHIR

Diajukan untuk melengkapi salah satu syarat untuk memperoleh gelar sarjana Teknik pada Jurusan Teknik Elektro Universitas Islam Indonesia Yogyakarta

oleh :

Nama : **Dody Sujoko Aprian Azmi**

No. Mahasiswa : **02 524 002**

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

YOGYAKARTA

2007

LEMBAR PENGESAHAN PEMBIMBING

INVERSE KINEMATICS MANIPULATOR 3D BERBASIS LOGIKA FUZZY

TUGAS AKHIR

Disusun oleh :

Nama : Dody Sujoko Aprian Azmi

No. Mahasiswa : 02 524 002

Yogyakarta, Juli 2007

Pembimbing I	Pembimbing II
(Ir. Hj. Budi Astuti, MT)	(Dwi Ana Ratna Wati, ST)

LEMBAR PENGESAHAN PENGUJI

INVERSE KINEMATICS MANIPULATOR 3D BERBASIS LOGIKA FUZZY

TUGAS AKHIR

Oleh :

Nama : Dody Sujoko Aprian Azmi

No. Mahasiswa : 02 524 002

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Teknik Elektro
Fakultas Teknologi Industri Universitas Islam Indonesia
Jogjakarta, 24 Juli 2007

Tim Penguji

Ir. Hj. Budi Astuti, MT
Ketua

Dwi Ana Ratna Wati, ST
Anggota I

Medilla Kusrianto, ST
Anggota II

Mengetahui,
Ketua Jurusan Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia

UNIVERSITAS ISLAM INDONESIA ★ YOGYAKARTA ★ FAKULTAS TEKNOLOGI INDUSTRI

Ir. Hari Purwono, ST, M.Sc

HALAMAN PERSEMBAHAN

Kupersembahkan Tugas Akhir ini Dengan Penuh rasa Hormat Kepada :

Alhamdulillah atas segala nikmat & karunia dari ALLAH SWT dan syukurku atas terselesaikannya karya ini, kupersembahkan dengan segala kerendahan hati kepadaMu ya Rabb, Penguasa Langit dan Bumi.

Papaku ***joko subagio*** *dan mamaku* ***Nezmailinda*** *tercinta,*

"Terima kasih selalu memberikan dukungan baik moril dan materiil dan selalu memberikan yang terbaik buat diri saya. Dan tak henti hentinya melimpahi kasih sayang dan doa".

Adikku ***Nita*** *dan* ***Sisi***

"Eratkan tangan kita untuk membahagiakan papamama tercinta"

Untuk Seseorang yang sangat aku sayangi Untuk menemaniku dalam keadaan sedih senang, tangis tawa,

Semoga semua hari-hari yang telah terlewati menjadi kenangan yang indah dan tak terlupakan
Amien Ya Rabbal 'Aalamien.....
Dan

Untuk Almamaterku serta untuk Bangsa dan Negaraku Yang Tercinta

MOTTO

- **"Allah tidak membebani seseorang melebihi kemampuan yang diberikan-Nya, dan sesudah kesukaran Allah pasti akan memberikan kelapangan." (Qs. Ath Thalaaq : 7)**
- **"Dan Allah pasti menolongmu dengan pertolongan yang kuat." (Qs. Al Fath : 3)**
- **"Apabila Ia memutus sesuatu Ia hanya berfirman, *"Jadilah"*, maka ia pun jadi." (Qs. Maryam : 35)**
- **"Allah mengilhamkan kepada hati dua pilihan : kejahatan dan kebaikan." (Qs. Asy Syams : 8)**
- Roger Dawson : "Senyuman adalah hal mudah untuk dilakukan namun memiliki makna yang besar dalam hidup ini."
- Hidup ini adalah bagaimana kita memaknainya.

KATA PENGANTAR

بسم الله الرحمن الرحيم

Assalamu'alaikum Wr. Wb.

Alhamdulillah, segala puji dan syukur ditunjukan hanya bagi Allah SWT, sang Esa pemilik alam semesta. Semoga kesejahteraan diberikan bagi rosul–Nya , Muhammad SAW, sang mustofa. Atas rahmat dan taufik–Nya penulis dapat menyelesaikan tugas akhir dengan judul **"Inverse Kinematics Manipulator 3D Berbasis Logika Fuzzy"** dengan lancar.

Adapun maksud dan tujuan penyusunan tugas akhir ini adalah untuk melengkapi salah satu syarat dalam menempuh gelar sarjana pada Jurusan Teknik Elektro Universitas Islam Indonesia, Yogyakarta. Disamping itu untuk menambah pengetahuan terhadap ilmu yang telah dipelajari di bangku perkuliahan untuk diterapkan di masyarakat.

Pada kesempatan ini dengan segala rasa syukur dan kerendahan hati penulis sampaikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Kedua orang tua dan adik - adikku tercinta yang telah memberikan dorongan serta doa restu dalam menggapai cita-cita penulis.
2. Bpk Fathul Wahid, ST. M.Sc. Selaku dekan Fakultas Teknologi Industri (FTI) Universitas Islam Indonesia (UII).
3. Bpk Tito Yuwono, ST. selaku Kajur Teknik Elektro.
4. Bpk Yusuf Azis Amrulloh, ST. selaku Sekjur Teknik Elektro.
5. Ibu Ir. Hj. Budi Astuti, MT. Selaku dosen pembimbing I atas bimbingan dan masukannya.
6. Ibu Dwi Ana Ratna Wati, ST. selaku dosen pembimbing II dan Ka.Lab pemrograman Matlab atas waktu, kebaikan, kesabaran dan ilmunya.
7. Ibu Sri Kusumadewi, S.Si. MT. Selaku Kajur Informatika atas buku-bukunya.

8. Dosen dan karyawan Fakultas Teknologi Industri UII, Ka.Lab dan laboran jurusan Teknik Elektro yang telah banyak memberikan ilmu yang sangat bermanfaat bagi penulis.
9. Seluruh Staf TU dan Perpustakaan Jurusan yang telah banyak membantu segala kelancaran penulis.
10. Untuk Rengganis Sekar Kinanti atas perhatian, semangat, doa, dan dukungan moril yang telah diberikan kepada penulis selama ini.
11. Seluruh mahasiswa jurusan Teknik Elektro UII, Ass. Lab Komputer Dasar (Komdas).
12. Teman-teman seperjuanganku (Ridi, Ida, Sony, Eko, Rama, nugroho, Fitrianto) yang telah memberi arti persahabatan selama ini.
13. Seluruh saudaraku dan teman-teman yang telah memberi motivasi, dorongan serta doa bagi penulis.
14. Seluruh pihak yang tidak dapat disebutkan satu-persatu, yang telah memberikan *support* dan doa.

Akhir kata penulis hanya dapat memanjatkan doa kehadirat Allah SWT, kiranya dapat melimpahkan rahmat dan kesejahteraan kepada semua pihak atas segala budi baik yang telah diberikan kepada penulis. Amien...

Wassalamu'alaikum Wr. Wb

Yogyakarta, Juli 2007

Dody Sujuko Aprian Azmi

ABSTRAKSI

Perkembangan teknologi robot saat ini sangat berkaitan dengan kepresisian sebuah robot yaitu ketepatan posisi end effector dan kecermatan robot dalam bergerak untuk melaksanakan pekerjaannya sesuai yang diperintahkan pengguna (*user*). Melalui PC (*Personal Computer*) yang sudah dilengkapi dengan program user interface sebuah lengan robot dapat dikendalikan pergerakannya sesuai dengan keinginan pemakai. Selain dapat mengendalikan, user dapat mempelajari persamaan persamaan *Inverse Kinematics* untuk mengetahui cara mencari posisi sudut end effector lengan robot. Oleh karena itu dengan persamaan *inverse kinematics* dapat mensimulasikan sebuah lengan robot dan menentukan posisi sudutnya. *Inverse kinematics* manipulator itu sendiri adalah pergerakkan lengan robot dengan input berupa koordinat orientasi dan menghasilkan output berupa sudut tiap joint dari setiap joint lengan robot.

Sebagai pengendalinya digunakan logika fuzzy. Logika fuzzy dirancang untuk difungsikan sebagai pengendali yang akan menghasilkan sudut untuk tiap jointnya. Manipulatornya itu sendiri terdiri dari 3 link dan 3 joint yang dapat bergerak secara 3 dimensi. Dalam perancangan *inverse kinematics* ini juga dibutuhkan metoda *forward kinematics* (sudut tiap joint dari lengan robot sudah diketahui sehingga dapat dicari posisi orientasi), metoda *forward* disini digunakan sebagai penentu posisi koordinat orientasi dari manipulator serta sebagai pembanding.

Metoda fuzzy yang digunakan adalah metoda Mamdani (*Max-Min*), pengendali fuzzy akan memberikan keluaran berupa sudut teta1, teta2, dan teta3 sesuai dengan input yaitu berupa koordinat orientasi x,y, dan z. Logika fuzzy dirancang menggunakan 5 mf dengan 122 rule yang dipakai, dengan memakai 122 rule maka dihasilkan error yang setelah dirata – ratakan hasilnya cukup kecil dari 20 data yang diambil secara acak.

DAFTAR ISI

Halaman Judul i
Lembar Pengesahan Pembimbing ii
Lembar Pengesahan Penguji iii
Halaman Motto iv
Kata Pengantar v
Abstraksi vii
Daftar Isi viii
Daftar Gambar xi
Daftar Tabel xiii

BAB I PENDAHULUAN

1.1 Latar Belakang Masalah 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan Penelitian 4
1.5 Sistematika Penulisan 4

BAB II DASAR TEORI

2.1 Sistem Kendali Fuzzy 6
2.1.1 Himpunan Fuzzy 6
2.1.2 Logika Fuzzy 8
2.1.3 Logika Fuzzy Untuk Sistem *Inverse Kinematics* Manipulator 3D 9
2.1.4 Fungsi Keanggotaan 10
2.1.5 Operator Dasar Zadeh Untuk Operasi Himpunan Fuzzy 20
2.1.6 Penalaran Monoton 21
2.1.7 Fuzzifikasi 22
2.1.8 Sistem Pengambilan Keputusan (*Fuzzy Inference System)* Dan Teknik *Reasoning* 23

2.1.8.1 Metode MAX-MIN 24
2.1.8.2 Metode MAX-DOT 25
2.1.9 Defuzzifikasi 26
2.1.9.1 Metode *Mean Of Maximum (MOM)* 26
2.1.9.2 Metode *Center Of Average (COA)* 27
2.1.10 Metode Mamdani 27
2.2 Matlab 31
2.2.1 *Graphical User Interface* (GUI) Pada Matlab 32
2.3 Definisi Robot 33
2.4 Struktur Robot 34
2.5 Penggerak Robot 34
2.6 *Kinematics* Manipulator 35
2.6.1 *Forward Manipulator Kinematics* 37
2.6.2 *Inverse Manipulator Kinematics* 37
BAB III PERANCANGAN SISTEM
3.1 Deskripsi Simulasi 44
3.2 Perancangan Sistem 45
3.2.1 Perancangan Sistem Fuzzy 45
3.2.2 Perancangan GUI (*Graphical User Interface*) 52
3.3 Hasil Analisis 58
3.3.1 Kebutuhan Proses 58
3.3.2 Kebutuhan Keluaran 59
BAB IV ANALISA DAN PEMBAHASAN
4.1 Pengambilan Data *Inverse* 122 Rule 61
4.2 Pengambilan Data *Inverse* 80 Rule 62
4.3 Pengambilan Data *Inverse* 38 Rule 63
4.4 Analisa Perbandingan Error Fuzzy Berdasarkan 122 Rule, 80 rule, dan 38 rule 64
4.5 Pengambilan Data *Inverse Kinematics* Manipulator 3D Bila x<y, x>y, dan x = y 64

4.5.1 Pengambilan Data *Inverse Kinematics* Manipulator 3D Bila x<y 65
4.5.2 Pengambilan Data *Inverse Kinematics* Manipulator 3D Bila x>y 66
4.5.3 Pengambilan Data *Inverse Kinematics* Manipulator 3D Bila x = y 67
4.5.4 Analisa Perbandingan Data *Inverse Kinematics* Manipulator x<y, x>y, dan x = y 68

BAB V PENUTUP

5.1 Kesimpulan 69
5.2 Saran 70

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1 Himpunan Fuzzy Pada Variabel Temperatur 8
Gambar 2.2 Representasi Linier Naik 11
Gambar 2.3 Representasi Linier Turun 12
Gambar 2.4 Kurva Segitiga 13
Gambar 2.5 Kurva Trapesium 13
Gambar 2.6 Daerah ' Bahu ' Pada Variabel TEMPERATUR 14
Gambar 2.7 Himpunan Fuzzy Dengan Kurva – S (pertumbuhan) 15
Gambar 2.8 Himpunan Fuzzy Dengan Kurva – S (Penyusutan) 16
Gambar 2.9 Karakteristik Fungsi Kurva – S 16
Gambar 2.10 Karakteristik Fungsional Kurva π 18
Gambar 2.11 Karakteristik Fungsional Kurva Beta 19
Gambar 2.12 Karakteristik Fungsional Kurva GAUSS 20
Gambar 2.13 Fungsi Implikasi : MIN 25
Gambar 2.14 Fungsi Implikasi : DOT 25
Gambar 2.15 Diagram Blok Perancangan Logika Fuzzy – Metode Mamdani 31
Gambar 2.16 Layout Editor Pada GUI 32
Gambar 2.17 Contoh Manipulator Lengan Robot 35
Gambar 2.18 *Inverse kinematics* 3 Dimensi 38
Gambar 3.1 Flowchart Jalannya Program *Inverse Kinematics* Manipulator 3D 41
Gambar 3.2 Diagram Rancangan Sistem *Inverse Kinematics* 42
Gambar 3.3 Input Variabel X Dan Y 47
Gambar 3.4 Input Variabel Z 48
Gambar 3.5 Output Variabel Teta 1 48
Gambar 3.6 Ouput Variabel Teta 2 Dan Teta 3 49
Gambar 3.7 Blok Diagram Sistem Fuzzy 49
Gambar 3.8 Fis Editor *Inverse Kinematics* Manipulator 3D Pada MATLAB 51

Gambar 3.9 Rancangan Simulasi *Inverse Kinematics* Manipulator 3D Berbasis Logika Fuzzy 53
Gambar 3.10 Rancangan Dasar GUI *Inverse Kinematics* Manipulator 3D Berbasis Logika Fuzzy.....54
Gambar 3.11 Tampilan GUI *Inverse Kinematics* Manipulator 3D Berbasis Logika Fuzzy Yang Sudah Dieksekusi.....57
Gambar 4.1 Tampilan GUI Bila x < y 65
Gambar 4.2 Tampilan GUI Bila x > y 66
Gambar 4.3 Tampilan GUI Bila x = y 67

DAFTAR TABEL

Tabel 3.1 Kombinasi Data *Forward* 46
Tabel 4.1 Pengujian Sistem Fuzzy Untuk 122 Rule Fuzzy 61
Tabel 4.2 Pengujian Sistem Fuzzy Untuk 80 Rule Fuzzy 62
Tabel 4.3 Pengujian Sistem Fuzzy Untuk 38 Rule Fuzzy 63
Tabel 4.4 Pengujian Sistem Fuzzy bila x<y 65
Tabel 4.5 Pengujian Sistem Fuzzy bila x>y 66
Tabel 4.6 Pengujian Sistem Fuzzy bila x=y 67

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi di segala bidang kehidupan pada masa sekarang ini sudah sangat pesat, terutama untuk bidang industri. Saat ini banyak industri yang telah menggunakan teknologi yang canggih. Salah satunya yaitu penggunaan teknologi robot. Dengan menggunakan teknologi robot produsen dapat mempercepat proses produksi, lebih presisi, mengefisienkan waktu dan masih banyak manfaat lainnya yang didapat dari penggunaan teknologi robot. Di Indonesia sendiri teknologi robotika masih jauh tertinggal jika dibandingkan negara-negara tetangga dan negara maju lainnya. Begitu juga dengan sebuah komputer atau yang lebih sering disebut PC (Personal Computer).

Perkembangan teknologi robot saat ini sangat berkaitan dengan kepresisian sebuah robot yaitu ketepatan posisi end effector dan kecermatan robot dalam bergerak untuk melaksanakan pekerjaannya sesuai yang diperintahkan pengguna (user). Melalui PC (Personal Computer) yang sudah dilengkapi dengan program user interface sebuah lengan robot dapat dikendalikan pergerakannya sesuai dengan keinginan pemakai. Selain dapat mengendalikan, user dapat mempelajari persamaan persamaan *Inverse Kinematics* untuk mengetahui cara mencari posisi sudut end

effector lengan robot. Oleh karena itu dengan persamaan inverse kinematics kita dapat mensimulasikan sebuah lengan robot dan menentukan posisi sudutnya.

Untuk merancang *inverse kinematics* manipulator 3D digunakan matlab sebagai bahasa pemrograman. Matlab adalah bahasa yang memiliki *high performance* untuk komputasi teknis. Matlab merupakan integrasi komputasi, visualisasi, dan pemrograman dalam lingkungan yang *easy to use* yang mana problem dan solusi diekspresikan dalam notasi matematis yang umum. Dengan kemampuan tersebut, Matlab menjadi perangkat bantu yang sangat bermanfaat dalam berbagai bidang ilmu teknik diantaranya dalam bidang robotika. Dengan menggunakan matlab penulis dapat merancang logika fuzzy yang telah tersedia pada fis editor matlab

Logika fuzzy digunakan sebagai pengendali dari manipulator 3D (lengan robot) secara inverse kinematics untuk menghasilkan output berupa sudut untuk tiap joint pada manipulator tersebut. Logika fuzzy adalah suatu cara yang tepat untuk memetakan suatu ruang input ke dalam suatu ruang output. Antara input dan output terdapat suatu relasi yang dapat menghubungkan keduanya untuk saling bekerja menghasilkan suatu keputusan. Dalam perancangan *inverse kinematics* manipulator 3D ini ruang input yang digunakan yaitu berupa koordinat orientasi x,y, dan z dengan output berupa sudut untuk tiap joint manipulator teta1, teta2, dan teta3.

1.2 Rumusan Masalah

Berdasarkan uraian yang telah dijelaskan diatas, maka dapat diambil suatu rumusan masalah sebagai berikut : "Bagaimana merancang simulasi dari inverse kinematics manipulator 3D berbasis logika fuzzy ?"

1.3 Batasan Masalah

Dengan adanya batasan masalah maka perancangan *invers kinematics* manipulator 3D ini dapat lebih disederhanakan dan diarahkan penelitiannya dan pembuatannya agar sistem tidak menyimpang dari apa yang diteliti dan dikembangkan. Batasan-batasannya adalah sebagai berikut :

1. Tidak membahas sistem mekanis pada lengan robot *inverse kinematics* manipulator 3D.
2. Pembuatan sistem ini menggunakan perangkat lunak Matlab 7.
3. Tampilan dengan GUI.
4. Input dari sistem ini yaitu koordinat dengan ouputnya berupa sudut.
5. Input variabel z (koordinat z) pada sistem fuzzy memiliki numeris yang bernilai positif (+) secara keseluruhan dengan batasan 0 hingga 30.
6. Output untuk variabel teta 2 (θ_2) dan teta 3 (θ_3) pada sistem fuzzy memiliki batasan mulai dari -90 hingga +90 dan tidak bergerak bebas.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian dan perancangan sistem ini adalah :

1. Merancang dan mensimulasikan sebuah lengan robot yang dapat bergerakkan sesuai dengan masukan berupa nilai koordinat posisi end effector pada user interface dikomputer sebagai sarana untuk mempelajari *inverse kinematics* manipulator 3D berbasis logika fuzzy
2. Mempelajari, menganalisis, dan mendesign sistem fuzzy menggunakan toolbox pada Matlab sebagai media pelatihan.
3. Dapat memahami dengan jelas dan benar tentang konsep logika fuzzy yang digunakan sebagai media kontrol.

1.5 Sistematika Penulisan

Sistematika penulisan tugas akhir ini terdiri dari 5 bab bagian isi laporan, dengan penjelasan bab sebagai berikut :

BAB I : PENDAHULUAN

Berisi tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penulisan, dan sistematika penulisan.

BAB II : LANDASAN TEORI

Bab ini memuat teori-teori yang berhubungan dengan penelitian dan juga berisi dasar teori yang berhubungan dengan fungsi yang akan digunakan dalam sistim.

BAB III : PERANCANGAN SISTEM

Bab ini menjelaskan metode-metode perancangan yang digunakan, cara mensimulasikan rancangan dan pengujian sistem yang telah dibuat, pembagian fungsi kerja dalam diagram blok serta berisi lebih terperinci tentang apa yang telah disampaikan pada proposal Tugas Akhir ini.

BAB IV : PENGUJIAN, ANALISIS DAN PEMBAHASAN

Bab ini membahas tentang hasil pengujian dan analisis dari sistem yang dibuat dibandingkan dengan dasar teori sistem atau sistem yang lain yang dapat dijadikan sebagai pembanding .

BAB V : PENUTUP

Bab ini memuat kesimpulan dan saran-saran dari proses perancangan, pensimulasian sistem, serta keterbatasan-keterbatasan yang ditemukan dan juga asumsi-asumsi yang dibuat selama melakukan tugas akhir.

BAB II

DASAR TEORI

2.1 Sistem Kendali Fuzzy

2.1.1 Himpunan Fuzzy

Himpunan fuzzy yang menjadi dasar dari sistem fuzzy merupakan pengembangan dari teori tradisional yang hanya membagi keanggotaannya menjadi dua keadaan saja yaitu aktif atau 1 dan tidak aktif atau 0. Teori klasik ini kemudian dimodifikasi pertama kali oleh Zadeh menjadi suatu teori himpunan yang beranggotakan nilai pada interval dari 0 sampai 1. Himpunan fuzzy ini lebih kompleks dari teori sebelumnya karena memiliki lebih banyak informasi, namun ketepatannya lebih tinggi dengan prosedur dan aturan yang mendasarinya.

Untuk mendefinisikan suatu himpunan, setiap elemen pada semesta wacana harus diberi nilai yang menunjukkan apakah elemen tersebut dalam himpunan atau tidak. Untuk himpunan bukan fuzzy, nilai ini terbatas pada 0 dan 1, yang menunjukkan anggota himpunan dan bukan anggota himpunan. Untuk suatu semesta wacana U, himpunan fuzzy ditentukan oleh fungsi keanggotaan yang memetakan anggota-anggota semesta wacana U ke suatu derajad yang nilainya berada pada interval 0 dan 1. Representasi himpunan fuzzy F dalam U ditunjukkan dalam pasangan elemen dan tingkat dari nilai keanggotaan :

$$F = \{(u,\mu_F(u) \mid u \varepsilon U\} \qquad (2.1)$$

Apabila U kontinyu, himpunan fuzzy F ditulis sebagai :

$$F = \int_U \mu_F (u)/u \tag{2.2}$$

Dan bila U diskret, himpunan fuzzy ditunjukkan dalam :

$$F = \sum \mu_F (u_i)/u_i \tag{2.3}$$

Himpunan fuzzy memiliki dua cara untuk mendefinisikan keanggotaannya, numeris dan fungsional. Definisi numeris menunjukkan derajad fungsi keanggotaan sebagai vektor nilai dengan dimensi sesuai tingkat diskretisasi, sedangkan secara fungsional ditunjukkan dalam persamaan analitik untuk tiap elemen yang didefinisikan. Ada beberapa hal yang perlu diketahui dalam memahami sistem fuzzy, yaitu :

a. Variabel fuzzy

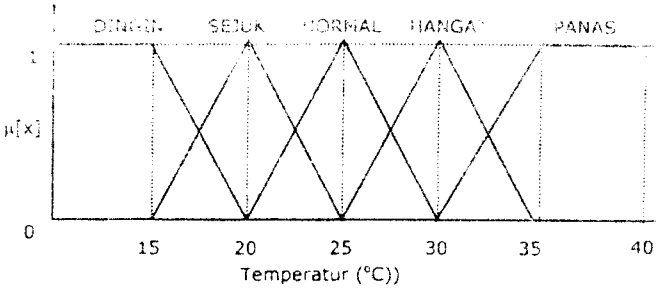
   Variabel fuzzy merupakan variabel yang hendak dibahas dalam suatu sistem fuzzy. Contoh : umur, temperatur, permintaan, dsb.

b. Himpunan fuzzy

   Himpunan fuzzy merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel fuzzy.

   Contoh :

   Variabel temperatur, terbagi menjadi 5 himpunan fuzzy, yaitu : DINGIN, SEJUK, NORMAL, HANGAT, dan PANAS



Gambar 2.1 Himpunan Fuzzy Pada Variabel Temperatur

c. Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variable fuzzy, contohnya : semesta pembicaraan untuk variable temperature : [0 40].

c. Domain

Domain himpunan fuzzy adalah keseluruhan nilai yang diizinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan fuzzy. Contohnya : DINGIN = [0, 20], SEJUK = [15, 25], NORMAL = [20, 30],HANGAT = [25, 35], PANAS = [30, 40].

2.1.2 Logika Fuzzy

Sebagaimana halnya himpunan fuzzy yang merupakan perluasan dari himpunan klasik, logika fuzzy merupakan perluasan dari logika klasik. Logika klasik hanya memiliki dua nilai kebenaran, yakni benar dan salah. Sebagaimana hubungan antara himpunan klasik dengan logika klasik, maka terdapat kaitan pula antara himpunan fuzzy dengan logika fuzzy, contohnya operasi gabungan (*union*) pada

himpunan berkaitan dengan logika *OR*, operasi irisan dengan logika *AND* dan operasi komplemen dengan logika *NOT*. Selanjutnya derajad keanggotaan pada himpunan fuzzy berkaitan dengan nilai kebenaran sebuah pernyataan pada logika fuzzy.

Logika fuzzy adalah suatu cara yang tepat untuk memetakan suatu ruang input ke dalam suatu ruang output. Antara input dan output terdapat suatu relasi. yang dapat menghubungkan keduanya untuk saling bekerja menghasilkan suatu keputusan.

Alasan digunakannya logika Fuzzy, antara lain :

1. Konsep logika fuzzy mudah dimengerti. Konsep matematis yang mendasari penalaran fuzzy sangat sederhana dan mudah dimengerti.
2. Logika fuzzy sangat fleksible.
3. Logika fuzzy memiliki toleransi terhadap data-data yang tidak tepat.
4. Logika fuzzy mampu memodelkan fungsi-fungsi non linear yang sangat kompleks.
5. Logika fuzzy dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Logika fuzzy dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
7. Logika fuzzy didasarkan pada bahasa alami.

2.1.3 Logika Fuzzy Untuk Sistem Inverse Kinematics Manipulator 3D

Beberapa istilah yang digunakan pada simulasi lengan robot *inverse kinematics* manipulator 3D berbasis logika fuzzy antara lain untuk sebaran posisi

koordinat yang dituju oleh ujung manipulator yaitu berupa titik koordinat x,y, dan z adalah negatif big (jauh), negatif small (dekat), zero (sedang), positif small (dekat), positif big (jauh), sedang untuk sudut teta 1(θ_1), teta 2(θ_2), dan teta 3(θ_3), adalah negatif big (jauh), negatif small (dekat), zero (sedang), positif small (dekat), positif big (jauh). Jelas istilah tersebut dapat menimbulkan kemenduaan (*ambiquity*) dalam pengertiannya. Logika fuzzy dapat mengubah kemenduaan tersebut kedalam model matematis sehingga dapat diproses lebih lanjut untuk dapat diterapkan dalam sebuah sistem kendali. Menggunakan teori himpunan fuzzy bahasa dapat diwakili oleh sebuah daerah yang mempunyai jangkauan tertentu yang menunjukkan derajat keanggotaannya. Untuk pengaturan simulasi *inverse kinematics* manipulator 3D derajat keanggotaannya itu adalah α (x) untuk x adalah posisi koordinat yang dituju oleh ujung manipulator. Derajat keanggotaan tersebut meempunyai nilai bergradasi sehingga mengurangi lonjakan pada sistem.

Sistem pengendalian fuzzy yang dirancang mempunyai tiga masukan dan enam keluaran, dimana masukan yaitu berupa posisi koordinat yang dituju oleh ujung manipulator yaitu berupa titik koordinat x,y, dan z dan keluaran berupa sudut teta 1(θ_1), teta 2(θ_2), dan teta 3(θ_3)

2.1.4 Fungsi keanggotaan

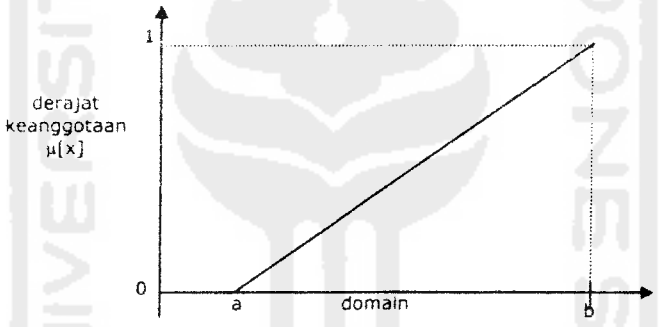
Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik input data kedalam nilai keanggotaannya atau derajat keanggotaan yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat

digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang bisa digunakan.

a. Representasi Linier

Pada representasi linier, pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Ada 2 keadaan himpunan fuzzy yang linier diantaranya :

1. Kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak kekanan menuju ke nilai domain yang memilki derajat keanggotaan lebih tinggi

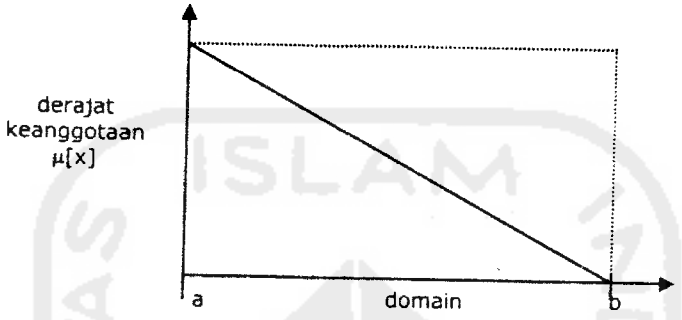


Gambar 2.2 Representasi Linier Naik

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \\ (x-a) / (b-a); & a \leq x \leq b \\ 1; & x \geq b \end{cases} \qquad (2.4)$$

2. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memilki derajat keanggotaan lebih rendah.



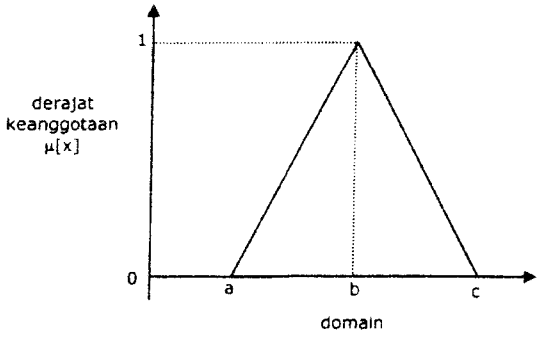
Gambar 2.3 Representasi Linier Turun

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} (b-x) / (b-a); & a \leq x \leq b \\ 0; & x \geq b \end{cases} \quad (2.5)$$

b. Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (linier). Seperti terlihat pada gambar berikut :



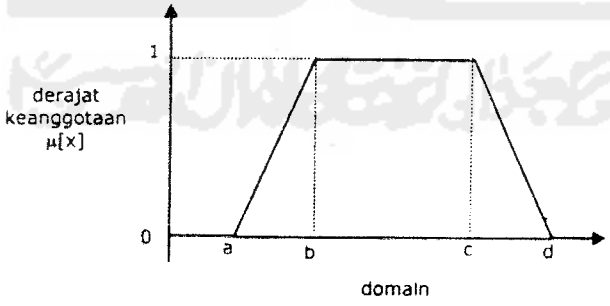
Gambar 2.4 Kurva Segitiga

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ (x-a) / (b-a); & a \leq x \leq b \\ (c-x) / (c-b); & b \leq x \leq c \end{cases} \quad (2.6)$$

c. Representasi Kurva Trapesium

Kurva trapesium pada dasarnya seperti bentuk kurva segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1.



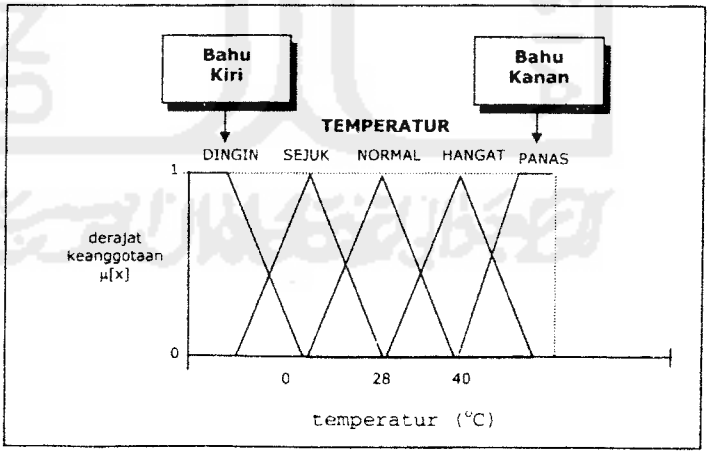
Gambar 2.5 Kurva Trapesium

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq d \\ (x-a) / (b-a); & a \leq x \leq b \\ 1; & b \leq x \leq c \\ (d-x) / (d-c); & c \leq x \leq d \end{cases} \quad (2.7)$$

d. Representasi Kurva Bentuk Bahu

Daerah yang terletak di tengah – tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun (misalkan :DINGIN bergerak ke SEJUK bergerak ke HANGAT bergerak ke PANAS). Himpunan fuzzy ' bahu ' digunakan unuk mengakhiri variabel suatu daerah fuzzy.Dimana bahu kiri bergerak dari benar ke salah, sedangkan bahu kanan bergerak dari salah ke benar. Berikut contoh yang menunjukkan variabel TEMPERATUR dengan daerah bahunya.

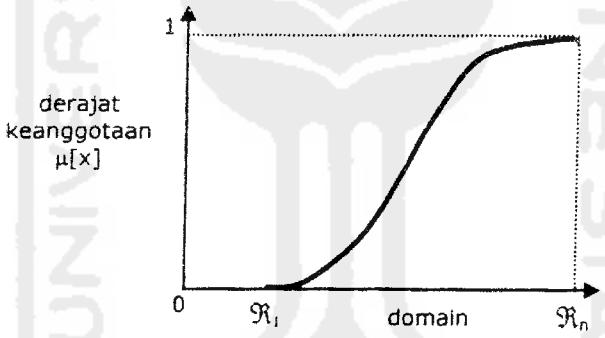


Gambar 2.6 Daerah ' Bahu ' Pada Variabel TEMPERATUR

e. Representasi Kurva - S

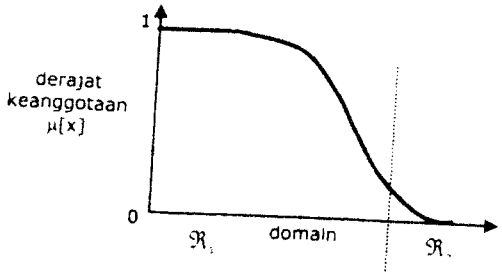
Kurva pertumbuhan dan penyusutan merupakan kurva – S atau *sigmoid* yang berhubungan dengan kenaikan dan penurunan permukaan secara tak linier. Berikut adalah keterangan mengenai kurva pertumbuhan dan kurva penyusutan diantaranya :

1. Kurva – S untuk pertumbuhan akan bergerak dari sisi paling kiri (nilai keanggotaan = 0) ke sisi paling kanan (nilai keanggotaan = 1). Fungsi keanggotaannya akan bertumpu pada 50% nilai keanggotaaannya yang sering disebut dengan titikinfleksi.



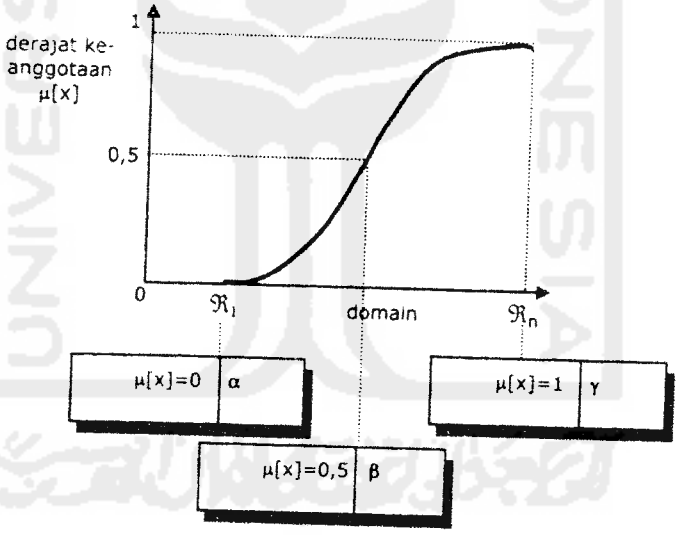
Gambar 2.7 Himpunan Fuzzy Dengan Kurva – S (pertumbuhan)

2. Kurva – S penyusutan akan bergerak dari sisi paling kanan (nilai keanggotaan = 1) ke sisi paling kiri (nilai keanggotaan = 0).



Gambar 2.8 Himpunan Fuzzy Dengan Kurva – S (Penyusutan)

Kurva – S didefinisikan dengan menggunakan 3 parameter, yaitu : nilai keanggotaan nol (α), nilai keanggotaan lengkap (γ), dan titik infleksi atau crossover (β) yaitu titik yang memiliki domain 50% benar.



Gambar 2.9 Karakteristik Fungsi Kurva - S

Fungsi keanggotaan pada kurva pertumbuhan adalah :

$$S(x;\alpha;\beta;\gamma) = \begin{cases} 0 ; & x \le \alpha \\ 2((x-\alpha)/(\gamma-\alpha))^2 ; & \alpha \le x \le \beta \\ 1-2((\gamma-x)/(\gamma-\alpha))^2; & \beta \le x \le \gamma \\ 1 ; & x \ge \gamma \end{cases} \quad (2.8)$$

Sedangkan fungsi keanggotaan pada kurva penyusutan adalah :

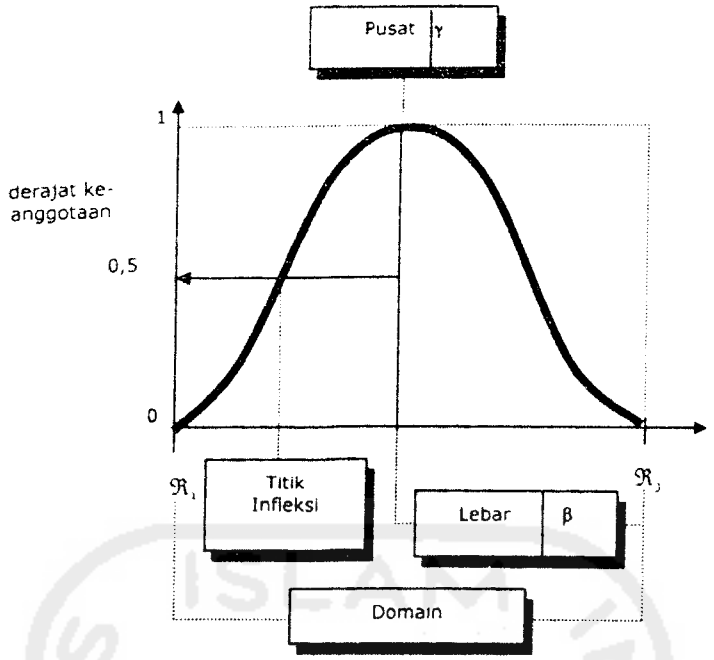
$$S(x;\alpha;\beta;\gamma) = \begin{cases} 1 ; & x \le \alpha \\ 1-2((x-\alpha)/(\gamma-\alpha))^2 ; & \alpha \le x \le \beta \\ 2((\gamma-x)/(\gamma-\alpha))^2; & \beta \le x \le \gamma \\ 0 ; & x \ge \gamma \end{cases} \quad (2.9)$$

f. Representasi Kurva Bentuk Lonceng (Bell Curve)

Kurva berbentuk lonceng ini terbagi atas 3 kelas, yaitu : himpunan fuzzy π, beta, dan Gauss. Perbedaan ketiga kurva ini terletak pada gradiennya.

1. Kurva π

Kurva π berbentuk lonceng dengan drajat keanggotaan 1 terletak pada pusat dengan domain (γ), dan lebar kurva (β).



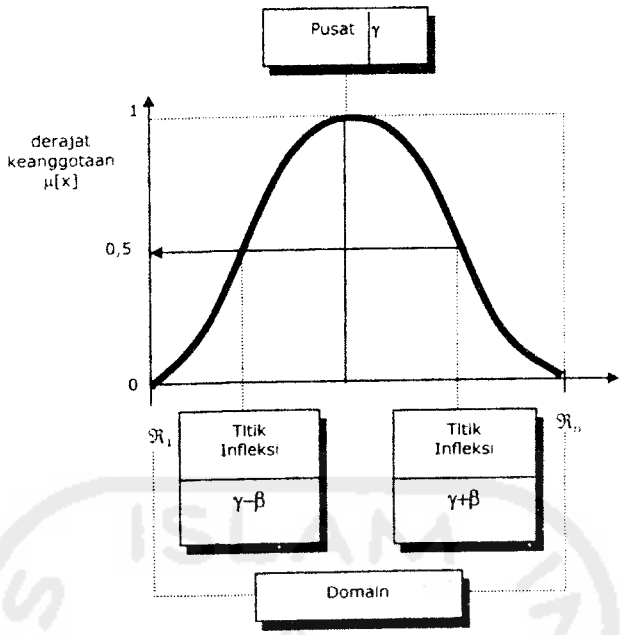
Gambar 2.10 Karakteristik Fungsional Kurva π

Fungsi Keanggotaan :

$$\beta\Pi(x,\beta,\gamma) = \begin{cases} S\left(x;\gamma-\beta,\gamma-\frac{\beta}{2},\gamma\right) & \rightarrow x \le \gamma \\ 1-S\left(x;\gamma-\beta,\gamma-\frac{\beta}{2},\gamma,\beta\right) & \rightarrow x > \gamma \end{cases} \quad (2.10)$$

2. Kurva Beta

Kurva beta juga berbentuk lonceng namun lebih rapat. Kurva ini didefinisikan dengan 2 parameter, yaitu nilai pada domain yang menunjukkan pusat kurva (γ), dan setengah lebar kurva (β).



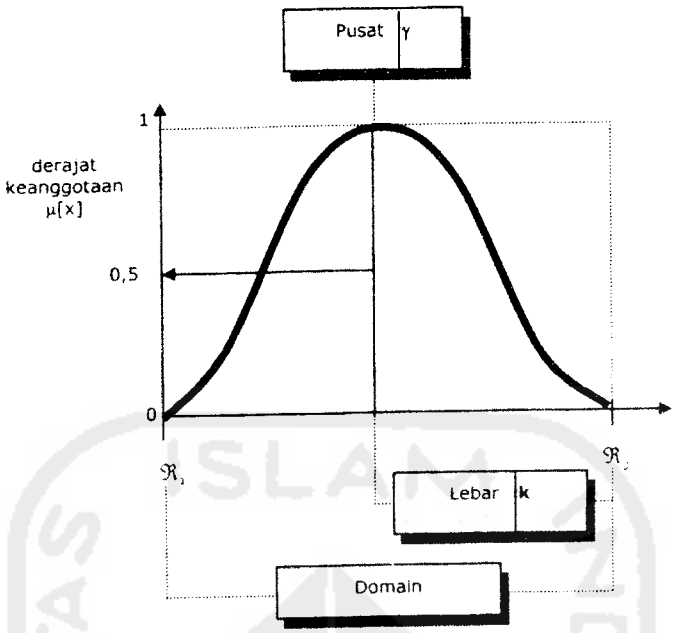
Gambar 2.11 Karakteristik Fungsional Kurva Beta

Fungsi Keanggotaaan :

$$B(x, \gamma, \beta) = \frac{1}{1+\left(\frac{x-\gamma}{\beta}\right)^2} \qquad (2.11)$$

2. Kurva GAUSS

Kurva GAUSS juga menggunakan (γ) untuk menunjukkan nilai domain pada pusat kurva.



Gambar 2.12 Karakteristik Fungsional Kurva GAUSS

Fungsi Keanggotaan :

$$G(x;k;\gamma) = e^{-k(\gamma-x)2} \quad (2.12)$$

2.1.5 Operator Dasar Zadeh Untuk Operasi Himpunan Fuzzy

Ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasikan dan memodifikasi himpunan fuzzy. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan sering dikenal dengan nama *fire strength* atau α-predikat. Ada 3 operator dasar yang diciptakan oleh zadeh, yaitu :

1. Operator AND

Operator ini berhubungan dengan operasi interseksi pada himpunan. α-predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan – himpunan yang bersangkutan.

$$\mu_{A\cap B} = \min(\mu_A[x], \mu_B[y]) \quad (2.13)$$

2. Operator OR

Operator ini berhubungan dengan operasi union pada himpunan. . α-predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan – himpunan yang bersangkutan.

$$\mu_{A\cup B} = \max(\mu_A[x], \mu_B[y]) \quad (2.13)$$

3. Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan. .α-predikat sebagi hasil operasi dengan operator NOT diperoleh dengan mengurangi nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

$$\mu_{A'} = 1 - \mu_A[x] \quad (2.14)$$

2.1.6 Penalaran Monoton

Metode penalaran secara monoton digunakan sebagai dasar untuk teknik implikasi fuzzy. Meskipun penalaran ini sudah jarang sekali digunakan, namun

terkadang masih digunakan untuk penskalaan fuzzy. Jika 2 daerah fuzzy direlasikan dengan implikasi sederhana sebagai berikut :

IF x is A THEN y is B

Transfer fungsi :

$y = f((x,A),B)$

maka system fuzzy dapat berjalan tanpa harus melalau komposisi dan dekomposisi fuzzy. Nilai output dapat diestiminasi secara langsung dari nilai keanggotaan yang berhubungan dengan antesendennnya.

2.1.7 Fuzzifikasi

Fuzzifikasi adalah proses pemetaan dari variabel masukan ke dalam himpunan dalam suatu semesta wacana. Proses fuzzifikasi ditunjukkan melalui persamaan :

X = Fuzzifier (Xo)

Xo merupakan vektor masukan himpunan tegas, x adalah vektor himpunan *fuzzy* hasil proses fuzzifikasi, dan fuzzifier adalah operator fuzzifikasi.

Beberapa strategi fuzzifikasi :

a. Fuzzy singleton

   Nilai fuzzy ditunjukkan melalui suatu nilai tegas. Strategi ini yang paling sering digunakan karena natural dan paling mudah diimplementasikan.

b. Fuzzy number

Nilai fuzzy diberikan melalui suatu nilai tertentu dengan interval kepastian. Fuzzy number merupakan nilai subyektif karena nilai yang diberikan adalah Penilaian, bukan hasil pengukuran.

c. Fuzzy random

Nilai yang dihasilkan merupakan nilai random yang bisa dikatakan sebagai nilai objektif karena menggunakan teori probabilitas dalam merepresentasikan data.

2.1.8 Sistem Pengambilan Keputusan (*Fuzzy Inference System*) Dan Teknik *Reasoning*

Inferensi Fuzzifikasi dalam hal ini digunakan untuk merumuskan pemetaan himpunan input ke himpunan output dengan prinsip logika fuzzy. Inferensi ini menjadi basis peraturan dalam pengambilan keputusan. Aturan yang digunakan menggunakan aturan jika-maka *(if-then)*.

Dalam penalaran logika fuzzy ada dua tipe utama inferensi fuzzy :

1. Inferensi berdasar komposisi. Semua aturan dalam inferensi ini dikombinasikan dalam suatu relasi fuzzy dan dilihat sebagai sebuah aturan *fuzzy if-then* tunggal.
2. Inferensi berdasar aturan individu. Setiap aturan menghasilkan sebuah keluaran himpunan fuzzy dan secara keseluruhan merupakan komposisi keluaran himpunan fuzzy yang dihasilkan setiap aturan.

Teknik reasoning digunakan untuk menentukan nilai yang akan digunakan sebagai masukan aksi kendali yang tepat. Dua buah teknik reasoning yang paling sering digunakan adalah MAX-MIN dan MAX-DOT.Untuk A' masukan himpunan fuzzy dan B' keluaran himpunan *fuzzy*.

2.1.8.1 Metode MAX-MIN

Inferensi maksimum-minimum dapat dijelaskan dengan permasalahan sebuah basis aturan dengan dua aturan :

Aturan 1 : jika x adalah A1 dan y adalah B1 maka z adalah C1

Aturan 2 : jika x adalah A2 dan y adalah B2 maka z adalah C2

Setiap aturan diberi bobot dengan kuat obot tertentu, kuat bobot ini sangat dipengaruhi oleh operator penghubung dalam aturan. Jika aturan dihubungkan dengan operator '*AND*' maka operasi yang dilakukan adalah irisan (minmum), sedangkan jika aturan dihubungkan dengan operator '*OR*' maka operasi yang dilakukan adalah gabungan (maksimum), jika kuat bobot dari aturan ke-i dilambangkan dengan αi, masukan *fuzzy* adalah xo dan yo, maka keanggotaan xo adalah himpunan A dilambangkan $\mu_A(x_o)$.

$$\alpha_1 = \mu A_1(x_o) \quad \mu B_1(y_o) \tag{2.15}$$

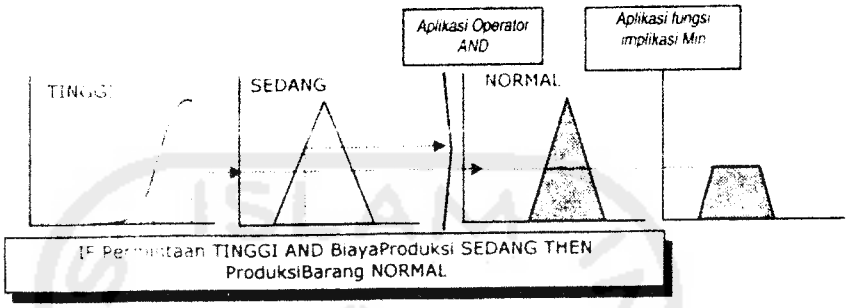
$$\alpha_2 = \mu A_2(x_o) \wedge \mu B_2(y_o) \tag{2.16}$$

'$\wedge$' adaalah operator '*AND*' atau minimum dan kuat penyulutan untuk aturan-aturan diatas adalah :

$$\mu C(w) = (\alpha_1 \wedge \mu C_1(w)) \vee (\alpha_2 \wedge \mu C_2(w)) \tag{2.17}$$

Dengan μC(w) adalah derajad anggota keluaran.

Min (*minimum*). Fungsi ini akan memotong output himpunan fuzzy. Berikut contoh fungsi implikasi MIN :



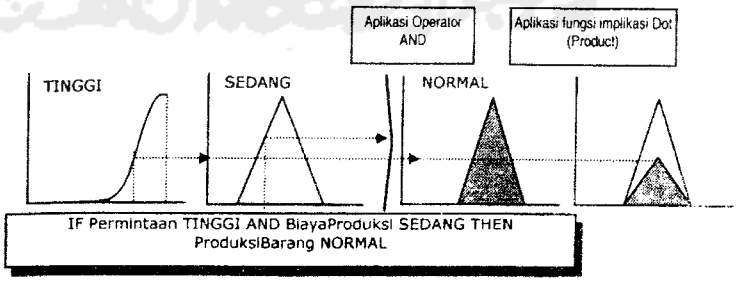
Gambar 2.13 Fungsi Implikasi : MIN

2.1.8.2 Metode MAX-DOT

Metode ini menggunakan operasi perkalian sebagai implikasi fuzzy, maka didapat persamaan sebagai berikut :

$$\mu C(w) = (\alpha_1 \,.\, \mu C_1(w)) \vee (\alpha_2 \,.\, \mu C_2(w)) \qquad (2.18)$$

'$\vee$' adalah operator '*OR*' atau maksimum, dengan μC(w) adalah derajad anggota keluaran. DOT (*product*) dimana fungsi ini akan menskala output himpunan fuzzy. Berkut salah satu contoh fungsi dot :



Gambar 2.14 Fungsi Implikasi : DOT

2.1.9 Defuzzifikasi

Defuzzifikasi didefinisikan sebagai proses pemetaan dari himpunan fuzzy hasil inferensi ke dalam suatu aksi kendali non fuzzy. Metode yang sering digunakan dalam defuzzifikasi adalah metode rerata maksimum (*mean of maximum*/MOM) dan metode rerata pusat (*center of average*/COA).

2.1.9.1 Metode *Mean Of Maximum* (MOM)

Metode ini memberikan hasil defuzzifikasi (z*) dengan rumusan :

$$z^* = \frac{\sum_{j=1}^{m} \alpha j \, Hj \, Wj}{\sum_{j=1}^{n} \alpha j \, Hj} \qquad (2.19)$$

Dengan :

- z* = hasil defuzzifikasi
- Hj = ketingginan maksimum fungsi keanggotaan keluaran untuk aturan ke-j.
- Wj = titik berat keanggotaan keluaran aturan ke-j.
- αj = kuat penyulutan *(free strength)* pada aturan ke-j.
- n = menunjukkan jumlah aturan.
- j = aturan ke- ...

2.1.9.2 Metode *Center Of Average* (COA)

Metode ini memberikan hasil defuzzifikasi (z*) dengan rumusan :

$$z^* = \frac{\sum_{j=1} \alpha j \, Mj}{\sum_{j=1}^{n} \alpha j \, Aj} \tag{2.20}$$

Dengan :

- z* = hasil defuzzifikasi
- Mj = momen fungsi keanggotaan himpunan fuzzy keluaran untuk aturan ke-j.
- αj = kuat penyulutan *(free strength)* pada aturan ke-j.
- Aj = merupakan luas daerah keluaran untuk aturan ke-j.

2.1.10 Metode Mamdani

Metode Mamdani sering dikenal sebagai metode MAX-MIN. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Untuk mendapatkan output, diperlukan 4 tahapan :

1. **Pembentukan himpunan fuzzy**

   Variabel input maupun variabel output dibagi menjadi satu atau lebih himpunan fuzzy.

2. **Aplikasi fungsi implikasi**

   Fungsi implikasi yang digunakan adalah MIN.

3. **Komposisi aturan**

Inferensi diperoleh dari kumpulan dan korelasi antar aturan. Ada 3 metode yang digunakan dalam melakukan inferensi sistem fuzzy, yaitu : *max, additive* dan *probabilistik OR* (probor).

- **Metode *Max* (Maksimum)**

Pada metode ini, solusi himpunan fuzzy diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah fuzzy, dan mengaplikasikannya ke output dengan menggunakan operator *OR* (*union*). Jika semua proposisi telah dievaluasi, maka output akan berisi suatu himpunan fuzzy yang merefleksikan konstibusi dari tiap-tiap proposisi. Secara umum dapat dituliskan :

$$\mu_{sf}[x_i] = \max (\mu_{sf}[x_i], \mu_{kf}[x_i]) \quad (2.21)$$

Dengan :

$\mu_{sf}[x_i]$ = nilai keanggotaan solusi fuzzy sampai aturan ke-i;

$\mu_{kf}[x_i]$ = nilai keanggotaan konsekuen fuzzy aturan ke-i.

- **Metode *Additive* (*Sum*)**

Pada metode ini, solusi himpunan fuzzy diperoleh dengan cara melakukan *bounded-sum* terhadap semua output daerah fuzzy. Secara umum dapat dituliskan :

$$\mu_{sf}[x_i] = \min (1, \mu_{sf}[x_i] + \mu_{kf}[x_i]) \quad (2.22)$$

Dengan :

$\mu_{sf}[x_i]$ = nilai keanggotaan solusi fuzzy sampai aturan ke-i;

$\mu_{kf}[x_i]$ = nilai keanggotaan konsekuen fuzzy aturan ke-i.

➢ **Metode Probabilistik *OR* (probor)**

Pada metode ini, solusi himpunan fuzzy diperoleh dengan cara melakukan *product* terhadap semua output daerah fuzzy. Secara umum dapat dituliskan :

$$\mu_{sf}[x_i] = (\mu_{sf}[x_i] + \mu_{kf}[x_i]) - (\mu_{sf}[x_i] * \mu_{kf}[x_i]) \quad (2.23)$$

Dengan :

$\mu_{sf}[x_i]$ = nilai keanggotaan solusi fuzzy sampai aturan ke-i;

$\mu_{kf}[x_i]$ = nilai keanggotaan konsekuen fuzzy aturan ke-i.

4. Penegasan (defuzzy)

Input dari proses defuzzy adalah suatu himpunan fuzzy yang diperoleh dari komposisi aturan-aturan fuzzy, sedangkan output yang dihasilkan merupakan suatu bilangan pada domain himpunan fuzzy tersebut. Sehingga jika diberikan suatu himpunan fuzzy dala range tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai output.

Ada beberapa metode defuzzy yang bisa dipakai pada komposisi aturan MAMDANI, antara lain :

- **Metode Centroid (*Composite Moment*)**

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil titik pusat (z*) daerah fuzzy. Secara umum dituliskan :

$$z^* = \frac{\int_z z\mu(z)dz}{\int_z z\mu(z)dz} \quad \text{untuk variabel kontinu,} \tag{2.24}$$

$$z^* = \frac{\sum_{j=1}^{n} zj\ \mu(zj)}{\sum_{j=1}^{n} \mu(zj)} \quad \text{untuk variabel diskret.} \tag{2.25}$$

- **Metode *Mean of Maximum* (MOM)**

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.

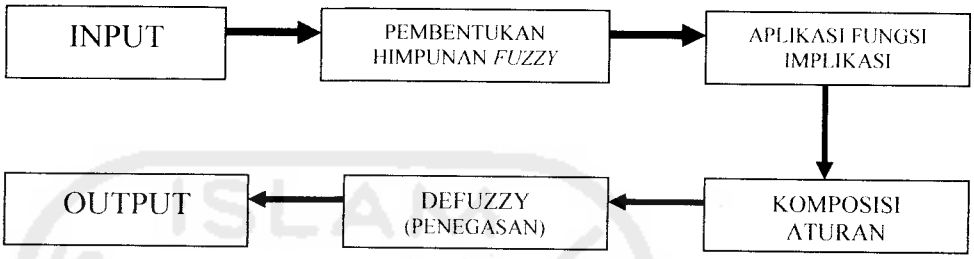
- **Metode *Largest of Maximum* (LOM)**

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.

- **Metode *Smallest of Maximum* (SOM)**

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum.

Keuntungan dari penggunakan metode mamdani ini terletak pada kemudahan penalarannya yang dapat diterima oleh banyak pihak, dari proses pembentukan himpunan fuzzy hingga proses defuzzifikasi.



Gambar 2.15 Diagram blok Perancangan Logika Fuzzy – Metode Mamdani

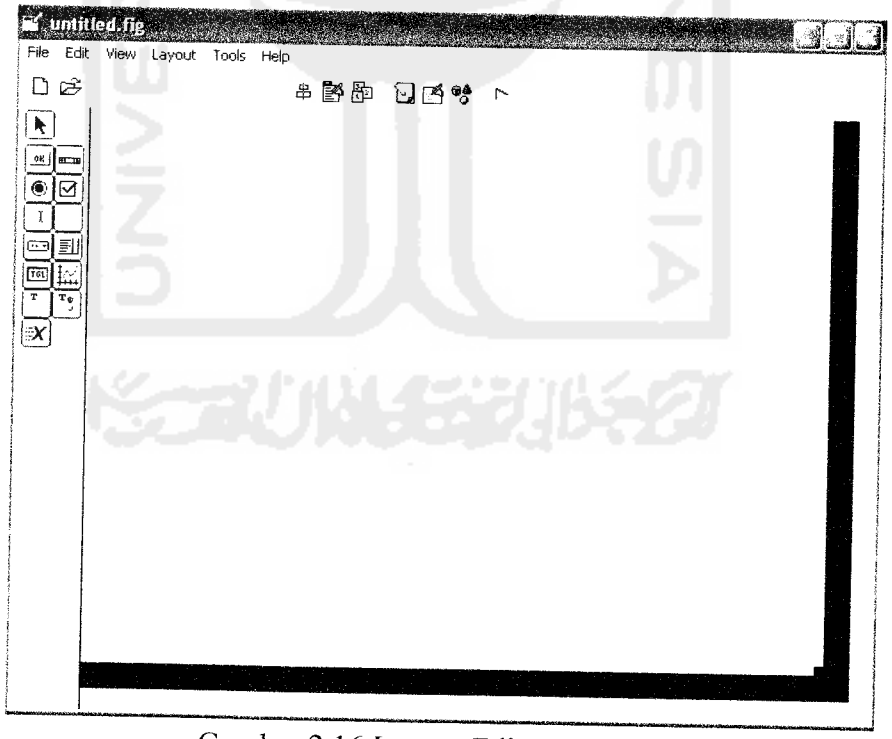
2.2 Matlab

Matlab dari Mathwork Inc. Software dilengkapi dengan berbagai *toolbox* yang memudahkan pemakai untuk melakukan perhitungan. Bahkan pada saat ini sudah dikembang t*oolbox* khusus simulasi yang diberi nama simulink. Aplikasi Matlab dalam bidang pengaturan dilengkapi dengan *toolbox* dimana *toolbox* ini dilengkapi berbagai faktor pendukung yang digunakan dalam analisis sistem kendali. Beberapa fungsi pendukung yang digunakan untuk menganalisa sustu sistem adalah *feedback, step, rootlocus, series,* dan lain-lain. Untuk menganalisa suatu sistem Matlab hanya memerlukan masukan beberapa fungsi yang dituliskan ke dalam bebtuk *Transformasi Laplace* (dalam kawasan-s) atau matriks. Matlab merupakan bahasa canggih untuk komputasi teknik yang memiliki kemampuan penghitungan, visualisasi dan pemograman. Beberapa kegunaan lain dari matlab ini adalah untuk pengembangan

algoritma, pemodelan, simulasi termasuk pembuatan antarmuka grafis/*Graphical User Interface* (GUI).

2.2.1 *Graphical User Interface* (GUI) pada Matlab.

Pada Matlab disediakan *tool* untuk pembuatan GUI (*Graphical User Interface*) yang merupakan suatu antarmuka/alat penghubung bergambar ke suatu program, fasilitas ini dinamakan GUIDE (*GUI Development Environment*). Ketika membuka GUI pada GUIDE, akan ditampilkan sebuah *layout editor,* dimana terdapat *control panel* untuk seluruh *tools* pada GUIDE. Dengan layout editor lebih memudahkan untuk menampilkan sebuah GUI dengan mudah dan cepat dengan men-*drag* komponen seperti, *push buttons*, *pop-up menus*, atau *axes* dari *component palette* ke *layout area.*



Gambar 2.16 Layout Editor pada GUI

GUIDE toolset :

- *Layout Editor* : tempat untuk menambah dan menyusun objek di jendela GUI
- *Alignment Tools* : untuk mengatur posisi masing – masing objek
- *Property Inspector* : memeriksa dan mengeset nilai – nilai (sifat) objek
- *Object Browser* : memantau hirarki objek – objek yang telah ada di *layout*
- *Menu Editor* : membuat menu dan submenu dari GUI

Ketika membuat layout GUI kemudian menyimpannya, maka MATLAB akan membuat *M-file* secara otomatis yang nantinya akan digunakan untuk pemrograman, nama *M-file* harus sama dengan nama figure dari GUI tersebut.

2.3 Definisi Robot

Robot menurut Karel Capek berasal dari kata *robota* (bahasa Ceko) yang berarti budak/pekerja.Definisi robot secara umum adalah suatu alat multi fungsi otomatis yang dapat diprogram untuk melakukan pekerjaan tertentu.

Unsur utama dalam kata "robot" adalah :

1. Seperangkat peralatan/mesin/*device*.
2. Dapat diprogram.
3. Bekerja/bergerak secara otomatis.
4. Mampu melaksanakan tugas tertentu sesuai program.

2.4 Struktur Robot

Pada umumnya sebuah robot memiliki struktur sebagai berikut :

1. Manipulator
   - Badan (*body*)
   - Pergelangan (*wrist*)
   - Lengan (*arm*)
2. End Effector

   Penjepit (*gripper*), bor, dll.
3. Penggerak (*Driver*)

   Motor, hidrolik, pneumatik.
4. Sensor

   Posisi, panas, kecepatan, dll.
5. Pengendali (*Controller*)

   PC/komputer, mikrokontroler, mikroposesor, dll.

2.5 Penggerak Robot

Macam-macam tenaga penggerak pada robot, yaitu :

1. Elektris

   Tenaga penggeraknya menggunakan jenis motor DC, motor AC, motor stepper dan motor servo.
2. Hidrolis

   Penggerak robot pada jenis ini menggunakan semacam cairan untuk

enggerakkan. Tenaganya diperoleh dari pompa hidrolik dan pergerakkanya diatur oleh *solenoid valve* yang kemudian diteruskan ke bagian aktuator melalui piston.

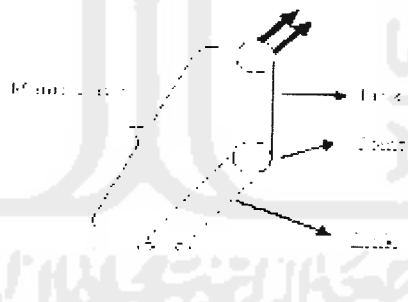
3. Pneumatis

Sama seperti penggerak hidrolis tetapi medium yang dipakai untuk menyalurkan tenaga berupa gas/udara.

3. Mekanis

2.6 Kinematics Manipulator

Kinematics manipulator adalah ilmu analisis guna mempelajari gerakanmanipulator dengan mengabaikan gaya yang menyebabkan gerakan tersebut.Sedangkan manipulator adalah sekumpulan *link* yang terhubung pada *joint*.



Gambar 2.17 Contoh manipulator lengan robot

Derajat kebebasan (*Degrees Of Freedom*/DOF) adalah banyaknya gerakan yang dapat dilakukan oleh manipulator dalam arah yang berbeda tergantung pada

banyak. *joint* yang dipakai. Sedangkan ruang untuk sebuah robot beroperasi dinamakan ruang kerja (work volume) yang fungsinya merupakan tempat daerah dimana end effector sebuah robot masih dapat mencapainya. Ruang kerja itu sendiri dipengaruhi oleh :

- Konfigurasi fisik robot
- Ukuran link dan wrist
- Batasan gerak joint

Dan ruang kerja memiliki macam –macam bentuk :

- Polar , seperti Bola
- Silindris , seperti silinder
- Kartesian , bentuk balok

Disamping ruang kerja dan derajat kebebasan dari sebuah robot juga memiliki konfigurasi, dimana konfigurasi itu sendiri merupakan pola susunan link dan joint yang menghasilkan suatu karakteristik gerakan tertentu. Macam – macam konfigurasi dari robot itu sendiri adalah :

1. Konfigurasi Articulated / Revolute / jointed
2. Konfigurasi Silindris
3. Konfigurasi Speris / Polar
4. Konfigurasi Kartesian

Kinematics manipulator dibedakan menjadi dua bagian yaitu :

2.6.1 Forward Manipulator Kinematics

Pada bagian ini sudut tiap *joint* dari lengan robot sudah diketahui sehingga dapat dicari posisi orientasi. Untuk mencari *Forward Kinematics* dapat digunakan rumus persamaan geometri atau persamaan matriks transformasi :

Analisis secara geometri untuk *Forward Kinematics* 3 dimensi :

$$\text{Sumbu } x = \cos\ \theta_1 \{ L_2 \cos(\theta 2) + L_3 \cos(\theta 2 + \theta 3)\} \quad (2.26)$$

$$\text{Sumbu } y = \sin\ \theta_1 \{ L_2 \cos(\theta 2) + L_3 \cos(\theta 2 + \theta 3)\} \quad (2.27)$$

$$\text{Sumbu } z = L_1 + \{ L_2 \sin(\theta 2) + \{ L_3 \cos(\theta 2 + \theta 3)\} \quad (2.28)$$

Sedangkan secara *Matriks Transformasi* dari suatu link terhadap link sebelumnya dicari dengan :

$$^{i-1}_{i}T = \begin{pmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i * \cos\alpha_{i-1} & \cos\theta_i * \cos\alpha_{i-1} & -\sin\alpha_{i-1} & \sin\alpha_{i-1}{}^{d} \\ \sin\theta_i * \sin\alpha_{i-1} & \cos\theta_i * \sin\alpha_{i-1} & \cos\alpha_{i-1} & \cos\alpha_{i-1}{}^{d} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.29)$$

2.6.2 Inverse Manipulator Kinematics

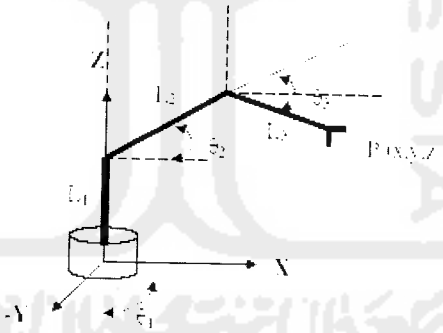
Sedangkan pada bagian ini sudah diketahui posisi orientasi untuk mencari sudut tiap joint. dari tiap *joint* lengan robot.

Untuk mencari *Inverse Kinematics* dapat dicari dengan berbagai cara diantaranya :

- *Geometric Approaches* (Lee and Ziegler [1984])
- Aljabar Matriks
- *Inverse Transform* (Paul et al. [1981])
- *Screw Algebra* (Kohli *and* Soni [1975])
- *Iterative* (Uicker et al. [1964])
- *Dual Quaternian* (Yang *and* Freudenstein [1964])
- *Dual Matrices* (Denavit [1956])

Analisis secara matematika Geometri untuk *Inverse Kinematics* 3D dimensi :

Untuk mencari persamaan ini pertama-tama harus diketahui nilai titik koordinat x dan y, dan juga z (jumlah semua sudut lengan robot yang diinginkan terhadap sumbu x positif).



Gambar 2.18 *Inverse kinematics* 3 Dimensi

Keterangan :

L_1 : panjang lengan pertama

L_2 : panjang lengan kedua

L_3 : panjang lengan ketiga

θ_1 : besar sudut yang dihasilkan dari putaran sumbu z pada lengan pertama

θ_2 : besar sudut yang dihasilkan antara lengan kedua dengan lengan pertama

θ_3 : besar sudut yang dihasilkan antara lengan ketiga dengan lengan kedua

Dari gambar diatas dapat kita tentukan rumus inverse kinematicsnya untuk tiap sudut manipulator :

$$x^2 + y^2 = L_2^2 + L_3^2 - 2 * L_1 * \cos(180 + \theta_2) \tag{2.30}$$

Dari $\cos(180+ \theta_2) = -\cos(\theta_2)$, kita dapatkan :

$$\theta_2 = \cos^{-1}\left(\frac{x^2 + y^2 - L_2^2 - L_3^2}{2*L_2*L_3}\right) \tag{2.31}$$

Sehingga dari persamaan diatas dapat pula rumus untuk θ_3 :

$$\theta_3 = \cos^{-1}\left(\frac{x^2 + (y-L_1)^2 - L_2^2 - L_3^2}{-2*L_2*L_3}\right) \tag{2.32}$$

Sedangkan untuk θ_1 didapat persamaan :

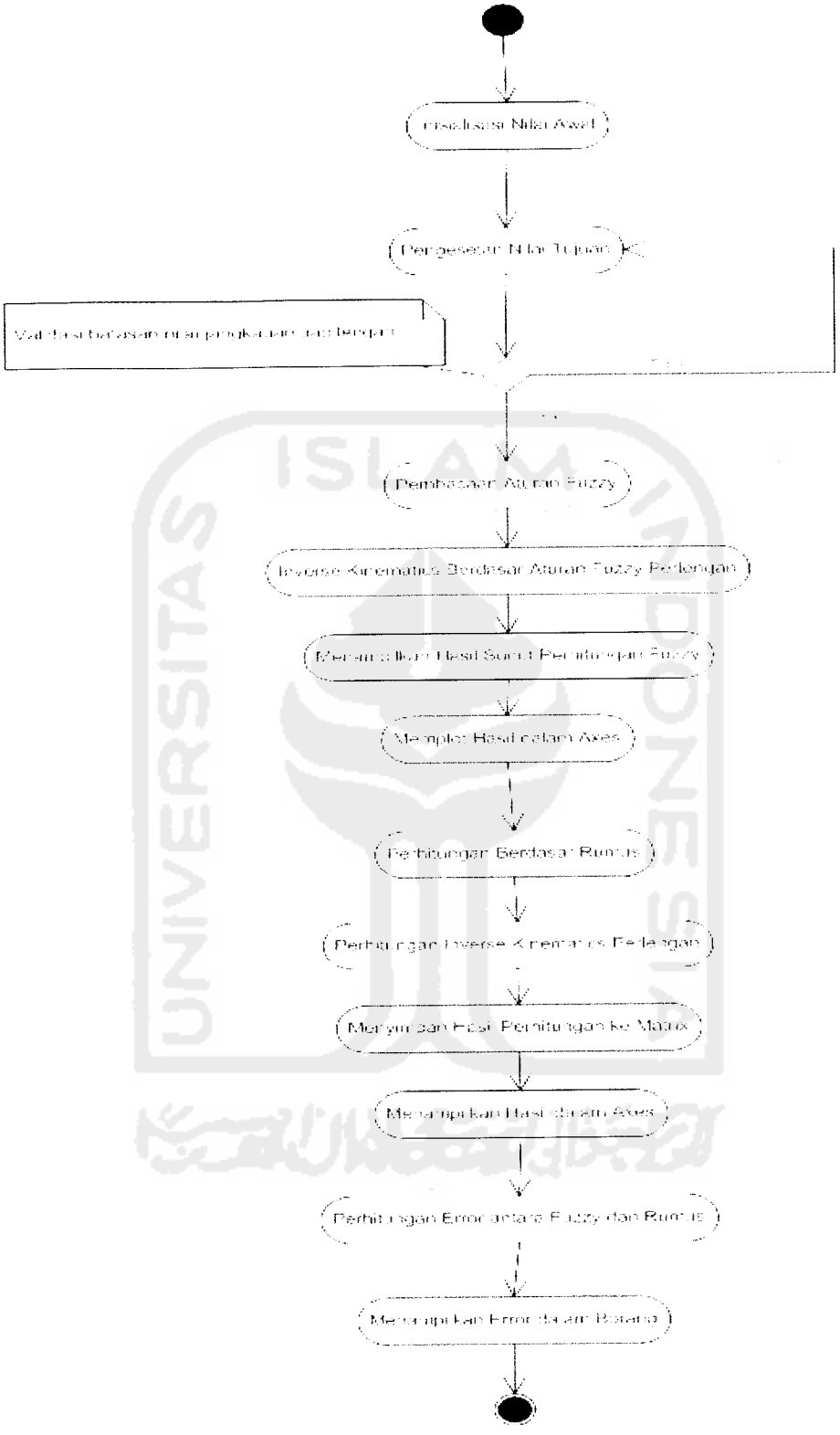
$$\theta_1 = \text{arc tg}\left(\frac{X}{-Y}\right) \tag{2.33}$$

BAB III

PERANCANGAN SISTEM

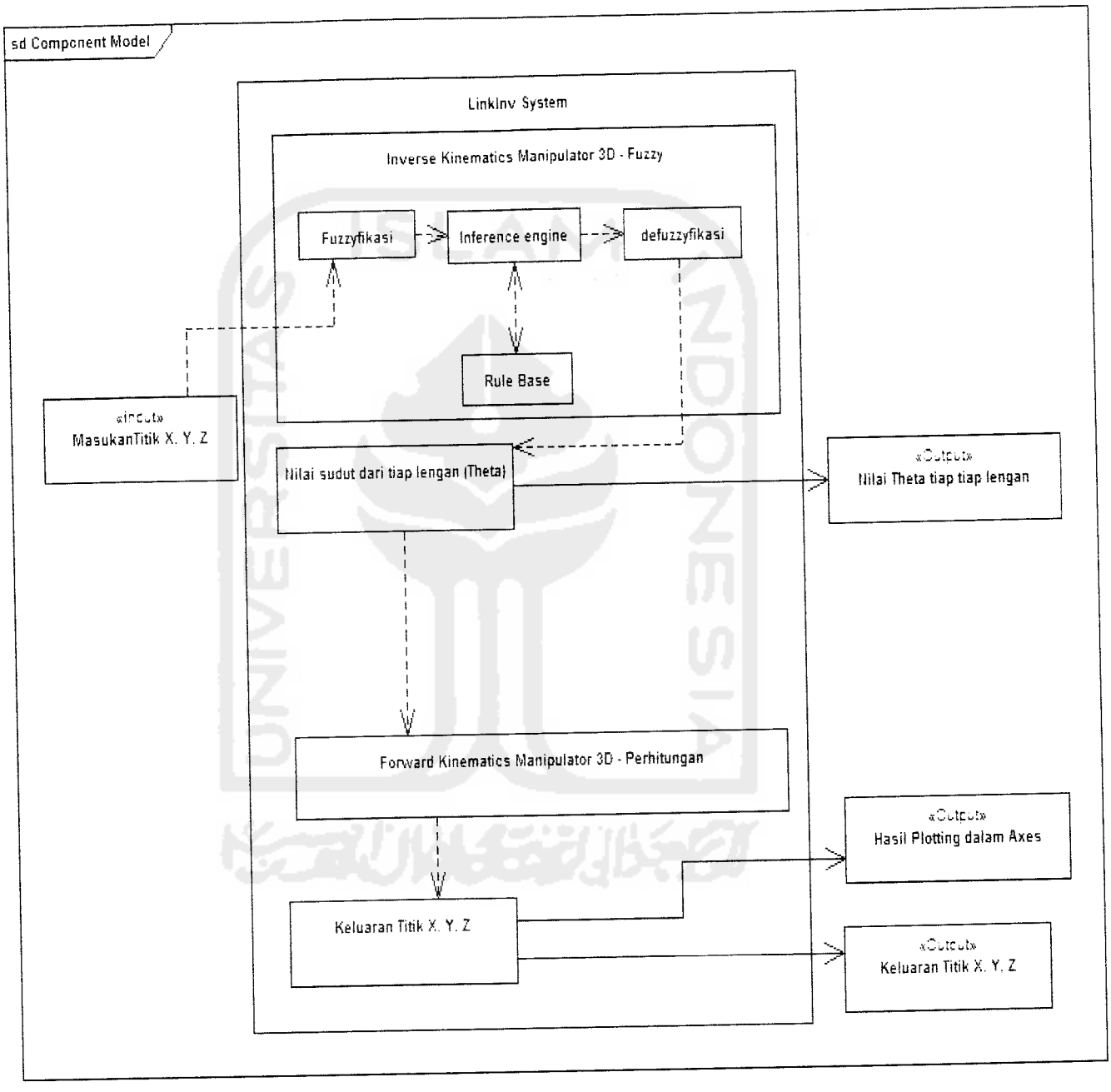
Dalam bab III akan dibahas mengenai perancangan sistem logika fuzzy pada *inverse kinematics* manipulator yang akan divisualisasikan menggunakan GUI Matlab dan program yang menghubungkan antara sistem tersebut dengan program *M-File* sehingga keduanya saling berinteraksi berdasarkan teori – teori yang telah dibahas pada bab sebelumnya. Untuk dapat lebih jelasnya bagaimana cara kerja program keseluruhan dapat dilihat pada gambar 3.1 merupakan flowchart jalannya program *inverse kinematics* manipulator 3D secara keseluruhan. Rancangan program yang terlihat pada diagram alir (flowchart) pada gambar 3.1 dapat dijelaskan bahwa input berupa koordinat x,y, dan z dimasukkan sebagai inisialisasi nilai awal lalu akan dilakukan pengesetan nilai tujuan atau nilai awal.

Apabila nilai awal yang diinginkan terjangkau sesuai validasi batasan nilai jangkauan perlengan maka nilai dapat diproses dan program diteruskan, namun apabila tidak terjangkau maka nilai awal yang dimasukkan tadi akan dikembalikan kenilai yang dimasukkan sebelumnya keinisialisasi nilai awal. Maka program diproses yang secara garis besar pertama program diproses dengan menggunakan perancangan sistem fuzzy secara *invers kinematics* dan yang kedua program diproses menggunakan perhitungan rumus secara *inverse kinematics* yang digunakan sebagai pembanding untuk menentukan berapa error dari sistem fuzzy tersebut.



Gambar 3.1 Flowchart jalannya program *inverse kinematics* manipulator 3D

Untuk rancangan sistem inverse kinematics manipulator 3D dapat dilihat pada diagram rancangan dibawah ini :



Gambar 3.2 Diagram rancangan sistem *inverse kinematics* manipulator 3D

Dari diagram rancangan sistem inverse kinematics manipulator 3D berbasis logika fuzzy diatas maka dapat diuraikan yang pertama adalah proses perancangan inverse kinematics manipulator 3D menggunakan logika fuzzy. Dalam perancangan ini terdapat tiga tahap yaitu proses fuzzyfikasi, didalam proses ini dilakukan pemetaan dari variabel masukan kedalam himpunan dalam suatu semesta wacana. Setelah dilakukan pemetaan variabel kedalam suatu himpunan maka masuk dalam tahap proses yang kedua yaitu inference engine, pada proses ini dilakukan pembentukan rule – rule fuzzy dengan menyusun variabel x,y, dan z sebagai input dan variabel teta1, teta2, dan teta3 sebagai output secara inverse dalam sebuah komposisi rule fuzzy atau pada rule base, dengan acuan pengelompokan data secara forward sebagai acuannya dan batasan perancangan rule fuzzy sesuai dengan himpunan fuzzy yang telah ditetapkan.

Setelah rule – rule fuzzy dirancang secara keseluruhan, maka rule – rule fuzzy diuji. Pengujian dilakukan untuk melihat hasil sudut perlengan yang berupa teta1, teta2, dan teta3, lalu akan ditampilkan dalam bentuk nilai sudut tiap lengan. Maka masuk pada proses yang ketiga yaitu defuzzyfikasi, pada proses ini dilakukan pemetaan dari himpunan fuzzy hasil inferensi ke dalam suatu aksi kendali non fuzzy, aksi kendali non fuzzy itu sendiri adalah proses perhitungan secara forward dengan menggunakan rumus forward dengan input berupa teta1,2,dan3 setelah itu dilakukan pemrosesan untuk menghasilkan output berupa koordinat x,y,dan z.

3.1 Deskripsi Simulasi

Simulasi pada perancangan ini menggunakan perangkat Matlab, dimana langkah awal yang harus dilakukan adalah mengelompokkan aturan – aturan fuzzy yang akan digunakan sebagai penggerak lengan robot (*inverse kinematics*) dengan cara menentukan himpunan fuzzy, dimana fungsi himpunan fuzzy disini untuk menentukan sejauh mana lengan dari robot (*inverse kinematics*) yang terdiri dari 3 link dapat menjangkau posisi yang diinginkan dengan batasan jangkauan yang telah ditentukan.

Kemudian dilakukan pemilihan aturan – aturan fuzzy untuk ruang input dan ruang output yang tepat untuk dapat menggerakkan ketiga link, dimana aturan – aturan lalu disusun didalam Fis editor yang tersedia didalam perangkat Matlab dengan cara membentuk himpunan fuzzy terlebih dahulu yang ada didalam Fis editor sebagai awal dari perancangan simulasi *inverse kinematics* 3D, setelah itu kita dapat memasukkan satu per satu variabel – variabel yang dibutuhkan dalam perancangan Fis untuk menyusun aturan – aturan fuzzy yang nantinya aturan – aturan tersebut diinteraksikan dengan *M-file* dengan cara memanggilnya kembali pada *M-file* untuk dapat diprogram didalam GUI Matlab yang telah dirancang sebagai visualisasi dari simulasi *inverse kinematics* itu sendiri. Lalu dilakukan perancangan sistem dengan menggunakan perhitungan rumus *inverse kinematics* sebagai pembanding untuk menentukan error pada sistem fuzzy sebagai acuan atas perancangan aturan – aturan fuzzy yang telah disusun.

3.2 Perancangan Sistem

3.2.1 Perancangan Sistem Fuzzy

Sistem fuzzy yang digunakan pada sistem simulasi *inverse kinematics* manipulator 3D ini menggunakan sistem inferensi Mamdani. Rule fuzzy sendiri memiliki bobot 1 untuk tiap rulenya, dimana tiap inputnya memakai 5 mf (multiply factor), sehingga didapat jumlah rule secara keseluruhan adalah 125 rule dengan cara pengalian 5x5x5 dan rule yang dipakai hanya 122 rule. Untuk input berupa koordinat x, y, dan z dibagi masing – masing 5 mf dengan nama keanggotaan diantaranya : NB(negative big)/jauh, NS(negative small)/dekat, Z(zero)/sedang, PS(positif small)/dekat, PB(positif big)/jauh, sedangkan untuk output juga memakai 5 mf dengan nama keanggotaan diantaranya : NB(negative big)/jauh, NS(negative small)/dekat, Z(zero)/sedang, PS(positif small)/dekat, PB(psitif big)/jauh

Untuk melakukan perancangan rule / aturan fuzzy yang berisi variabel – variabel keanggotaan NB, NS, Z, PS, PB maka pertama dilakukan pengambilan data – data berupa angka secara forward yang berguna untuk menyusun aturan –aturan fuzzy yang kita inginkan. berikut adalah beberapa bentuk kombinasi data digunakan untuk penyusunan fuzzy :

Tabel 3.1 Kombinasi Data *Forward*

input			output		
THETA1	THETA2	THETA3	X	Y	Z
0	-17.52	-69.82	10	0	0
-135	19.4	-81.3	-10	-10	7.5
-90	70.5	-118.69	0	-10	15
0	89.93	-92.79	10	0	22.5
0	174.26	-168.52	0	0	15
45	65.48	-63.18	10	10	22.5
180	26.39	-110.41	-10	0	7.5
90	89.93	-92.79	0	10	22.5
-135	65.48	-63.18	-10	-10	22.5
135	19.4	-81.3	-10	10	7.5
135	52.48	-88.85	-10	10	15
-135	-26.43	-32.33	-10	10	0
-90	-17.52	-69.82	0	-10	0
-45	-26.43	-32.33	10	-10	0
180	-17.52	-69.82	-10	0	0
0	-40.54	-98.92	0	0	0
90	-17.52	-69.82	0	10	0
45	-26.43	-32.33	10	10	0
-90	26.39	-110.41	0	-10	7.5
-45	19.4	-81.3	10	-10	7.5
0	-15.96	-148.08	0	0	7.5
0	26.39	-110.41	10	0	7.5
90	26.39	-110.41	0	10	7.5
45	19.4	-81.3	10	10	7.5
-135	52.48	-88.85	-10	-10	15
-45	52.48	-88.85	10	-10	15
180	70.65	-118.69	-10	0	15
90	70.65	-118.69	0	10	15
45	52.48	-88.85	10	10	15
0	151.64	-123.28	0	0	22.5
180	89.93	-92.79	-10	0	22.5
135	65.48	-63.18	-10	10	22.5
-90	69.08	-19.09	0	-10	30
90	69.08	-19.09	0	10	30
0	69.08	-19.09	10	0	30
0	121.79	-63.58	0	0	30
180	69.08	-19.09	-10	0	30

Dari data diatas kita dapat menyusun aturan fuzzy satu per satu, misalkan untuk data output pertama : bila teta1 adalah 0 = Z (sedang / zero), teta2 adalah -17,52 = NS (negatif dekat / negatif small), teta3 adalah -69,82 = NB (negatif jauh / negatif big), maka didapat aturannya sebagai berikut :

teta1 is Z and teta2 is NS and teta3 is NB

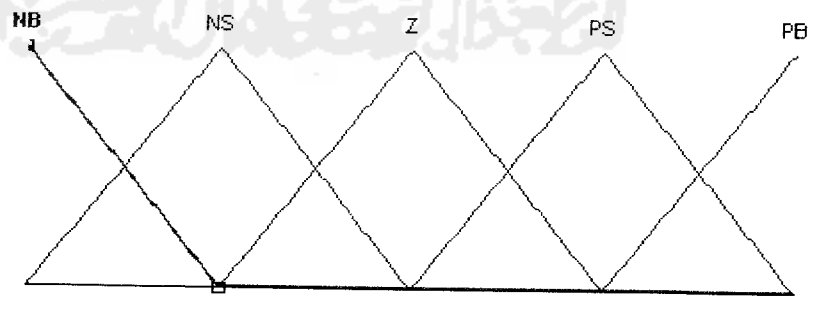
Sedangkan untuk data inputnya : bila x adalah 10 = PS (positif small / positif dekat), y adalah 0 = Z (zero / sedang), dan z adalah 0 = Z (zero / sedang), maka didapat kombinasi :

x is PS and y is Z and z is Z

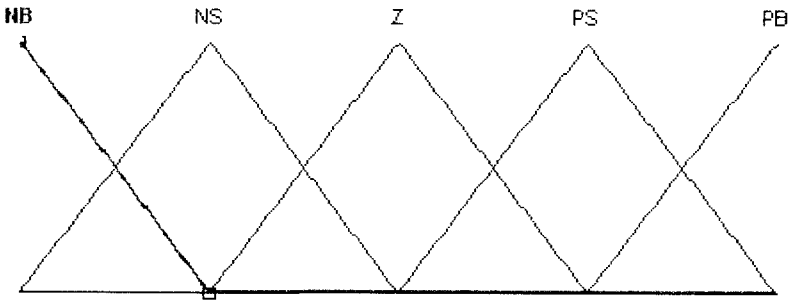
Maka didapat sebuah aturan / rule fuzzy :

If x is PS and y is Z and z is Z then teta1 is Z and teta2 is NS and teta3 is NB

Setelah itu rulenya diuji apabila outputnya masih kurang bagus, maka rulenya diperiksa ulang sampai menghasilkan output yang paling mendekati output dari rumus. Sehingga didapat kombinasi rule yang diinginkan, semua itu tidak terlepas dari peran *member function*, berikut *member function* yang digunakan pada perancangan invers kinematik manipulator 3D berbasis logika fuzzy :

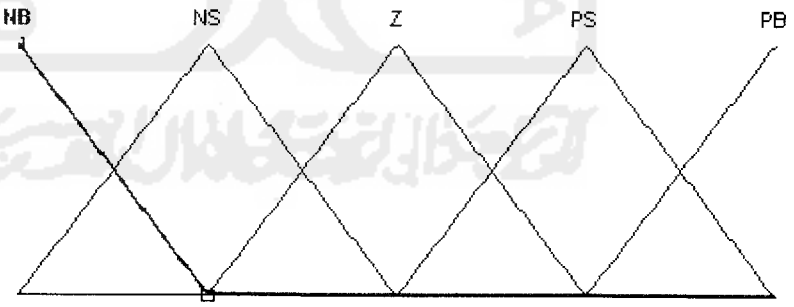


Gambar 3.3 Input Variabel X Dan Y

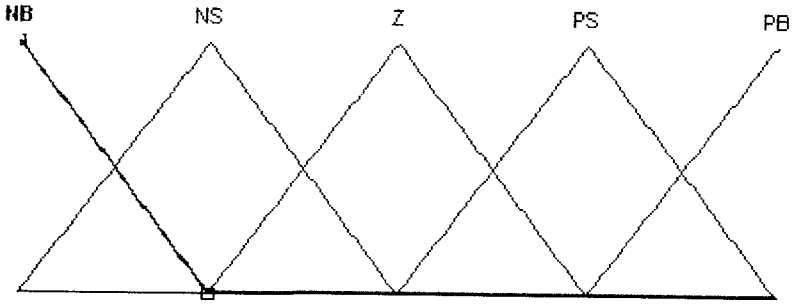


Gambar 3.4 Input Variabel Z

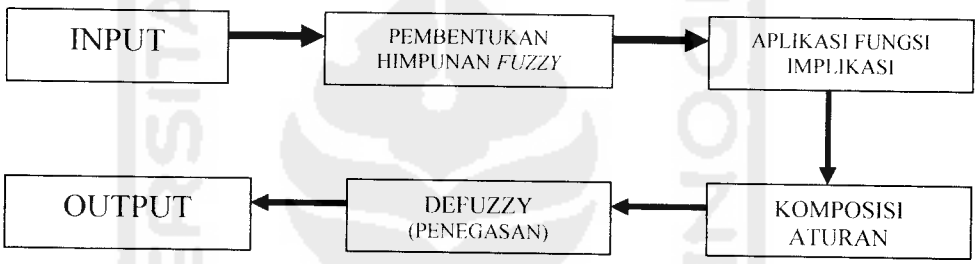
Sedangkan untuk output teta 1, teta 2, dan teta 3 juga terdiri dari 5 mf dengan keanggotaan fuzzy sama seperti ruang input Penyusunan rule fuzzynya itu sendiri dapat disusun dengan menggunakan acuan data *forward* yang inputnya berupa sudut teta 1, teta 2, dan teta 3 dan dikelompokkan berdasarkan sudut – sudut tersebut sebagai input dari data forward yang menghasilkan output berupa titik koordinat x,y, dan z. Berikut adalah gambar variabel output teta 1, teta 2, teta 3:



Gambar 3.5 Output Variabel Teta 1



Gambar 3.6 Ouput Variabel Teta 2 Dan Teta 3



Gambar 3.7 Blok Diagram Sistem Fuzzy

Sistem simulasi *inverse kinematics* manipulator 3D ini dirancang menggunakan logika fuzzy dengan menggunakan fungsi segitiga sebagai representasi input untuk posisi koordinat yang dituju yaitu berupa titik x,y,dan z serta untuk representasi output yaitu berupa sudut teta 1(θ_1), teta 2 (θ_2), dan teta 3 (θ_3)

Fungsi implikasi yang digunakan adalah Min, dimana fungsi ini akan memotong output himpunan fuzzy. Pada sistem fuzzy ini digunakan 122 aturan dari hasil relasi dari domain input yang berupa posisi koordinat yang dituju yaitu berupa titik x,y,dan z. Berikut adalah 1 perwakilan aturan –aturan yang digunakan :

1. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

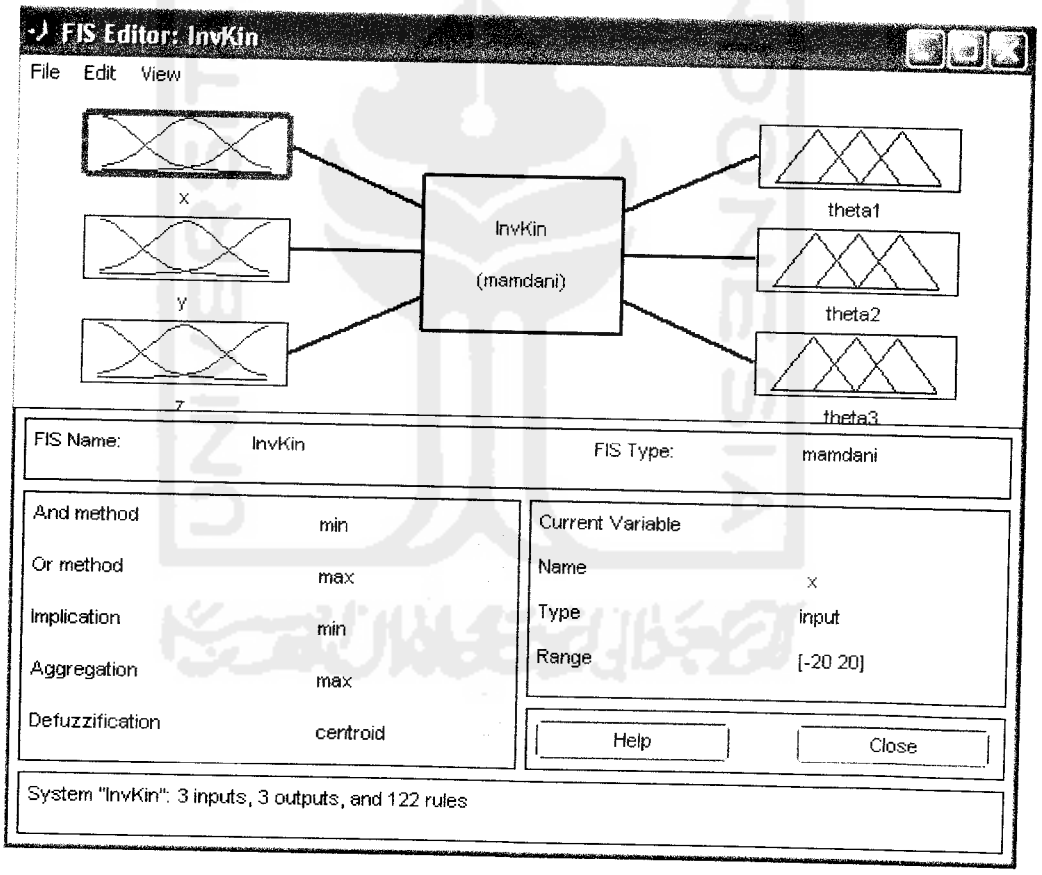
Keterangan aturan – aturan fuzzy :

1. Posisi koordinat x,y, dan z merupakan posisi koordinat yang akan dituju oleh ujung manipulator *inverse kinematics* 3D.
2. Sudut teta 1 merupakan sudut yang dibentuk akibat adanya perputaran terhadap sumbu z, sudut teta 2 merupakan sudut yang terbentuk antara lengan pertama dan lengan kedua, sedangkan teta 3 merupakan besar sudut yang dihasilkan antara lengan kedua dan lengan pertama.
3. Aturan secara keseluruhan dapat dilihat pada lampiran aturan fuzzy.

Jadi input mempunyai 5 derajat keanggotaan dan output juga mempunyai 5 derajat keanggotaan, yaitu :

- Posisi koordinat untuk x,y,dan z yaitu negatif big (jauh), negatif small (dekat), zero (sedang), positif small (dekat), positif big (jauh)
- Sudut teta 1, teta 2, dan teta 3 yaitu negative big (jauh), negatif small (dekat), zero (sedang), positif small (dekat), positif big (jauh)

Sistem fuzzy ini digunakan setelah melakukan beberapa percobaan dengan menggunakan fasilitas sistem fuzzy yang ada di MATLAB, alasan dipilihnya sistem fuzzy tersebut yaitu karena dengan menggunakan sistem fuzzy ini didapat hasil output seperti yang diinginkan, yaitu adanya perubahan nilai output setiap nilai input yang dimasukkan. Dimana proses kerja dari sistem fuzzy adalah mencari yang terbaik dari setiap keputusan yang diambil. Berikut adalah perancanagn sistem fuzzy pada MATLAB:



Gambar 3.8. Fis Editor *Inverse Kinematics* Manipulator 3D Pada MATLAB

Dari perancangan logika fuzzy pada *inverse kinematics* manipulator 3D terjadi error pada setiap titik yang dituju oleh manipulator yang menggunakan pengendali fuzzy hal itu tidak dapat dipungkiri karena logika fuzzy merupakan logika yang dirancang berdasarkan bahasa alami yang terdiri dari aturan – aturan, sehingga mengatur sedikit atau banyaknya jumlah aturan – aturan dapat memperkecil persentase error. Error yang terjadi pada perancangan fuzzy inverse kinematics itu sendiri dapat dihitung dengan membandingkan titik koordinat yang dituju oleh ujung manipulator yang menggunakan kendali fuzzy dengan titik yang dituju oleh ujung manipulator yang menggunakan rumus. Berikut adalah persamaan rumus jarak untuk membandingkan kedua titik koordinat diatas :

$$\text{Distance} = \sqrt{\left[(X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2 \right]} \times 100\% \qquad (4.1)$$

Keterangan : X_1, Y_1, Z_1 = Koordinat yang dituju oleh ujung manipulator dengan fuzzy

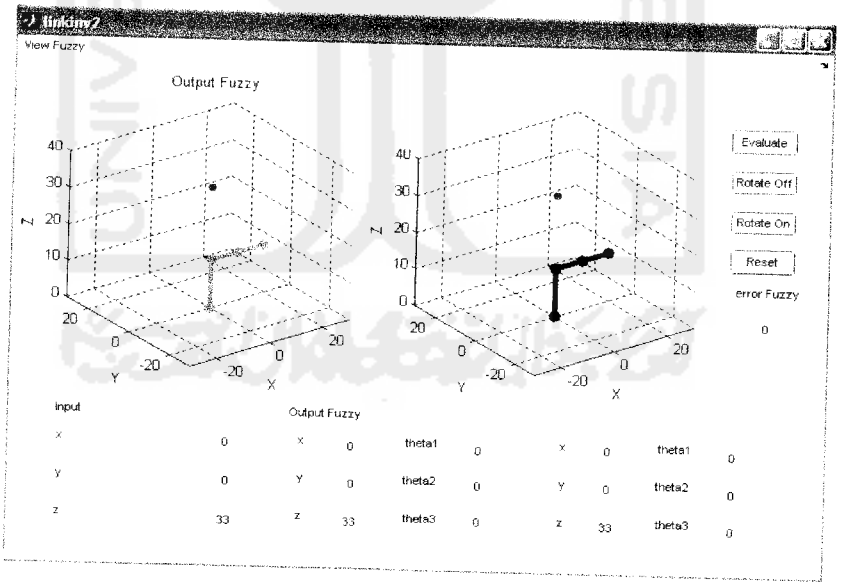
X_2, Y_2, Z_2 = Koordinat yang dituju oleh ujung manipulator dengan rumus

3.2.2 Perancangan GUI (Graphical User Interface)

GUI ini merupakan inti dari simulasi *inverse kinematics* manipulator 3D berbasis logika fuzzy, nilai input yang dimasukkan yaitu berupa posisi koordinat yang dituju oleh ujung link dari manipulator yaitu berupa koordinat x,y, dan z yang akan diproses dengan sistem fuzzy yang ada. Hasil dari proses ini akan ditampilkan yang berupa sudut teta 1, teta 2, dan teta 3. Pada GUI invers kinematics manipulator 3D

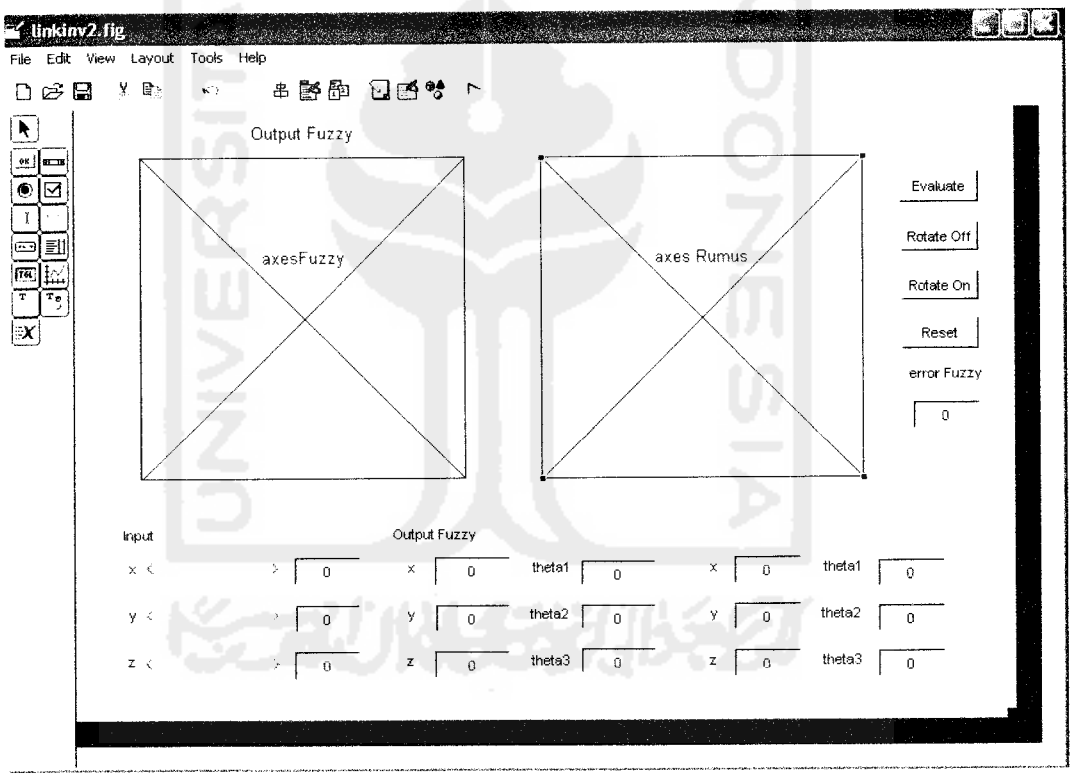
terdapat output yaitu berupa koordinat x,y, dan z , dimana fungsi dari output itu sendiri adalah untuk membandingkan antara masukan berupa koordinat tadi apakah sudah mendekati sesuai sesuai dengan masukannya itu sendiri bila menggunakan logika fuzzy. Dan yang kedua adalah sebagai pembanding seberapa besar error dari sistem fuzzy yang dihasilkan dengan cara membandingkan antara output dari rancangan menggunakan fuzzy dengan ouput manipulator yang dirancang menggunakan rumus *forward kinematics*.

Untuk menampilkan simulasi *inverse kinematics* manipulator 3D berbasis logika fuzzy ini menggunakan fasilitas menu yang telah disediakan oleh GUI, yaitu 2 axes, 16 edit text, 17 static text, 3 slider, dan 4 push button yang akan menampilkan seluruh simulasi yang berlangsung, dimana simulasi ini akan menampilkan simulasi *invers kinematics* manipulator 3D berbasis logika fuzzy.



Gambar 3.9 Rancangan Simulasi *Inverse Kinematics* Manipulator 3D Berbasis Logika Fuzzy

Dalam simulasi ini menggunakan sistem fuzzy dalam bentuk bahasa pemrograman pada *M-file*. Tetapi program sistem fuzzy ini dibuat berdasarkan sistem fuzzy yang telah dilakukan percobaan – percobaan dengan menggunakan fasilitas sistem fuzzy pada MATLAB. Program perancangan fuzzy dilakukan pada fis editor MATLAB, didalam fis editor aturan fuzzy satu per satu dirancang dan kemudian dipanggil pada *M-file* melalui GUI (Graphical User Interface) sebagai visualisasi dari hasil hasil fuzzy yang dirancang.



Gambar 3.10 Rancangan Dasar GUI *Inverse Kinematics* Manipulator 3D Berbasis Logika Fuzzy

Komponen – komponen GUI yang digunakan adalah sebagai berikut :

1. *Axes*

   disini terdapat 2 axes diantaranya axes untuk mengvisualisasikan manipulator yang menggunakan fuzzy dan yang satunya axes untuk menampilkan manipulator yang menggunakan rumus, dimana axes ini akan menerima perintah tampilan *Plot* dari *M-File.*

2. *Static Text*

   disini terdapat 17 static text dimana 1 untuk menampilkan output fuzzy, 1 untuk tulisan error fuzzy, 3 untuk tulisan x,y, dan z pada input, 6 untuk tulisan x,y, dan z serta teta1, teta2,dan teta3 pada output fuzzy, dan 6 untuk tulisan x,y, dan z setra teta1,teta2, dan teta3 pada output rumus. Caranya yaitu dengan mengklik pada komponen *Static Text* kemudian isikan *String property* sesuai dengan nama yang kita inginkan dan pada *Tag property*nya di abaikan saja karena komponen ini tidak menghasilkan *callback* pada *M-file.*

3. *Slider*

   disini terdapat 3 slider penentu posisi tiap koordinat x,y,dan z, range dari slider dapat diatur diproperty inspector.

4. *Panel*

   disini memakai 3 panel yaitu untuk input fuzzy dan rumus, untuk output fuzzy dan untuk output rumus

5. *Edit Text*

   Pada GUI program ini terdapat 16 edit text diantaranya 3 untuk input x,y, dan z,3 untuk output x,y,dan z fuzzy serta 3 untuk output teta1,teta2,dan teta3 fuzzy, 3 untuk output x,y, dan z rumus serta 3 untuk output teta1,teta2, dan teta3 rumus, ditambah satu untuk tampilan error

6. *Push Button*

   Pada GUI ini terdapat 4 push button diantaranya untuk tombol evaluate (tag property :evalbutton), tombol Rotate off (tag property : rotOff_button), tombol rotate on (tag property : rotOn_button), serta tombol reset (tag property : Reset_botton)

7. *Menu Editor*

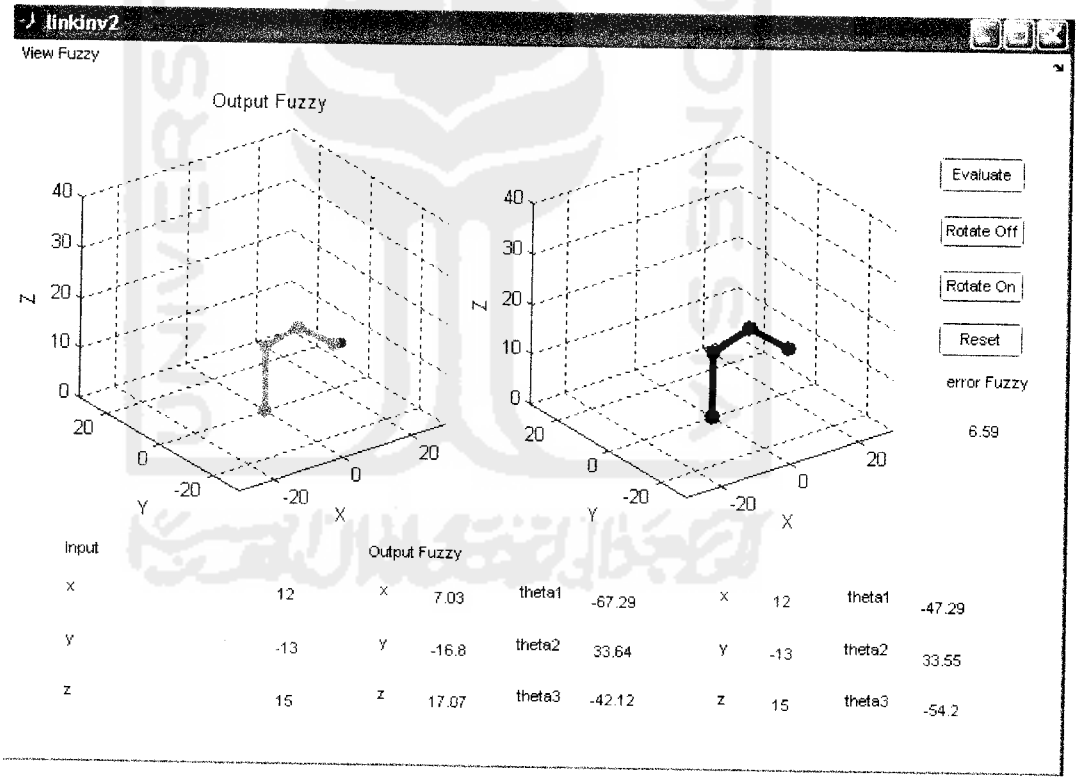
   Di GUI program ini terdapat sebuah menu editor yang berisi block diagram, rule view, surf view, member function x,y, dan z

Untuk menambahkan komponen yang dibutuhkan ke *Layout Area* klik komponen yang di perlukan dari komponen *pallette*, kemudian *drag* dan letakan pada *Layout Area*. Setelah komponen-komponen tersebut disusun di dalam *Layout Area* kemudian kita rapikan penempatannya dengan *align object* dari *menu tools* dengan cara pilih komponen yang ingin dirapikan sambil menekan *Ctrl* kemudian lakukan pengaturan sesuai dengan yang dikehendaki

Setelah melakukan pemrograman kita lakukan pengujian GUI dengan cara mengaktifkan GUI. Pengaktifan GUI melalui *M-file* dapat dilakukan dengan cara

klik *icon run* atau tekan F5 atau pilih *debug run.* Hasil pengaktifan GUI yang telah dibuat akan menampilkan seperti gambar (3.6).

Tag property berguna untuk mendiskripsikan nama fungsi yang akan di panggil. Setelah komponen selesai diatur nilainya kemudian disimpan dan diaktifkan dengan memilih *run*, dengan cara klik *icon run* atau pilih *tool, run* hal ini akan menghasilkan dua file, yaitu *file * fig (file figure)* dan *file * m (untuk pemrograman GUI)* biasa disebut dengan *M-File*. Komponen-komponen yang harus di program adalah *Edit Text, popup menu* dan *Push Button.* Program lengkap dari komponen-komponen GUI Matlab terlampir pada lampiran.



Gambar 3.11 Tampilan GUI *Inverse Kinematics* Manipulator 3D Berbasis Logika Fuzzy Yang Sudah Dieksekusi

3.3 Hasil Analisis

Dari data – data yang diperoleh melalui survei dan literatur – literatur selama penelitian maka dapat dilakukan proses analisa, maka didapatkan hasil analisa yang terdiri dari kebutuhan proses dan kebutuhan keluaran

3.3.1 Kebutuhan Proses

Masukan data yang dibutuhkan dalam penggunaan perangkat lunak ini adalah sebagai berikut :

1. Data posisi koordinat yang dapat yang dapat dijangkau oleh ujung manipulator.

   Inverse kinematics manipulator 3D dirancang menggunakan 3 link yang masing – masing linknya memiliki panjang L1 = 13, L2 = 10, L3 = 10 dengan posisi koordinat awal untuk x = 0, y = 0, dan z = 33 dan 3 dejarat kebebasan. Sehingga posisi x, y, dan z yang akan dituju sangat bervariasi sesuai dengan masukan nilai yang diinginkan dan posisi output yang diinginkan akan mendekati tepat bila menggunakan logika fuzzy. Posisi koordinat yang dapat dijangkau adalah posisi koordinat yang masih didalam ruang lingkup dibawah jangkauan terjauh dari ujung link manipulator *inverse kinematics* 3D.

2. Data sudut teta 1(θ_1) , teta 2 (θ_2), teta 3 (θ_3)

   Sudut yang dihasilkan juga bervariasi tergantung dari posisi koordinat yang dituju oleh ujung manipulator. Dimana besar sudut terjauh negatif (-)

dan terjauh positif (+) adalah -90 derajat dan +90 derajat untuk teta 2 (θ_2) dan teta 3 (θ_3) serta -180 derajat dan + 180 derajat untuk teta1 (θ_1).

3.3.2 Kebutuhan Keluaran

Data keluaran yang diperoleh dari proses simulasi adalah berapa besarnya sudut sudut teta 1(θ_1) , teta 2 (θ_2), teta 3 (θ_3) *inverse kinematics* manipulator 3D, dimana kemudian data sudut yang telah dari hasil *inverse kinematics* menggunakan logika fuzzy diforward kembali dengan menggunakan rumus untuk mendapatkan hasil output berupa koordinat x,y, dan z sebagai acuan buat koordinat input *inverse kinematics* manipulator 3D.

BAB IV

PENGAMATAN DAN ANALISA

Pada simulasi *inverse kinematics* manipulator 3D berbasis logika fuzzy ini menggunakan 2 metoda kinematics manipulator diantaranya *inverse kinematics* manipulator yang dikendalikan oleh logika fuzzy dan *forward kinematics* untuk menentukan titik koordinat orientasi x,y, dan z. Sistem kerjanya antara lain koordinat yang telah ditentukan yaitu x,y, dan z lalu diproses secara *inverse kinematics* dengan pengendali logika fuzzy untuk menghasilkan sudut teta1, teta2, dan teta3, setelah sudut ditemukan maka hasil yang berupa sudut teta1, teta2, dan teta3 tadi diforwardkan kembali dengan menggunakan rumus *forward kinematics* agar dapat menghasilkan titik koordinat orientasi sebagai pembanding seberapa besar error yang dihasilkan oleh fuzzy yang sudah dirancang, untuk menentukan error pada *invers kinematics* manipulator 3D dapat dihitung dengan rumus jarak antara titik yang dituju oleh ujung manipulator yang menggunakan pengendali fuzzy dengan ujung manipulator yang menggunakan rumus *forward kinematics*. Dengan rumus dapat dilihat pada perancanagn fuzzy pada bab III.

Pada bab ini akan dilakukan pengambilan data mulai 122 rule, 80 rule, dan 38 rule, serta akan dilakukan pengambilan data bila dilihat dari sudut pandang $x<y$, $x>y$, dan $x = y$. untuk pengambilan data mulai dari 122 rule, 80 rule, dan 38 rule data yang input yang digunakan sama yang diambil secara acak.

4.1 Pengambilan data *inverse* untuk 122 rule

Berikut adalah data – data *inverse kinematics* manipulator 3D yang terdiri dari 122 rule yang dirancang :

Tabel 4.1 Pengujian sistem fuzzy dengan 122 rule

NO	INPUT FUZZY & RUMUS			OUTPUT									ERROR
				FUZZY						RUMUS			
	X	Y	Z	TETA 1	TETA 2	TETA 3	X	Y	Z	TETA 1	TETA 2	TETA 3	
1	0	0	33	0	73.77	45	-2.02	0	31.37	0	90	0	2.60%
2	10	0	23	0	75.5	-69.62	12.45	0	23.71	0	90	-90	2.55%
3	0	10	23	90	75.5	-69.62	0	12.45	23.71	90	90	-90	2.55%
4	12	-13	15	-67.29	33.64	-42.12	7.03	-16.8	17.07	-47.29	33.55	-54.2	6.59%
5	0	-1	25	-13.18	74.24	36.15	-0.75	0.18	32	-90	138.22	-105.96	7.13%
6	9	4	9	37.74	19.69	-73.77	12.08	9.35	8.27	23.96	35.79	-115.79	6.22%
7	14	1	16	13.86	36.8	-46.83	17.33	4.28	17.25	4.09	56.21	-88.28	4.84%
8	10	3	22	30.12	69.19	-74.46	11.81	6.85	21.19	16.7	87.2	-92.87	4.33%
9	13	-13	20	-47.4	23.7	-25.59	12.96	-14.1	16.69	-45	31.23	-20.77	3.49%
10	1	0	30	0	75.41	30.53	-0.23	0	32.29	0	118.26	-63.26	2.60%
11	5	0	30	0	72.97	0	5.86	0	32.12	0	101.24	-55.25	2.29%
12	9	0	30	0	75.41	-30.53	9.6	0	29.73	0	78	-31.79	0.66%
13	5	12	21	63.46	38.07	-37.7	7.98	15.99	19.23	67.38	71.86	-80.5	5.29%
14	-2	-2	22	-0.32	68.99	35.75	1.04	-0.01	32.01	-135	134.41	-123.71	10.65%
15	13	10	10	59.35	16.36	-35.12	9.72	16.4	12.6	37.57	23.16	-67.05	7.65%
16	7	0	18	0	48.76	-19.22	15.29	0	25.45	0	100.06	-129.05	11.15%
17	11	2	25	23.15	41.55	-40.75	16.07	6.87	19.77	10.3	81.93	-69.82	8.77%
18	7	6	25	52.26	32.28	-9.84	10.83	13.99	22.16	40.6	93.3	-81.66	9.31%
19	11	2	3	24.23	-5.92	-48.16	14.42	6.49	3.87	10.3	-0.4	-82.82	5.71%
20	12	14	18	52.26	30	-29.73	11.42	14.76	18.05	49.4	32.38	-34.41	0.95%

Dari data diatas maka dapat dirata – rata errornya yaitu :

$$\text{rata- rata error} = \frac{105{,}33}{20}\ \% = 5{,}26\ \%$$

4.2 Pengambilan data *inverse* untuk 80 rule

Berikut adalah data – data *inverse kinematics* manipulator menggunakan 80 rule fuzzy :

Tabel 4.2 Pengujian sistem fuzzy dengan 80 rule

NO	INPUT FUZZY & RUMUS			OUTPUT									ERROR
				FUZZY						RUMUS			
	X	Y	Z	TETA 1	TETA 2	TETA 3	X	Y	Z	TETA 1	TETA 2	TETA 3	
1	0	0	33	0	0	0	20	0	13	0	90	0	28.28%
2	10	0	23	0	75.5	-75.5	12.5	0	22.68	0	90	-90	2.52%
3	0	10	23	90	75.5	-75.5	0	12.5	22.68	90	90	-90	2.52%
4	12	-13	15	-67.29	33.64	-42.12	7.03	-16.8	17.07	-47.29	33.55	-54.2	6.59%
5	0	-1	25	0	0	0	20	0	13	-90	138.22	-105.96	23.35%
6	9	4	9	37.74	19.69	-73.77	12.08	9.35	8.27	23.96	35.79	-115.79	6.22%
7	14	1	16	13.86	45.6	-73.77	15.35	3.79	15.42	4.09	56.21	-88.28	3.15%
8	10	3	22	30.12	69.19	-74.46	11.81	6.85	21.19	16.7	87.2	-92.87	4.33%
9	13	-13	20	-47.4	23.7	-25.59	12.96	-14.1	16.69	-45	31.23	-20.77	3.49%
10	1	0	30	0	0	0	20	0	13	0	118.26	-63.26	25.5%
11	5	0	30	0	0	0	20	0	13	0	101.24	-55.25	22.67
12	9	0	30	0	0	0	20	0	13	0	78	-31.79	20.25%
13	5	12	21	63.46	38.07	-37.7	7.98	15.99	19.23	67.38	71.86	-80.5	5.29%
14	-2	-2	22	-44.96	57.89	-70.12	10.68	-10.66	19.35	-135	134.41	-123.71	15.58%
15	13	10	10	59.35	0	-35.12	9.27	15.64	7.25	37.57	23.16	-67.05	7.3%
16	7	0	18	0	48.76	-73.77	15.66	0	16.29	0	100.06	-129.05	8.82%
17	11	2	25	23.15	52.49	-52.49	14.79	6.33	20.93	10.3	81.93	-69.82	7.05%
18	7	6	25	52.26	73.77	-73.77	7.83	10.12	22.26	40.6	93.3	-81.66	4.84%
19	11	2	3	24.23	-5.92	-48.16	14.42	6.49	3.87	10.3	-0.4	-82.82	5.71%
20	12	14	18	45	24.43	-24.43	13.51	13.51	17.14	49.4	32.38	-34.41	1.81%

Dari data diatas maka dapat dirata – rata errornya yaitu :

$$\text{rata- rata error} = \frac{209{,}27\ \%}{20} = 10{,}26\ \%$$

4.3 Pengambilan data *inverse* untuk 38 rule

Berikut adalah data – data *inverse kinematics* manipulator menggunakan 38 rule fuzzy :

Tabel 4.3 Pengujian sistem fuzzy dengan 38 rule

NO	INPUT FUZZY & RUMUS			OUTPUT									ERROR
				FUZZY						RUMUS			
	X	Y	Z	TETA 1	TETA 2	TETA 3	X	Y	Z	TETA 1	TETA 2	TETA 3	
1	0	0	33	0	0	0	20	0	13	0	90	0	28.28%
2	10	0	23	0	75.5	-75.5	12.5	0	22.68	0	90	-90	2.52%
3	0	10	23	90	75.5	-75.5	0	12.5	22.68	90	90	-90	2.52%
4	12	-13	15	0	0	0	20	0	13	-47.29	33.55	-54.2	15.39%
5	0	-1	25	0	0	0	20	0	13	-90	138.22	-105.96	23.35%
6	9	4	9	0	69.1	-69.1	13.57	0	22.34	23.96	35.79	-115.79	14.66%
7	14	1	16	37.39	69.44	-69.44	10.73	8.2	22.36	4.09	56.21	-88.28	10.15%
8	10	3	22	30.12	74.46	-74.46	10.97	6.36	23.63	16.7	87.2	-92.87	3.56%
9	13	-13	20	0	0	0	20	0	13	-45	31.23	-20.77	16.34%
10	1	0	30	0	0	0	20	0	13	0	118.26	-63.26	25.5%
11	5	0	30	0	0	0	20	0	13	0	101.24	-55.25	22.67%
12	9	0	30	0	0	0	20	0	13	0	78.	-31.79	20.25%
13	5	12	21	63.46	37.7	-37.7	8	16.02	19.12	67.38	71.86	-80.5	5.36%
14	-2	-2	22	-82.68	70.12	-70.12	1.71	-13.29	22.4	-135	134.41	-123.71	11.89%
15	13	10	10	0	0	0	20	0	13	37.57	23.16	-67.05	12.57%
16	7	0	18	0	72.09	-72.09	13.08	0	22.52	0	100.06	-129.05	7.57%
17	11	2	25	23.15	52.49	-52.49	14.79	6.33	20.93	10.3	81.93	-69.82	7.05%
18	7	6	25	52.26	73.77	-73.77	7.83	10.12	22.6	40.6	93.3	-81.66	4.84%
19	11	2	3	0	-38.07	-51.27	7.99	0	-3.17	10.3	-0.4	-82.82	7.15%
20	12	14	18	45	24.43	-24.43	13.51	13.51	17.14	49.4	32.38	-34.41	1.81%

Dari data diatas maka dapat dirata – rata errornya yaitu :

$$\text{rata- rata error} = \frac{243{,}43\ \%}{20} = 12{,}17\ \%$$

4.4 Analisa Perbandinagan error fuzzy berdasarkan 122 rule, 80 rule, dan 38 rule.

Dari pengambilan data - data diatas yaitu 122 rule, 80 rule, dan 38 rule terlihat bahwa dengan menggunakan rule yang lebih banyak maka error yang di hasilkan relati lebih kecil. Pada kelompok pertama yang menggunakan 122 rule terlihat rata – rata errornya lebih rendah karena aturan – aturan yang digunakan lebih lengkap, namun untuk 80 dan 38 rule hasil rata – rata errornya lebih besar bahkan ada beberapa titik tidak dapat terjangkau oleh manipulator yang menggunakan logika fuzzy, hal ini dikarenakan ada beberapa titik dapat kelompok 80 dan 38 rule tidak dimaping sehingga fuzzynya tidak terjangkau sehingga membuat error yang lebih besar.

4.5 Pengambilan data error fuzzy pada *inverse kinematics* manipulator bila x<y, x>y, dan x = y.

Pengambilan data dengan cara seperti memilki tujuan yang sama yaitu untuk melihat seberapa besar error yang diperoleh dari setiap percobaan. Namun disini data masukan untuk setiap variabel x,y, dan z dikombinasikan. Data yang diambil sampelnya hanya 5 data lalu dibandingkan.

4.5.1 Pengambilan data *inverse kinematics* manipulator 3D bila x<y

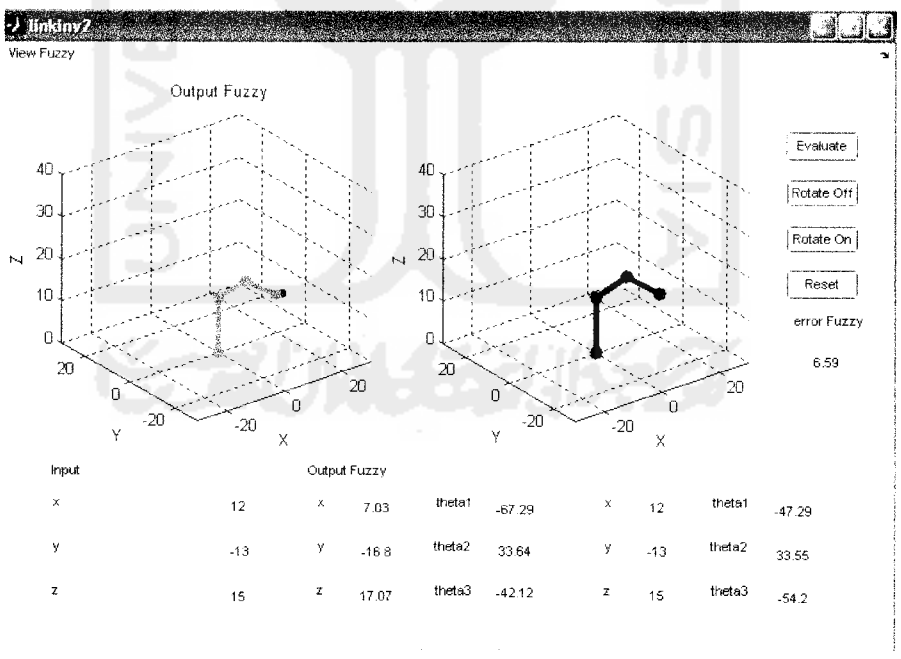
Tabel 4.4 Pengujian sistem fuzzy bila x <y

NO	INPUT FUZZY & RUMUS			OUTPUT									ERROR
				FUZZY						RUMUS			
	X	Y	Z	TETA 1	TETA 2	TETA 3	X	Y	Z	TETA 1	TETA 2	TETA 3	
1	12	14	18	52.26	30	-29.73	11.42	14.76	18.05	49.4	32.38	-34.41	0.95%
2	12	-13	15	-67.29	33.64	-42.12	7.03	-16.8	17.07	47.29	33.55	-54.2	6.59%
3	5	12	21	63.46	38.07	-37.7	7.98	15.99	19.23	67.38	71.86	-80.5	5.29%
4	0	10	23	90	75.5	69.62	0	12.45	23.71	90	90	-90	2.55%
5	8	13	16	59.88	30.5	-35.68	9.32	16.07	17.17	58.39	50.06	-77.88	3.54%

Dari data diatas maka dapat dirata – rata errornya yaitu :

$$\text{rata- rata error} = \frac{18{,}92}{5}\ \% = 3{,}78\ \%$$

Berikut tampilan simulasi pada GUI bila x < y dengan menggunakan salah satu sampel pada tabel diatas :



Gambar 4.1 Tampilan GUI Bila x < y

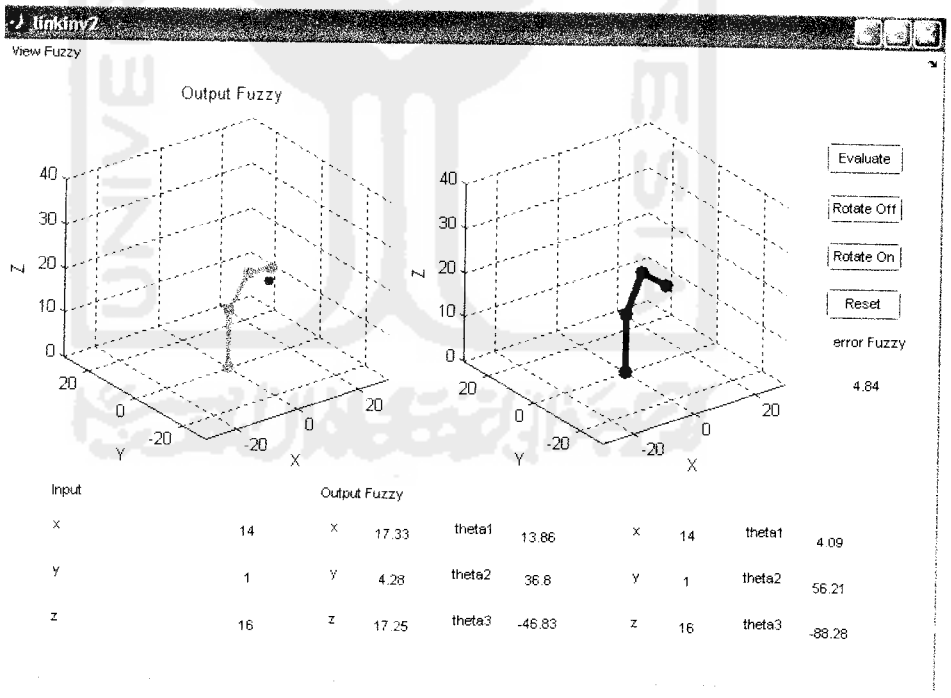
4.5.2 Pengambilan data *inverse kinematics* manipulator 3D bila x>y

Tabel 4.5 Pengujian sistem fuzzy bila x >y

NO	INPUT FUZZY & RUMUS			OUTPUT									ERROR
				FUZZY						RUMUS			
	X	Y	Z	TETA 1	TETA 2	TETA 3	X	Y	Z	TETA 1	TETA 2	TETA 3	
1	9	0	30	0	75.41	-30.53	9.6	0	29.73	0	78	-31.79	0.66%
2	10	0	23	0	75.5	-69.62	12.45	0	23.71	0	90	-90	2.55%
3	10	3	22	30.12	69.19	-74.46	11.81	6.85	21.19	16.7	87.2	-92.87	4.33%
4	14	1	16	13.86	36.8	-46.83	17.33	4.28	17.25	4.09	56.21	-88.28	4.84%
5	5	0	30	0	0	0	5.86	0	32.12	0	101.24	-55.25	2.29%

Dari data diatas maka dapat dirata – rata errornya yaitu :

rata- rata error = $\frac{14{,}67\ \%}{5} = 2{,}93\ \%$



Gambar 4.2 Tampilan GUI Bila x > y

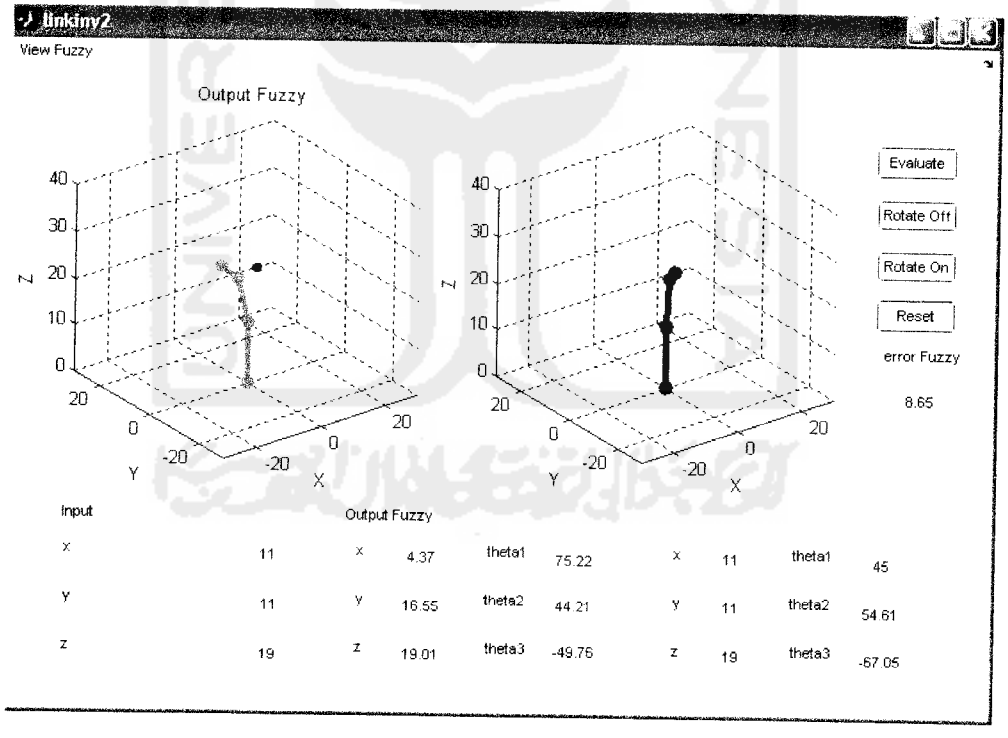
4.5.3 Pengambilan data *inverse kinematics* manipulator 3D bila x = y

Tabel 4.5 Pengujian sistem fuzzy bila x = y

NO	INPUT FUZZY & RUMUS			OUTPUT									ERROR
				FUZZY						RUMUS			
	X	Y	Z	TETA 1	TETA 2	TETA 3	X	Y	Z	TETA 1	TETA 2	TETA 3	
1	0	0	33	0	73.77	45	-2.02	0	31.37	0	90	0	2.60%
2	9	9	15	78.04	45.32	-75.41	3.25	15.34	15.1	45	58.82	-99.79	8.56%
3	7	7	24	59.88	42.12	-13.98	8.15	14.04	24.42	45	90.29	-84.55	7.15%
4	11	11	19	72.22	44.21	-49.76	4.37	16.55	19.01	45	54.61	-67.05	8.65%
5	6	6	10	52.26	3.14	-73.77	8.14	10.52	4.11	45	43.79	-126.51	7.72%

Dari data diatas maka dapat dirata – rata errornya yaitu :

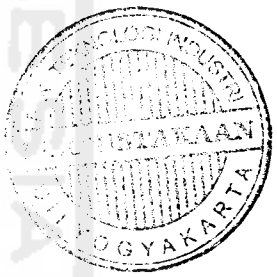
rata- rata error = $\frac{34{,}68}{5}$ % = 6,93 %



Gambar 4.3 Tampilan GUI Bila x = y

4.5.4 Analisa perbandingan Pengambilan data *inverse kinematics* manipulator 3D bila x < y, x > y, dan x = y

Dari pengambilan data berdasarkan x < y, x > y, da x = y, terlihat bahwa apabila x > y menghasilkan error yang paling kecil hal ini dipengaruhi oleh kombinasi data input baik itu x,y, dan z, dimana pada saat kombinasi data x lebih besar dari pada memiliki titik maping yang memiliki error yang cukup rendah, itu semua dikembalikan lagi pada susunan rule yang telah dirancang, apabila disuatu titik yang dipetakan sesuai dengan kombinasi masukannya maka errornya relatif kecil karena hampir mendekati titik yang dituju.



BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan pengujian pada simulasi *inverse kinematics* manipulator 3D berbasis logika fuzzy dengan tampilan visualisasi berupa GUI Matlab maka kita dapat menarik beberapa kesimpulan diantaranya :

1. Fungsi keanggotaan fuzzy sangat mempengaruhi presisi keluaran yang dihasilkan, semakin banyak fungsi keanggotaannya, maka presisi terhadap hasil keluaran yang diinginkan akan semakin baik
2. Dari pengambilan 20 sampel data dengan input yang sama yang telah rata – ratakan berdasarkan 3 kelompok rule fuzzy yaitu 122 rule, 80 rule, dan 38 rule maka terlihat bahwa menggunakan 122 rule menghasilkan persentase error yang relatif kecil hal ini dikarenakan oleh banyaknya aturan fuzzy yang digunakan untuk memetakan setiap titik yang akan dijangkau oleh manipulator dibandingkan menggunakan 80 dan 38 rule
3. Sistem simulasi *inverse kinematics* manipulator 3D ini dirancang menggunakan logika fuzzy dengan menggunakan fungsi segitiga sebagai representasi input untuk posisi koordinat yang dituju yaitu berupa titik x,y,dan z serta untuk representasi output yaitu berupa sudut teta 1(θ_1), teta 2 (θ_2), dan teta 3 (θ_3).

4. Sistem fuzzy yang digunakan pada sistem simulasi *inverse kinematics* manipulator 3D ini menggunakan sistem inferensi Mamdani. Rule fuzzy sendiri memiliki bobot 1 untuk tiap rulenya, dimana tiap inputnya memakai 5 mf (*member function*), sehingga didapat jumlah rule secara keseluruhan adalah 125 rule dengan cara pengalian 5x5x5 dan rule yang dipakai hanya 122 rule

5.2 Saran

1. Agar lebih interaktif, bagi yang ingin menambahkan, sebaiknya *software* yang telah ada sekarang ditambah lagi dengan beberapa algoritma lain, seperti algoritma jaringan syaraf tiruan. Kemudian membandingkan dan mencari keunggulan dari masing-masing algoritma tersebut.
2. Bagi yang ingin mengembangkan, Simulasi *inverse kinematics* manipulator 3D berbasis logika fuzzy ini menggunakan 3 link, maka sebaiknya ada yang mampu mengembangkan *invers kinematics* manipulator 3D yang memakai 4 link dengan metoda logika fuzzy yang hasil errornya lebih kecil lagi.

DAFTAR PUSTAKA

1. Arhami, Muhammad , S.Si., M. Kom & Desiani, Anita , S.Si., M. Kom ,. 2004. *Pemrograman MATLAB*. Penerbit Andi, Yogyakarta,
2. Craig, John J., 1955. *Introduction to Robotics Mechanics & Control*. USA:Addison-Wesley Publishing Company, Inc.
3. Dwane Hanselman & Bruce Littlefield,. 2000. *MATLAB Bahasa Komputasi Teknis*. Yogyakarta : Andi.
4. Fu, K. S., 1987. *Robotics: control, sensing, vision and intelligence*. Singapore: McGraw-Hill,Inc.
5. Halliday-Resnick,. 1987. *Fisika Jilid 1, edisi ke 3.*
6. Sri Kusumadeawi dan, Hari Purnomo. 2004. *Aplikasi Logika Fuzzy untuk Pendukung Keputusan*. Penerbit Graha Ilmu. Yogyakarta.
7. Sri Kusumadewi. 2002. *Analisis & Desain Sistem Fuzzy Menggunakan Toolbox MATLAB*.
8. Silaban, Pantur, Ph.D. & Sucipto, Erwin, Drs, M.Sc.. 1985. *Fisika Jilid 1, edisi ke 3*. Penerbit erlangga. Bandung.
9. Setiawan, Hendra, 2004. *Bahan Mata Kuliah Robotika*. Yogyakarta: Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia.

LAMPIRAN ATURAN – ATURAN FUZZY YANG DIGUNAKAN

X	Y	Z	TETA 1	TETA 2	TETA 3
NB	NB	NB	Z	Z	Z
		NS	Z	Z	Z
		Z	Z	Z	Z
		PS	Z	Z	Z
		PB	Z	Z	Z
	NS	NB	Z	Z	Z
		NS	Z	Z	Z
		Z	Z	Z	Z
		PS	NB	PB	NS
		PB	Z	Z	Z
	Z	NB	Z	Z	Z
		NS	Z	Z	Z
		Z	Z	Z	Z
		PS	Z	Z	Z
		PB	Z	Z	Z
	PS	NB	Z	Z	Z
		NS	PB	Z	Z
		Z	Z	Z	Z
		PS	Z	Z	Z
		PB	PB	PS	Z
	PB	NB	Z	Z	Z
		NS	Z	Z	Z
		Z	Z	Z	Z
		PS	Z	Z	Z
		PB	Z	Z	Z
NS	NB	NB	Z	Z	Z
		NS	Z	Z	Z
		Z	Z	Z	Z
		PS	Z	Z	Z
		PB	Z	Z	Z
	NS	NB	NB	NS	NS
		NS	NB	Z	NB
		Z	NB	PS	NB
		PS	NB	PB	NB
		PB	Z	Z	Z
	Z	NB	PB	Z	NB
		NS	PB	PS	NB
		Z	PB	PS	NB
		PS	PB	PB	NB
		PB	PB	PB	NB
	PS	NB	PB	NS	NS
		NS	PB	PS	NB
		Z	PB	PS	NB

		PS	PB	PB	NB
		PB	Z	Z	Z
	PB	NB	Z	Z	Z
		NS	Z	Z	Z
		Z	PS	Z	Z
		PS	Z	Z	Z
		PB	Z	Z	Z
Z	NB	NB	Z	Z	Z
		NS	Z	Z	Z
		Z	Z	Z	Z
		PS	Z	Z	Z
		PB	Z	Z	Z
	NS	NB	NS	Z	NB
		NS	NS	Z	NB
		Z	NS	PB	NB
		PS	NS	PB	NB
		PB	NS	PB	NS
	Z	NB	Z	NS	NB
		NS	Z	NS	NB
		Z	Z	PB	NB
		PS	Z	PB	PB
		PB	Z	PB	PS
	PS	NB	PS	Z	NB
		NS	PS	PS	NB
		Z	PS	PS	NB
		PS	PS	PB	NB
		PB	PS	PB	NS
	PB	NB	Z	Z	Z
		NS	Z	Z	Z
		Z	Z	Z	Z
		PS	Z	Z	Z
		PB	Z	Z	Z
	NB	NB	Z	Z	Z
		NS	Z	Z	Z
		Z	Z	Z	Z
		PS	Z	Z	Z
		PB	Z	Z	Z
	NS	NB	Z	NS	NS
		NS	NS	Z	NB
		Z	NS	PS	NB
		PS	Z	PB	Z
		PB	Z	Z	Z
		NB	Z	NS	NB
		NS	Z	PS	NB

PS	Z	Z	Z	PS	NB
		PS	Z	PB	NB
		PB	Z	PB	NS
	PS	NB	PS	NS	NS
		NS	PS	Z	NB
		Z	PS	PS	NB
		PS	PS	PB	NB
		PB	Z	Z	Z
	PB	NB	Z	Z	Z
		NS	Z	Z	Z
		Z	Z	Z	Z
		PS	Z	Z	Z
		PB	Z	Z	Z
PB	NB	NB	Z	Z	Z
		NS	Z	Z	Z
		Z	Z	Z	Z
		PS	Z	Z	Z
		PB	Z	Z	Z
	NS	NB	Z	Z	Z
		NS	Z	Z	Z
		Z	Z	Z	Z
		PS	Z	Z	Z
		PB	Z	Z	Z
	Z	NB	Z	Z	Z
		NS	Z	Z	Z
		Z	Z	PS	NB
		PS	Z	Z	Z
		PB	Z	Z	Z
	PS	NB	Z	Z	Z
		NS	Z	Z	Z
		Z	Z	Z	Z
		PS	Z	Z	Z
		PB	Z	Z	Z
	PB	NB	Z	Z	Z
		NS	Z	Z	Z
		Z	Z	Z	Z
		PS	Z	Z	Z
		PB	Z	Z	Z

Keterangan :

NB = Negatif Big

NS = Negatif Small

Z = Zero

PS = Positif Small

PB = Positif Big

LAMPIRAN ATURAN – ATURAN FUZZY YANG DIGUNAKAN

1. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

2. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

3. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

4. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is NEGATIF DEKAT and SUDUT TETA – 3 is NEGATIF JAUH

5. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

6. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

7. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

8. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

9. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

10. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

11. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

12. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is SEDANG then

SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

13. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

14. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

15. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

16. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is NEGATIF DEKAT and SUDUT TETA– 2 is POSITIF JAUH and SUDUT TETA – 3 is NEGATIF JAUH

17. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is POSITIF JAUH and SUDUT TETA – 3 is NEGATIF JAUH

18. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

19. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is POSITIF JAUH and SUDUT TETA – 3 is NEGATIF JAUH

20. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is POSITIF DEKAT and SUDUT TETA– 2 is POSITIF JAUH and SUDUT TETA – 3 is NEGATIF JAUH

21. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

22. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

23. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is POSITIF JAUH and SUDUT TETA– 2 is POSITIF DEKAT and SUDUT TETA – 3 is NEGATIF JAUH

24. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

25. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is POSITIF DEKAT and SUDUT TETA– 2 is POSITIF JAUH and SUDUT TETA – 3 is NEGATIF JAUH

26. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

27. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

28. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

29. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is POSITIF DEKAT

then SUDUT TETA – 1 is NEGATIF JAUH and SUDUT TETA– 2 is POSITIF JAUH and SUDUT TETA – 3 is NEGATIF JAUH

30. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is POSITIF JAUH and SUDUT TETA– 2 is POSITIF DEKAT and SUDUT TETA – 3 is NEGATIF JAUH

31. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is POSITIF JAUH and SUDUT TETA– 2 is POSITIF DEKAT and SUDUT TETA – 3 is NEGATIF JAUH

32. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

33. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

34. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

35. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is NEGATIF JAUH and SUDUT TETA– 2 is NEGATIF DEKAT and SUDUT TETA – 3 is NEGATIF DEKAT

36. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is NEGATIF DEKAT and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is NEGATIF JAUH

37. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is NEGATIF DEKAT and SUDUT TETA – 3 is NEGATIF DEKAT

38. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

39. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is POSITIF JAUH and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is NEGATIF JAUH

40. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is NEGATIF DEKAT and SUDUT TETA – 3 is NEGATIF JAUH

41. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is POSITIF JAUH and SUDUT TETA– 2 is NEGATIF DEKAT and SUDUT TETA – 3 is NEGATIF DEKAT

42. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is POSITIF DEKAT and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is NEGATIF JAUH

43. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

44. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is POSITIF DEKAT and SUDUT TETA– 2 is NEGATIF DEKAT and SUDUT TETA – 3 is NEGATIF DEKAT

45. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

46. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is NEGATIF JAUH then

SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

47. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

48. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

49. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

50. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

51. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

52. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

53. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

54. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is NEGATIF DEKAT and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is NEGATIF JAUH

55. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is NEGATIF DEKAT and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is NEGATIF JAUH

56. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

57. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is NEGATIF DEKAT and SUDUT TETA – 3 is NEGATIF JAUH

58. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is POSITIF DEKAT and SUDUT TETA – 3 is NEGATIF JAUH

59. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

60. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is POSITIF JAUH and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

61. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is POSITIF DEKAT and SUDUT TETA– 2 is POSITIF DEKAT and SUDUT TETA – 3 is NEGATIF JAUH

62. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is POSITIF DEKAT and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is NEGATIF JAUH

63. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is NEGATIF DEKAT

then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

64. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

65. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

66. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

67. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

68. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is NEGATIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

69. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

70. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

71. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

72. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

73. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is NEGATIF JAUH and SUDUT TETA– 2 is POSITIF DEKAT and SUDUT TETA – 3 is NEGATIF JAUH

74. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is NEGATIF DEKAT and SUDUT TETA– 2 is POSITIF DEKAT and SUDUT TETA – 3 is NEGATIF JAUH

75. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

76. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

77. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is POSITIF JAUH and SUDUT TETA– 2 is POSITIF DEKAT and SUDUT TETA – 3 is NEGATIF JAUH

78. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is POSITIF DEKAT and SUDUT TETA – 3 is NEGATIF JAUH

79. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

80. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is SEDANG then SUDUT

TETA – 1 is POSITIF DEKAT and SUDUT TETA– 2 is POSITIF DEKAT and SUDUT TETA – 3 is NEGATIF JAUH

81. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is POSITIF DEKAT and SUDUT TETA– 2 is POSITIF DEKAT and SUDUT TETA – 3 is NEGATIF JAUH

82. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

83. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

84. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

85. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

86. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is SEDANG then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

87. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

88. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

89. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

90. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

91. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

92. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

93. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

94. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is POSITIF JAUH and SUDUT TETA – 3 is POSITIF JAUH

95. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is POSITIF JAUH and SUDUT TETA– 2 is POSITIF JAUH and SUDUT TETA – 3 is NEGATIF JAUH

96. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

97. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is POSITIF DEKAT

then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

98. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is POSITIF JAUH and SUDUT TETA– 2 is POSITIF JAUH and SUDUT TETA – 3 is NEGATIF JAUH

99. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

100. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

101. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

102. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is POSITIF DEKAT then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

103. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

104. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

105. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

106. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is NEGATIF JAUH and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

107. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

108. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

109. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is NEGATIF DEKAT and SUDUT TETA– 2 is POSITIF JAUH and SUDUT TETA – 3 is NEGATIF DEKAT

110. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

111. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is NEGATIF DEKAT and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

112. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

113. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is POSITIF DEKAT and SUDUT TETA– 2 is POSITIF JAUH and SUDUT TETA – 3 is NEGATIF DEKAT

114. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is POSITIF JAUH

then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

115. If POSISI KOORDINAT – X is NEGATIF JAUH and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

116. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

117. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is POSITIF JAUH and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

118. If POSISI KOORDINAT – X is POSITIF DEKAT and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is POSITIF JAUH and SUDUT TETA – 3 is NEGATIF DEKAT

119. If POSISI KOORDINAT – X is POSITIF JAUH and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is SEDANG and SUDUT TETA – 3 is SEDANG

120. If POSISI KOORDINAT – X is SEDANG and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is SEDANG and SUDUT TETA– 2 is POSITIF JAUH and SUDUT TETA – 3 is POSITIF DEKAT

121. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is SEDANG and POSISI KOORDINAT – Z is POSITIF JAUH then SUDUT TETA – 1 is POSITIF JAUH and SUDUT TETA– 2 is POSITIF JAUH and SUDUT TETA – 3 is NEGATIF JAUH

122. If POSISI KOORDINAT – X is NEGATIF DEKAT and POSISI KOORDINAT – Y is POSITIF DEKAT and POSISI KOORDINAT – Z is NEGATIF DEKAT then SUDUT TETA – 1 is POSITIF JAUH and SUDUT TETA– 2 is POSITIF DEKAT and SUDUT TETA – 3 is NEGATIF JAUH

LAMPIRAN – LAMPIRAN PROGRAM

```
varargout = linkinv2(varargin)
```

```
_Singleton = 1;
_State = struct('gui_Name',       mfilename, ...
                'gui_Singleton',  gui_Singleton, ...
                'gui_OpeningFcn', @linkinv2_OpeningFcn, ...
                'gui_OutputFcn',  @linkinv2_OutputFcn, ...
                'gui_LayoutFcn',  [] , ...
                'gui_Callback',   []);
iargin && ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});


iargout
[varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});

gui_mainfcn(gui_State, varargin{:});
```

```
linkinv2_OpeningFcn(hObject, eventdata, handles, varargin)
```

```
  0;
  0;
  33;
 [13 10 10];
 [0 0 0];
 [0 180 0];

tion = CartePosition(l,A,B);
= position(1,:);
= position(2,:);
= position(3,:);
= position(4,:);
 [V0(:,1) V1(:,1) V2(:,1) V3(:,1)];
 [V0(:,1) V1(:,2) V2(:,2) V3(:,2)];
 [V0(:,3) V1(:,3) V2(:,3) V3(:,3)];

s(handles.axesCCD);
:3(x,y,z,'-ko',...
                'LineWidth',4,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','k',...
                'MarkerSize',5)
 on
 3(xd,yd,zd,'-ro',...
                'LineWidth',2,...
                'MarkerEdgeColor','r',...
                'MarkerFaceColor','r',...
                'MarkerSize',5)
 off
bel('X');
bel('Y');
bel('Z');
s([-30 30 -30 30 0 40]);
 normal;
 on;

s(handles.axesFuzzy);
:3(x,y,z,'-ko',...
                'LineWidth',4,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','k',...
                'MarkerSize',5)
 on
:3(xd,yd,zd,'-ro',...
                'LineWidth',2,...
                'MarkerEdgeColor','r',...
                'MarkerFaceColor','r',...
                'MarkerSize',5)
 off
```

```
el('X');
el('Y');
el('Z');
([-30 30 -30 30 0 40]);
;
;

handles.xslider,'Value',xd)
handles.yslider,'Value',yd)
handles.zslider,'Value',zd)
handles.xbox_input,'String',xd)
handles.ybox_input,'String',yd)
handles.zbox_input,'String',zd)
handles.xoutFuzzybox,'String',0)
handles.youtFuzzybox,'String',0)
handles.zoutFuzzybox,'String','0')
handles.xOut,'String',0)
handles.yOut,'String',0)
handles.zOut,'String','0')
handles.theta1Fbox,'String',0)
handles.theta2Fbox,'String',0)
handles.theta3Fbox,'String',0)
handles.theta1box,'String',0)
handles.theta2box,'String',0)
handles.theta3box,'String',0)
handles.deltaFUZZYbox,'String',0)


les.output = hObject;

ata(hObject, handles);



    varargout = linkinv2_OutputFcn(hObject, eventdata, handles)



rgout{1} = handles.output;

    zoutFuzzybox_Callback(hObject, eventdata, handles)
```

```
zoutFuzzybox_CreateFcn(hObject, eventdata, handles)
```

```
spc && isequal(get(hObject,'                    '), get↙
                    '))
set(hObject,'                    ','                    ');
```

```
xoutFuzzybox_Callback(hObject, eventdata, handles)
```

```
xoutFuzzybox_CreateFcn(hObject, eventdata, handles)
```

```
spc && isequal(get(hObject,'                    '), get↙
                    '))
set(hObject,'                    ','                    ');
```

```
youtFuzzybox_Callback(hObject, eventdata, handles)
```

```
youtFuzzybox_CreateFcn(hObject, eventdata, handles)
```

```
spc && isequal(get(hObject,'[illegible]'), get
[illegible]'))
set(hObject,'[illegible]','[illegible]');
```

```
Reset_button_Callback(hObject, eventdata, handles)
```

```
0;
0;
33;
[13 10 10];
[0 180 0];
[0 90 0];

tion = CartePosition(1,A,B);
position(1,:);
position(2,:);
position(3,:);
position(4,:);
[V0(:,1) V1(:,1) V2(:,1) V3(:,1)];
[V0(:,1) V1(:,2) V2(:,2) V3(:,2)];
[V0(:,3) V1(:,3) V2(:,3) V3(:,3)];


(handles.axesCCD);
3(x,y,z,'-k.',...
              '[illegible]',4,...
              'Ma[illegible]','.',...
              'Ma[illegible]','.',...
              'Ma[illegible]',5)
3(xd,yd,zd,'-[illegible]',...
              '[illegible]',2,...
```

```
                'MarkerEdgeColor','r',...
                'MarkerFaceColor','b',...
                'MarkerSize',5)
  off
el('X');
el('Y');
el('Z');

([-30 30 -30 30 0 40]);
;
;

(handles.axesFuzzy);
3(x,y,z,'-bo',...
                'LineWidth',4,...
                'MarkerEdgeColor','b',...
                'MarkerFaceColor','b',...
                'MarkerSize',5)
on
3(xd,yd,zd,'-ro',...
                'LineWidth',2,...
                'MarkerEdgeColor','r',...
                'MarkerFaceColor','r',...
                'MarkerSize',5)
  off
el('X');
el('Y');
el('Z');
([-30 30 -30 30 0 40]);
  ;
  on;


handles.xslider,'Value',xd)
handles.yslider,'Value',yd)
handles.zslider,'Value',zd)
handles.xbox_input,'String',xd)
handles.ybox_input,'String',yd)
handles.zbox_input,'String',zd)
handles.xoutFuzzybox,'String',0)
handles.youtFuzzybox,'String',0)
handles.zoutFuzzybox,'String','X')
handles.xOut,'String',0)
handles.yOut,'String',0)
handles.zOut,'String','X')
handles.theta1Fbox,'String',0)
handles.theta2Fbox,'String',0)
handles.theta3Fbox,'String',0)
handles.theta1box,'String',0)
handles.theta2box,'String',0)
handles.theta3box,'String',0)
handles.deltaFUZZYbox,'String',0)
```

```
ta(hObject, handles);

      rotOn_button_Callback(hObject, eventdata, handles)

te3d

      rotOff_button_Callback(hObject, eventdata, handles)

te3d

      xslider_Callback(hObject, eventdata, handles)

 get(hObject,'
 get(handles.yslider,'
 get(handles.zslider,'
handles.xbox_input,'
handles.ybox_input,'
handles.zbox_input,'

= readfis('
a = evalfis([xd;yd;zd],fis);
F = [theta(1),0;theta(2),0;theta(3),0];
handles.theta1Fbox,'
handles.theta2Fbox,'
handles.theta3Fbox,'
les.tetaF=tetaF*pi/180;

les.teta_akhirF=[];
F=handles.tetaF;
_awal1F=handles.teta_akhirF;
ksF=1;
tF=0:1:10;
 isempty(teta_awal1F);
    teta_awal1F=zeros(3,2);
teta_awalF=teta_awal1F(:,indeksF);
```

```
eta_akhirF=handles.tetaF(:,indeksF);
        axes(handles.axesFuzzy)
        cla
        axis
        handles.pF=plot3(0,0,0,'
                ',4,...
                ',5);
        hold
        plot3(xd,yd,zd,'
                        ',2,...
                        ',5)
        hold
        xlabel('
        ylabel('
        zlabel('
        axis([-30 30 -30 30 0 40]);
        grid

teta_akhirF=[tetaF(1,indeksF);0;0];
    kF=1:length(tF)
    teta1F=trajectory(teta_awalF',teta_akhirF',10,tF(kF));
    link_parameterF=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1F;0]];
    pointF=[0;0;0];
    [keluaranF,xF,yF,zF]=data_robot(link_parameterF,pointF);
    set(handles.pF,'
    set(handles.xoutFuzzybox,'
    set(handles.youtFuzzybox,'
    set(handles.zoutFuzzybox,'
    MF(kF) = getframe;

teta_awalF=teta_akhirF;
teta_akhirF=[tetaF(1,indeksF);tetaF(2,indeksF);0];
    kF=1:length(tF)
    teta1F=trajectory(teta_awalF',teta_akhirF',10,tF(kF));
    link_parameterF=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1F;0]];
    pointF=[0;0;0];
    [keluaranF,xF,yF,zF]=data_robot(link_parameterF,pointF);
    set(handles.pF,'
    set(handles.xoutFuzzybox,'
    set(handles.youtFuzzybox,'
    set(handles.zoutFuzzybox,'
    MF(kF) = getframe;

teta_awalF=teta_akhirF;
teta_akhirF=[tetaF(1,indeksF);tetaF(2,indeksF);tetaF(3,indeksF)];
    kF=1:length(tF)
    teta1F=trajectory(teta_awalF',teta_akhirF',10,tF(kF));
```

```
    link_parameterF=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;tetaIF;0]];
    pointF=[0;0;0];
    [keluaranF,xF,yF,zF]=data_robot(link_parameterF,pointF);
    set(handles.pF,'
    set(handles.xoutFuzzybox,'
    set(handles.youtFuzzybox,'
    set(handles.zoutFuzzybox,'
    MF(kF) = getframe;

handles.teta_awalF(:,indeksF)=teta_akhirF;
les.teta_akhirF=handles.teta_awalF;


les.teta_akhir=[];
[xd yd zd];
k=[xyz(1);xyz(2);xyz(3)];
1=atan2(titik(2,1),titik(1,1))*180/pi;
k1=[sqrt(xyz(1)^2+xyz(2)^2);xyz(3)-13];
a23,str] = invers_kinematik(titik1(1),titik1(2));
isempty(teta23)
teta_1=[teta1;teta23(:,1)];
teta_2=[teta1;teta23(:,2)];
teta=[teta_1 teta_2];
set(handles.theta1box,'
set(handles.theta2box,'
set(handles.theta3box,'
handles.teta=teta*pi/180;
teta=handles.teta;
teta_awal1=handles.teta_akhir;
indeks=1;
    t=0:1:10;
      isempty(teta_awal1);
       teta_awal1=zeros(3,2);
    teta_awal=teta_awal1(:,indeks);
    teta_akhir=handles.teta(:,indeks);

        axes(handles.axesCCD)
        cla
        axis
        handles.p=plot3(0,0,0,'
                '
                '
                '
                '
        hold
        plot3(xd,yd,zd,'
                        '
                        '
                        '
                        '
        hold
        xlabel('X')
```

```
        ylabel('y')
        zlabel('z')
        axis([-30 30 -30 30 0 40]);
        grid on

    teta_akhir=[teta(1,indeks);0;0];
        k=1:length(t)
        teta1=trajectory(teta_awal',teta_akhir',10,t(k));
        link_parameter=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1;0]];
        point=[0;0;0];
        [keluaran,x,y,z]=data_robot(link_parameter,point);
        set(handles.p,'XData',x,'YData',y,'ZData',z);
        set(handles.xOut,'String',(0.01*round(100*(keluaran(1)))));
        set(handles.yOut,'String',(0.01*round(100*(keluaran(2)))));
        set(handles.zOut,'String',(0.01*round(100*(keluaran(3)))));
        M(k) = getframe;

    teta_awal=teta_akhir;
    teta_akhir=[teta(1,indeks);teta(2,indeks);0];
        k=1:length(t)
        teta1=trajectory(teta_awal',teta_akhir',10,t(k));
        link_parameter=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1;0]];
        point=[0;0;0];
        [keluaran,x,y,z]=data_robot(link_parameter,point);
        set(handles.p,'XData',x,'YData',y,'ZData',z);
        set(handles.xOut,'String',(0.01*round(100*(keluaran(1)))));
        set(handles.yOut,'String',(0.01*round(100*(keluaran(2)))));
        set(handles.zOut,'String',(0.01*round(100*(keluaran(3)))));
        M(k) = getframe;

    teta_awal=teta_akhir;
    teta_akhir=[teta(1,indeks);teta(2,indeks);teta(3,indeks)];
        k=1:length(t)
        teta1=trajectory(teta_awal',teta_akhir',10,t(k));
        link_parameter=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1;0]];
        point=[0;0;0];
        [keluaran,x,y,z]=data_robot(link_parameter,point);
        set(handles.p,'XData',x,'YData',y,'ZData',z);
        set(handles.xOut,'String',(0.01*round(100*(keluaran(1)))));
        set(handles.yOut,'String',(0.01*round(100*(keluaran(2)))));
        set(handles.zOut,'String',(0.01*round(100*(keluaran(3)))));
        M(k) = getframe;

    handles.teta_awal(:,indeks)=teta_akhir;
handles.teta_akhir=handles.teta_awal;

msgbox('Titik di luar jangkauan','','help')


 [keluaranF(1) keluaranF(2) keluaranF(3)];
 [xd yd zd];
anceF = Distance(VD,VF);
```

```
handles.deltaFUZZYbox,'String',distanceF)
ata(hObject, handles);


      xslider_CreateFcn(hObject, eventdata, handles)



sequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor',[.9 .9 .9]);


      yslider_Callback(hObject, eventdata, handles)



 get(handles.xslider,'Value');
 get(hObject,'Value');
 get(handles.zslider,'Value');
handles.xbox_input,'String',(0.01*round(100*(xd))))
handles.ybox_input,'String',(0.01*round(100*(yd))))
handles.zbox_input,'String',(0.01*round(100*(zd))))


= readfis('tugasKin');
a = evalfis([xd;yd;zd],fis);
F = [theta(1),0;theta(2),0;theta(3),0];
handles.theta1Fbox,'String',(0.01*round(100*tetaF(1,1))))
handles.theta2Fbox,'String',(0.01*round(100*tetaF(2,1))))
handles.theta3Fbox,'String',(0.01*round(100*tetaF(3,1))))
les.tetaF=tetaF*pi/180;

les.teta_akhirF=[];
F=handles.tetaF;
_awal1F=handles.teta_akhirF;
ksF=1;
tF=0:1:10;
   isempty(teta_awal1F);
    teta_awal1F=zeros(3,2);

teta_awalF=teta_awal1F(:,indeksF);
teta_akhirF=handles.tetaF(:,indeksF);
```

```
        axes(handles.axesFuzzy)
        cla
        axis equal
        handles.pF=plot3(0,0,0,'-r*','LineWidth',4,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','r',...
                'MarkerSize',5);
        hold on
        plot3(xd,yd,zd,'-ro',...
                        'LineWidth',2,...
                        'MarkerEdgeColor','r',...
                        'MarkerFaceColor','r',...
                        'MarkerSize',5)
        hold off
        xlabel('X')
        ylabel('Y')
        zlabel('Z')
        axis([-30 30 -30 30 0 40]);
        grid on

teta_akhirF=[tetaF(1,indeksF);0;0];
    kF=1:length(tF)
    teta1F=trajectory(teta_awalF',teta_akhirF',10,tF(kF));
    link_parameterF=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1F;0]];
    pointF=[0;0;0];
    [keluaranF,xF,yF,zF]=data_robot(link_parameterF,pointF);
    set(handles.pF,'XData',xF,'YData',yF,'ZData',zF);
    set(handles.xoutFuzzybox,'String',(0.01*round(100*(keluaranF(1)))))
    set(handles.youtFuzzybox,'String',(0.01*round(100*(keluaranF(2)))))
    set(handles.zoutFuzzybox,'String',(0.01*round(100*(keluaranF(3)))))
    MF(kF) = getframe;


teta_awalF=teta_akhirF;
teta_akhirF=[tetaF(1,indeksF);tetaF(2,indeksF);0];
    kF=1:length(tF)
    teta1F=trajectory(teta_awalF',teta_akhirF',10,tF(kF));
    link_parameterF=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1F;0]];
    pointF=[0;0;0];
    [keluaranF,xF,yF,zF]=data_robot(link_parameterF,pointF);
    set(handles.pF,'XData',xF,'YData',yF,'ZData',zF);
    set(handles.xoutFuzzybox,'String',(0.01*round(100*(keluaranF(1)))))
    set(handles.youtFuzzybox,'String',(0.01*round(100*(keluaranF(2)))))
    set(handles.zoutFuzzybox,'String',(0.01*round(100*(keluaranF(3)))))
    MF(kF) = getframe;


teta_awalF=teta_akhirF;
teta_akhirF=[tetaF(1,indeksF);tetaF(2,indeksF);tetaF(3,indeksF)];
    kF=1:length(tF)
    teta1F=trajectory(teta_awalF',teta_akhirF',10,tF(kF));
    link_parameterF=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1F;0]];
```

```
    pointF=[0;0;0];
    [keluaranF,xF,yF,zF]=data_robot(link_parameterF,pointF);
    set(handles.pF,'XData',xF,'YData',yF,'ZData',zF);
    set(handles.xoutFuzzybox,'String',(0.01*round(100*(keluaranF(1)))))
    set(handles.youtFuzzybox,'String',(0.01*round(100*(keluaranF(2)))))
    set(handles.zoutFuzzybox,'String',(0.01*round(100*(keluaranF(3)))))
    MF(kF) = getframe;

handles.teta_awalF(:,indeksF)=teta_akhirF;
les.teta_akhirF=handles.teta_awalF;


les.teta_akhir=[];
[xd yd zd];
k=[xyz(1);xyz(2);xyz(3)];
1=atan2(titik(2,1),titik(1,1))*180/pi;
k1=[sqrt(xyz(1)^2+xyz(2)^2);xyz(3)-13];
a23,str] = invers_kinematik(titik1(1),titik1(2));
isempty(teta23)
teta_1=[teta1;teta23(:,1)];
teta_2=[teta1;teta23(:,2)];
teta=[teta_1 teta_2];
set(handles.theta1box,'String',(0.01*round(100*(teta(1,1)))));
set(handles.theta2box,'String',(0.01*round(100*(teta(2,1)))));
set(handles.theta3box,'String',(0.01*round(100*(teta(3,1)))));
handles.teta=teta*pi/180;
teta=handles.teta;
teta_awal1=handles.teta_akhir;
indeks=1;
    t=0:1:10;
      isempty(teta_awal1);
       teta_awal1=zeros(3,2);
    teta_awal=teta_awal1(:,indeks);
    teta_akhir=handles.teta(:,indeks);

        axes(handles.axesCCD)
        cla
        axis equal
        handles.p=plot3(0,0,0,'-o','LineWidth',4,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','r',...
                'MarkerSize',5);
        hold on
        plot3(xd,yd,zd,'-o',...
                        'LineWidth',2,...
                        'MarkerEdgeColor','m',...
                        'MarkerFaceColor','r',...
                        'MarkerSize',5)
        hold off
        xlabel('X')
        ylabel('Y')
```

```
        zlabel('  ')
        axis([-30 30 -30 30 0 40]);
        grid  on

    teta_akhir=[teta(1,indeks);0;0];
        k=1:length(t)
        teta1=trajectory(teta_awal',teta_akhir',10,t(k));
        link_parameter=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1;0]];
        point=[0;0;0];
        [keluaran,x,y,z]=data_robot(link_parameter,point);
        set(handles.p,'      ',x,'      ',y,'      ',z);
        set(handles.xOut,'      ',(0.01*round(100*(keluaran(1)))));
        set(handles.yOut,'      ',(0.01*round(100*(keluaran(2)))));
        set(handles.zOut,'      ',(0.01*round(100*(keluaran(3)))));
        M(k) = getframe;

    teta_awal=teta_akhir;
    teta_akhir=[teta(1,indeks);teta(2,indeks);0];
        k=1:length(t)
        teta1=trajectory(teta_awal',teta_akhir',10,t(k));
        link_parameter=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1;0]];
        point=[0;0;0];
        [keluaran,x,y,z]=data_robot(link_parameter,point);
        set(handles.p,'      ',x,'      ',y,'      ',z);
        set(handles.xOut,'      ',(0.01*round(100*(keluaran(1)))));
        set(handles.yOut,'      ',(0.01*round(100*(keluaran(2)))));
        set(handles.zOut,'      ',(0.01*round(100*(keluaran(3)))));
        M(k) = getframe;

    teta_awal=teta_akhir;
    teta_akhir=[teta(1,indeks);teta(2,indeks);teta(3,indeks)];
        k=1:length(t)
        teta1=trajectory(teta_awal',teta_akhir',10,t(k));
        link_parameter=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1;0]];
        point=[0;0;0];
        [keluaran,x,y,z]=data_robot(link_parameter,point);
        set(handles.p,'      ',x,'      ',y,'      ',z);
        set(handles.xOut,'      ',(0.01*round(100*(keluaran(1)))));
        set(handles.yOut,'      ',(0.01*round(100*(keluaran(2)))));
        set(handles.zOut,'      ',(0.01*round(100*(keluaran(3)))));
        M(k) = getframe;

    handles.teta_awal(:,indeks)=teta_akhir;
handles.teta_akhir=handles.teta_awal;

msgbox('      ','','      ')


 [keluaranF(1) keluaranF(2) keluaranF(3)];
 [xd yd zd];
anceF = Distance(VD,VF);
```

```
handles.deltaFUZZYbox,'String',distanceF)
ita(hObject, handles);


     yslider_CreateFcn(hObject, eventdata, handles)



sequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor',[.9 .9 .9]);


    zslider_Callback(hObject, eventdata, handles)



 get(handles.xslider,'Value');
 get(handles.yslider,'Value');
 get(hObject,'Value');
handles.xbox_input,'String',(0.01*round(100*(xd))))
handles.ybox_input,'String',(0.01*round(100*(yd))))
handles.zbox_input,'String',(0.01*round(100*(zd))))


= readfis('linkinv');
a = evalfis([xd;yd;zd],fis);
F = [theta(1),0;theta(2),0;theta(3),0];
handles.theta1Fbox,'String',(0.01*round(100*tetaF(1,1))))
handles.theta2Fbox,'String',(0.01*round(100*tetaF(2,1))))
handles.theta3Fbox,'String',(0.01*round(100*tetaF(3,1))))
les.tetaF=tetaF*pi/180;

les.teta_akhirF=[];
F=handles.tetaF;
_awal1F=handles.teta_akhirF;
ksF=1;
tF=0:1:10;
  isempty(teta_awal1F);
   teta_awal1F=zeros(3,2);
;
teta_awalF=teta_awal1F(:,indeksF);
teta_akhirF=handles.tetaF(:,indeksF);
        axes(handles.axesFuzzy)
```

```
        cla
        axis square
        handles.pF=plot3(0,0,0,'-o','LineWidth',4,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',5);
        hold on
        plot3(xd,yd,zd,'-o',...
                        'LineWidth',2,...
                        'MarkerEdgeColor','k',...
                        'MarkerFaceColor','g',...
                        'MarkerSize',5)
        hold off
        xlabel('X')
        ylabel('Y')
        zlabel('Z')
        axis([-30 30 -30 30 0 40]);
        grid on

teta_akhirF=[tetaF(1,indeksF);0;0];
    kF=1:length(tF)
    teta1F=trajectory(teta_awalF',teta_akhirF',10,tF(kF));
    link_parameterF=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1F;0]];
    pointF=[0;0;0];
    [keluaranF,xF,yF,zF]=data_robot(link_parameterF,pointF);
    set(handles.pF,'XData',xF,'YData',yF,'ZData',zF);
    set(handles.xoutFuzzybox,'String',(0.01*round(100*(keluaranF(1)))))
    set(handles.youtFuzzybox,'String',(0.01*round(100*(keluaranF(2)))))
    set(handles.zoutFuzzybox,'String',(0.01*round(100*(keluaranF(3)))))
    MF(kF) = getframe;
end

teta_awalF=teta_akhirF;
teta_akhirF=[tetaF(1,indeksF);tetaF(2,indeksF);0];
    kF=1:length(tF)
    teta1F=trajectory(teta_awalF',teta_akhirF',10,tF(kF));
    link_parameterF=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1F;0]];
    pointF=[0;0;0];
    [keluaranF,xF,yF,zF]=data_robot(link_parameterF,pointF);
    set(handles.pF,'XData',xF,'YData',yF,'ZData',zF);
    set(handles.xoutFuzzybox,'String',(0.01*round(100*(keluaranF(1)))))
    set(handles.youtFuzzybox,'String',(0.01*round(100*(keluaranF(2)))))
    set(handles.zoutFuzzybox,'String',(0.01*round(100*(keluaranF(3)))))
    MF(kF) = getframe;
end

teta_awalF=teta_akhirF;
teta_akhirF=[tetaF(1,indeksF);tetaF(2,indeksF);tetaF(3,indeksF)];
    kF=1:length(tF)
    teta1F=trajectory(teta_awalF',teta_akhirF',10,tF(kF));
    link_parameterF=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1F;0]];
    pointF=[0;0;0];
```

```
        [keluaranF,xF,yF,zF]=data_robot(link_parameterF,pointF);
        set(handles.pF,'String',xF,'String',yF,'String',zF);
        set(handles.xoutFuzzybox,'String',(0.01*round(100*(keluaranF(1)))))
        set(handles.youtFuzzybox,'String',(0.01*round(100*(keluaranF(2)))))
        set(handles.zoutFuzzybox,'String',(0.01*round(100*(keluaranF(3)))))
        MF(kF) = getframe;

    handles.teta_awalF(:,indeksF)=teta_akhirF;
    les.teta_akhirF=handles.teta_awalF;


    los.teta_akhir=[];
    [xd yd zd];
    k=[xyz(1);xyz(2);xyz(3)];
    1=atan2(titik(2,1),titik(1,1))*180/pi;
    =[sqrt(xyz(1)^2+xyz(2)^2);xyz(3)-13];
    a23,str] = invers_kinematik(titik1(1),titik1(2));
    isempty(teta23)
    teta_1=[teta1;teta23(:,1)];
    teta_2=[teta1;teta23(:,2)];
    teta={teta_1 teta_2};
    set(handles.theta1box,'String',(0.01*round(100*(teta(1,1)))));
    set(handles.theta2box,'String',(0.01*round(100*(teta(2,1)))));
    set(handles.theta3box,'String',(0.01*round(100*(teta(3,1)))));
    handles.teta=teta*pi/180;
    teta=handles.teta;
    teta_awal1=handles.teta_akhir;
    indeks=1;
    t=0:1:10;
        isempty(teta_awal1);
        teta_awal1=zeros(3,2);

    teta_awal=teta_awal1(:,indeks);
    teta_akhir=handles.teta(:,indeks);

        axes(handles.axesCCD)
        cla
        axis equal
        handles.p=plot3(0,0,0,'o',...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',5);
        hold on
        plot3(xd,yd,zd,'-o',...
                'LineWidth',2,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','r',...
                'MarkerSize',5)
        hold on
        xlabel('x')
        ylabel('y')
        zlabel('z')
```

```
ta(hObject, handles);

    zslider_CreateFcn(hObject, eventdata, handles)

sequal(get(hObject,'                    '), get(0,'                                    '))
set(hObject,'                ',[.9 .9 .9]);

    zbox_input_Callback(hObject, eventdata, handles)

StrVal = get(hObject,'      ');
d_val = str2double(zNewStrVal);

isempty(z_sld_val) | (z_sld_val < 0) | (z_sld_val > 34),
zOldVal = get(handles.zslider,'      ');
set(hObject,'      ',zOldVal)

set(handles.zslider,'      ',z_sld_val)

    zbox_input_CreateFcn(hObject, eventdata, handles)

spc && isequal(get(hObject,'                    '), get
                                                    '))
set(hObject,'                ','      ');

    ybox_input_Callback(hObject, eventdata, handles)
```

```
StrVal = get(hObject,'String');
d_val = str2double(yNewStrVal);


isempty(y_sld_val) | (y_sld_val < -20) | (y_sld_val > 20),
yOldVal = get(handles.yslider,'Value');
set(hObject,'String',yOldVal)

set(handles.yslider,'Value',y_sld_val)


      ybox_input_CreateFcn(hObject, eventdata, handles)




spc && isequal(get(hObject,'BackgroundColor'), get↙
(0.94,0.94,0.94))
set(hObject,'BackgroundColor','white');


      xbox_input_Callback(hObject, eventdata, handles)




StrVal = get(hObject,'String');
d_val = str2double(xNewStrVal);


isempty(x_sld_val) | (x_sld_val < -20) | (x_sld_val > 20),
xOldVal = get(handles.xslider,'Value');
set(hObject,'String',xOldVal)

set(handles.xslider,'Value',x_sld_val)


      xbox_input_CreateFcn(hObject, eventdata, handles)
```

```
spc && isequal(get(hObject,'[illegible]'), get↙
[illegible]'))
set(hObject,'[illegible]','[illegible]');
```

```
    theta3box_Callback(hObject, eventdata, handles)
```

```
    theta3box_CreateFcn(hObject, eventdata, handles)
```

```
spc && isequal(get(hObject,'[illegible]'), get↙
[illegible]'))
set(hObject,'[illegible]','[illegible]');
```

```
    theta2box_Callback(hObject, eventdata, handles)
```

```
    theta2box_CreateFcn(hObject, eventdata, handles)
```

```
spc && isequal(get(hObject,'[illegible]'), get↙
[illegible]'))
set(hObject,'[illegible]','[illegible]');

    thetalbox_Callback(hObject, eventdata, handles)

    thetalbox_CreateFcn(hObject, eventdata, handles)

spc && isequal(get(hObject,'[illegible]'), get↙
[illegible]'))
set(hObject,'[illegible]','[illegible]');

    theta3Fbox_Callback(hObject, eventdata, handles)

    theta3Fbox_CreateFcn(hObject, eventdata, handles)

spc && isequal(get(hObject,'[illegible]'), get↙
[illegible]'))
set(hObject,'[illegible]','[illegible]');
```

```
theta2Fbox_Callback(hObject, eventdata, handles)

theta2Fbox_CreateFcn(hObject, eventdata, handles)

spc && isequal(get(hObject,'[illegible]'), get↙
[illegible]'))
set(hObject,'[illegible]','[illegible]');

theta1Fbox_Callback(hObject, eventdata, handles)

theta1Fbox_CreateFcn(hObject, eventdata, handles)

spc && isequal(get(hObject,'[illegible]'), get↙
[illegible]'))
set(hObject,'[illegible]','[illegible]');

deltaFUZZYbox_Callback(hObject, eventdata, handles)
```

```
deltaFUZZYbox_CreateFcn(hObject, eventdata, handles)

spc && isequal(get(hObject,'[illegible]'), get↙
[illegible]'))
set(hObject,'[illegible]','[illegible]');

FuzzyMenu_Callback(hObject, eventdata, handles)

BlockMenu_Callback(hObject, eventdata, handles)

= readfis('[illegible]');
re(5)
fis(fis)

RuleMenu_Callback(hObject, eventdata, handles)

view('[illegible]')

SurfMenu_Callback(hObject, eventdata, handles)
```

```
iew('[illegible]')


    MFXMenu_Callback(hObject, eventdata, handles)


= readfis('[illegible]');
re(6)
mf(fis,'[illegible]',1)


    MFYMenu_Callback(hObject, eventdata, handles)


= readfis('[illegible]');
re(7)
mf(fis,'[illegible]',2)


    MFZMenu_Callback(hObject, eventdata, handles)


= readfis('[illegible]');
re(8)
mf(fis,'[illegible]',3)


    evalbutton_Callback(hObject, eventdata, handles)


StrVal = get(handles.xbox_input,'[illegible]');
d_val = str2double(xNewStrVal);

isempty(x_sld_val) | (x_sld_val < -20) | (x_sld_val > 20),
xOldVal = get(handles.xslider,'[illegible]');
set(handles.xbox_input,'[illegible]',xOldVal)
xd = xOldVal;

set(handles.xslider,'[illegible]',x_sld_val)
xd = x_sld_val;

StrVal = get(handles.ybox_input,'[illegible]');
d_val = str2double(yNewStrVal);
```

```
 sempty(y_sld_val) | (y_sld_val < -20) | (y_sld_val > 20),
'OldVal = get(handles.yslider,'Value');
set(handles.ybox_input,'String',yOldVal)
yd = yOldVal;

set(handles.yslider,'Value',y_sld_val)
yd = y_sld_val;

StrVal = get(handles.zbox_input,'String');
d_val = str2double(zNewStrVal);

isempty(z_sld_val) | (z_sld_val < 0) | (z_sld_val > 30),
zOldVal = get(handles.zslider,'Value');
set(handles.zbox_input,'String',zOldVal)
zd = zOldVal;

set(handles.zslider,'Value',z_sld_val)
zd = z_sld_val;

handles.xbox_input,'String',(0.01*round(100*(xd))))
handles.ybox_input,'String',(0.01*round(100*(yd))))
handles.zbox_input,'String',(0.01*round(100*(zd))))


= readfis('InvFin');
a = evalfis([xd;yd;zd],fis);
F = [theta(1),0;theta(2),0;theta(3),0];
handles.theta1Fbox,'String',(0.01*round(100*tetaF(1,1))))
handles.theta2Fbox,'String',(0.01*round(100*tetaF(2,1))))
handles.theta3Fbox,'String',(0.01*round(100*tetaF(3,1))))
les.tetaF=tetaF*pi/180;

les.teta_akhirF=[];
F=handles.tetaF;
_awal1F=handles.teta_akhirF;
ksF=1;
tF=0:1:10;
 isempty(teta_awal1F);
    teta_awal1F=zeros(3,2);

teta_awalF=teta_awal1F(:,indeksF);
teta_akhirF=handles.tetaF(:,indeksF);
        axes(handles.axesFuzzy)
        cla
        axis square
        handles.pF=plot3(0,0,0,'-o','LineWidth',4,...
                'MarkerEdgeColor','b',...
                'MarkerFaceColor','r',...
                'MarkerSize',5);
        hold on
        plot3(xd,yd,zd,'-ro',...
                        'LineWidth',2,...
```

```
                                                    ',' ',...
                                                    ',' ',...
                                                    ',5)
        hold on
        xlabel('X')
        ylabel('Y')
        zlabel('Z')
        axis([-30 30 -30 30 0 40]);
        grid on


teta_akhirF=[tetaF(1,indeksF);0;0];
    kF=1:length(tF)
    teta1F=trajectory(teta_awalF',teta_akhirF',10,tF(kF));
    link_parameterF=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1F;0]];
    pointF=[0;0;0];
    [keluaranF,xF,yF,zF]=data_robot(link_parameterF,pointF);
    set(handles.pF,'String',xF,'String',yF,'String',zF);
    set(handles.xoutFuzzybox,'String',(0.01*round(100*(keluaranF(1)))))
    set(handles.youtFuzzybox,'String',(0.01*round(100*(keluaranF(2)))))
    set(handles.zoutFuzzybox,'String',(0.01*round(100*(keluaranF(3)))))
    MF(kF) = getframe;



teta_awalF=teta_akhirF;
teta_akhirF=[tetaF(1,indeksF);tetaF(2,indeksF);0];
    kF=1:length(tF)
    teta1F=trajectory(teta_awalF',teta_akhirF',10,tF(kF));
    link_parameterF=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1F;0]];
    pointF=[0;0;0];
    [keluaranF,xF,yF,zF]=data_robot(link_parameterF,pointF);
    set(handles.pF,'String',xF,'String',yF,'String',zF);
    set(handles.xoutFuzzybox,'String',(0.01*round(100*(keluaranF(1)))))
    set(handles.youtFuzzybox,'String',(0.01*round(100*(keluaranF(2)))))
    set(handles.zoutFuzzybox,'String',(0.01*round(100*(keluaranF(3)))))
    MF(kF) = getframe;


teta_awalF=teta_akhirF;
teta_akhirF=[tetaF(1,indeksF);tetaF(2,indeksF);tetaF(3,indeksF)];
    kF=1:length(tF)
    teta1F=trajectory(teta_awalF',teta_akhirF',10,tF(kF));
    link_parameterF=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1F;0]];
    pointF=[0;0;0];
    [keluaranF,xF,yF,zF]=data_robot(link_parameterF,pointF);
    set(handles.pF,'String',xF,'String',yF,'String',zF);
    set(handles.xoutFuzzybox,'String',(0.01*round(100*(keluaranF(1)))))
    set(handles.youtFuzzybox,'String',(0.01*round(100*(keluaranF(2)))))
    set(handles.zoutFuzzybox,'String',(0.01*round(100*(keluaranF(3)))))
    MF(kF) = getframe;
```

```
andles.teta_awalF(:,indeksF)=teta_akhirF;
es.teta_akhirF=handles.teta_awalF;



les.teta_akhir=[];
[xd yd zd];
k=[xyz(1);xyz(2);xyz(3)];
l=atan2(titik(2,1),titik(1,1))*180/pi;
k1=[sqrt(xyz(1)^2+xyz(2)^2);xyz(3)-13];
a23,str] = invers_kinematik(titik1(1),titik1(2));
isempty(teta23)
teta_1=[teta1;teta23(:,1)];
teta_2=[teta1;teta23(:,2)];
teta=[teta_1 teta_2];
set(handles.theta1box,'          ',(0.01*round(100*(teta(1,1)))));
set(handles.theta2box,'          ',(0.01*round(100*(teta(2,1)))));
set(handles.theta3box,'          ',(0.01*round(100*(teta(3,1)))));
handles.teta=teta*pi/180;
teta=handles.teta;
teta_awal1=handles.teta_akhir;
indeks=1;
    t=0:1:10;
       isempty(teta_awal1);
        teta_awal1=zeros(3,2);

    teta_awal=teta_awal1(:,indeks);
    teta_akhir=handles.teta(:,indeks);

        axes(handles.axesCCD)
        cla
        axis
        handles.p=plot3(0,0,0,'
                ',4,...
                '
                ',...
                '
                ',...
                '
                ',5);
        hold
        plot3(xd,yd,zd,'
                ',2,...
hanc                    '
                ',...
                '
                ',...
msgb                    '
                ',5)
        hold
        xlabel('
        ylabel('
        zlabel('
        axis([-30 30 -30 30 0 40]);
 [ke    grid
 [xd
ance
```

```
    teta_akhir=[teta(1,indeks);0;0];
        k=1:length(t)
        teta1=trajectory(teta_awal',teta_akhir',10,t(k));
        link_parameter=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1;0]];
        point=[0;0;0];
        [keluaran,x,y,z]=data_robot(link_parameter,point);
        set(handles.p,'[illegible]',x,'[illegible]',y,'[illegible]',z);
        set(handles.xOut,'[illegible]',(0.01*round(100*(keluaran(1)))));
        set(handles.yOut,'[illegible]',(0.01*round(100*(keluaran(2)))));
        set(handles.zOut,'[illegible]',(0.01*round(100*(keluaran(3)))));
        M(k) = getframe;

    teta_awal=teta_akhir;
    teta_akhir=[teta(1,indeks);teta(2,indeks);0];
        k=1:length(t)
        teta1=trajectory(teta_awal',teta_akhir',10,t(k));
        link_parameter=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1;0]];
        point=[0;0;0];
        [keluaran,x,y,z]=data_robot(link_parameter,point);
        set(handles.p,'[illegible]',x,'[illegible]',y,'[illegible]',z);
        set(handles.xOut,'[illegible]',(0.01*round(100*(keluaran(1)))));
        set(handles.yOut,'[illegible]',(0.01*round(100*(keluaran(2)))));
        set(handles.zOut,'[illegible]',(0.01*round(100*(keluaran(3)))));
        M(k) = getframe;
    end
    teta_awal=teta_akhir;
    teta_akhir=[teta(1,indeks);teta(2,indeks);teta(3,indeks)];
        k=1:length(t)
        teta1=trajectory(teta_awal',teta_akhir',10,t(k));
        link_parameter=[[0;0;pi/2;0;0],[0;0;0;10;10],[0;13;0;0;0],[0;teta1;0]];
        point=[0;0;0];
        [keluaran,x,y,z]=data_robot(link_parameter,point);
        set(handles.p,'[illegible]',x,'[illegible]',y,'[illegible]',z);
        set(handles.xOut,'[illegible]',(0.01*round(100*(keluaran(1)))));
        set(handles.yOut,'[illegible]',(0.01*round(100*(keluaran(2)))));
        set(handles.zOut,'[illegible]',(0.01*round(100*(keluaran(3)))));
        M(k) = getframe;
    end
    handles.teta_awal(:,indeks)=teta_akhir;
handles.teta_akhir=handles.teta_awal;

msgbox('[illegible]','','[illegible]')



 [keluaranF(1) keluaranF(2) keluaranF(3)];
 [xd yd zd];
anceF = Distance(VD,VF);
```

```
handles.deltaFUZZYbox,'String',distanceF)
data(hObject, handles);


zOut_Callback(hObject, eventdata, handles)


zOut_CreateFcn(hObject, eventdata, handles)


spc && isequal(get(hObject,'BackgroundColor'), get↙
(0.9400,0.9400,0.9400))
set(hObject,'BackgroundColor','white');


xOut_Callback(hObject, eventdata, handles)


xOut_CreateFcn(hObject, eventdata, handles)


spc && isequal(get(hObject,'BackgroundColor'), get↙
(0.9400,0.9400,0.9400))
set(hObject,'BackgroundColor','white');
```

```
    yOut_Callback(hObject, eventdata, handles)


    yOut_CreateFcn(hObject, eventdata, handles)


spc && isequal(get(hObject,'Background
set(hObject,'Background','white');
```

```
base_T_wrist = kinematiks_N (link_parameter)

T_wrist = eye(4,4);
= size (link_parameter);
= 1 : N
base_T_wrist = base_T_wrist*kinematik (link_parameter(k,:));
```

```
    T = kinematik (parameter)

parameter(1);
ameter(2);
ameter(3);
parameter(4);
[cos(teta),          -sin(teta),           0,          a;
sin(teta)*cos(alfa), cos(teta)*cos(alfa), -sin(alfa), -sin(alfa)*d;
sin(teta)*sin(alfa), cos(teta)*sin(alfa), cos(alfa),  cos(alfa)*d;
0,                    0,                   0,          1 ];
```

```
% [teta,str] = invers_kinematik(x,y)

[10,10];
s_point=sqrt(x^2+y^2);
eta2=((x^2+y^2-link(1)^2-link(2)^2)/(2*link(1)*link(2)));

s_teta2>=-1 & cos_teta2<=1
eta= atan2(y,x);
os_w=(x^2+y^2+link(1)^2-link(2)^2)/(2*link(1)*radius_point);
eta1=[beta+acos(cos_w),beta-acos(cos_w)];
eta2=[-acos(cos_teta2),acos(cos_teta2)];
eta=[teta1;teta2]*180/pi;
str=[];

str=('Titik di luar jangkauan');
eta=[];
```

```
    end_point=gambar_robot(link_parameter,point)
 = size (link_parameter);
 = 1 : N
=kinematiks_N(link_parameter(1:k,:));
oint(:,k) = T*[0;0;0;1];

_point=robot(link_parameter,point);
oint=[tool_point(1,1),tool_point(2,1)];
oint(1,:),tool_point(1)];
oint(2,:),tool_point(2)];

ot(x,y,'-r','LineWidth',2,...
            'MarkerEdgeColor','k',...
            'MarkerFaceColor','r',...
            'MarkerSize',3)
```

```
% length = Distance(A,B)
A(:,1);
A(:,2);
A(:,3);
B(:,1);
B(:,2);
B(:,3);
h = 0.01*round(100*(sqrt(((x1-x2)^2)+((y1-y2)^2)+((z1-z2)^2))));
```

```
    [end_point,x,y,z]=data_robot(link_parameter,point)

 = size (link_parameter);
 = 1 : N
=kinematiks_N(link_parameter(1:k,:));
oint(:,k) = T*[0;0;0;1];

_point=robot(link_parameter,point);
oint=[tool_point(1,1),tool_point(2,1),tool_point(3,1)];
int(1,:),tool_point(1)];
int(2,:),tool_point(2)];
int(3,:),tool_point(3)];
```

```
    post = CartePosition(l,A,B)

l(:,1);
l(:,2);
l(:,3);
A(:,1);
A(:,2);
A(:,3);
B(:,1);
B(:,2);
B(:,3);


d1=pi/180*A1;
d2=pi/180*A2;
d3=pi/180*A3;









;

cos(s_rad1)*(L2*cos(s_rad2)));
sin(s_rad1)*(L2*cos(s_rad2)));
l1+(L2*sin(s_rad2)));

cos(s_rad1)*(L2*cos(s_rad2)+L3*cos(s_rad2+s_rad3)));
sin(s_rad1)*(L2*cos(s_rad2)+L3*cos(s_rad2+s_rad3)));
l1+(L2*sin(s_rad2))+(L3*sin(s_rad2+s_rad3)));

= [0 0 0
   x1 y1 z1
   x2 y2 z2
   x3 y3 z3];
```

```
    tool_point=robot(link_parameter,point)

=kinematiks_N(link_parameter);
int;1];
point=B_T_W*P;
```

```
function teta=trajectory(teta_awal,teta_akhir,tf,t)

a=[teta_awal',zeros(length(teta_awal),1),3*(teta_akhir'-teta_awal')...
/tf^2,-2*(teta_akhir'-teta_awal')/tf^3];
time=[1;t;t^2;t^3];
teta=a*time;
```