



**SOLUSI *TRAVELLING SALESMAN PROBLEM* MENGGUNAKAN
ALGORITMA FUZZY EVOLUSI
(Studi Kasus PT. Jalur Nugraha Ekakurir (JNE) Semarang)**

skripsi

disajikan sebagai salah satu syarat
untuk memperoleh gelar Sarjana Sains
Program Studi Matematika

Oleh

Dinar Anggit Wicaksana

4111409006

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI SEMARANG
2013**

PERNYATAAN KEASLIAN TULISAN

Dengan ini saya menyatakan bahwa isi skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya tidak terdapat karya yang diterbitkan oleh orang lain, kecuali yang secara tertulis dirujuk dalam skripsi ini dan disebutkan dalam daftar pustaka.

Semarang, Agustus 2013

Dinar Anggit Wicaksana
NIM 4111409006

PENGESAHAN

Skripsi yang berjudul

Solusi *Travelling Salesman Problem* Menggunakan Algoritma Fuzzy
Evolusi (Studi Kasus PT. Jalur Nugraha Ekakurir (JNE) Semarang)

disusun oleh

Dinar Anggit Wicaksana
4111409006

telah dipertahankan di hadapan sidang Panitia Ujian Skripsi FMIPA UNNES pada
tanggal 23 Agustus 2013.

Panitia:

Ketua

Sekretaris

Prof. Dr. Wiyanto, M.Si
NIP. 196310121988031001

Drs. Arief Agoestanto, M.Si
NIP.196807221993031005

Ketua Penguji

Riza Arifudin, S.Pd., M.Cs
NIP. 198005252005011001

Anggota Penguji/
Pembimbing Utama

Anggota Penguji/
Pembimbing Pendamping

Alamsyah, S.Si., M.Kom
NIP. 197405172006041001

Zaenal Abidin, S.Si., M.Cs
NIP. 198205042005011001

MOTTO DAN PERSEMBAHAN

Motto:

- ❖ Kesalahan terbesar yang mungkin diperbuat seseorang adalah tidak berbuat apa-apa (John C. Maxwell)
- ❖ Sesuatu yang belum dikerjakan, seringkali tampak mustahil; kita baru yakin kalau kita melakukannya dengan baik. (Evelyn Underhill)
- ❖ Ketika kebahagiaan menjadi harga mutlak untuk masa depan kita, hanya kerja keras yang disertai doalah kata kuncinya

Persembahan:

- Tuhan Yang Maha Esa
- Bapak, Ibu, Adik, beserta keluarga tercinta yang tak henti-hentinya memberikan doa, semangat, dan dukungan.
- Meilia Mira Lestanti dan keluarganya
- Teman-teman seperjuangan matematika '09.

KATA PENGANTAR

Segala puji hanya bagi Tuhan Yang Maha Esa yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Solusi *Travelling Salesman Problem* Menggunakan Algoritma Fuzzy Evolusi (Studi Kasus PT. Jalur Nugraha Ekakurir (JNE) Semarang)”.

Dalam penyusunan skripsi ini, penulis memperoleh bantuan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada :

1. Prof. M. Fathur Rohman, M.Hum, Rektor Universitas Negeri Semarang.
2. Prof. Dr. Wiyanto, M.Si, Dekan Fakultas Matematika dan Ilmu pengetahuan Alam Universitas Negeri Semarang.
3. Drs. Arief Agoestanto, M.Si, Ketua Jurusan Matematika Fakultas Matematika dan Ilmu pengetahuan Alam Universitas Negeri Semarang.
4. Alamsyah, S.Si., M.Kom selaku Dosen pembimbing I yang telah memberikan bimbingan, arahan, dan saran kepada penulis selama penyusunan skripsi.
5. Zaenal Abidin, S.Si., M.Cs, selaku Dosen pembimbing II yang telah memberikan bimbingan, arahan, dan saran kepada penulis selama penyusunan skripsi.
6. Dosen Penguji Utama yang telah memberikan inspirasi, kritik, saran dan motivasi kepada penulis, sehingga penulis dapat menyelesaikan skripsi
7. Pak Haryanto yang telah menjadi pembimbing lapangan di PT. Jalur Nugraha Ekakurir sehingga penulis dapat mengerti proses pengiriman barang untuk sampai kepada *supplier*.

8. Bapak dan Ibu dosen yang telah memberikan bekal ilmu yang tak ternilai harganya selama belajar di Fakultas Matematika dan Ilmu pengetahuan Alam Universitas Negeri Semarang.
9. Bapak, ibu dan adikku tercinta yang senantiasa mendoakan serta memberikan dukungan baik secara moral maupun spiritual.
10. Dek Meilia Mira Lestanti yang selama ini memberikan dukungan, semangat serta inspirasi untuk penulis.

Akhirnya penulis berharap semoga skripsi ini dapat bermanfaat bagi penulis pada khususnya dan pembaca pada umumnya.

Semarang, Agustus 2013

Dinar Anggit Wicaksana
NIM 4111409006

ABSTRAK

Wicaksana, Dinar Anggit. 2013. Solusi *Traveling Salesman Problem* Menggunakan Algoritma Fuzzy Evolusi (Studi Kasus PT. Jalur Nugraha Ekakurir (JNE) Semarang). Skripsi, Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang. Pembimbing Utama: Alamsyah S.Si., M.Kom. dan Pembimbing Pendamping: Zaenal Abidin, S.Si., M.Cs.

Kata Kunci: Algoritma Fuzzy Evolusi, *Traveling Salesman Problem*.

Traveling Salesman Problem (TSP) adalah problem mencari rute optimal bagi seorang salesman yang berkeliling mengunjungi n kota dengan setiap kota dikunjungi satu kali kecuali kota asal. Skripsi ini akan meneliti salah satu kasus TSP pada masalah pengiriman barang di PT. Jalur Nugraha Ekakurir (JNE) Semarang dengan tujuan 28 alamat penerima di wilayah Kota Semarang. Pada penelitian ini, digunakan algoritma fuzzy evolusi dalam MATLAB untuk menyelesaikan masalah TSP.

Permasalahan pada skripsi ini adalah bagaimana rute jaringan TSP yang mempunyai jarak minimum dalam pengiriman barang dengan menggunakan algoritma Fuzzy Evolusi di PT. Jalur Nugraha Ekakurir (JNE) Semarang, bagaimana hasil pencarian jarak minimum dari jaringan TSP dalam pengiriman barang di PT. Jalur Nugraha Ekakurir (JNE) Semarang menggunakan algoritma Fuzzy Evolusi.

Pengambilan data dilakukan dengan cara dokumentasi data dari PT. Jalur Nugraha Ekakurir (JNE) Semarang. Data yang diambil berupa list alamat rumah penerima barang di wilayah Kota Semarang, selanjutnya dilakukan pencarian koordinat masing-masing lokasi dengan bantuan situs GetLatlon.Yohman.com sehingga koordinat lokasi dapat diketahui. Analisis data dilakukan dengan menggunakan mekanisme algoritma fuzzy Evolusi yang diaplikasikan dalam program MATLAB. Penentuan probabilitas *crossover* (pc), probabilitas mutasi (pm), jumlah kromosom dalam 1 generasi, dan maksimum generasi memberikan pengaruh yang signifikan terhadap solusi optimal yang bisa didapatkan.

Dari hasil analisis dengan algoritma fuzzy evolusi diperoleh solusi optimal algoritma fuzzy evolusi menggunakan masukkan populasi 100 dan generasi 1000 lebih baik dari solusi optimal yang didapatkan dengan masukkan populasi dan generasinya secara berturut-turut adalah (100 dan 100), (100 dan 200), (100 dan 500), (200 dan 100), (500 dan 100) dan (1000 dan 100). Kemudian didapatkan rute terbaiknya 1 – 8 – 13 – 19 – 15 – 20 – 5 – 7 – 28 – 21 – 17 – 10 – 25 – 22 – 26 – 14 – 24 – 11 – 6 – 18 – 27 – 3 – 16 – 23 – 9 – 2 – 4 – 12 – 1 dan panjang jalur terbaiknya adalah 35,18 Km dengan memasukkan populasi 100 dan generasi 1000.

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
PERNYATAAN KEASLIAN TULISAN	ii
PENGESAHAN	iii
MOTTO DAN PERSEMBAHAN	iv
KATA PENGANTAR	v
ABSTRAK	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
DAFTAR LAMPIRAN	xv
BAB	
1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Pembatasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
1.6 Sistematika Penulisan	5
2. LANDASAN TEORI	8
2.1 Teori Graf	8

2.1.1 Pengertian Graf	8
2.1.2 Graf Tak Berarah.....	9
2.1.3 Graf Berarah	10
2.1.4 Graf Berbobot	11
2.2 <i>Traveling Salesman Problem</i> (TSP).....	11
2.3 Algoritma Genetika.....	14
2.4 <i>Fuzzy</i>	29
2.4.1 Logika Fuzzy.....	29
2.4.2 Himpunan Fuzzy.....	31
2.4.3 Fungsi Keanggotaan Fuzzy.....	34
2.4.4 Fungsi Implikasi.....	38
2.4.5 Sistem Inferensi Fuzzy.....	40
2.4.5.1 Metode Mamdani.....	40
2.5 Algoritma Fuzzy Evolusi	41
2.6 Algoritma Fuzzy Evolusi Xu	42
2.7 Arsitektur Algoritma Fuzzy Evolusi	49
2.8 Matlab	50
2.8.1 Beberapa Bagian dari Window Matlab	50
2.8.2 Meminta Bantuan	51
2.8.3 Interrupting dan Terminating dalam Matlab	51
2.8.4 Variabel pada Matlab	51
3. METODE PENELITIAN	53
3.1 Menemukan Masalah.....	53

3.2 Merumuskan Masalah	53
3.3 Pengambilan Data	54
3.4 Analisis dan Pemecahan Masalah	55
3.5 Penarikan Simpulan	58
4 HASIL PENELITIAN DAN PEMBAHASAN	59
4.1 Hasil Penelitian	59
4.1.1 Implementasi Program	60
4.1.2 Hasil Simulasi Program.....	62
4.1.2.1 Perhitungan menggunakan masukan populasi 100 dan generasi 100	62
4.1.2.2 Perhitungan menggunakan masukan populasi 100 dan generasi 200	67
4.1.2.3 Perhitungan menggunakan masukan populasi 100 dan generasi 500	69
4.1.2.4 Perhitungan menggunakan masukan populasi 100 dan generasi 1000	71
4.1.2.5 Perhitungan menggunakan masukan populasi 200 dan generasi 100	73
4.1.2.6 Perhitungan menggunakan masukan populasi 500 dan generasi 100	75
4.1.2.7 Perhitungan menggunakan masukan populasi 1000 dan generasi 100	77

4.1.3 Analisis Penyelesaian Jaringan Traveling Salesman Problem Menggunakan Aplikasi Algoritma Fuzzy Evolusi dalam Pengiriman Barang di P.T. Jalur Nugraha Ekakurir Semarang..	79
4.2 Pembahasan	82
5 PENUTUP	85
5.1 Simpulan	85
5.2 Saran	86
DAFTAR PUSTAKA	87
LAMPIRAN	89

DAFTAR GAMBAR

Gambar	Halaman
2.1 Graf dengan Lima Titik dan Tujuh Sisi	9
2.2 Graf Lengkap dengan 4 Simpul, Tiap Sisi Diberi Bobot.....	12
2.3 Tiga Buah Sirkuit Hamilton	13
2.4 Diagram Alir Algoritma Genetika	16
2.5 Ilustrasi Seleksi dengan Mesin Roulette	21
2.6 Contoh Gen Sebelum dan Setelah Mutasi dengan Pengkodean Pohon...	28
2.7 Contoh Pemetaan Input - Output.....	30
2.8 Himpunan Fuzzy untuk Variabel Umur	32
2.9 Himpunan Fuzzy pada Variabel Temperatur.....	33
2.10 Representasi Linier Naik.....	35
2.11 Representasi Turun	35
2.12 Representasi Kurva Segitiga.....	36
2.13 Representasi Kurva Bahu	37
2.14 Representasi Kurva S	38
2.15 Fungsi Implikasi: MIN	39
2.16 Fungsi Implikasi: DOT	40
2.17 Proses Defuzzifikasi	41
2.18 Semesta Pembicaraan dan Domain untuk Variabel Populasi	46
2.19 Semesta Pembicaraan dan Domain untuk Variabel Generasi.....	46
2.20 Semesta Pembicaraan dan Domain untuk Variabel Probabilitas	

Crossover	47
2.21 Semesta Pembicaraan dan Domain untuk Variabel Probabilitas Mutasi..	48
2.22 Alur Proses Sistem Fuzzy Mamdani	49
2.23 Arsitektur Algoritma Fuzzy Evolusi	49
3.1 Tampilan Getlatlon	54
3.2 Hasil Pencarian Tempat	55
3.3 Flow Chart Rancangan Sistem	56
4.1.Tampilan TSP	61
4.2.Tampilan Koordinat Kota atau Alamat Dituju.....	62
4.3.Hasil Pencarian Probabilitas Mutasi dan Probabilitas <i>Crossover</i>	63
4.4. Grafik Fungsi AND	64
4.5. Tampilan TSP Setelah Dijalankan.....	65
4.6. Tampilan Grafik Koordinat Alamat Tujuan.....	65
4.7. Proses Perhitungan dengan Panjang Jalur Terbaik 35,18.....	81

DAFTAR TABEL

Tabel	Halaman
2.1 Aturan untuk Nilai Probabilitas Rekombinasi (<i>Crossover</i>).....	43
2.2 Aturan untuk Nilai Probabilitas Mutasi.....	44
4.1. Hasil Perhitungan Menggunakan Populasi 100 dan Generasi 100.....	66
4.2. Hasil Perhitungan Menggunakan Populasi 100 dan Generasi 200.....	68
4.3. Hasil Perhitungan Menggunakan Populasi 100 dan Generasi 500.....	70
4.4. Hasil Perhitungan Menggunakan Populasi 100 dan Generasi 1000.....	72
4.5. Hasil Perhitungan Menggunakan Populasi 200 dan Generasi 100.....	74
4.6. Hasil Perhitungan Menggunakan Populasi 500 dan Generasi 100.....	76
4.7. Hasil Perhitungan Menggunakan Populasi 1000 dan Generasi 500.....	78
4.8. Tabel Hasil Panjang Jalur Terbaik.....	80
4.9. Tabel Hasil Probabilitas Mutasi dan Probabilitas Crossover.....	81

DAFTAR LAMPIRAN

Lampiran	Halaman
1. Nama, Alamat dan Kode Lokasi Penerima Barang dari PT. Jalur Nugraha Ekakurir.....	89
2. Kode Lokasi, Koordinat X, Koordinat Y pada Alamat Penerima Barang dari PT. Jalur Nugraha Ekakurir.....	90
3. Jarak Antartitik Koordinat	92
4. Tampilan Simulasi Matlab	95
5. Coding Pada Matlab	97
6. Semesta Pembicaraan, Domain, Fungsi Keanggotaan dan Aturan Fuzzy...	114

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Proses pendistribusian barang adalah kegiatan yang tidak pernah lepas dari kehidupan. Jarak yang jauh serta penyebaran masyarakat yang meluas menjadi salah satu alasan bagi masyarakat untuk menggunakan jasa pengiriman barang daripada mengantar sendiri barang yang akan dikirimkan. Permasalahan pendistribusian barang menjadi poin penting bagi perusahaan penyedia jasa pengiriman barang. Hal ini sangat memerlukan pertimbangan dan perhitungan yang tepat karena berkaitan dengan biaya transportasi yang harus dikeluarkan dalam proses pendistribusian.

PT. Jalur Nugraha Ekakurir (JNE) merupakan salah satu perusahaan yang bergerak dalam bidang pengiriman barang di Indonesia. PT. Jalur Nugraha Ekakurir (JNE) sendiri memiliki cabang di setiap kota di seluruh Indonesia. Dalam mengirimkan barang dari pusat ke pelanggan di berbagai tempat dan di banyak kota, perlu adanya suatu sistem yang mampu meminimalisasi biaya pengiriman sehingga akan didapatkan keuntungan yang paling maksimal. Permasalahan seperti ini merupakan masalah model jaringan yang sama dengan permasalahan pada pedagang kaki lima atau biasa disebut *Travelling Salesman Problem* (TSP).

TSP merupakan salah satu masalah optimalisasi. TSP adalah suatu permasalahan untuk menemukan siklus Hamilton yang memiliki total bobot sisi minimum. TSP bertujuan mencari rute dari kota asal ke kota-kota yang dituju dengan syarat setiap kota hanya dapat dikunjungi satu kali kecuali kota awal. Banyak algoritma yang diterapkan pada permasalahan TSP diantaranya adalah *nearest neighbor heuristic*, *cheapest insertion heuristic*, *two way exchange improvement heuristic*, *nearest insertion heuristic*, *genetic*, *ant colony optimization*, dan *branch and bound method*.

Terdapat algoritma lain yang dapat digambarkan sebagai metode untuk menemukan solusi dari suatu permasalahan TSP, yaitu algoritma Fuzzy Evolusi. Algoritma Fuzzy Evolusi merupakan perpaduan antara algoritma genetika (*evolutionary algorithm*) dengan *system fuzzy*. Tahapan-tahapan yang ada dalam algoritma fuzzy evolusi adalah sama dengan tahapan yang ada dalam algoritma genetika namun untuk parameter-parameter genetika seperti probabilitas *crossover* dan probabilitas mutasinya dihasilkan melalui sistem fuzzy. Algoritma ini didasarkan pada proses genetik yang ada dalam makhluk hidup, yaitu perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti prinsip seleksi alam atau “siapa yang kuat, dia yang bertahan (*survive*)”. Dengan meniru teori evolusi ini, algoritma Fuzzy Evolusi dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam dunia nyata seperti permasalahan *task assignment* pada sistem terdistribusi, penjadwalan, *time tabling*, transportasi, dan *knapsack* (Entin, 2006).

Selain dengan menggunakan penyelesaian secara manual, seiring dengan keberhasilan pengembangan dan penerapan teknologi, masalah-masalah optimalisasi dapat diselesaikan secara lebih cepat dan mudah. Perhitungan yang begitu rumit jika diselesaikan secara manual, sekarang dapat diselesaikan dalam waktu yang relatif lebih singkat. Terdapat beberapa *software* komputer yang memungkinkan permasalahan-permasalahan optimalisasi terselesaikan secara lebih mudah dan lebih cepat. Beberapa program tersebut antara lain dengan menggunakan *software* Lindo, Matlab, QM for Windows, SkyNet, Solver, dan WinQSB.

Dari latar belakang yang telah disebutkan di atas, dalam skripsi ini penulis ingin mencoba menyelesaikan permasalahan jaringan TSP yang terdapat pada suatu perusahaan. Dengan dipilihnya penyelesaian jaringan TSP melalui algoritma Fuzzy Evolusi, diharapkan akan diperoleh solusi permasalahan jaringan TSP paling optimal sehingga dapat memaksimalkan keuntungan perusahaan melalui jarak yang paling minimal.

1.2 Rumusan Permasalahan

Permasalahan yang akan dikaji dalam penelitian ini adalah sebagai berikut.

1. Bagaimana rute jaringan TSP yang mempunyai jarak minimum dalam pengiriman barang dengan menggunakan algoritma Fuzzy Evolusi di PT. Jalur Nugraha Ekakurir (JNE) Semarang?

2. Bagaimana hasil pencarian jarak minimum dari jaringan TSP dalam pengiriman barang di PT. Jalur Nugraha Ekakurir (JNE) Semarang menggunakan algoritma Fuzzy Evolusi?

1.3 Pembatasan Masalah

Dalam penyusunan skripsi ini, penulis membahas tentang penentuan jarak minimum dan rute optimal dari jaringan TSP pada pengiriman barang di PT. Jalur Nugraha Ekakurir (JNE) menggunakan algoritma Fuzzy Evolusi.

1. Data penelitian diambil dari PT. Jalur Nugraha Ekakurir (JNE) Semarang yang terletak di Jalan Sultan Agung No. 102 Candi Baru Semarang, berupa *list* alamat para *supplier* yang akan menerima barang kiriman.
2. Permasalahan diasumsikan sebagai sebuah TSP simetris, dimana jarak dari kota 1 ke kota 2 sama dengan jarak dari kota 2 ke kota 1.
3. Perhitungan jarak antara lokasi dalam penelitian dilakukan dengan bantuan Getlatlon, yaitu Getlatlon.Yohman.Com dan observasi.

1.4 Tujuan Penelitian

Adapun tujuan yang diharapkan pada penelitian ini adalah sebagai berikut.

1. Menentukan rute optimal jaringan TSP yang mempunyai jarak minimum dalam pengiriman barang dengan menggunakan algoritma Fuzzy Evolusi di PT. Jalur Nugraha Ekakurir (JNE) Semarang.

2. Menentukan hasil pencarian jarak minimum dari jaringan TSP dalam pengiriman barang di PT. Jalur Nugraha Ekakurir (JNE) Semarang menggunakan algoritma Fuzzy Evolusi.

1.5 Manfaat Penelitian

Adapun manfaat yang diharapkan pada penelitian ini adalah sebagai berikut.

1. Dapat menambah wawasan dan pengetahuan tentang pendistribusian dan transportasi suatu jaringan.
2. Dapat menerapkan algoritma Fuzzy Evolusi dalam menyelesaikan TSP.
3. Dapat menerapkan *software* Matlab dalam penyelesaian jaringan distribusi barang.

1.6 Sistematika Penulisan

Sistematika berguna untuk memudahkan dalam memahami jalan pemikiran skripsi secara keseluruhan. Penulisan skripsi ini secara garis besar dibagi menjadi 3 bagian yaitu sebagai berikut.

1. Bagian awal skripsi

Bagian ini berisikan halaman judul, abstrak, halaman pengesahan, halaman motto dan persembahan, kata pengantar, daftar isi, daftar tabel, daftar gambar dan daftar lampiran.

2. Bagian isi skripsi

Bagian ini berisikan 5 bab, yaitu:

BAB I. PENDAHULUAN

Berisi gambaran secara global tentang isi skripsi yaitu latar belakang masalah, rumusan permasalahan, pembatasan masalah, tujuan penelitian dan manfaat penelitian serta sistematika penulisan.

BAB II. LANDASAN TEORI

Landasan teori akan menguraikan tentang definisi maupun pemikiran-pemikiran yang dijadikan kerangka teoritis dalam mendasari pemecahan dari permasalahan pada penelitian ini yaitu masalah rute optimal dan jarak minimal dari jaringan TSP dalam pengiriman barang di PT. Jalur Nugraha Ekakurir (JNE) Semarang yang akan diselesaikan dengan algoritma Fuzzy Evolusi dan *software* Matlab. Bagian ini dibagi menjadi beberapa sub bab yaitu Teori Graf, *Travelling Salesman Problem* (TSP), Algoritma Genetika, Logika Fuzzy, Algoritma Fuzzy Evolusi, Algoritma Fuzzy Evolusi Xu, Arsitektur Algoritma Fuzzy Evolusi dan Matlab.

BAB III. METODE PENELITIAN

Bab ini menguraikan langkah-langkah kerja yang akan ditempuh, meliputi menemukan masalah, pengambilan data, analisis dan pemecahan masalah, serta penarikan simpulan.

BAB IV. HASIL PENELITIAN DAN PEMBAHASAN

Bab ini berisi pembahasan dari permasalahan yang dikemukakan. Bab ini dibagi menjadi dua sub bab, yaitu hasil penelitian dan pembahasan. Hasil penelitian berisi hasil perhitungan dan analisis data yang diperoleh dari studi pustaka maupun pemecahan kasus penentuan rute dan jarak minimum dari

jaringan TSP pada pengiriman barang di PT. Jalur Nugraha Ekakurir (JNE) Semarang dengan menggunakan algoritma Fuzzy Evolusi.

BAB V. PENUTUP

Bab ini dibagi menjadi dua sub bab, yaitu simpulan dan saran. Simpulan berisi tentang menyimpulkan secara garis besarnya saja isi dalam skripsi, sedangkan saran berupa komentar, sanggaan yang bersifat menyarankan baik kepada pemerintah, instansi dll tergantung dengan variabel yang ada dalam skripsi.

3. Bagian akhir skripsi

Bagian ini berisi daftar pustaka sebagai acuan penulisan dan lampiran yang mendukung kelengkapan skripsi.

BAB 2

LANDASAN TEORI

2.1 Teori Graf

2.1.1 Pengertian Graf

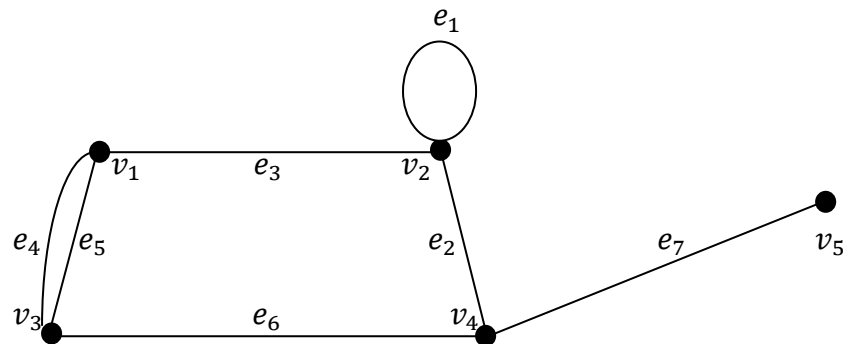
Graf G didefinisikan sebagai pasangan himpunan (V, E) , dengan notasi $G = (V, E)$, yang dalam hal ini V adalah himpunan tidak kosong dari simpul-simpul (*vertices* atau *nodes*) dan E adalah himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang simpul (Munir, 2005: 356).

Setiap garis berhubungan dengan satu atau dua titik. Titik-titik tersebut dinamakan Titik Ujung. Garis yang hanya berhubungan dengan satu titik ujung disebut *loop*. Dua garis berbeda yang menghubungkan titik yang sama disebut Garis Paralel (Siang, 2002: 186).

Menurut Rosen (2003: 539) *loop* adalah sisi yang berasal dari suatu titik yang kembali lagi ke titik itu sendiri yang tidak diperbolehkan dalam graf rangkap

Sebuah graf linier (atau secara sederhana disebut graf) $G = (V, E)$ adalah suatu sistem yang terdiri atas suatu himpunan objek $V = \{v_1, v_2, \dots\}$ yang disebut himpunan titik, dan sebuah koleksi $E = \{e_1, e_2, \dots\}$ yang merupakan koleksi sisi sedemikian hingga tiap sisi e_k dikaitkan dengan suatu pasangan tak-terurut (v_i, v_j) . titik v_i, v_j yang berkaitan dengan e_k disebut titik-titik ujung sisi e_k (Sutarno dkk, 2003: 59).

Untuk lebih jelasnya, Sutarno (2003: 59) memberikan contoh graf yang direpresentasikan dengan diagram pada Gambar 2.1.



Gambar 2.1 Graf dengan Lima Titik dan Tujuh Sisi

2.1.2 Graf Tak Berarah

2.1.2.1 Path dan Sirkuit

Suatu *walk* dari v ke w adalah barisan titik-titik yang berhubungan dan garis secara selang-seling, diawali dari titik v dan diakhiri pada titik w .

Walk dengan panjang n dari v ke w dituliskan sebagai berikut: $v_0 e_1 v_1 e_2 v_2 \dots v_{n-1} e_n v_n$ dengan $v_0 = v$, $v_n = w$ dan v_1 adalah titik-titik ujung garis e_1 .

Path dengan panjang n dari v ke w adalah *walk* dari v ke w yang semua garisnya berbeda. *Path* dari v ke w dituliskan sebagai $v = v_0 e_1 v_1 e_2 v_2 \dots v_{n-1} e_n v_n = w$ dengan $e_i \neq e_j$ untuk $i \neq j$.

Path sederhana dengan panjang n dari v ke w adalah *path* dari v ke w yang semua titiknya berbeda. *Path* sederhana dari v ke w berbentuk $v = v_0 e_1 v_1 e_2 v_2 \dots v_{n-1} e_n v_n = w$ dengan $e_i \neq e_j$ untuk $i \neq j$ dan $v_k \neq v_m$ untuk $k \neq m$.

Sirkuit dengan panjang n adalah *path* yang dimulai dan diakhiri pada titiknya berbeda. Sirkuit sederhana berbentuk $v_0e_1v_1e_2v_2 \dots v_{n-1}e_nv_n$ dengan $e_i \neq e_j \neq j$ dan $v_k \neq v_m$ untuk $k \neq m$ kecuali $v_0 = v_n$ (Siang, 2002: 210).

2.1.2.2 Graf Terhubung dan Tidak Terhubung

Misalkan G adalah suatu graf. Dua titik v dan w dalam G dikatakan terhubung bila dan hanya bila ada walk dari v ke w .

Graf G dikatakan terhubung bila dan hanya bila setiap dua titik dalam G terhubung. Graf G dikatakan tidak terhubung bila dan hanya bila ada dua titik dalam G yang tidak terhubung (Siang, 2002: 215).

2.1.2.3 Sirkuit Hamilton

Suatu graf terhubung G disebut sirkuit Hamilton bila ada sirkuit yang mengunjungi setiap titiknya tepat satu kali (kecuali titik awal yang sama dengan titik akhirnya) (Siang, 2002: 220).

2.1.3 Graf Berarah

Suatu graf berarah G terdiri dari himpunan titik-titik $V(G):\{v_1, v_2, \dots\}$, himpunan garis-garis $E(G):\{e_1, e_2, \dots\}$, dan suatu fungsi ψ yang mengawankan setiap garis dalam $E(G)$ ke suatu pasangan berurutan titik (v_i, v_j) .

Jika $e_k = (v_i, v_j)$ adalah suatu garis dalam G , maka v_i disebut juga titik awal e_k dan v_j disebut titik akhir e_k . arah garis adalah dari v_i ke v_j .

Jumlah garis yang keluar dari titik v_i disebut derajat keluar (*out degree*) titik v_i (simbol $d^+(v_i)$), sedangkan jumlah garis yang menuju ke titik v_i disebut derajat masuk (*in degree*) titik v_i , yang disimbolkan sebagai $d^-(v_i)$ (Siang, 2002: 226).

2.1.4 Graf Berbobot

Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot). Bobot pada tiap sisi dapat menyatakan jarak antara dua buah kota, waktu tempuh antara dua buah kota, biaya perjalanan yang kita tempuh, dan sebagainya (Sutarno dkk, 2003: 107).

2.2 *Travelling Salesman Problem (TSP)*

Travelling Salesman Problem (TSP) termasuk ke dalam persoalan yang sangat terkenal dalam teori graf. Nama persoalan ini diilhami oleh masalah seorang pedagang yang berkeliling mengunjungi sejumlah kota. Deskripsi persoalannya adalah sebagai berikut: diberikan sejumlah kota dan jarak antar kota. Tentukan sirkuit terpendek yang harus dilalui oleh seorang pedagang bila pedagang itu berangkat dari sebuah kota asal dan menyinggahi setiap kota tepat satu kali dan kembali lagi ke kota asal keberangkatan.

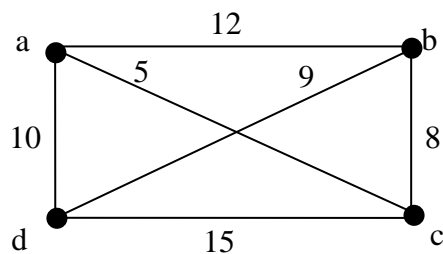
Kota dapat dinyatakan sebagai simpul graf, sedangkan sisi menyatakan jalan yang menghubungkan antar dua buah kota. Bobot pada sisi menyatakan jarak antara dua buah kota. Persoalan perjalanan pedagang tidak lain adalah menentukan sirkuit Hamilton yang memiliki bobot minimum pada sebuah graf terhubung.

Pada persoalan TSP ini, jika setiap simpul mempunyai sisi ke simpul yang lain, maka graf yang merepresentasikannya adalah graf lengkap berbobot. Pada sembarang graf lengkap dengan n buah simpul ($n > 2$), jumlah sirkuit Hamilton yang berbeda adalah $\frac{(n-1)!}{2}$. Rumus ini dihasilkan dari kenyataan bahwa dimulai

dari sembarang simpul kita mempunyai $n - 1$ buah sisi untuk dipilih dari simpul pertama, $n - 2$ sisi dari simpul kedua, $n - 3$ dari simpul ketiga, dan seterusnya. Ini adalah pilihan yang independen, sehingga kita memperoleh $(n - 1)!$ pilihan. Jumlah itu harus dibagi dengan 2, karena tiap sirkuit Hamilton terhitung dua kali, sehingga semuanya ada $\frac{(n-1)!}{2}$ buah sirkuit Hamilton.

Contoh:

Tinjau graf lengkap dengan $n = 4$ simpul seperti yang ditunjukkan pada Gambar 2.2.



Gambar 2.2 Graf Lengkap dengan 4 Simpul. Tiap Sisi Diberi Bobot

Penyelesaian:

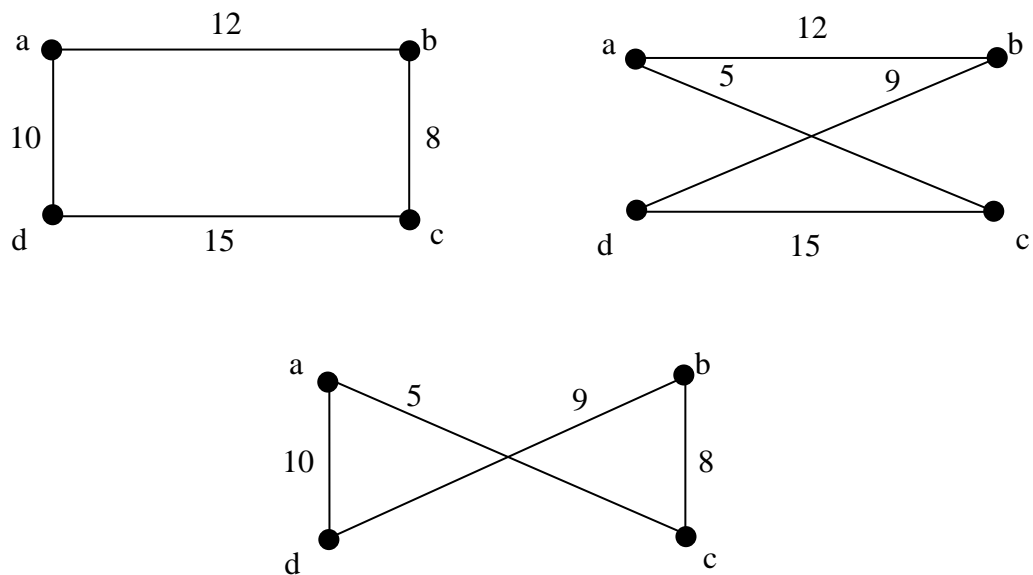
Graf di atas memiliki $\frac{(4-1)!}{2} = 3$ sirkuit Hamilton, yaitu:

$S_1 = (a, b, c, d, a)$ atau (a, d, c, b, a) dengan panjang rute $10 + 12 + 8 + 15 = 45$

$S_2 = (a, c, d, b, a)$ atau (a, b, d, c, a) dengan panjang rute $12 + 5 + 9 + 15 = 41$

$S_3 = (a, c, b, d, a)$ atau (a, d, b, c, a) dengan panjang rute $10 + 5 + 9 + 8 = 32$

Gambar sirkuit Hamilton-nya adalah pada Gambar 2.3.



Gambar 2.3 Tiga Buah Sirkuit Hamilton

Jadi, sirkuit Hamilton terpendek adalah $S_3 = (a, c, b, d, a)$ atau (a, d, b, c, a) dengan panjang rute $10 + 5 + 9 + 8 = 32$. Ini adalah solusi persoalan TSP untuk graf berbobot pada graf gambar 4 (Munir, 2005: 421).

Salah satu cara termudah untuk menyelesaikan TSP yaitu dengan menggunakan algoritma *brute force*. Hal yang dilakukan yaitu mencoba semua kombinasi dan mencari rute yang paling murah. Tetapi hal tersebut memerlukan waktu yang sangat lama karena banyaknya jumlah kombinasi yang ada. Sebagai contoh, jumlah kombinasi rute untuk 20 kota adalah $20! = 2,4 \times 10^{18}$. Jumlah yang sangat besar untuk suatu algoritma pencarian.

Metode konvensional lain dalam menyelesaikan TSP yaitu dengan menggunakan algoritma greedy. Hal yang dilakukan yaitu memilih kota yang belum dikunjungi yang mempunyai biaya paling rendah pada setiap langkah.

Namun, dengan menggunakan algoritma greedy solusi yang dihasilkan tidak menjamin bahwa solusi tersebut optimal.

2.3 Algoritma Genetika

Algoritma genetika adalah suatu algoritma pencarian yang berbasis pada mekanisme seleksi alam dan genetika. Algoritma genetika merupakan salah satu algoritma yang sangat tepat digunakan dalam menyelesaikan masalah optimasi kompleks, yang sulit dilakukan oleh metode konvensional (Desiani, 2006: 187).

Algoritma genetika pertama kali diperkenalkan oleh John Holland (1975) dari Universitas Michigan. John Holland mengatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan ke dalam terminologi genetika. Goldberg mendefinisikan algoritma genetika ini sebagai suatu pencarian algoritma berdasarkan pada mekanisme seleksi alam dan genetika alam (Desiani, 2006: 187).

Beberapa definisi penting dalam algoritma genetika, yaitu:

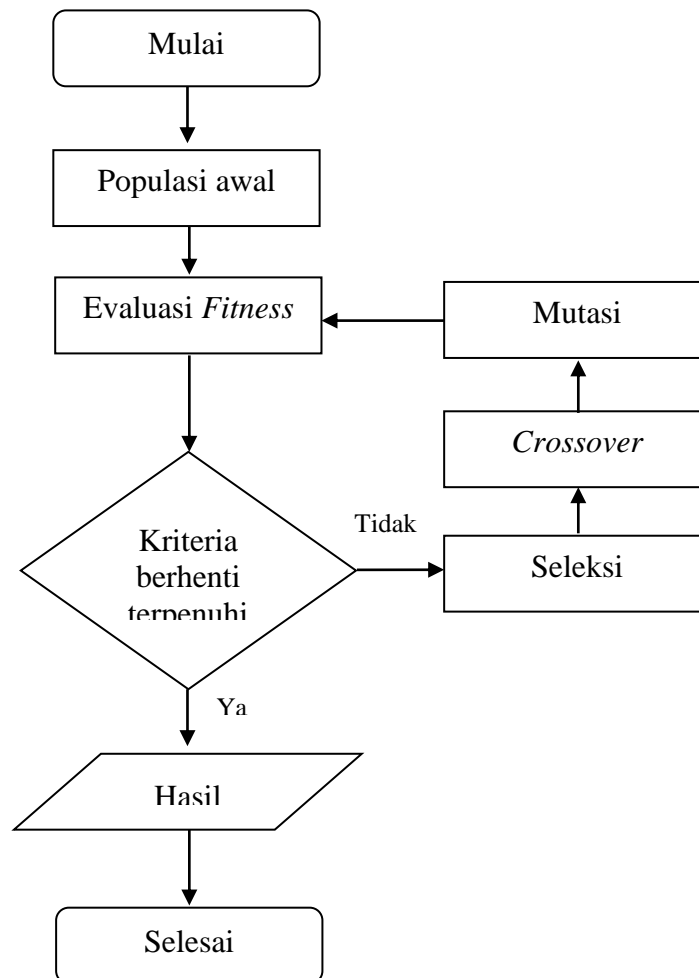
1. Genotip (Gen) adalah sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Dalam algoritma genetika, gen ini bisa bernilai biner, float, integer maupun karakter.
2. Allel adalah nilai dari gen.
3. Kromosom adalah gabungan gen-gen yang membentuk nilai tertentu.
4. Individu menyatakan satu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.

5. Populasi merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi.
6. Generasi menyatakan satu satuan siklus proses evolusi.
7. Nilai *Fitness* menyatakan seberapa baik nilai dari suatu individu atau solusi yang didapatkan.

Ciri-ciri permasalahan yang dikerjakan dengan menggunakan algoritma genetika adalah:

1. Mempunyai fungsi tujuan optimalisasi non linear dengan banyak kendala yang juga non linear.
2. Mempunyai kemungkinan solusi yang jumlahnya tak berhingga.
3. Membutuhkan solusi “*real-time*” dalam arti solusi bisa didapatkan dengan cepat sehingga dapat diimplementasikan untuk permasalahan yang mempunyai perubahan yang cepat seperti optimasi pada pembebanan kanal pada komunikasi seluler.
4. Mempunyai multi-obyektif dan multi-kriteria, sehingga diperlukan solusi yang dapat secara bijak diterima oleh semua pihak (Basuki, 2003: 4).

Algoritma genetika secara umum dapat diilustrasikan dalam diagram alir yang dapat dilihat pada Gambar 2.4.



Gambar 2.4 Diagram Alir Algoritma Genetika

Keterangan Gambar 2.4:

1. Populasi awal

Proses ini merupakan proses yang digunakan untuk membangkitkan populasi awal secara random sehingga didapatkan solusi awal.

2. Evaluasi *fitness*

Proses ini merupakan proses untuk mengevaluasi setiap populasi dengan menghitung nilai *fitness* setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti.

3. Seleksi

Proses seleksi merupakan proses untuk menentukan individu-individu mana saja yang akan dipilih untuk dijadikan *crossover*.

4. *Crossover*

Proses *crossover* ini merupakan proses untuk menambah keanekaragaman suatu populasi.

5. Mutasi

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam satu kromosom.

6. Kriteria berhenti

Kriteria berhenti merupakan kriteria yang digunakan untuk menghentikan proses algoritma genetika.

7. Hasil

Hasil merupakan seleksi optimum yang didapat oleh algoritma genetika.

Struktur umum algoritma genetika dapat didefinisikan dengan langkah-langkah sebagai berikut:

1. Membangkitkan populasi awal

Populasi awal ini dibangkitkan secara random sehingga didapatkan solusi awal. Populasi itu sendiri terdiri dari sejumlah kromosom yang mempresentasikan solusi yang diinginkan.

2. Membentuk generasi baru

Dalam membentuk generasi baru digunakan tiga operator yaitu operator reproduksi/seleksi, *crossover* dan mutasi. Proses ini dilakukan berulang-ulang hingga didapatkan jumlah kromosom yang cukup untuk membentuk generasi baru dimana generasi baru ini merupakan representasi dari solusi baru.

3. Evaluasi solusi

Proses ini akan mengevaluasi setiap populasi dengan menghitung nilai fitness setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti. Bila kriteria berhenti belum terpenuhi maka akan dibentuk lagi generasi baru dengan mengulangi langkah 2. Beberapa kriteria berhenti yang sering digunakan antara lain:

- a. Berhenti pada generasi tertentu.
- b. Berhenti setelah dalam beberapa generasi berturut-turut didapatkan nilai *fitness* tertinggi tidak berubah.
- c. Berhenti bila dalam n generasi berikut tidak didapatkan nilai *fitness* yang lebih tinggi (Wibowo, 2009:13).

Komponen-komponen utama algoritma genetika:

1. Teknik pengkodean

Pengkodean adalah suatu teknik untuk menyatakan populasi awal sebagai calon solusi suatu masalah ke dalam suatu kromosom sebagai suatu kunci pokok persoalan ketika menggunakan algoritma genetika.

Teknik pengkodean ini meliputi pengkodean gen dan kromosom. Gen merupakan bagian dari kromosom. Satu gen bisa mewakili satu variabel. Gen dapat direpresentasikan dalam bentuk string bit, pohon, array bilangan real, daftar aturan, elemen permutasi, elemen program, atau representasi lainnya yang dapat diimplementasikan untuk operator genetika.

2. Prosedur inisialisasi

Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian harus dilakukan inisialisasi kromosom dilakukan secara acak, namun demikian harus tetap memperhatikan domain solusi dan kendala permasalahan yang ada.

3. Evaluasi *fitness*

Evaluasi *fitness* merupakan dasar untuk proses seleksi. Langkah-langkahnya yaitu string dikonversi ke parameter fungsi, fungsi obyektifnya dievaluasi, kemudian mengubah fungsi obyektif tersebut kedalam fungsi *fitness*, dimana untuk maksimasi problem, *fitness* sama dengan fungsi obyektifnya. Output dari fungsi *fitness* dipergunakan sebagai dasar untuk menseleksi individu pada generasi berikutnya.

Untuk permasalahan minimalisasi, nilai *fitness* adalah inversi dari nilai maksimal yang diharapkan. Proses inversi dapat dilakukan dengan rumusan

$tness = \frac{1}{f(x)}$, dengan x merupakan kromosom (Basuki, 2003:17).

4. Seleksi

Seleksi ini bertujuan memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling baik.

Ada beberapa metode seleksi dari induk, antara lain:

a. *Rank-based Fitness*

Pada *Rank-based fitness*, populasi diurutkan menurut nilai obyektifnya. Nilai *fitness* tiap-tiap individu hanya tergantung pada posisi individu tersebut dalam urutan, dan tidak dipengaruhi oleh nilai obyektifnya.

b. *Roulette Wheel Selection*

Metode seleksi dengan mesin roulette ini merupakan metode yang paling sederhana dan sering dikenal dengan nama *stochastic sampling with replacement*. Cara kerja metode ini adalah sebagai berikut:

- 1). Dihitung nilai *fitness* dari masing-masing individu (f_i , dimana i adalah individu ke-1 sampai dengan ke- n)
- 2). Dihitung total *fitness* semua individu
- 3). Dihitung probabilitas masing-masing individu
- 4). Dari probabilitas tersebut dihitung jatah masing-masing individu pada angka 1 sampai 100
- 5). Dibangkitkan bilangan random antara 1 sampai 100
- 6). Dari bilangan random yang dihasilkan, tentukan individu mana yang terpilih dalam proses seleksi

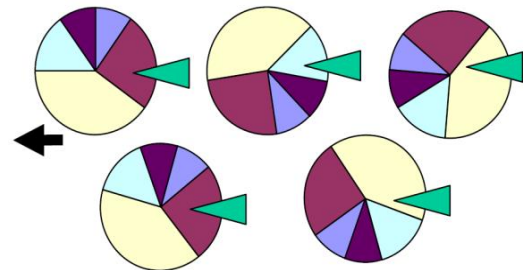
Individu 1: <i>fitness</i> = 10%	Jatah untuk individu 1: 1-10
Individu 2: <i>fitness</i> = 25%	Jatah untuk individu 2: 11-35
Individu 3: <i>fitness</i> = 40%	Jatah untuk individu 3: 36-75
Individu 4: <i>fitness</i> = 15%	Jatah untuk individu 4: 75-90
Individu 5: <i>fitness</i> = 10%	Jatah untuk individu 5: 91-100



Dibangkitkan bilangan random
antara 1-100 sebanyak 5 kali

Individu Terpilih

Random 30 → individu 2
 Random 88 → individu 4
 Random 64 → individu 3
 Random 18 → individu 2
 Random 44 → individu 3



Gambar 2.5 Ilustrasi Seleksi dengan Mesin Roulette

c. *Stochastic Universal Sampling*

Pada metode ini, individu-individu dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga tiap-tiap segmen individu memiliki ukuran yang sama dengan ukuran *fitness*-nya seperti halnya pada seleksi roda roulette. Kemudian diberikan sejumlah *pointer* sebanyak individu yang ingin diseleksi pada garis tersebut. Andaikan N adalah jumlah individu yang akan diseleksi, maka jarak

antar *pointer* adalah $1/N$, dan posisi *pointer* pertama diberikan secara acak pada *range* $[1, 1/N]$.

d. *Truncation Selection*

Seleksi ini biasanya digunakan oleh populasi yang jumlahnya sangat besar. Pada metode ini, individu-individu diurutkan berdasarkan nilai *fitness*-nya. Hanya individu-individu yang terbaik saja yang akan diseleksi sebagai induk. Parameter yang digunakan dalam metode ini adalah suatu nilai ambang trunk yang mengindikasikan ukuran populasi yang akan diseleksi sebagai induk yang berkisar antara 50%-100%. Individu-individu yang ada di bawah nilai ambang ini tidak akan menghasilkan keturunan.

e. *Tournament Selection*

Pada metode seleksi dengan turnamen, ditetapkan suatu nilai tour untuk individu-individu yang dipilih secara acak dari suatu populasi. Individu-individu yang terbaik dalam kelompok ini akan diseleksi sebagai induk. Parameter yang digunakan pada metode ini adalah ukuran tour yang bernilai antara 2 sampai N (jumlah individu dalam suatu populasi).

5. *Crossover*

Crossover (perkawinan silang) bertujuan menambah keanekaragaman string dalam suatu populasi. Beberapa jenis *crossover* tersebut adalah:

a. *Crossover* Diskret

Proses *crossover* dilakukan dengan menukar nilai variabel antar kromosom induk. Misalkan ada 2 individu dengan 3 variabel, yaitu:

Induk 1:	12	25	5
Induk 2:	123	4	34

Untuk tiap-tiap variabel induk yang menyumbangkan variabelnya ke anak dipilih secara acak dengan probabilitas yang sama.

Sampel 1:	2	2	1
Sampel 2:	1	2	1

Kromosom baru yang terbentuk:

Anak 1:	123	4	5
Anak 2:	12	4	5

b. *Crossover Intermediate* (menengah)

Crossover menengah merupakan metode *crossover* yang hanya dapat digunakan untuk variabel real. Nilai variabel anak dipilih di sekitar dan antara nilai-nilai variabel induk. Anak dihasilkan menurut aturan sebagai berikut:

$$\text{Anak} = \text{induk 1} + \alpha (\text{induk 2} - \text{induk 1})$$

Dengan α adalah faktor skala yang dipilih secara random pada interval $[-d, 1+d]$, biasanya $d = 0,25$. Tiap-tiap variabel pada anak merupakan hasil *crossover* variabel-variabel menurut aturan di atas dengan nilai α dipilih ulang untuk tiap variabel.

Misalkan ada 2 individu dengan 3 variabel, yaitu:

Induk 1: 12 25 5

Induk 2: 123 4 34

Misalkan nilai alpha yang terpilih adalah:

Sampel 1: 0,5 1,1 -0,1

Sampel 2: 0,1 0,8 0,5

Kromosom baru yang terbentuk

Anak 1: 67,5 1,9 2,1

Anak 2: 23,1 8,2 19,5

c. *Crossover* Garis

Pada dasarnya *crossover* garis ini sama dengan *crossover* menengah, hanya saja nilai alpha untuk semua variabel sama. Misalkan ada 2 individu dengan 3 variabel, yaitu:

Induk 1: 12 25 5

Induk 2: 123 4 34

Alpha yang dipilih adalah:

Sampel 1: 0,5

Sampel 2: 0,1

Kromosom baru yang terbentuk :

Anak 1: 67,5 14,5 19,5

Anak 2: 23,1 22,9 7,9

d. *Crossover* dengan Permutasi

Pada penyilangan permutasi ini kromosom-kromosom anak diperoleh dengan cara memilih sub barisan *tour* dari satu induk dengan

tetap menjaga urutan dan posisi sejumlah gen yang mungkin terhadap induk yang lainnya.

Contoh *crossover* dengan permutasi:

Misal

Induk 1: (1 2 3 | 4 5 6 7 | 8 9)

Induk 2: (4 5 3 | 1 8 7 6 | 9 2)

Anak 1: (x x x | 1 8 7 6 | x x)

Anak 2: (x x x | 4 5 6 7 | x x)

Dari sini kita memperoleh pemetaan:

1-4, 8-5, 7-6, 6-7

Kemudian *copy* sisa gen di induk 1 ke anak 1 dengan menggunakan pemetaan yang sudah ada.

Anak 1: (1-4 2 3 | 1 8 7 6 | 8-5 9)

Anak 1: (4 2 3 | 1 8 7 6 | 5 9)

Lakukan hal yang sama untuk anak 2

Anak 2: (1-4 5-8 3 | 1 8 7 6 | 9 2)

e. *Order Crossover*

Order crossover merupakan cara *crossover* dengan menukar kromosom dengan tetap menjaga urutan gen yang bukan bagian dari kromosom tersebut.

Contoh *order crossover* adalah:

Misalkan ada 3 kromosom induk yang akan dilakukan *crossover*:

Kromosom[1] = [ABCD]

Kromosom [2] = [BACD]

Kromosom [3] = [ACDB]

Proses *crossover*:

$$\begin{aligned} \text{Kromosom}[1] &= \text{Kromosom}[1] \times\!< \text{Kromosom}[2] \\ &= [\text{ABCD}] \times\!< [\text{BACD}] \\ &= [\text{ACBD}] \end{aligned}$$

$$\begin{aligned} \text{Kromosom}[2] &= \text{Kromosom}[2] \times\!< \text{Kromosom}[3] \\ &= [\text{BACD}] \times\!< [\text{BCDA}] \\ &= [\text{BCDA}] \end{aligned}$$

$$\begin{aligned} \text{Kromosom}[3] &= \text{Kromosom}[3] \times\!< \text{Kromosom}[1] \\ &= [\text{BCDA}] \times\!< [\text{ABCD}] \\ &= [\text{BCAE}] \end{aligned}$$

6. Mutasi

Mutasi merupakan proses mengubah nilai satu atau beberapa gen dalam satu kromosom. Mutasi ini berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi.

a. Mutasi dengan pengkodean biner

Mutasi dengan pengkodean biner merupakan operasi yang sangat sederhana. Proses yang dilakukan adalah menginversi nilai bit pada posisi tertentu yang dipilih secara acak (atau dengan menggunakan skema tertentu) pada kromosom.

Contoh mutasi pada pengkodean biner

Kromosom sebelum mutasi : 1 0 0 1 0 1 1 1

Kromosom sesudah mutasi : 1 0 0 1 0 0 1 1

b. Mutasi dengan pengkodean permutasi

Proses mutasi yang dilakukan dalam pengkodean biner tidak dapat dilakukan pada pengkodean permutasi karena konsistensi urutan permutasi harus diperhatikan. Salah satu cara yang dapat dilakukan adalah dengan memilih dua posisi (*locus*) dari kromosom dan kemudian nilainya saling dipertukarkan.

Contoh mutasi dalam pengkodean permutasi

Kromosom sebelum mutasi : 1 2 3 4 5 6 7 8 9

Kromosom sesudah mutasi : 1 2 7 4 6 5 8 3 9

c. Mutasi dengan pengkodean nilai

Proses mutasi dalam pengkodean nilai dapat dilakukan dengan berbagai cara, salah satunya yaitu dengan memilih sembarang posisi gen pada kromosom, nilai yang ada tersebut kemudian ditambahkan atau dikurangkan dengan suatu nilai kecil tertentu yang diambil secara acak.

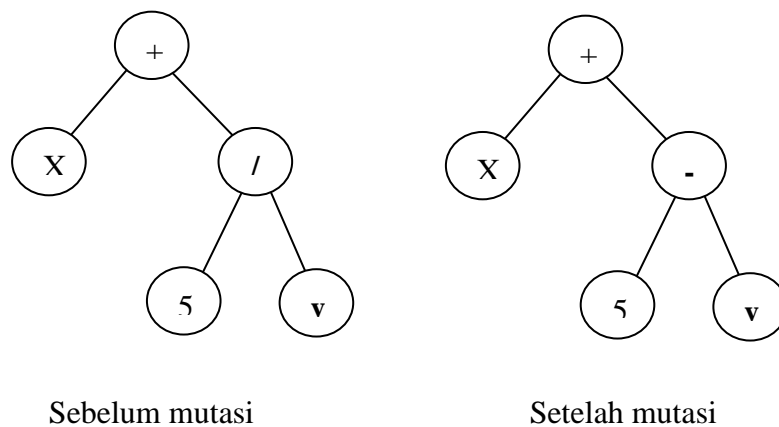
Contoh mutasi dalam pengkodean nilai riil dengan nilai yang ditambahkan atau dikurangkan adalah 0,1

Kromosom sebelum mutasi : 1,43 1,09 4,51 9,11 6,94

Kromosom sesudah mutasi : 1,43 1,19 4,51 9,01 6,94

d. Mutasi dengan pengkodean pohon

Mutasi dalam pengkodean pohon dapat dilakukan antara lain dengan cara mengubah operator (+, -, *, /) atau nilai yang terkandung dalam suatu *vertex* pohon yang dipilih. Atau dapat juga dilakukan dengan memilih dua *vertex* dari pohon dan saling mempertukarkan operator atau nilainya. Contoh mutasi dalam pengkodean pohon seperti pada gambar 2.6.



Gambar 2.6 Contoh Gen Sebelum dan Setelah Mutasi dengan Pengkodean Pohon

e. *Swapping Mutation*

Proses mutasi dengan cara ini dilakukan dengan menentukan jumlah kromosom yang akan mengalami mutasi dalam satu populasi melalui parameter *mutation rate* (pm). Proses mutasi dilakukan dengan cara menukar gen yang telah dipilih secara acak dengan gen sesudahnya, jika gen tersebut berada di akhir kromosom, maka ditukar dengan gen yang pertama.

Pertama hitung panjang total gen yang ada pada suatu populasi:

Panjang total gen = jumlah gen dalam 1 kromosom * jumlah kromosom

Untuk memilih posisi gen yang akan mengalami mutasi dilakukan dengan membangkitkan bilangan acak antara 1 sampai Panjang total gen untuk dilakukan proses mutasi (Wibowo, 2003: 14).

2.4 Fuzzy

2.4.1 Logika Fuzzy

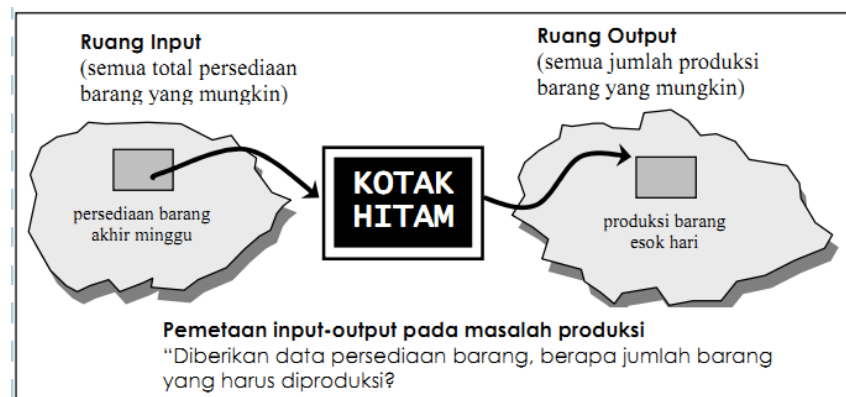
Konsep logika fuzzy pertama kali diperkenalkan pada tahun 1965 oleh Prof. Lotfi A. Zadeh, seorang professor dari University of California di Berkly. Dasar logika fuzzy adalah teori himpunan fuzzy. Pada teori himpunan fuzzy, peranan derajat keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangatlah penting. Nilai keanggotaan atau derajat keanggotaan (membership values) yang nilainya terletak di antara selang [0,1] menjadi ciri utama dari penalaran dengan logika fuzzy tersebut.

Logika fuzzy adalah suatu cara yang tepat untuk memetakan suatu ruang input ke dalam suatu ruang output. Sebagai contoh:

1. Manajer pergudangan mengatakan pada manajer produksi seberapa banyak persediaan barang pada akhir minggu ini, kemudian manajer produksi akan menetapkan jumlah barang yang harus diproduksi esok hari.
2. Pelayan restoran memberikan pelayanan terhadap tamu, kemudian tamu akan memberikan tip yang sesuai atas baik tidaknya pelayan yang diberikan.

3. Anda mengatakan pada saya seberapa sejuk ruangan yang anda inginkan, saya akan mengatur putaran kipas yang ada pada ruangan ini.
4. Penumpang taksi berkata pada sopir taksi seberapa cepat laju kendaraan yang diinginkan, sopir taksi akan mengatur pijakan gas taksinya.

Salah satu contoh pemetaan suatu input-output dalam bentuk grafis seperti terlihat pada Gambar 2.7.



Gambar 2.7 Contoh Pemetaan Input-Output (Kusumadewi, 2006: 154)

Antara input dan output terdapat satu kotak hitam yang harus memetakan input ke output yang sesuai.

Menurut Cox, ada beberapa alasan mengapa orang menggunakan logika fuzzy, antara lain:

- 2.4.1.1** Konsep logika fuzzy mudah dimengerti. Karena konsep matematis yang mendasari penalaran fuzzy cukup mudah dimengerti.
- 2.4.1.2** Logika fuzzy sangat fleksibel, artinya mampu beradaptasi dengan perubahan-perubahan, dan ketidakpastian yang menyertai permasalahan.
- 2.4.1.3** Logika fuzzy memiliki toleransi terhadap data yang tidak tepat.

2.4.1.4 Logika fuzzy mampu memodelkan fungsi-fungsi nonlinier yang sangat kompleks.

2.4.1.5 Logika fuzzy dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus pelatihan.

2.4.1.6 Logika fuzzy dapat bekerjasama dengan teknik-teknik kendali konvensional.

2.4.1.7 Logika fuzzy didasarkan pada bahasa alami.

2.4.2 Himpunan Fuzzy

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan $\mu_A[x]$, memiliki 2 kemungkinan, yaitu:

1. satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
2. nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Prinsip dasar dan persamaan matematika dari teori himpunan fuzzy adalah pengelompokan objek dalam batas yang samar. Himpunan fuzzy merupakan sebuah generalisasi dari himpunan *crisp*. Kalau pada himpunan *crisp*, nilai keanggotaan hanya ada 2 kemungkinan, yaitu 0 atau 1. Sedangkan himpunan fuzzy didasarkan pada gagasan untuk memperluas jangkauan fungsi karakteristik sedemikian hingga fungsi tersebut akan mencakup bilangan real pada interval $[0,1]$. Nilai keanggotaan pada himpunan fuzzy menunjukkan bahwa suatu item dalam semesta pembicaraan tidak hanya berada pada 0 atau 1, melainkan juga

nilai yang terletak diantaranya. Dengan kata lain, nilai kebenaran dari suatu item tidak hanya benar atau salah.

Pada himpunan fuzzy terdapat 2 atribut, yaitu:

1. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti : MUDA, PAROBAYA, TUA.
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel, seperti : 40, 25, 50, dsb.

Ada beberapa hal yang perlu diketahui dalam memahami sistem fuzzy, yaitu:

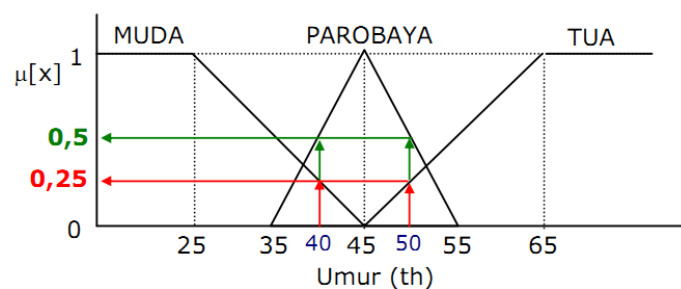
1. Variabel fuzzy

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

2. Himpunan fuzzy

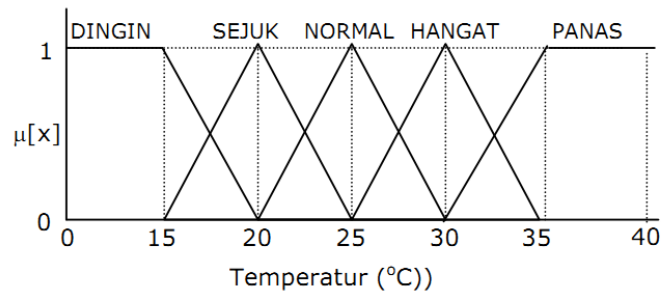
Himpunan fuzzy merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel fuzzy. Contoh:

- Variabel umur, terbagi menjadi 3 himpunan fuzzy, yaitu: MUDA, PAROBAYA, dan TUA (Gambar 2.8).



Gambar 2.8 Himpunan Fuzzy untuk Variabel Umur

- Variabel temperatur, terbagi menjadi 5 himpunan fuzzy, yaitu: DINGIN, SEJUK, NORMAL, HANGAT, dan PANAS (Gambar 2.9).



Gambar 2.9 Himpunan Fuzzy pada Variabel Temperatur

3. Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel fuzzy. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya. Contoh:

- Semesta pembicaraan untuk variabel umur: $[0 + \infty)$.
- Semesta pembicaraan untuk variabel temperatur: $[0 40]$.

4. Domain

Domain himpunan fuzzy adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan fuzzy. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif maupun negatif. Contoh domain himpunan fuzzy:

- Muda = $[0 \ 45]$.
- Pabobaya = $[35 \ 55]$.
- Tua = $[45 \ +\infty)$.
- Dingin = $[0 \ 20]$.
- Sejuk = $[15 \ 25]$.
- Normal = $[20 \ 30]$.
- Hangat = $[25 \ 35]$.
- Panas = $[30 \ 40]$.

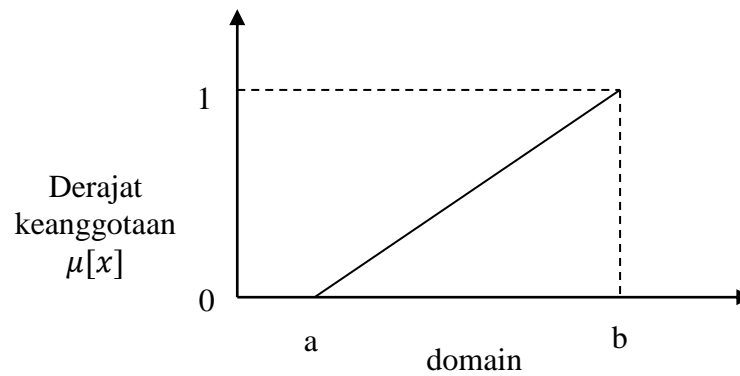
2.4.3 Fungsi Keanggotaan Fuzzy

Fungsi keanggotaan fuzzy (membership function) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (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, diantaranya sebagai berikut :

a. Representasi Linier

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

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



Gambar 2.10 Representasi Linier Naik

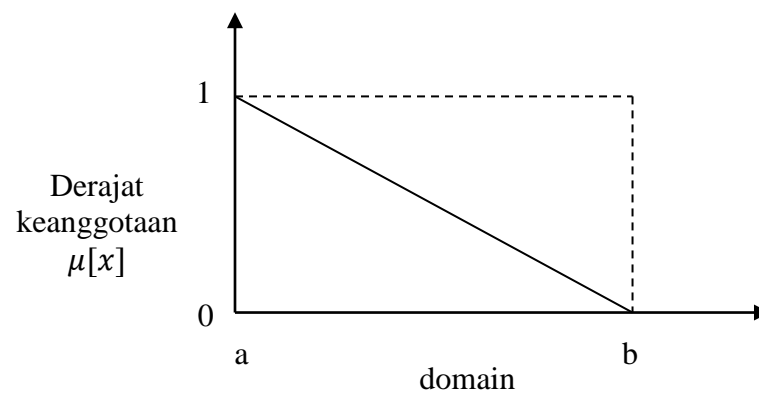
Fungsi keanggotaan:

$$\mu[x] \begin{cases} 0; & x \leq a \\ \frac{x-a}{b-a}; & a \leq x \leq b \\ 1; & x \geq b \end{cases}$$

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

Fungsi keanggotaan:

$$\mu[x] \begin{cases} \frac{b-x}{b-a}; & a \leq x \leq b \\ 0; & x \geq b \end{cases}$$



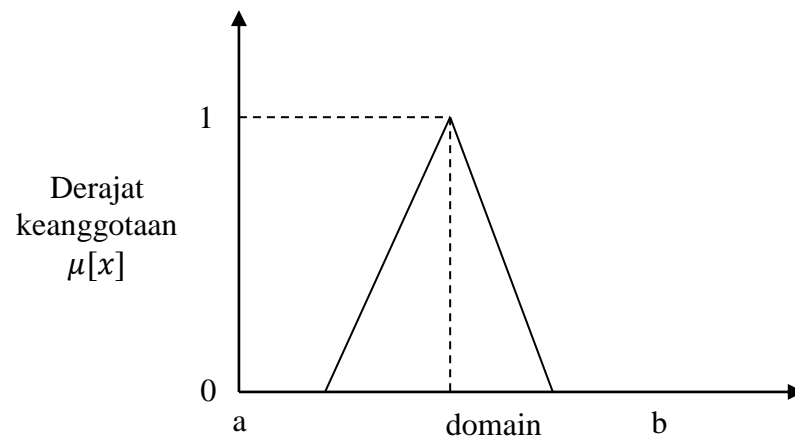
Gambar 2.11 Representasi Turun

b. Representasi Kurva Segitiga

Kurva Segitiga pada dasarnya merupakan gabungan antara 2 garis (linier).

Fungsi keanggotaan:

$$\mu[x] \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a}; & a \leq x \leq b \\ \frac{c-x}{c-b}; & b \leq x \leq c \end{cases}$$



Gambar 2.12 Representasi Kurva Segitiga

c. Representasi Bentuk Bahu

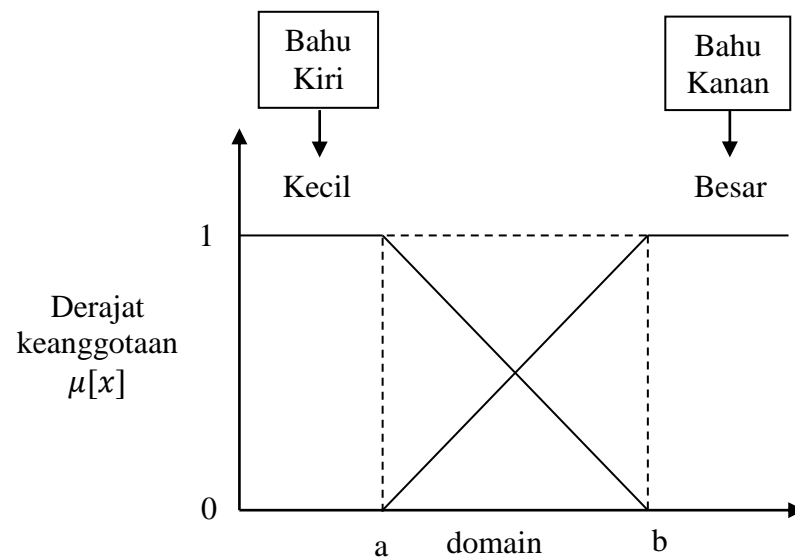
Pada representasi kurva bahu, daerah yang terletak ditengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun (misalkan: DINGIN bergerak ke SEJUK bergerak ke HANGAT dan bergerak ke PANAS). Tapi terkadang, salah satu sisi dari variabel tersebut tidak mengalami perubahan. Sebagai contoh, apabila telah mencapai kondisi PANAS, kenaikan temperatur akan tetap pada kondisi PANAS. Himpunan fuzzy 'bahu', bukan

segitiga, digunakan untuk mengakhiri variabel suatu daerah fuzzy. Bahu kiri bergerak dari benar ke salah, sebaliknya bahu kanan bergerak dari salah ke benar.

Fungsi keanggotaan:

$$\mu_{Kecil}[x] \begin{cases} 0; & x > b \\ \frac{b-x}{b-a}; & a \leq x \leq b \\ 1; & x < a \end{cases}$$

$$\mu_{Besar}[x] \begin{cases} 0; & x < a \\ \frac{x-a}{b-a}; & a \leq x \leq b \\ 1; & x > b \end{cases}$$

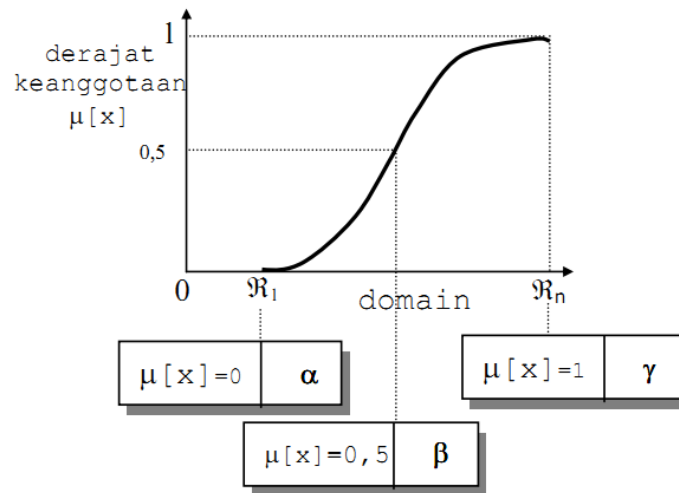


Gambar 2.13 Representasi Kurva Bahu

d. Representasi Kurva S

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 2.14

menunjukkan karakteristik kurva-S dalam bentuk skema.



Gambar 2.14 Representasi Kurva S

Fungsi keanggotaan:

$$s(x; \alpha, \beta, \gamma) \begin{cases} 0; & x \leq \alpha \\ 2 \left(\frac{x - \alpha}{\beta - \alpha} \right)^2; & \alpha \leq x \leq \beta \\ 1 - 2 \left(\frac{\gamma - \beta}{\gamma - \alpha} \right)^2; & \beta \leq x \leq \gamma \\ 1; & x \geq \gamma \end{cases}$$

2.4.4 Fungsi Implikasi

Tiap-tiap aturan (proposisi) pada basis pengetahuan fuzzy akan berhubungan dengan suatu relasi fuzzy. Bentuk umum dari aturan yang digunakan dalam fungsi implikasi adalah:

IF x is A THEN y is B

dengan x dan y adalah skalar, dan A dan B adalah himpunan fuzzy. Proposisi

yang mengikuti IF disebut sebagai anteseden, sedangkan proposisi yang mengikuti THEN disebut sebagai konsekuen. Proposisi ini dapat diperluas dengan menggunakan operator fuzzy, seperti:

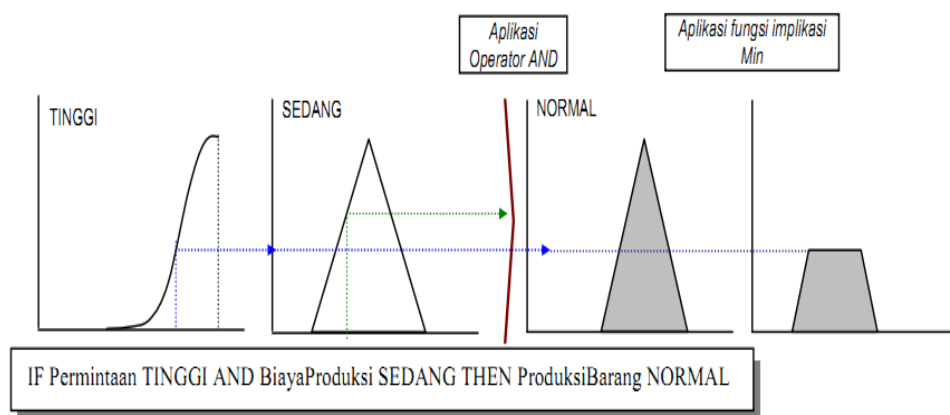
$$IF (x_1 \text{ is } A_1) \bullet (x_2 \text{ is } A_2) \bullet \dots \bullet (x_n \text{ is } A_n) \text{ Then } y \text{ is } B$$

Dengan \bullet adalah operator (misal: *OR* atau *AND*).

Secara umum, ada 2 fungsi implikasi yang dapat digunakan, yaitu:

1. Min (*minimum*).

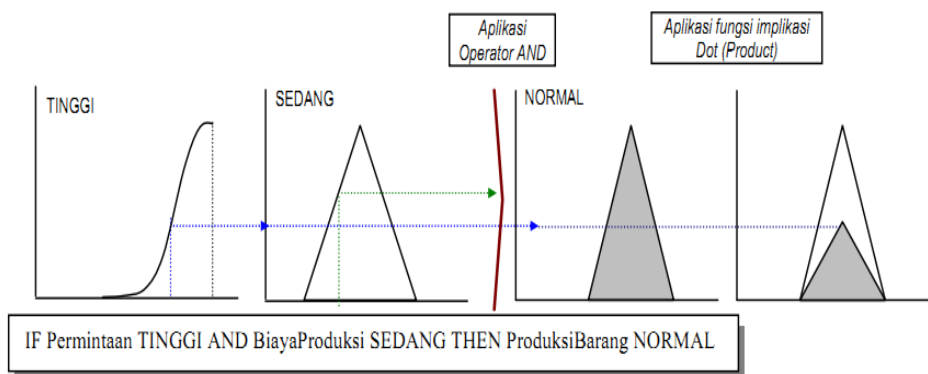
Fungsi ini akan memotong output himpunan fuzzy. Gambar 2.15 menunjukkan salah satu contoh penggunaan fungsi min.



Gambar 2.15 Fungsi Implikasi: *MIN* (Kusumadewi, 2003: 180)

2. Dot (*product*).

Fungsi ini akan menskala output himpunan fuzzy. Gambar 2.16 menunjukkan salah satu contoh penggunaan fungsi dot.



Gambar 2.16 Fungsi Implikasi: *DOT* (Kusumadewi, 2003:180)

2.4.5 Sistem Inferensi Fuzzy

Terdapat 3 metode pada sistem inferensi fuzzy yaitu: metode Tsukamoto, metode Mamdani dan metode Sugeno. Dalam skripsi ini metode yang dipakai adalah metode Mamdani yang akan dijelaskan sebagai berikut:

2.4.5.1 Metode Mamdani

Metode Mamdani sering juga dikenal dengan nama Metode Max-Min. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Untuk mendapatkan output, diperlukan 4 tahapan:

1. Pembentukan himpunan fuzzy

Pada Metode Mamdani, baik variabel input maupun variabel output dibagi menjadi satu atau lebih himpunan fuzzy.

2. Aplikasi fungsi implikasi

Pada Metode Mamdani, fungsi implikasi yang digunakan adalah Min.

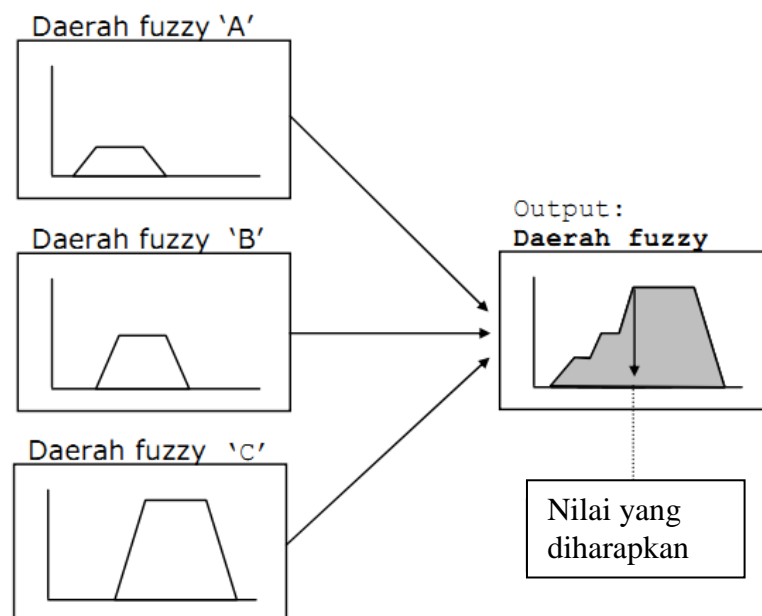
3. Komposisi aturan

Tidak seperti penalaran monoton, apabila sistem terdiri-dari beberapa aturan, maka 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).

4. Penegasan (defuzzifikasi)

Input dari proses defuzzifikasi 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 dalam range tertentu, maka harus dapat diambil suatu nilai crisp tertentu sebagai output seperti terlihat pada Gambar 2.17.



Gambar 2.17 Proses Defuzzifikasi

2.5 Algoritma Fuzzy Evolusi

Algoritma fuzzy evolusi adalah sebuah teknik komputasi gabungan antara algoritma genetika dan logika fuzzy. Metode ini hampir sama dengan metode

algoritma genetika, namun parameter-parameter yang dipakai dihasilkan dari sebuah sistem fuzzy. Dalam algoritma fuzzy evolusi, proses yang terjadi atau alur proses sama seperti dengan algoritma genetika, yang dikenalkan oleh John Holland dari Universitas Michigan (1975), dimana algoritma genetika merupakan teknik pencarian heuristik berdasar mekanisme evolusi biologis yang meniru dari teori Darwin dan operasi genetika pada kromosom. (Bindu & Tanwar, 2012:418) dari pada memilih nilai acak dari orang tua, aturan fuzzy didefinisikan untuk memilih aturan yang optimal. Sistem yang diusulkan adalah untuk mengoptimalkan proses hasil dari algoritma genetika dalam kasus DPX pindah silang. Dalam algoritma fuzzy evolusi terdapat enam tahap utama, yaitu:

1. Representasi kromosom.
2. Inisialisasi Populasi.
3. Fungsi evaluasi.
4. Seleksi.
5. Operator genetika, meliputi operator rekombinasi (*crossover*) dan mutasi.
6. Penentuan parameter, yaitu parameter kontrol algoritma genetika, yaitu: ukuran populasi (*popsize*), peluang *crossover* (P_c), dan peluang mutasi (p_m).
Dalam penentuan parameter ini dilakukan proses sistem fuzzy untuk mendapatkan nilai yang akan digunakan sebagai parameter.

2.6 Algoritma Fuzzy Evolusi Xu

Pada algoritma fuzzy evolusi terdapat berbagai metode yang digunakan dalam perpaduan antara algoritma genetika dan sistem fuzzy. Diantaranya adalah

model yang dikemukakan oleh Xu, yaitu algoritma fuzzy evolusi dengan sistem fuzzy digunakan pada penentuan parameter adalah menggunakan sistem inferensi fuzzy metode mamdani. Metode Mamdani ini juga dikenal dengan metode Max-Min. Metode ini dikenalkan oleh Ebrahim Mamdani (1975). Metode Mamdani yang digunakan oleh Xu untuk algoritma fuzzy evolusi adalah menggunakan dua buah masukan dan dua buah keluaran. Dua buah masukan yang digunakan adalah:

1. Jumlah Populasi yang digunakan.
2. Jumlah generasi yang akan diproses.

Sedangkan dua buah keluaran yang akan dihasilkan adalah:

1. Nilai probabilitas rekombinasi (*crossover*).
2. Nilai probabilitas mutasi.

Dalam menentukan nilai yang dihasilkan melalui sistem fuzzy perlu dibuat aturan-aturan fuzzy yang digunakan untuk penentuan hasil. Dalam model Xu aturan fuzzy yang digunakan didasarkan dari masukan jumlah populasi yang diinginkan serta jumlah maksimum generasi. Dari dua masukan tersebut akan menghasilkan nilai untuk probabilitas rekombinasi dan probabilitas mutasi. Aturan yang ditentukan oleh Xu dapat dilihat dalam Tabel 2.2 dan 2.3.

Tabel 2.1 Aturan untuk Nilai Probabilitas Rekombinasi (*Crossover*)

Pm	<i>Population size</i>		
<i>Generation</i>	<i>Small</i>	<i>Medium</i>	<i>Large</i>
<i>Short</i>	<i>Medium</i>	<i>Small</i>	<i>Small</i>
<i>Medium</i>	<i>Large</i>	<i>Large</i>	<i>Medium</i>
<i>Long</i>	<i>Very Large</i>	<i>Very Large</i>	<i>Large</i>

Tabel 2.2 Aturan untuk Nilai Probabilitas Mutasi

Pm	Population size		
Generation	<i>Small</i>	<i>Medium</i>	<i>Large</i>
<i>Short</i>	<i>Large</i>	<i>Medium</i>	<i>Small</i>
<i>Medium</i>	<i>Medium</i>	<i>Small</i>	<i>Very Small</i>
<i>Long</i>	<i>Small</i>	<i>Very Small</i>	<i>Very Small</i>

Dengan aturan pada Tabel 2.1 dan 2.2, jumlah populasi dan maksimum generasi yang dimasukkan akan dianalisa dengan metode Mamdani, sehingga didapatkan nilai probabilitas *crossover* dan probabilitas mutasi yang mana akan dipakai dalam iterasi. Dalam algoritma fuzzy evolusi, aturan-aturan fuzzy yang telah dibuat harus sudah diimplementasikan terlebih dahulu sebelum proses iterasi dilakukan. Dari model Xu yang digunakan sesuai Tabel 2.2 dan 2.3 didapatkan sebanyak sembilan aturan, yaitu:

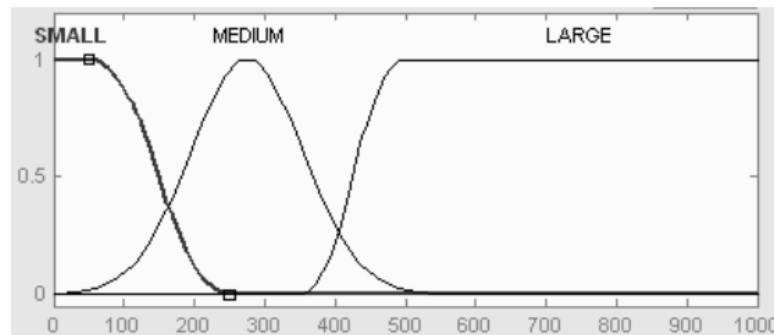
- *IF (Populasi is SMALL) AND (Generasi is SHORT)*
THEN (ProbCrossover is MEDIUM) AND (ProbMutasi is LARGE).
- *IF (Populasi is MEDIUM) AND (Generasi is SHORT)*
THEN (ProbCrossover is SMALL) AND (ProbMutasi is MEDIUM).
- *IF (Populasi is LARGE) AND (Generasi is SHORT)*
THEN (ProbCrossover is SMALL) AND (ProbMutasi is SMALL).
- *IF (Populasi is SMALL) AND (Generasi is MEDIUM)*
THEN (ProbCrossover is LARGE) AND (ProbMutasi is MEDIUM).

- *IF (Populasi is MEDIUM) AND (Generasi is MEDIUM)
THEN (ProbCrossover is LARGE) AND (ProbMutasi is SMALL).*
- *IF (Populasi is LARGE) AND (Generasi is MEDIUM)
THEN (ProbCrossover is MEDIUM) AND (ProbMutasi is VERYSMALL).*
- *IF (Populasi is SMALL) AND (Generasi is LONG)
THEN (ProbCrossover is VERYLARGE) AND (ProbMutasi is SMALL).*
- *IF (Populasi is MEDIUM) AND (Generasi is LONG)
THEN (ProbCrossover is VERYLARGE) AND (ProbMutasi is
VERYSMALL).*
- *IF (Populasi is LARGE) AND (Generasi is LONG)
THEN (ProbCrossover is LARGE) AND (ProbMutasi is VERYSMALL).*

Aturan-aturan yang dikembangkan oleh Xu diimplementasikan dalam sistem fuzzy mamdani, tetapi perlu diperhatikan supaya sistem fuzzy mamdani dapat menghasilkan hasil tentunya diperlukan semesta pembicaraan dan domain yang memberikan nilai batas untuk setiap himpunan yang ada pada tiap variabel. Misal nilai untuk semesta pembicaraan pada variabel populasi adalah [0 1000], yang berarti dalam variabel populasi memiliki batas semesta pembicaraan mulai

batas nilai nol (0) sampai nilai seribu (1000). Sedangkan misal domain untuk himpunan SMALL pada variabel populasi adalah [50 250], yang berarti batas populasi dikatakan SMALL jika bernilai antara lima puluh (50) dan dua ratus lima puluh (250). Adapun semesta pembicaraan dan domain yang digunakan dalam model Xu, ditentukan oleh peneliti karena hal ini belum ditemukan studi literatur yang menjelaskan tentang hal ini. Gambar 2.18 sampai dengan 2.21

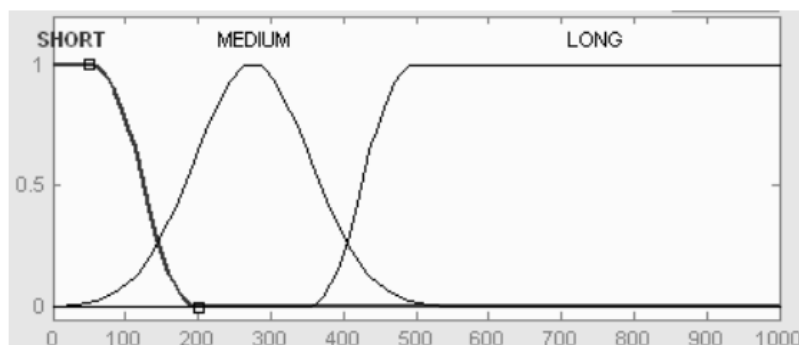
adalah gambar yang menjelaskan tentang semesta pembicaraan dan domain yang digunakan peneliti:



Gambar 2.18 Semesta Pembicaraan dan Domain untuk Variabel Populasi (Muzid, 2008:35)

Pada semesta pembicaraan dan domain untuk populasi, aturan nilai yang digunakan adalah sebagai berikut:

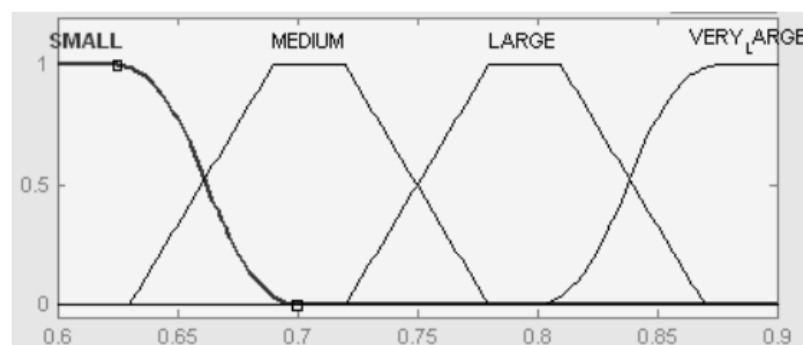
- Semesta pembicaraan: [0 1000]
- Domain SMALL: [50 250]
- Domain Medium: [80 275]
- Domain LARGE: [350 500]



Gambar 2.19 Semesta Pembicaraan dan Domain untuk Variabel Generasi (Muzid, 2008:35)

kemudian untuk semesta pembicaraan dan domain untuk variable generasi, aturan nilai yang digunakan adalah sebagai berikut:

- Semesta pembicaraan: [0 1000]
- Domain SHORT: [50 200]
- Domain MEDIUM: [80 275]
- Domain LONG: [350 500]

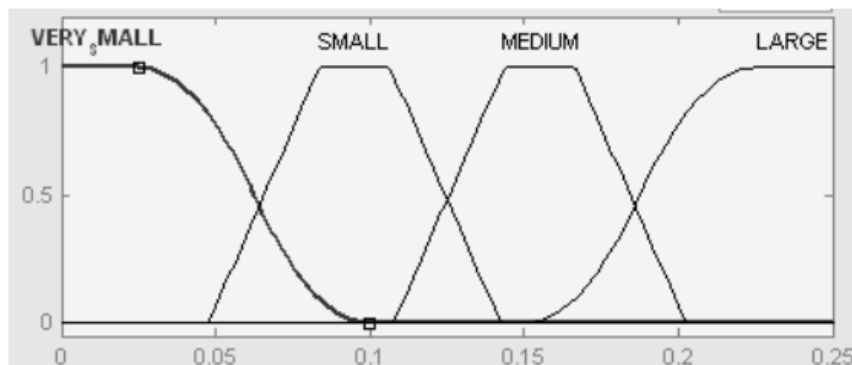


Gambar 2.20 Semesta Pembicaraan dan Domain

untuk Variabel Probabilitas *Crossover* (Muzid, 2008:35)

Pada umumnya probabilitas untuk *crossover* adalah antara 0.6 sampai 0.9. sehingga pada semesta pembicaraan dan domain untuk hasil output yaitu nilai probabilitas crossover, aturan nilai yang digunakan adalah sebagai berikut:

- Semesta pembicaraan: [0.6 0.9]
- Domain SMALL: [0.625 0.7]
- Domain MEDIUM: [0.63 0.7 0.72 0.78]
- Domain LARGE: [0.72 0.78 0.8 0.87]
- Domain VARY LARGE: [0.8 0.875]



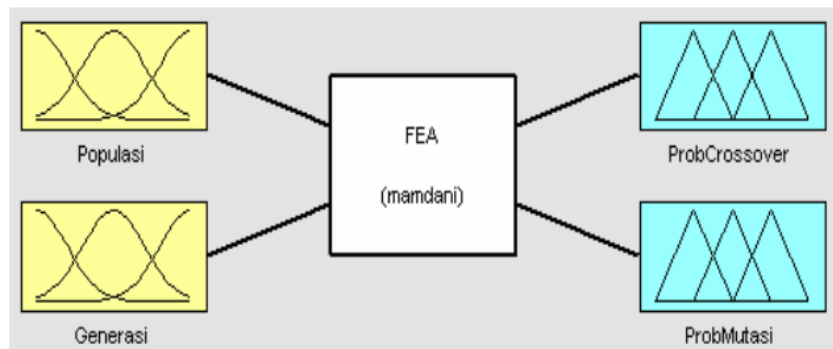
Gambar 2.21 Semesta Pembicaraan dan Domain

untuk Variabel Probabilitas *Mutasi* (Muzid, 2008:36)

Sedangkan untuk probabilitas mutasi pada umumnya sangat kecil, sekitar 1 dibagi dengan jumlah gen yang digunakan. Artinya peluang mutasi hanya terjadi pada kisaran satu gen saja pada tiap individu atau dengan kata lain probabilitas mutasi mendekati nol (0). Sehingga pada semesta pembicaraan dan domain untuk nilai probabilitas mutasi, aturan nilai yang digunakan adalah sebagai berikut.

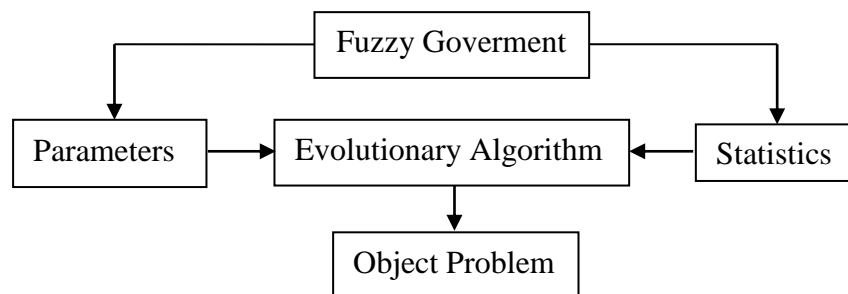
- Semesta pembicaraan: [0 0.25]
- Domain VERY SMALL: [0.025 0.1]
- Domain SMALL: [0.047 0.083 0.1 0.14]
- Domain MEDIUM: [0.1 0.14 0.167 0.2]
- Domain LARGE: [0.15 0.225]

Gambar 2.22 adalah gambar proses sistem fuzzy Mamdani yang digunakan pada penentuan nilai fuzzy untuk parameter probabilitas crossover dan mutasi pada algoritma fuzzy evolusi.



Gambar 2.22 Alur Proses Sistem Fuzzy Mamdani (Muzid, 2008:36)

2.7 Arsitektur Algoritma Fuzzy Evolusi



Gambar 2.23 Arsitektur Algoritma Fuzzy Evolusi

Dari Gambar 2.23 dapat dilihat bahwa proses yang digunakan dalam algoritma fuzzy evolusi bahwa nilai statistik dari populasi dan generasi yang ada dimasukkan dalam proses fuzzy sehingga menghasilkan parameter yang kemudian akan digunakan dalam proses algoritma genetika sehingga akan menghasilkan keluaran akhir (Kusumadewi, 2007:90).

2.8 MATLAB

Matlab merupakan bahasa pemrograman yang hadir dengan fungsi dan karakteristik yang berbeda dengan bahasa pemrograman lain yang sudah ada lebih dahulu seperti Delphi, Basic maupun C++. MATLAB merupakan bahasa pemrograman level tinggi yang dikhususkan untuk kebutuhan komputasi teknis, visualisasi dan pemrograman seperti komputasi matematik, analisis data, pengembangan algoritma, simulasi dan pemodelan dan grafik-grafik perhitungan.

MATLAB hadir dengan membawa warna yang berbeda. Hal ini karena MATLAB membawa keistimewaan dalam fungsi-fungsi matematika, fisika, statistik, dan visualisasi. MATLAB dikembangkan oleh MathWorks, yang pada awalnya dibuat untuk memberikan kemudahan mengakses data matrik pada proyek LINPACK dan EISPACK. Saat ini MATLAB memiliki ratusan fungsi yang dapat digunakan sebagai *problem solver* mulai dari simple sampai masalah-masalah yang kompleks dari berbagai disiplin ilmu (Firmansyah, 2007).

2.8.1. Beberapa Bagian dari *Window* MATLAB

Adapun beberapa bagian dari *window* yang terdapat dalam program MATLAB meliputi:

1. *Current Directory*

Bagian dari *window* ini menampilkan isi dari direktori kerja saat menggunakan MATLAB.

2. *Command History*

Bagian ini berfungsi untuk menyimpan perintah-perintah apa saja yang sebelumnya dilakukan oleh pengguna terhadap MATLAB.

3. *Command Window*

Bagian ini merupakan tempat untuk menjalankan fungsi, variabel, mendeklarasikan variabel, menjalankan proses-proses, serta melihat isi variabel.

4. *Workspace*

Bagian ini berfungsi untuk menampilkan seluruh variabel-variabel yang sedang aktif pada saat pemakaian MATLAB (Firmansyah, 2007).

2.8.2. Meminta Bantuan

MATLAB menyediakan fungsi *help* yang berisikan tutorial lengkap mengenai MATLAB dan segala keunggulannya. *User* dapat menjalankan fungsi ini dengan menekan tombol pada *toolbar* atau menulis perintah '*helpwin*' pada *command window*. MATLAB juga menyediakan fungsi *demos* yang berisikan video tutorial MATLAB serta contoh-contoh program yang bisa dibuat dengan MATLAB (Firmansyah, 2007).

2.8.3. Interrupting dan Terminating dalam MATLAB

Untuk menghentikan proses yang sedang berjalan pada MATLAB dapat dilakukan dengan menekan tombol *Ctrl+C*. Sedangkan untuk keluar dari MATLAB dapat dilakukan dengan menuliskan perintah *exit* atau *quit* pada *command window* atau dengan menekan menu *exit* pada bagian menu *file* dari *menu bar* (Firmansyah, 2007).

2.8.4. Variabel pada MATLAB

MATLAB hanya memiliki dua jenis tipe data yaitu *numeric* dan *string*. Dalam MATLAB setiap variabel akan disimpan dalam bentuk matrik. *User* dapat

langsung menuliskan variabel baru tanpa harus mendeklarasikannya terlebih dahulu pada *command window* (Firmansyah, 2007).

BAB 3

METODE PENELITIAN

Metode penelitian merupakan suatu cara yang digunakan dalam penelitian sehingga pelaksanaan penelitian dapat dipertanggungjawabkan secara ilmiah. Dengan metode penelitian data yang diperoleh semakin lengkap untuk memecahkan masalah yang dihadapi. Pada penelitian ini langkah-langkah yang dilakukan adalah sebagai berikut.

3.1 Menemukan Masalah

Dalam tahap ini dicari sumber pustaka dan dipilih dari sumber pustaka suatu masalah. Untuk lebih memperjelas pembahasan, maka dipilih suatu kasus yang terjadi di suatu perusahaan yang berkaitan langsung dengan permasalahan yang akan diangkat.

3.2 Merumuskan masalah

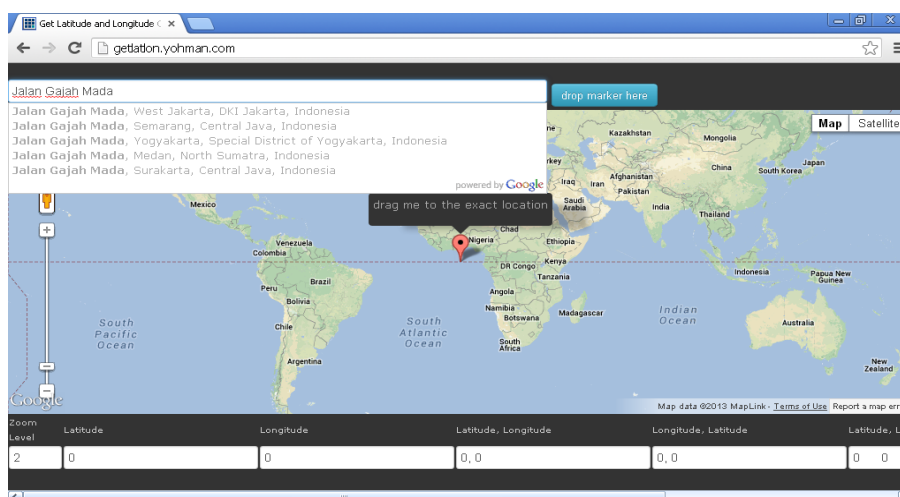
Masalah yang ditemukan kemudian dirumuskan kedalam pertanyaan yang harus diselesaikan sebagai berikut.

1. Bagaimana rute optimal jaringan TSP yang mempunyai jarak minimum dalam pengiriman barang dengan menggunakan algoritma Fuzzy Evolusi di PT. Jalur Nugraha Ekakurir (JNE) Semarang?
2. Bagaimana hasil pencarian jarak minimum dari jaringan TSP dalam pengiriman barang di PT. Jalur Nugraha Ekakurir (JNE) Semarang menggunakan algoritma Fuzzy Evolusi?

3.3 Pengambilan Data

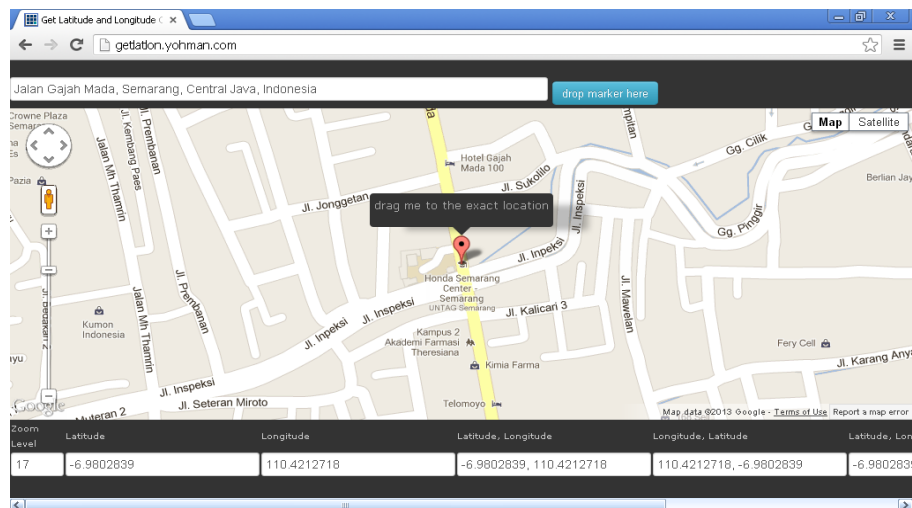
Dalam penelitian ini, penulis memperoleh data dari PT. Jalur Nugraha Ekakurir (JNE) Semarang yang kemudian akan dilakukan pengolahan. Data ini berupa data pengiriman barang oleh kurir dari PT. Jalur Nugraha Ekakurir (JNE) Semarang beserta alamatnya. Untuk memperoleh data jarak antar lokasi dilakukan proses pencarian jarak yang diambil dari Google Maps melalui situs <http://getlatlon.yohman.com>. Metode ini dilakukan karena dengan cara ini akan didapatkan titik koordinat yaitu garis lintang (*latitude*) dan garis bujur (*longitude*) antar lokasi secara lebih akurat tanpa harus mengeluarkan banyak waktu dan biaya dalam pencariannya. Adapun langkah-langkahnya adalah sebagai berikut.

- Membuka situs <http://getlatlon.yohman.com/>
- Pada menu **drop here** ketik semarang lalu pilih semarang, Central Java, Indonesia. Di Bagian Pencarian atau *search* ketik kata kunci yang berhubungan dengan Kota Semarang, misal Jalan Gajah Mada, maka akan muncul seperti Gambar 3.1.



Gambar 3.1 Tampilan Getlatlon

- Setelah muncul semua informasi yang berhubungan dengan Jalan Gajah Mada, selanjutnya pilih alamat yang diinginkan kemudian klik tombol **drop marker here**.



Gambar 3.2 Hasil Pencarian Tempat

- d. Catat hasil latitude dan longitude pada menu yang tertera di bawah peta yaitu -6.9802839 dan 110.4212718.

3.4 Analisis dan Pemecahan Masalah

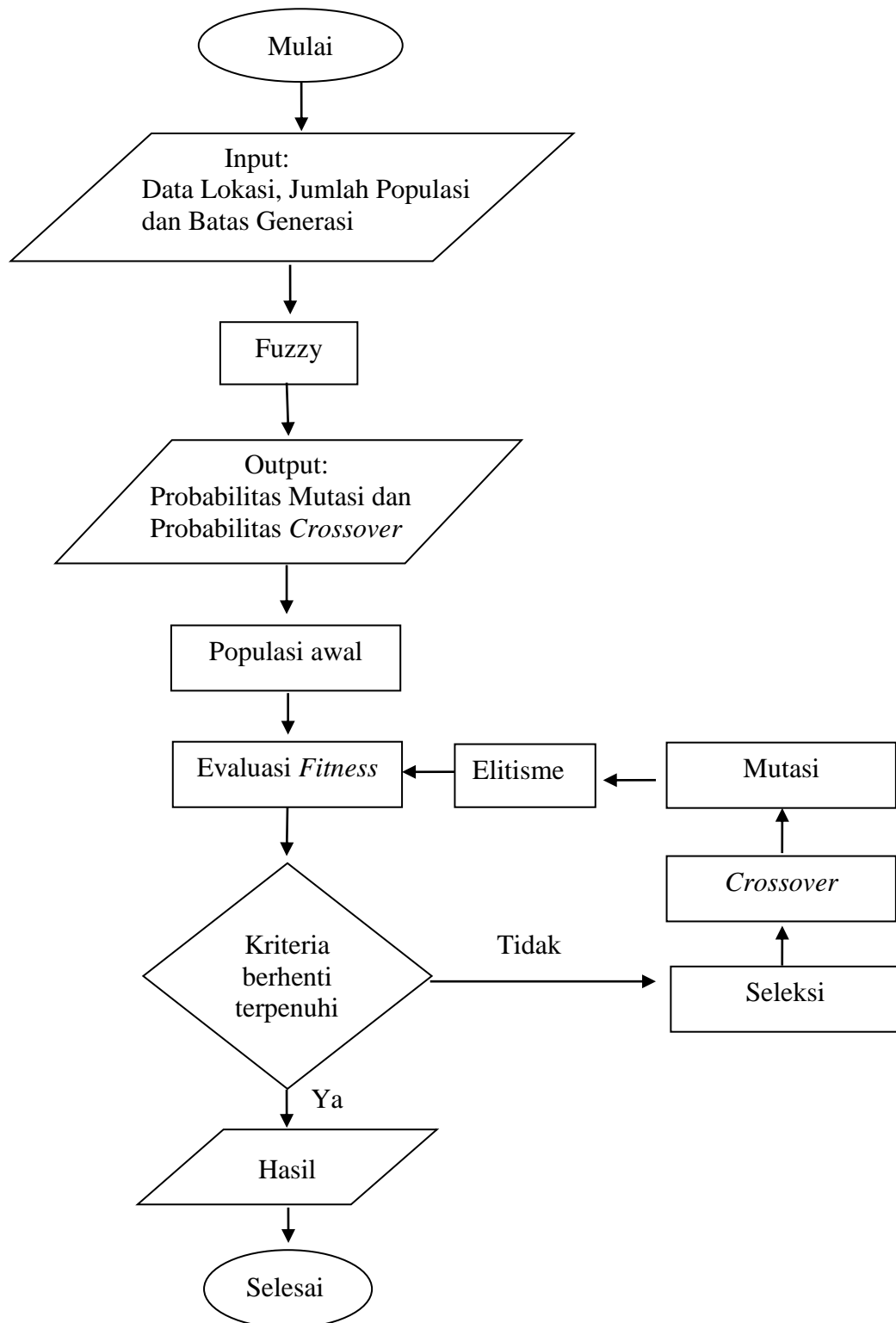
Sebelum analisis dan pemecahan masalah, penulis memiliki beberapa asumsi dalam penelitian ini adalah sebagai berikut.

1. Jalur yang dilewati melalui jalan yang rata.
2. Jalur pengiriman melewati jalan Negara, Propinsi, maupun jalan Kota.

Dari berbagai sumber yang sudah menjadi bahan kajian, diperoleh suatu pemecahan masalah di atas. Selanjutnya dilakukan langkah-langkah pemecahan masalah sebagai berikut:

- a. Pembentukan model

Menyajikan titik-titik yang harus dilalui dalam jaringan TSP berdasarkan data perusahaan beserta jarak antar titiknya. Kemudian masukkan data ke dalam program yang telah disiapkan. Gambar 3.3 menjelaskan alur kerja program.

Gambar 3.3 *Flow Chart* Rancangan Sistem

b. Alur proses algoritma fuzzy evolusi

1. Tahap pemasukkan data

Merupakan pemasukkan data berupa data titik koordinat yang diperoleh dari getlatlon.yohman.com berupa longitude dan latitude, kemudian memasukan jumlah populasi dan batas generasi.

2. Tahap proses fuzzy

Proses ini merupakan penentuan nilai probabilitas mutasi dan probabilitas *crossover* dari masukan jumlah populasi dan batas generasi yang akan diproses menggunakan system inferensi fuzzy metode Mamdani sehingga diperoleh nilai probabilitas mutasi dan probabilitas *crossover*.

3. Tahap proses populasi awal/ inialisasi populasi

Proses ini merupakan proses yang digunakan untuk membangkitkan populasi awal secara random sehingga didapatkan solusi awal.

4. Tahap evaluasi *fitness*

Proses ini merupakan proses untuk mengevaluasi setiap populasi dengan menghitung nilai *fitness* setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti.

5. Tahap kriteria berhenti

Kriteria berhenti terpenuhi bila telah mencapai batas generasi yang telah ditentukan. Apabila belum mencapai batas generasi maka dilakukan tahap seleksi.

6. Tahap seleksi

Proses ini merupakan proses untuk menentukan individu mana saja yang akan dipilih untuk dijadikan *crossover*. Proses seleksi yang digunakan menggunakan metode *roulette-wheel selection*.

7. Tahap *crossover*

Proses ini merupakan proses untuk menambah keanekaragaman suatu populasi dengan cara memindah silangkan dua buah kromosom. Proses *crossover* yang digunakan menggunakan metode *order crossover* yaitu

merupakan cara *crossover* dengan menukar kromosom dengan tetap menjaga urutan gen yang bukan bagian dari kromosom tersebut.

8. Tahap mutasi

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam satu kromosom. Proses mutasi yang digunakan menggunakan metode swapping mutation yaitu probabilitas mutasi telah ditentukan

9. Tahap *elitisme*

Merupakan tahapan untuk menjaga agar individu bernilai fitness tertinggi tidak hilang selama proses evolusi.

10. Perolehan hasil

Hasil yang diperoleh berupa rute jalur, panjang jalur terbaik, nilai fitness tertinggi dan waktu eksekusi.

c. Mencari penyelesaian masalah

Pada tahap ini dilakukan pencarian rute optimal dan jarak minimal yang dapat ditempuh dalam pengiriman barang dengan syarat semua alamat dilalui tepat satu kali kecuali titik asal yang sama dengan titik akhir. Setelah diketahui jarak antara titik menggunakan Getlatlon, akan dicari hasil perhitungan rute optimal dan jarak minimal dari jaringan TSP beserta gambar rute tersebut. Proses ini memerlukan ketelitian yang tinggi karena jika terjadi suatu kesalahan kecil saja akan berakibat pada ketidaktepatan dalam perhitungan rute dan jarak dari jaringan TSP terbaik. Masalah minimasi ini akan dicari dengan menggunakan algoritma Fuzzy Evolusi.

3.5 Penarikan Simpulan

Langkah terakhir dalam metode penelitian adalah penarikan simpulan yang diperoleh dari hasil langkah pemecahan masalah.

BAB 4

HASIL PENELITIAN DAN PEMBAHASAN

4.1 Hasil Penelitian

Penelitian ini mengkaji tentang pengiriman barang di PT. Jalur Nugraha Ekakurir (JNE) Semarang dengan permasalahannya yaitu menentukan rute jaringan *travelling salesman problem* (TSP) terbaik dengan jarak pendistribusian terkecil dengan algoritma fuzzy evolusi menggunakan aplikasi yang telah dibuat dengan bantuan Matlab.

Dalam penelitian ini yang akan dicari adalah panjang rute yang dilalui untuk pendistribusian barang yang akan dikirimkan JNE menuju para penerima barang yang berada di wilayah Kota Semarang. Permasalahan TSP pada penelitian ini bukanlah masalah TSP murni, karena masih terdapat beberapa jalan yang dilewati lebih dari satu kali. Hal ini dikarenakan tidak adanya jalan lain yang bisa dipilih untuk melanjutkan pendistribusikan barang dari rumah penerima satu ke penerima selanjutnya.

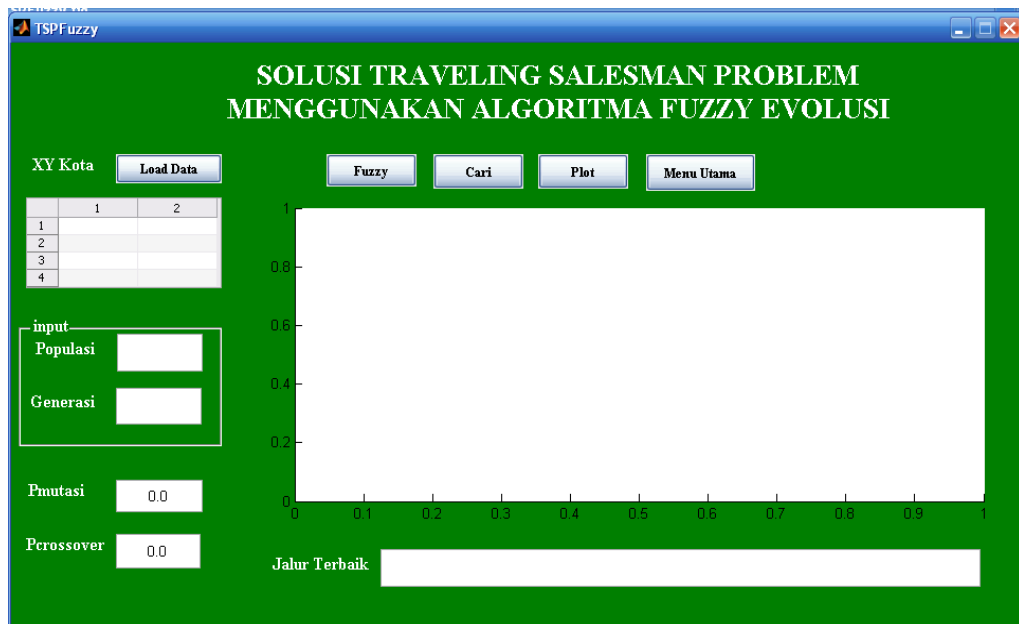
Penulis memperoleh data dari PT. Jalur Nugraha Ekakurir Semarang berupa list nama penerima beserta alamat lengkapnya seperti yang tersaji dalam Lampiran 1, kemudian dilakukan proses pencarian koordinat titik dengan bantuan situs GetLatlon.Yohman.com yang sudah terintegrasi dengan *Google Maps*. Situs GetLatlon.Yohman.com merupakan situs pencari koordinat lokasi di bumi, dengan sumbu horisontal X adalah garis bujur (Longitude), sedangkan sumbu

vertikal Y merupakan garis lintang (latitude) yang berjalan melalui Observatorium Greenwich di Inggris.

Koordinat semua titik dalam pendistribusian barang menuju rumah penerima barang yang telah diberikan oleh PT. Jalur Nugraha Ekakurir Semarang disajikan pada Lampiran 2. Dari koordinat yang telah diketahui pada Lampiran 2, kemudian dapat dicari jarak antar lokasi dalam penelitian ini. Dalam penelitian ini, perhitungan jarak antar lokasi dilakukan dengan bantuan Google Maps yang telah menyediakan fasilitas berupa pengukuran jarak. Hasil perhitungan jarak antar semua lokasi terlampir pada Lampiran 3.

4.1.1 Implementasi Program

Setelah perangkat lunak kajian Algoritma Fuzzy Evolusi pada *Travelling Salesman Problem* selesai dibangun, maka tahap selanjutnya adalah tahap uji coba program. Tahap uji coba tampilan adalah tahap pengujian dengan menjalankan program *Travelling Salesman Problem* yang sebagai masukan adalah titik koordinat tempat tujuan, jarak antar lokasi tempat tujuan, jumlah populasi dan batas generasi yang akan diproses. Dalam perangkat yang telah dibuat, terdapat beberapa tampilan antara lain: tampilan menu utama, tampilan TSP, tampilan ABOUT dan tampilan Help. Hasil pada tampilan menu utama, tampilan About dan tampilan Help dapat dilihat pada Lampiran 4. Untuk coding pada matlab dapat dilihat pada Lampiran 5. Tampilan TSP dapat dilihat pada Gambar 4.1.



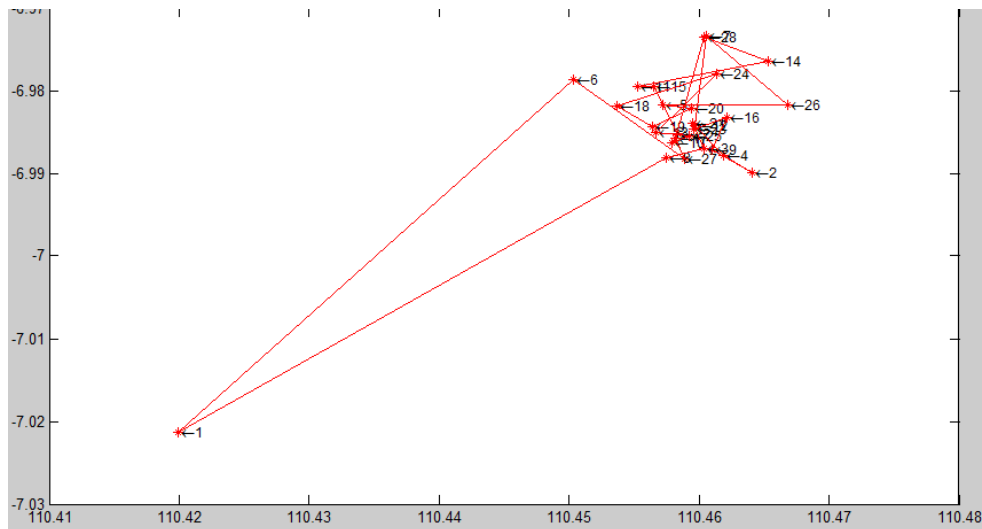
Gambar 4.1 Tampilan TSP

Pada tampilan TSP yang ada pada Gambar 4.1 berguna untuk melakukan proses pencarian rute menggunakan algoritma fuzzy evolusi dengan memasukkan data koordinat alamat dituju yang sebelumnya telah dimasukkan ke dalam excel, jumlah populasi dan batas generasi. Kemudian terdapat beberapa tombol beserta fungsinya antara lain:

1. Tombol **Load Data** (berfungsi untuk memasukkan data koordinat alamat dituju yang sebelumnya telah dimasukkan ke dalam excel).
2. Tombol **Fuzzy** (berfungsi untuk memberi keluaran berupa **Pmutasi** dan **Pcrossover** setelah menginputkan **Populasi** dan **Generasi**).
3. Tombol **Cari** (berfungsi untuk melakukan proses perhitungan menggunakan algoritma fuzzy evolusi).
4. Tombol **Plot** (berfungsi untuk menampilkan grafik koordinat kota/alamat yang akan dilalui setelah melakukan proses perhitungan).

5. Tombol **Menu Utama** (berfungsi untuk kembali pada tampilan menu utama).

Berikut merupakan grafik koordinat kota/alamat yang dituju dapat dilihat pada Gambar 4.2.



Gambar 4.2 Tampilan Koordinat Kota atau Alamat Dituju

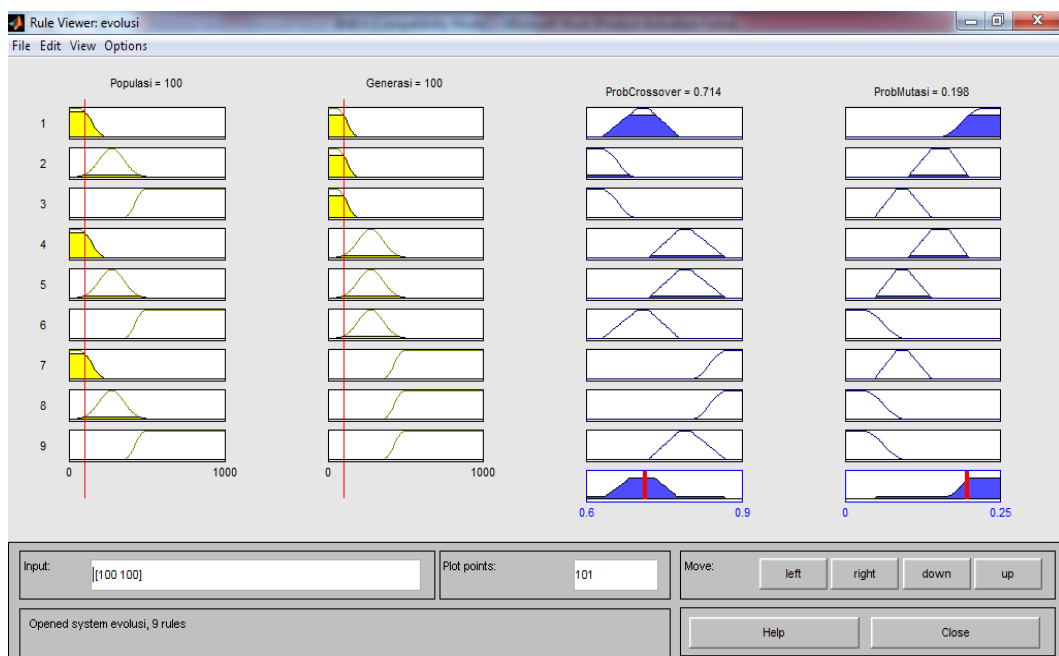
4.1.2 Hasil Simulasi Program

Perangkat lunak yang telah dirancang memerlukan pengujian data dengan melakukan proses pencarian rute dengan variasi jumlah populasi dan batas generasi yaitu: (100 dan 100), (100 dan 200), (100 dan 500), (100 dan 1000), (200 dan 100), (500 dan 100) dan (1000 dan 100). Kemudian dilakukan proses perhitungan sebanyak 10 kali dan diambil hasil jalur terbaik *minimum*.

4.1.2.1 Perhitungan menggunakan masukan populasi 100 dan generasi 100.

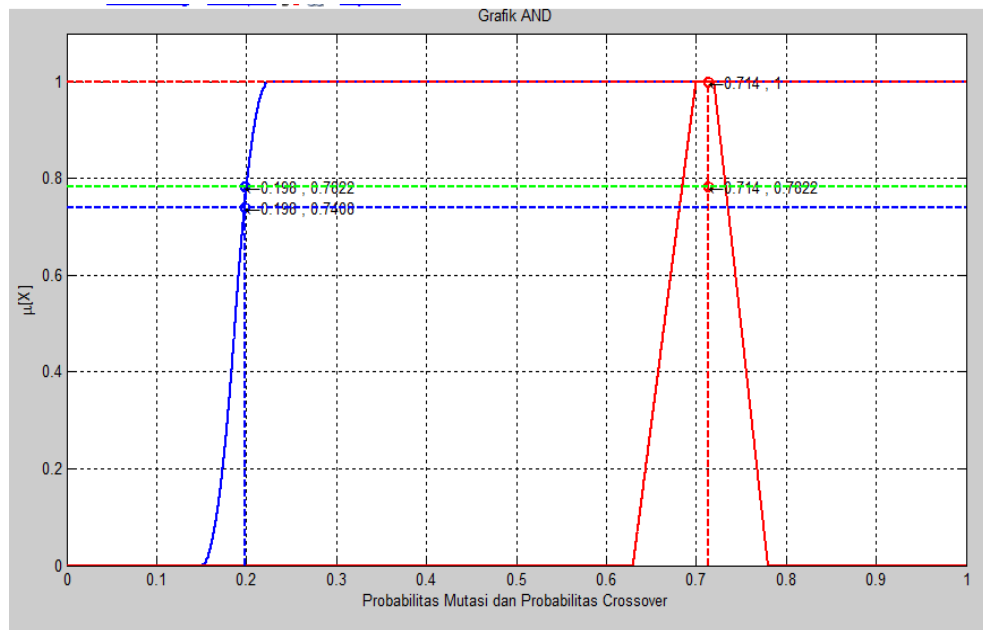
Untuk memulai perhitungan menggunakan perangkat lunak yang telah disediakan perlu melakukan langkah-langkah pemakaiannya antara lain:

1. Klik tombol **Load Data** kemudian pilih file excel yang sebelumnya telah dimasukkan data koordinat alamat tujuan.
2. Masukkan jumlah populasi pada **Populasi**= 100 dan batas generasi (**Generasi**) = 100.
3. Klik tombol **Fuzzy** maka pada **Pmutasi** dan **Pcrossover** akan muncul nilai 0,197845 dan 0,714007. Nilai Pmutasi dan Pcrossover dihasilkan melalui sistem fuzzy Mamdani dengan memasukkan populasi 100 dan generasi 100 dan menggunakan aturan fuzzy serta nilai fungsi keanggotaan fuzzy yang telah dijelaskan pada Lampiran 6. Gambar 4.3 menjelaskan proses kerja pencarian probabilitas mutasi dan probabilitas *crossover* menggunakan fuzzy mamdani.



Gambar 4.3 Hasil Pencarian Probabilitas Mutasi dan Probabilitas *Crossover*

Kemudian untuk nilai keanggotaan fuzzy pada probabilitas mutasi dan probabilitas crossover dapat dijelaskan pada Gambar 4.4.



Gambar 4.4 Grafik Fungsi AND

Gambar 4.4 menjelaskan bahwa dengan menggunakan hasil operasi logika fuzzy AND diperoleh

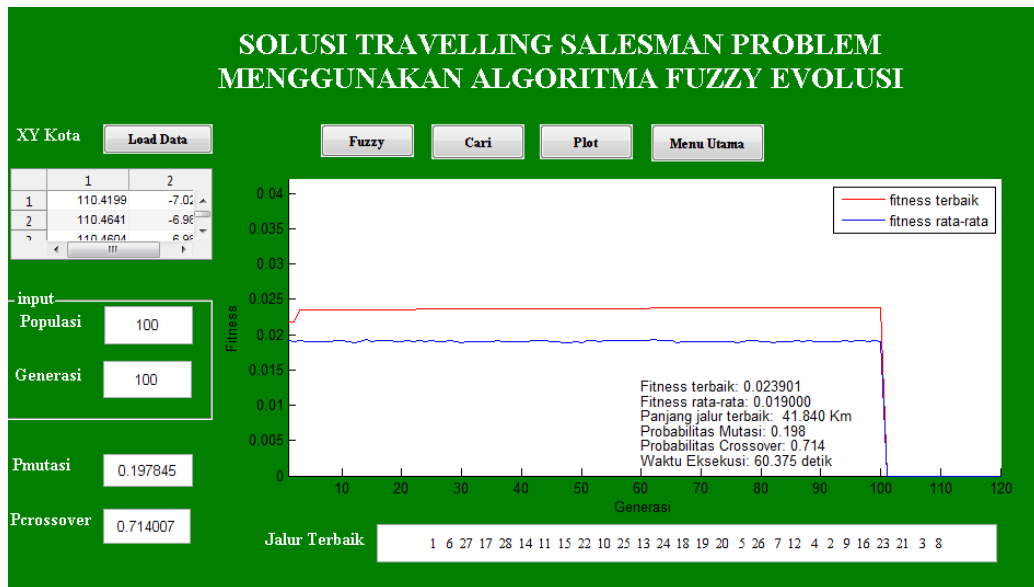
$$\min(\mu(P_{mutasi} = \text{VERY LARGE}), \mu(P_{crossover} = \text{medium}),$$

$$\mu(\text{populasi dan generasi}))$$

$$= \min(0.7408, 1, 0.7822) = 0.7408$$

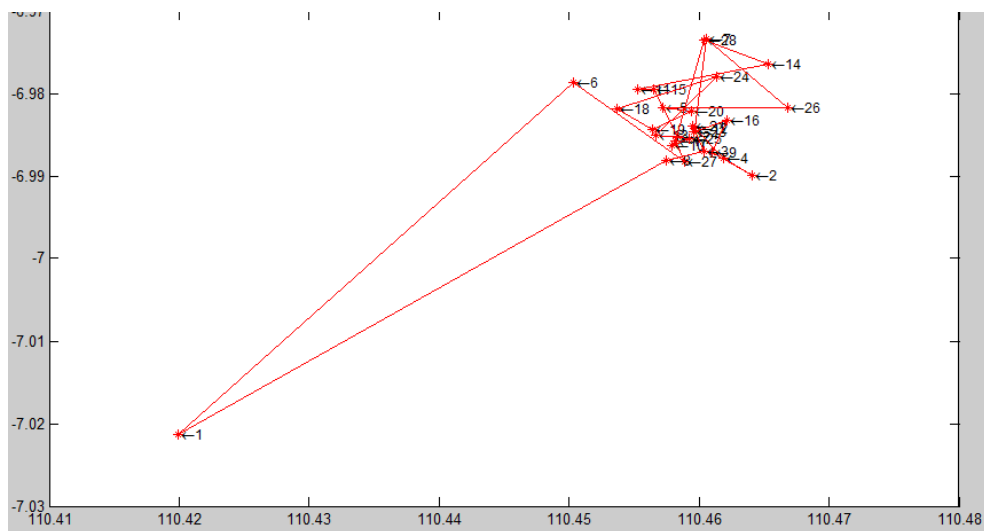
Jadi nilai keanggotaan fuzzy untuk nilai probabilitas mutasi 0,197845 dan nilai probabilitas crossover 0,714007 adalah 0,7408.

4. Klik tombol **Cari**. Maka akan dilakukan proses perhitungan seperti yang tertera pada Gambar 4.5.



Gambar 4.5 Tampilan TSP Setelah Dijalankan

Kemudian jika ingin melihat grafik koordinat alamat tujuan klik tombol **Plot** seperti tertera pada Gambar 4.6.



Gambar 4.6 Tampilan Grafik Koordinat Alamat Tujuan

Hasil setelah dilakukan perhitungan menggunakan populasi 100, generasi 100, probabilitas mutasi 0,197845 dan probabilitas *crossover* 0,714007 sebanyak 10 kali pada Tabel 4.1.

Tabel 4.1 Hasil Perhitungan menggunakan Populasi 100 dan Generasi 100

No	<i>Fitness</i> terbaik	<i>Fitness</i> rata-rata	Panjang jalur terbaik (Km)	Waktu (detik)	Jalur terbaik
1	0,023901	0,019	41,84	60,375	1 – 6 – 27 – 17 – 28 – 14 – 11 – 15 – 22 – 10 – 25 – 13 – 24 – 18 – 19 – 20 – 5 – 26 – 7 – 12 – 4 – 2 – 9 – 16 – 23 – 21 – 3 – 8 – 1
2	0,024125	0,019009	41,45	62,664	1 – 8 – 16 – 9 – 12 – 4 – 15 – 14 – 2 – 27 – 22 – 25 – 19 – 11 – 28 – 7 – 18 – 6 – 24 – 3 – 26 – 5 – 21 – 10 – 17 – 20 – 13 – 23 – 1
3	0,024056	0,019052	41,57	67,015	1 – 12 – 11 – 6 – 18 – 25 – 16 – 17 – 9 – 3 – 13 – 15 – 27 – 22 – 19 – 10 – 21 – 26 – 24 – 23 – 4 – 14 – 5 – 28 – 7 – 20 – 2 – 8 – 1
4	0,024361	0,019068	41,05	58,968	1 – 12 – 23 – 17 – 26 – 2 – 22 – 5 – 27 – 18 – 13 – 15 – 9 – 10 – 20 – 16 – 14 – 3 – 4 – 19 – 6 – 11 – 24 – 7 – 28 – 21 – 25 – 8 – 1
5	0,023685	0,01917	42,22	68,454	1 – 23 – 9 – 19 – 25 – 10 – 13 – 18 – 14 – 28 – 7 – 20 – 8 – 3 – 4 – 12 – 16 – 26 – 17 – 27 – 24 – 2 – 22 – 21 – 15 – 5 – 11 – 6 – 1
6	0,024108	0,019051	41,48	57,694	1 – 8 – 13 – 17 – 22 – 10 – 23 – 5 – 28 – 3 – 2 – 19 – 18 – 26 – 7 – 16 – 20 – 4 – 9 – 6 – 15 – 11 – 24 – 14 – 25 – 21 – 27 – 12 – 1
7	0,024704	0,01895	40,479	61,521	1 – 23 – 25 – 8 – 5 – 27 – 4 – 18 – 15 – 11 – 6 – 24 – 7 – 28 – 14 – 9 – 3 – 20 – 21 – 17 – 10 – 16 – 19 – 2 – 26

No	<i>Fitness</i> terbaik	<i>Fitness</i> rata-rata	Panjang jalur terbaik (Km)	Waktu (detik)	Jalur terbaik
					- 22 - 13 - 12 - 1
8	0,024734	0,019	40,43	58,981	1 - 8 - 25 - 27 - 17 - 14 - 26 - 2 - 3 - 4 - 18 - 11 - 15 - 24 - 20 - 16 - 7 - 5 - 28 - 10 - 21 - 19 - 6 - 23 - 9 - 22 - 13 - 12 - 1
9	0,02331	0,019211	42,9	61,254	1 - 8 - 13 - 5 - 28 - 24 - 7 - 14 - 26 - 3 - 20 - 27 - 16 - 25 - 19 - 22 - 4 - 9 - 12 - 7 - 23 - 11 - 15 - 18 - 21 - 10 - 2 - 6 - 1
10	0,024313	0,019068	41,13	62,181	1 - 8 - 15 - 16 - 24 - 5 - 18 - 13 - 17 - 22 - 27 - 10 - 3 - 19 - 21 - 9 - 12 - 20 - 25 - 14 - 26 - 4 - 2 - 28 - 7 - 11 - 6 - 23 - 1

Dari Tabel 4.1 diperoleh hasil rata-rata panjang jalur terbaik adalah 41,4549 Km nilai *fitness* terbaik yang terbesar adalah 0,024734, panjang jalur terpendek adalah 40,43 Km dan waktu eksekusi adalah 58,981 detik dengan jalur terbaiknya adalah 1 - 8 - 25 - 27 - 17 - 14 - 26 - 2 - 3 - 4 - 18 - 11 - 15 - 24 - 20 - 16 - 7 - 5 - 28 - 10 - 21 - 19 - 6 - 23 - 9 - 22 - 13 - 12 - 1.

4.1.2.2 Perhitungan menggunakan masukan populasi 100 dan generasi 200.

Hasil setelah dilakukan perhitungan menggunakan populasi 100, generasi 200, probabilitas mutasi 0,144176 dan probabilitas *crossover* 0,7936 sebanyak 10 kali pada Tabel 4.2. Dan untuk mencari nilai keanggotaan fuzzy probabilitas mutasi dan probabilitas *crossover* diperoleh:

$$\min(\mu(P_{\text{mutasi}} = \text{MEDIUM}), \mu(P_{\text{crossover}} = \text{LARGE})),$$

$$\mu(\text{populasi dan generasi}))$$

$$= \min(1, 1, 0.644) = 0.644$$

Jadi nilai keanggotaan fuzzy untuk nilai probabilitas mutasi 0,144176 dan nilai probabilitas crossover 0,7936 adalah 0,644.

Tabel 4.2 Hasil Perhitungan menggunakan Populasi 100 dan Generasi 200

No	Fitness terbaik	Fitness rata-rata	Panjang jalur terbaik (Km)	Waktu (detik)	Jalur terbaik
1	0,024131	0,019119	41,44	124,847	1 – 12 – 22 – 9 – 24 – 14 – 6 – 15 – 17 – 25 – 27 – 10 – 20 – 11 – 13 – 23 – 16 – 26 – 18 – 19 – 2 – 3 – 4 – 21 – 5 – 7 – 28 – 8 – 1
2	0,023889	0,019163	41,86	125,063	1 – 8 – 11 – 18 – 3 – 2 – 22 – 14 – 28 – 7 – 24 – 15 – 20 – 27 – 10 – 17 – 9 – 16 – 26 – 19 – 21 – 12 – 25 – 5 – 4 – 13 – 6 – 23 – 1
3	0,025394	0,019126	39,38	123,998	1 – 8 – 12 – 23 – 27 – 22 – 28 – 7 – 24 – 20 – 9 – 2 – 4 – 3 – 11 – 13 – 10 – 25 – 16 – 5 – 18 – 17 – 21 – 14 – 26 – 15 – 6 – 19 – 1
4	0,02448	0,019216	40,85	122,46	1 – 12 – 4 – 8 – 5 – 18 – 20 – 24 – 14 – 7 – 28 – 3 – 17 – 9 – 10 – 21 – 13 – 6 – 11 – 22 – 16 – 15 – 26 – 2 – 27 – 25 – 19 – 23 – 1
5	0,024225	0,01991	41,28	126,161	1 – 13 – 17 – 21 – 6 – 19 – 4 – 11 – 18 – 25 – 27 – 23 – 12 – 7 – 28 – 3 – 16 – 26 – 14 – 24 – 9 – 2 – 10 – 22 – 20 – 15 – 5 – 8 – 1
6	0,025253	0,019121	39,6	123,307	1 – 6 – 24 – 2 – 3 – 4 – 21 – 25 – 12 – 20 – 5 – 15 – 23 – 9 – 16 – 26 – 11 – 18 – 22 – 27 – 19 – 17 – 10 – 14 – 7 – 28 – 13 – 8 – 1
7	0,024869	0,019176	40,21	119,62	1 – 8 – 5 – 18 – 15 – 11 – 13 – 19 – 10 – 20 – 4 – 27 – 2 – 21 – 22 – 25 – 16 – 26 – 14 – 3 – 6 – 9 – 24 – 17 –

No	Fitness terbaik	Fitness rata-rata	Panjang jalur terbaik (Km)	Waktu (detik)	Jalur terbaik
					12 – 28 – 7 – 23 – 1
8	0,024981	0,019143	40,03	119,724	1 – 8 – 17 – 23 – 21 – 2 – 25 – 10 – 20 – 13 – 19 – 4 – 9 – 28 – 7 – 6 – 11 – 18 – 15 – 22 – 27 – 5 – 16 – 3 – 24 – 26 – 14 – 12 – 1
9	0,025253	0,01927	39,599	118,38	1 – 8 – 22 – 3 – 2 – 27 – 17 – 14 – 26 – 10 – 4 – 16 – 15 – 6 – 20 – 9 – 19 – 28 – 7 – 5 – 18 – 11 – 24 – 13 – 21 – 25 – 23 – 12 – 1
10	0,024039	0,019228	41,599	129,171	1 – 8 – 6 – 15 – 14 – 11 – 23 – 25 – 4 – 27 – 21 – 17 – 18 – 19 – 22 – 20 – 9 – 16 – 2 – 3 – 13 – 10 – 12 – 5 – 26 – 24 – 7 – 28 – 1

Dari Tabel 4.2 diperoleh hasil rata-rata panjang jalur terbaik adalah 40,5848 Km, nilai *fitness* terbaik yang terbesar adalah 0,025394, panjang jalur terpendek adalah 39,38 Km dan waktu eksekusi adalah 123,998 detik dengan jalur terbaiknya adalah 1 – 8 – 12 – 23 – 27 – 22 – 28 – 7 – 24 – 20 – 9 – 2 – 4 – 3 – 11 – 13 – 10 – 25 – 16 – 5 – 18 – 17 – 21 – 14 – 26 – 15 – 6 – 19 – 1.

4.1.2.3 Perhitungan menggunakan masukan populasi 100 dan generasi 500.

Hasil setelah dilakukan perhitungan menggunakan populasi 100, generasi 500, probabilitas mutasi 0,0888582 dan probabilitas *crossover* 0,862962 sebanyak 10 kali pada Tabel 4.3. Dan untuk nilai keanggotaan fuzzy probabilitas mutasi dan probabilitas *crossover* diperoleh:

$$\min(\mu(P_{\text{mutasi}} = \text{SMALL}), \mu(P_{\text{crossover}} = \text{VERY LARGE}),$$

$$\mu(\text{populasi dan generasi}))$$

$$= \min(1, 0,9488, 0.875) = 0.875$$

Jadi nilai keanggotaan fuzzy untuk nilai probabilitas mutasi 0,0888582 dan nilai probabilitas crossover 0,862962 adalah 0,875.

Tabel 4.3 Hasil Perhitungan menggunakan Populasi 100 dan Generasi 500

No	<i>Fitness</i> terbaik	<i>Fitness</i> rata-rata	Panjang jalur terbaik (Km)	Waktu (detik)	Jalur terbaik
1	0,026205	0,018893	38,16	317,541	1 – 6 – 5 – 19 – 22 – 18 – 13 – 11 – 15 – 16 – 4 – 2 – 27 – 10 – 25 – 17 – 9 – 21 – 26 – 20 – 24 – 14 – 23 – 7 – 28 – 12 – 3 – 8 – 1
2	0,027027	0,019193	37	316,335	1 – 8 – 10 – 2 – 21 – 19 – 12 – 27 – 25 – 17 – 11 – 15 – 14 – 24 – 16 – 5 – 6 – 18 – 13 – 22 – 20 – 28 – 7 – 26 – 9 – 3 – 4 – 23 – 1
3	0,026082	0,019049	38,34	320,522	1 – 8 – 28 – 7 – 24 – 26 – 21 – 23 – 9 – 2 – 27 – 20 – 11 – 4 – 14 – 3 – 22 – 17 – 10 – 13 – 19 – 18 – 6 – 15 – 5 – 16 – 25 – 12 – 1
4	0,026048	0,019168	38,390	314,889	1 – 21 – 22 – 18 – 6 – 19 – 13 – 25 – 16 – 7 – 28 – 24 – 14 – 11 – 5 – 15 – 9 – 23 – 12 – 17 – 2 – 27 – 4 – 3 – 26 – 20 – 10 – 8 – 1
5	0,025401	0,019405	39,369	311,045	1 – 6 – 19 – 27 – 22 – 5 – 13 – 18 – 12 – 2 – 4 – 16 – 3 – 20 – 23 – 25 – 10 – 21 – 17 – 26 – 15 – 11 – 24 – 7 – 28 – 9 – 14 – 8 – 1
6	0,02774	0,019174	36,049	305,744	1 – 6 – 18 – 22 – 17 – 19 – 7 – 28 – 24 – 14 – 26 – 20 – 12 – 23 – 25 – 9 – 4 – 3 – 15 – 5 – 16 – 11 – 13 – 10 – 21 – 27 – 2 – 8 – 1
7	0,026497	0,019355	37,74	314,59	1 – 6 – 18 – 17 – 4 – 3 – 12 – 10 – 27 – 19 – 16 – 23 – 20 – 24 – 9 – 26 – 14 – 7 – 28 – 11 – 15 – 5 – 13

No	<i>Fitness</i> terbaik	<i>Fitness</i> rata-rata	Panjang jalur terbaik (Km)	Waktu (detik)	Jalur terbaik
					- 25 - 21 - 22 - 2 - 8 - 1
8	0,026247	0,019181	38,1	313,364	1 - 6 - 11 - 16 - 25 - 7 - 28 - 15 - 23 - 17 - 24 - 14 - 26 - 9 - 5 - 19 - 18 - 13 - 21 - 27 - 22 - 10 - 20 - 12 - 3 - 4 - 2 - 8 - 1
9	0,025497	0,019277	39,22	315,052	1 - 12 - 25 - 24 - 15 - 18 - 5 - 19 - 7 - 28 - 14 - 26 - 10 - 13 - 9 - 20 - 22 - 27 - 17 - 23 - 3 - 2 - 4 - 16 - 21 - 6 - 11 - 8 - 1
10	0,025786	0,019082	38,78	313,87	1 - 12 - 23 - 9 - 2 - 27 - 4 - 26 - 5 - 20 - 18 - 15 - 28 - 21 - 25 - 13 - 7 - 14 - 24 - 11 - 6 - 16 - 3 - 10 - 19 - 17 - 22 - 8 - 1

Dari Tabel 4.3 diperoleh hasil rata-rata panjang jalur terbaik adalah 38,1148 Km nilai *fitness* terbaik yang terbesar adalah 0,02774, panjang jalur terpendek adalah 36,049 Km dan waktu eksekusi adalah 305,744 detik dengan jalur terbaiknya adalah 1 - 6 - 18 - 22 - 17 - 19 - 7 - 28 - 24 - 14 - 26 - 20 - 12 - 23 - 25 - 9 - 4 - 3 - 15 - 5 - 16 - 11 - 13 - 10 - 21 - 27 - 2 - 8 - 1.

4.1.2.4 Perhitungan menggunakan masukan populasi 100 dan generasi 1000.

Hasil setelah dilakukan perhitungan menggunakan populasi 100, generasi 1000, probabilitas mutasi 0,0872423 dan probabilitas *crossover* 0,866503 sebanyak 10 kali pada Tabel 4.4. Dan untuk nilai keanggotaan fuzzy probabilitas mutasi dan probabilitas *crossover* diperoleh:

$$\min(\mu(P_{\text{mutasi}} = \text{SMALL}), \mu(P_{\text{crossover}} = \text{VERY LARGE}),$$

$$\mu(\text{populasi dan generasi}))$$

$$= \min(1, 0,9772, 0.875) = 0.875$$

Jadi nilai keanggotaan fuzzy untuk nilai probabilitas mutasi 0,0872423 dan nilai probabilitas crossover 0,866503 adalah 0,875.

Tabel 4.4 Hasil Perhitungan menggunakan Populasi 100 dan Generasi 1000

No	<i>Fitness</i> terbaik	<i>Fitness</i> rata-rata	Panjang jalur terbaik (Km)	Waktu (detik)	Jalur terbaik
1	0,027825	0,019059	35,939	628,616	1 – 8 – 3 – 9 – 2 – 16 – 26 – 14 – 24 – 7 – 28 – 21 – 18 – 20 – 11 – 4 – 22 – 17 – 12 – 13 – 10 – 25 – 23 – 27 – 19 – 15 – 5 – 6 – 1
2	0,026233	0,019279	38.12	648,476	1 – 8 – 10 – 22 – 2 – 26 – 5 – 16 – 9 – 27 – 25 – 12 – 6 – 11 – 3 – 4 – 20 – 28 – 7 – 24 – 14 – 17 – 19 – 13 – 21 – 18 – 15 – 23 – 1
3	0,026233	0,019279	38,120	648,476	1 – 8 – 10 – 22 – 2 – 26 – 5 – 16 – 9 – 27 – 25 – 12 – 6 – 11 – 3 – 4 – 20 – 28 – 7 – 24 – 14 – 17 – 19 – 13 – 21 – 18 – 15 – 23 – 1
4	0,026925	0,019009	37,14	610,492	1 – 12 – 23 – 15 – 26 – 14 – 24 – 16 – 4 – 9 – 18 – 6 – 11 – 3 – 2 – 25 – 10 – 21 – 13 – 20 – 17 – 27 – 7 – 28 – 5 – 19 – 22 – 8 – 1
5	0,027071	0,019284	36,94	663,694	1 – 12 – 3 – 27 – 23 – 22 – 19 – 25 – 24 – 14 – 20 – 9 – 16 – 4 – 2 – 10 – 17 – 18 – 11 – 15 – 26 – 7 – 28 – 21 – 8 – 13 – 5 – 6 – 1
6	0,027633	0,01923	36,189	626,706	1 – 23 – 25 – 19 – 21 – 7 – 28 – 24 – 20 – 15 – 18 – 6 – 5 – 4 – 2 – 27 – 12 – 17 – 22 – 13 – 11 – 9 – 3 – 26 – 14 – 16 – 10 – 8 – 1
7	0,027323	0,019026	36,599	609,443	1 – 6 – 11 – 15 – 24 – 9 – 16 – 4 – 2 – 3 – 10 – 27 – 14 – 26 – 5 – 21 – 17 – 28 – 7 – 20 – 22 – 13 – 18 – 19 – 12 – 25 – 23 – 8 – 1

No	<i>Fitness</i> terbaik	<i>Fitness</i> rata-rata	Panjang jalur terbaik (Km)	Waktu (detik)	Jalur terbaik
8	0,028425	0,018959	35,18	635,698	1 – 8 – 13 – 19 – 15 – 20 – 5 – 7 – 28 – 21 – 17 – 10 – 25 – 22 – 26 – 14 – 24 – 11 – 6 – 18 – 27 – 3 – 16 – 23 – 9 – 2 – 4 – 12 – 1
9	0,027763	0,019427	36,019	686,621	1 – 8 – 23 – 25 – 12 – 22 – 9 – 4 – 3 – 2 – 26 – 14 – 20 – 21 – 17 – 18 – 11 – 5 – 15 – 19 – 10 – 27 – 13 – 16 – 24 – 7 – 28 – 6 – 1
10	0,027648	0,019041	36,169	668,626	1 – 8 – 10 – 18 – 15 – 16 – 26 – 24 – 14 – 28 – 7 – 27 – 13 – 19 – 23 – 25 – 17 – 22 – 5 – 11 – 6 – 20 – 21 – 9 – 4 – 2 – 3 – 12 – 1

Dari Tabel 4.4 diperoleh hasil rata-rata panjang jalur terbaik adalah 36,6415 Km, nilai *fitness* terbaik yang terbesar adalah 0,028425, panjang jalur terpendek adalah 35,180 Km dan waktu eksekusi adalah 635,698 detik dengan jalur terbaiknya adalah 1 – 8 – 13 – 19 – 15 – 20 – 5 – 7 – 28 – 21 – 17 – 10 – 25 – 22 – 26 – 14 – 24 – 11 – 6 – 18 – 27 – 3 – 16 – 23 – 9 – 2 – 4 – 12 – 1.

4.1.2.5 Perhitungan menggunakan masukan populasi 200 dan generasi 100

Hasil setelah dilakukan perhitungan menggunakan populasi 200, generasi 100, probabilitas mutasi 0,153006 dan probabilitas *crossover* 0,674734 sebanyak 10 kali pada Tabel 4.5. Dan untuk nilai keanggotaan fuzzy probabilitas mutasi dan probabilitas *crossover* diperoleh:

$$\min(\mu(P_{mutasi} = MEDIUM), \mu(P_{crossover} = SMALL),$$

$$\mu(\text{populasi dan generasi}))$$

$$= \min(1, 0,2178, 0,644) = 0,2178$$

Jadi nilai keanggotaan fuzzy untuk nilai probabilitas mutasi 0,153006 dan nilai probabilitas crossover 0,674734 adalah 0,2178.

Tabel 4.5 Hasil Perhitungan menggunakan Populasi 200 dan Generasi 100

No	<i>Fitness</i> terbaik	<i>Fitness</i> rata-rata	Panjang jalur terbaik (Km)	Waktu (detik)	Jalur terbaik
1	0,024190	0,019207	41,34	101,256	1 – 8 – 6 – 11 – 18 – 13 – 26 – 14 – 9 – 10 – 4 – 25 – 21 – 2 – 3 – 24 – 19 – 12 – 17 – 16 – 20 – 7 – 28 – 5 – 15 – 27 – 22 – 23 – 1
2	0,023832	0,018992	41,96	99,794	1 – 8 – 11 – 18 – 19 – 10 – 6 – 15 – 3 – 9 – 4 – 25 – 2 – 27 – 12 – 26 – 14 – 21 – 28 – 5 – 13 – 17 – 24 – 7 – 16 – 20 – 22 – 23 – 1
3	0,025157	0,019134	39,75	102,174	1 – 8 – 10 – 3 – 4 – 27 – 28 – 7 – 6 – 11 – 19 – 20 – 23 – 18 – 5 – 14 – 24 – 21 – 13 – 17 – 2 – 9 – 16 – 22 – 25 – 15 – 26 – 12 – 1
4	0,024039	0,018937	41,599	100,418	1 – 13 – 15 – 11 – 6 – 7 – 28 – 20 – 18 – 27 – 4 – 23 – 25 – 12 – 21 – 22 – 24 – 14 – 16 – 3 – 5 – 26 – 9 – 19 – 17 – 2 – 10 – 8 – 1
5	0,025202	0,019071	39,68	100,309	1 – 8 – 13 – 23 – 12 – 17 – 4 – 9 – 3 – 16 – 22 – 21 – 15 – 24 – 5 – 20 – 14 – 7 – 28 – 11 – 6 – 18 – 26 – 2 – 27 – 19 – 25 – 10 – 1
6	0,024378	0,019043	41,02	100,549	1 – 8 – 20 – 3 – 13 – 2 – 12 – 4 – 23 – 27 – 22 – 21 – 25 – 17 – 26 – 9 – 10 – 18 – 5 – 11 – 19 – 16 – 28 – 7 – 15 – 24 – 14 – 6 – 1
7	0,025183	0,019119	39,71	100,006	1 – 23 – 5 – 12 – 27 – 2 – 4 – 3 – 18 – 22 – 21 – 14 – 20 – 11 – 17 – 25 – 24 – 19 – 13 – 9 – 16 – 26 – 7 – 28 – 15 – 6 – 10 – 8 – 1

No	<i>Fitness</i> terbaik	<i>Fitness</i> rata-rata	Panjang jalur terbaik (Km)	Waktu (detik)	Jalur terbaik
8	0,024826	0,019108	40,28	99,485	1 – 6 – 20 – 15 – 28 – 7 – 24 – 18 – 19 – 10 – 13 – 11 – 26 – 16 – 17 – 14 – 25 – 12 – 2 – 3 – 9 – 5 – 23 – 4 – 22 – 21 – 27 – 8 – 1
9	0,025233	0,019149	39,63	100,526	1 – 23 – 20 – 14 – 24 – 7 – 16 – 5 – 18 – 11 – 21 – 17 – 12 – 22 – 25 – 6 – 28 – 15 – 27 – 13 – 19 – 26 – 9 – 3 – 4 – 2 – 10 – 8 – 1
10	0,025893	0,019071	38,62	100,133	1 – 8 – 9 – 27 – 20 – 13 – 17 – 10 – 19 – 15 – 16 – 28 – 7 – 14 – 24 – 26 – 12 – 25 – 22 – 11 – 5 – 21 – 6 – 18 – 3 – 4 – 2 – 23 – 1

Dari Tabel 4.5 diperoleh hasil rata-rata panjang jalur terbaik 40,3589 Km, nilai *fitness* terbaik yang terbesar adalah 0,025893, panjang jalur terpendek adalah 38,62 Km dan waktu eksekusi adalah 100,133 detik dengan jalur terbaiknya adalah 1 – 8 – 9 – 27 – 20 – 13 – 17 – 10 – 19 – 15 – 16 – 28 – 7 – 14 – 24 – 26 – 12 – 25 – 22 – 11 – 5 – 21 – 6 – 18 – 3 – 4 – 2 – 23 – 1.

4.1.2.6 Perhitungan menggunakan masukan populasi 500 dan generasi 100.

Hasil setelah dilakukan perhitungan menggunakan populasi 500, generasi 100, probabilitas mutasi 0,0886878 dan probabilitas *crossover* 0,651854 sebanyak 10 kali pada Tabel 4.6. Dan untuk nilai keanggotaan fuzzy probabilitas mutasi dan probabilitas *crossover* diperoleh:

$$\min(\mu(P_{\text{mutasi}} = \text{SMALL}), \mu(P_{\text{crossover}} = \text{SMALL}),$$

$$\mu(\text{populasi dan generasi}))$$

$$= \min(1, 0.7408, 0.7822) = 0.7408$$

Jadi nilai keanggotaan fuzzy untuk nilai probabilitas mutasi 0,0886878 dan nilai probabilitas crossover 0,651854 adalah 0,7408.

Tabel 4.6 Hasil Perhitungan menggunakan Populasi 500 dan Generasi 100

No	<i>Fitness</i> terbaik	<i>Fitness</i> rata-rata	Panjang jalur terbaik (Km)	Waktu (detik)	Jalur terbaik
1	0,024697	0,019053	40,49	223,077	1 – 6 – 20 – 13 – 17 – 22 – 21 – 25 – 12 – 10 – 18 – 19 – 16 – 2 – 27 – 8 – 28 – 7 – 15 – 11 – 4 – 3 – 5 – 14 – 26 – 24 – 9 – 23 – 1
2	0,024716	0,019145	40,46	220,164	1 – 6 – 7 – 28 – 13 – 19 – 24 – 14 – 17 – 2 – 26 – 21 – 5 – 16 – 9 – 4 – 3 – 22 – 27 – 15 – 11 – 18 – 23 – 20 – 25 – 10 – 8 – 12 – 1
3	0,024913	0,019037	40,14	223	1 – 6 – 5 – 26 – 14 – 24 – 21 – 4 – 9 – 22 – 25 – 11 – 15 – 16 – 2 – 13 – 18 – 7 – 28 – 19 – 8 – 10 – 20 – 3 – 17 – 23 – 27 – 12 – 1
4	0,026302	0,018953	38,02	219,192	1 – 23 – 15 – 24 – 14 – 11 – 5 – 27 – 20 – 12 – 6 – 18 – 7 – 28 – 19 – 10 – 21 – 17 – 25 – 22 – 13 – 4 – 3 – 9 – 16 – 26 – 2 – 8 – 1
5	0,024085	0,019056	41,52	223,598	1 – 21 – 13 – 11 – 18 – 6 – 5 – 10 – 27 – 23 – 22 – 3 – 2 – 19 – 17 – 24 – 14 – 20 – 15 – 25 – 12 – 4 – 26 – 9 – 16 – 28 – 7 – 8 – 1
6	0,025981	0,019122	38,49	219,854	1 – 12 – 17 – 11 – 15 – 26 – 18 – 5 – 13 – 6 – 16 – 9 – 20 – 10 – 27 – 21 – 25 – 19 – 3 – 2 – 4 – 24 – 14 – 7 – 28 – 23 – 22 – 8 – 1
7	0,024851	0,019069	40,24	226,765	1 – 23 – 3 – 19 – 25 – 14 – 26 – 6 – 18 – 9 – 2 – 4 – 12 – 17 – 5 – 20 – 10 – 27 – 15 – 16 – 13 – 21 – 22 – 11 – 28 – 7 – 24 – 8 – 1

No	<i>Fitness</i> terbaik	<i>Fitness</i> rata-rata	Panjang jalur terbaik (Km)	Waktu (detik)	Jalur terbaik
8	0,024734	0,019077	40,43	218,588	1 – 23 – 5 – 24 – 7 – 28 – 21 – 27 – 11 – 16 – 9 – 12 – 4 – 10 – 18 – 17 – 22 – 25 – 13 – 14 – 26 – 2 – 3 – 19 – 8 – 20 – 15 – 6 – 1
9	0,025419	0,01905	39,34	226,81	1 – 12 – 22 – 23 – 9 – 3 – 10 – 6 – 28 – 7 – 24 – 17 – 18 – 21 – 25 – 2 – 4 – 13 – 15 – 14 – 26 – 16 – 27 – 19 – 11 – 5 – 20 – 8 – 1
10	0,024624	0,019041	40,61	218,393	1 – 12 – 20 – 14 – 16 – 26 – 24 – 9 – 25 – 21 – 23 – 4 – 27 – 28 – 7 – 13 – 17 – 19 – 8 – 2 – 10 – 3 – 22 – 18 – 5 – 15 – 11 – 6 – 1

Dari Tabel 4.6 diperoleh hasil rata-rata panjang jalur terbaik adalah 39,974 Km, nilai *fitness* terbaik yang terbesar adalah 0,026302, panjang jalur terpendek adalah 38,02 Km dan waktu eksekusi adalah 219,192 detik dengan jalur terbaiknya adalah 1 – 23 – 15 – 24 – 14 – 11 – 5 – 27 – 20 – 12 – 6 – 18 – 7 – 28 – 19 – 10 – 21 – 17 – 25 – 22 – 13 – 4 – 3 – 9 – 16 – 26 – 2 – 8 – 1.

4.1.2.7 Perhitungan menggunakan masukan populasi 1000 dan generasi 100.

Hasil setelah dilakukan perhitungan menggunakan populasi 1000, generasi 100, probabilitas mutasi 0,0869803 dan probabilitas *crossover* 0,647113 sebanyak 10 kali pada Tabel 4.7. Dan untuk nilai keanggotaan fuzzy probabilitas mutasi dan probabilitas *crossover* diperoleh:

$$\min(\mu(P_{\text{mutasi}} = \text{SMALL}), \mu(P_{\text{crossover}} = \text{SMALL})),$$

$$\mu(\text{populasi dan generasi}))$$

$$= \min(1, 0.828, 0.7822) = 0.7822$$

Jadi nilai keanggotaan fuzzy untuk nilai probabilitas mutasi 0,0869803 dan nilai probabilitas crossover 0,647113 adalah 0.7822.

Tabel 4.7 Hasil Perhitungan menggunakan Populasi 1000 dan Generasi 100

No	<i>Fitness</i> terbaik	<i>Fitness</i> rata-rata	Panjang jalur terbaik (Km)	Waktu (detik)	Jalur terbaik
1	0,025013	0,019086	39,98	424,156	1 – 8 – 3 – 9 – 4 – 17 – 27 – 11 – 7 – 24 – 14 – 20 – 22 – 10 – 21 – 13 – 25 – 12 – 5 – 26 – 2 – 23 – 19 – 16 – 18 – 15 – 28 – 6 – 1
2	0,025846	0,019108	38,69	424,746	1 – 8 – 24 – 20 – 9 – 4 – 2 – 26 – 3 – 25 – 16 – 28 – 7 – 14 – 23 – 19 – 18 – 6 – 21 – 17 – 13 – 22 – 15 – 11 – 5 – 27 – 10 – 12 – 1
3	0,025833	0,019023	38,71	425,012	1 – 8 – 21 – 16 – 25 – 12 – 27 – 11 – 28 – 7 – 6 – 18 – 19 – 17 – 10 – 13 – 22 – 20 – 2 – 4 – 14 – 26 – 15 – 24 – 9 – 3 – 5 – 23 – 1
4	0,025381	0,019059	39,4	419,49	1 – 8 – 13 – 19 – 15 – 20 – 5 – 7 – 28 – 21 – 17 – 10 – 25 – 22 – 26 – 14 – 24 – 11 – 6 – 18 – 27 – 3 – 16 – 23 – 9 – 2 – 4 – 12 – 1
5	0,02631	0,01903	38,009	420,178	1 – 6 – 16 – 26 – 24 – 20 – 9 – 22 – 14 – 7 – 28 – 21 – 5 – 11 – 19 – 10 – 18 – 15 – 25 – 23 – 12 – 27 – 2 – 4 – 17 – 13 – 3 – 8 – 1
6	0,0251	0,018968	39,84	425,88	1 – 6 – 9 – 3 – 4 – 5 – 15 – 23 – 21 – 26 – 20 – 22 – 7 – 28 – 24 – 14 – 11 – 19 – 17 – 10 – 27 – 2 – 13 – 25 – 16 – 18 – 12 – 8 – 1
7	0,02432	0,01909	41,119	422,808	1 – 8 – 22 – 20 – 7 – 14 – 16 – 26 – 15 – 18 – 28 – 23 – 25 – 10 – 5 – 19 – 27 – 2 – 21 – 24 – 11 – 13 – 17 – 4 – 9 – 12 – 3 – 6 – 1
8	0,024759	0,019033	40,39	430,535	1 – 12 – 23 – 18 – 15 – 11 – 27 – 20 – 5 – 3 – 4 – 16

No	<i>Fitness</i> terbaik	<i>Fitness</i> rata-rata	Panjang jalur terbaik (Km)	Waktu (detik)	Jalur terbaik
					- 17 - 19 - 13 - 26 - 14 - 22 - 8 - 2 - 21 - 9 - 25 - 10 - 7 - 28 - 24 - 6 - 1
9	0,025233	0,019119	39,63	423,846	1 - 8 - 20 - 3 - 21 - 5 - 13 - 24 - 26 - 14 - 2 - 9 - 27 - 17 - 10 - 25 - 22 - 16 - 23 - 4 - 6 - 18 - 19 - 15 - 11 - 7 - 28 - 12 - 1
10	0,025196	0,01904	39,689	439,794	1 - 8 - 9 - 20 - 6 - 14 - 7 - 28 - 15 - 26 - 2 - 3 - 23 - 25 - 4 - 27 - 24 - 16 - 19 - 13 - 17 - 18 - 10 - 21 - 11 - 5 - 22 - 12 - 1

Dari Tabel 4.7 diperoleh hasil rata-rata panjang jalur terbaik adalah 39,5457 Km, nilai *fitness* terbaik yang terbesar adalah 0,02631, panjang jalur terpendek adalah 38,009 Km dan waktu eksekusi adalah 420,178 detik dengan jalur terbaiknya adalah 1 - 6 - 16 - 26 - 24 - 20 - 9 - 22 - 14 - 7 - 28 - 21 - 5 - 11 - 19 - 10 - 18 - 15 - 25 - 23 - 12 - 27 - 2 - 4 - 17 - 13 - 3 - 8 - 1.

4.1.3 Analisis Penyelesaian Travelling Salesman Problem Menggunakan Aplikasi Algoritma Fuzzy Evolusi dalam Pengiriman Barang di PT. Jalur Nugraha Ekakurir Semarang

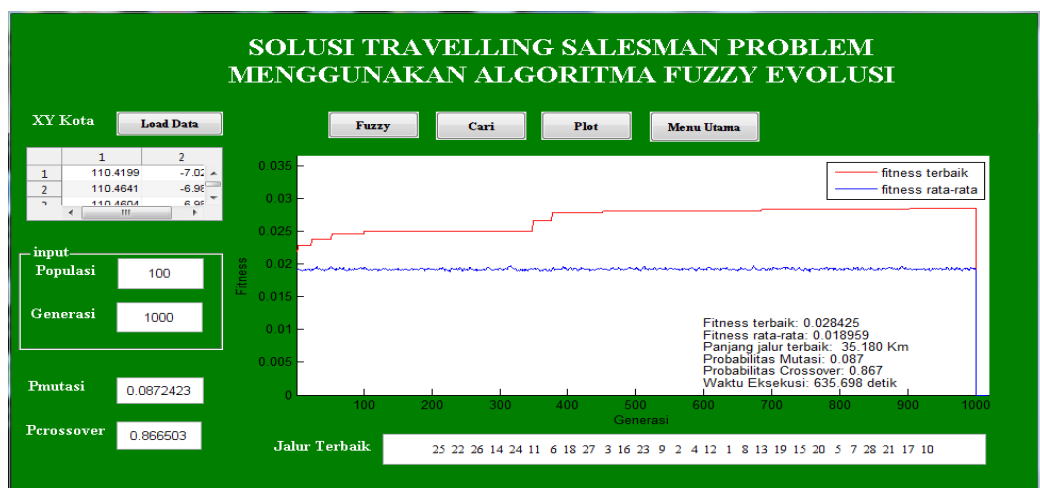
Dari hasil penelitian diperoleh bahwa solusi optimal permasalahan jaringan TSP dalam pengiriman barang oleh PT. Jalur Nugraha Ekakurir ke rumah penerima barang di wilayah Kota Semarang dengan menggunakan variasi populasi dan generasi pada algoritma Fuzzy Evolusi yang berbeda dapat dijelaskan pada Tabel 4.8.

Tabel 4.8 Tabel Hasil Panjang Jalur Terbaik

No	Populasi	Generasi	<i>Fitness</i> terbaik	Panjang jalur terbaik (Km)	Waktu (detik)	Jalur terbaik
1	100	100	0,024734	40,43	58,981	1 – 8 – 25 – 27 – 17 – 14 – 26 – 2 – 3 – 4 – 18 – 11 – 15 – 24 – 20 – 16 – 7 – 5 – 28 – 10 – 21 – 19 – 6 – 23 – 9 – 22 – 13 – 12 – 1
2	100	200	0,025394	39,38	123,998	1 – 8 – 12 – 23 – 27 – 22 – 28 – 7 – 24 – 20 – 9 – 2 – 4 – 3 – 11 – 13 – 10 – 25 – 16 – 5 – 18 – 17 – 21 – 14 – 26 – 15 – 6 – 19 – 1
3	100	500	0,02774	36,049	305,744	1 – 6 – 18 – 22 – 17 – 19 – 7 – 28 – 24 – 14 – 26 – 20 – 12 – 23 – 25 – 9 – 4 – 3 – 15 – 5 – 16 – 11 – 13 – 10 – 21 – 27 – 2 – 8 – 1
4	100	1000	0,028425	35,18	635,698	1 – 8 – 13 – 19 – 15 – 20 – 5 – 7 – 28 – 21 – 17 – 10 – 25 – 22 – 26 – 14 – 24 – 11 – 6 – 18 – 27 – 3 – 16 – 23 – 9 – 2 – 4 – 12 – 1
5	200	100	0,025893	38,62	100,133	1 – 8 – 9 – 27 – 20 – 13 – 17 – 10 – 19 – 15 – 16 – 28 – 7 – 14 – 24 – 26 – 12 – 25 – 22 – 11 – 5 – 21 – 6 – 18 – 3 – 4 – 2 – 23 – 1
6	500	100	0,026302	38,02	219,192	1 – 23 – 15 – 24 – 14 – 11 – 5 – 27 – 20 – 12 – 6 – 18 – 7 – 28 – 19 – 10 – 21 – 17 – 25 – 22 – 13 – 4 – 3 – 9 – 16 – 26 – 2 – 8 – 1
7	1000	100	0,02631	38,009	420,178	1 – 6 – 16 – 26 – 24 – 20 – 9 – 22 – 14 – 7 – 28 – 21 – 5 – 11 – 19 – 10 – 18 – 15 – 25 – 23 – 12 – 27 – 2 – 4 – 17 – 13 – 3 – 8 – 1

Dari ketujuh variasi populasi dan generasi pada algoritma fuzzy evolusi diperoleh bahwa dengan populasi 100 dan generasi 1000 mempunyai nilai *fitness* yang lebih

tinggi serta panjang jalur yang lebih minimal dari yang lain. Nilai *fitness* yang diperoleh adalah 0,028425, panjang jalur terbaik adalah 35,18 Km, waktu eksekusi adalah 635,698 dengan jalur terbaiknya 1 – 8 – 13 – 19 – 15 – 20 – 5 – 7 – 28 – 21 – 17 – 10 – 25 – 22 – 26 – 14 – 24 – 11 – 6 – 18 – 27 – 3 – 16 – 23 – 9 – 2 – 4 – 12 – 1. Gambar 4.7 menunjukkan proses perhitungan dengan panjang jalur terbaik 35,18 Km.



Gambar 4.7 Proses Perhitungan dengan Panjang Jalur Terbaik 35,18 Km

Kemudian untuk analisis probabilitas *crossover* dan probabilitas mutasi dapat dijelaskan pada Tabel 4.9.

Tabel 4.9 Tabel Hasil Probabilitas Mutasi dan Probabilitas *Crossover*

No	Populasi	Generasi	Probabilitas Mutasi	Probabilitas <i>Crossover</i>	Nilai Keanggotaan Fuzzy	Panjang jalur terbaik (Km)
1	100	100	0,197	0,714007	0,7408	40,43
2	100	200	0,144	0,7937	0,644	39,38
3	100	500	0,088	0,862962	0,875	36,049
4	100	1000	0,087	0,9665	0,875	35,18

No	Populasi	Generasi	Probabilitas Mutasi	Probabilitas <i>Crossover</i>	Nilai Keanggotaan Fuzzy	Panjang jalur terbaik (Km)
5	200	100	0,153	0,674734	0,2178	38,62
6	500	100	0,088	0,651854	0,7408	38,02
7	1000	100	0,087	0,647113	0,7822	38,009

Dari Tabel 4.8 dapat dilihat dengan populasi 100 dan generasi 1000 bahwa pada probabilitas mutasi bisa dikatakan lebih kecil dari keenam variasi yang lain walaupun untuk populasi 1000 dan generasi 100 juga mempunyai nilai probabilitas mutasi yang sama kecilnya. Kemudian untuk nilai probabilitas *crossover*nya pada populasi 100 dan generasi 1000 mempunyai nilai yang lebih tinggi dari pada nilai probabilitas *crossover* yang lainnya. Pada nilai keanggotaan fuzzy dengan populasi 100 dan generasi 500 mempunyai nilai keanggotaan yang sama besar dengan populasi 100 dan generasi 1000 namun pada populasi 100 dan generasi 1000 mempunyai panjang jalur lebih pendek dari pada menggunakan populasi 100 dan generasi 1000.

4.2 Pembahasan

Berdasarkan hasil penelitian yang telah dilakukan di PT. Jalur Nugraha Ekakurir Semarang diperoleh hasil pencarian koordinat titik lokasi penelitian dengan bantuan situs Getlatlon.yohman.com yang sudah terintegrasi dengan *Google Maps* menghasilkan koordinat yang cukup akurat. Hal ini mengakibatkan hasil pencarian jarak antara lokasi menjadi lebih tepat. Selain itu, penggunaan bantuan situs Getlatlon.yohman.com dan *Google Maps* bisa menghemat waktu

dan biaya dalam pencarian jarak antar lokasi penelitian. Ini membuktikan bahwa situs Getlatlon.com dan Google *Maps* layak dipilih untuk dijadikan suatu alat pencarian jarak antar lokasi.

Hasil pencarian solusi optimal dengan algoritma fuzzy evolusi dilakukan dengan menggunakan bantuan aplikasi software. Proses evolusi dihentikan pada generasi ke 100, 200, 500 dan 1000 dengan populasi 100, 200, 500 dan 1000. Hal ini karena diharapkan pada generasi dan populasi tersebut didapatkan nilai fitness terbaik dan juga panjang jalur yang minimal.

Solusi optimal dari permasalahan jaringan TSP dengan algoritma fuzzy evolusi pada penelitian ini menghasilkan rute terbaik pengiriman barang PT. Jalur Nugraha Ekakurir ke rumah *supplier* yang tersebar di wilayah Kota Semarang yaitu: “Jalur Nugraha Ekakurir Semarang (Jl. Sultan Agung) – Betty Ekowati (Jl. Parang Kusuma 1) – Indah Putri (Jl. Gusti putri No. 17) – Fresma (Jl. Bledok Kantil 1) – Desy Nourma (Jl. Seruni 4) – Prasetyo Kentjono (Jl. Parang Barong Raya N0.10) – Teta (Jl. Satrio Manah II/2) – Anjie Aristianty (Jl. Sido Asih 5 No. 15) – Elisabet Yania (Jl. Sido Asih 4/79) – Toko Pro Atk (Jl. Tlogosari Raya I/69) – Januar Wahyu (Jl. Parang Kusuma XI/13) – Nouva Alesia (Jl. Parang Kusumo VII/7) – Adi (Bank BRI KCP Tlogosari) – Vani (Jl. Parang Kesit) – Esti (Jl. Bugen) – Maulita (Jl. Grinsing) – Yudi Ucil (Jl. Tlogosari Raya 2 Blok H2) – Putri (Jl. Seruni 7/28) – Michelle Buison (Jl. Soekarno Hatta No.28) – Ida Hanifah (Jl. Malang Sari Cluster III) – Optik Audhifa (Jl. Parang Kembang Raya No.11) – Rendy Risk (Jl. Parang kembang X/30) – Awan Djati (Jl. Parang Baris 6 No. 10) – Ningrum (Jl. Tlogosari Raya 2/ 30) – Elisa Amalia (Jl. Lintang Trenggono V/2) –

Zerlin (Jl. Tirta Mukti 3 No. 1022) – Agung (Jl. Wahyu temurun 2 No. 15) – Ferdinand (Jl. Tlogosari Raya 2 No. 62) – Jalur Nugraha Ekakurir Semarang (Jl. Sultan Agung)” dengan jarak yang ditempuh 35,180 Km.

Berdasarkan hasil pencarian solusi optimal dari jaringan TSP dalam pengiriman barang PT. Jalur Nugraha Ekakurir Semarang ke rumah *supplier* di wilayah Kota Semarang diperoleh solusi optimal menggunakan algoritma fuzzy evolusi dengan populasi 100 dan generasi 1000 menghasilkan solusi optimal yang paling baik dibandingkan dengan 6 variasi lain. Dengan demikian maka penggunaan algoritma fuzzy evolusi dengan populasi 100 dan generasi 1000 dijadikan pilihan pada penyelesaian masalah optimasi terutama pada permasalahan TSP.

BAB 5

PENUTUP

5.1. Simpulan

1. Hasil perhitungan masalah Solusi optimal dari permasalahan jaringan TSP dengan algoritma fuzzy evolusi pada penelitian ini menghasilkan rute terbaik pengiriman barang PT. Jalur Nugraha Ekakurir ke rumah *supplier* yang tersebar di wilayah Kota Semarang yaitu: “Jalur Nugraha Ekakurir Semarang (Jl. Sultan Agung) – Betty Ekowati (Jl. Parang Kusuma 1) – Indah Putri (Jl. Gusti putri No. 17) – Fresma (Jl. Bledok Kantil 1) – Desy Nourma (Jl. Seruni 4) – Prasetyo Kentjono (Jl. Parang Barong Raya N0.10) – Teta (Jl. Satrio Manah II/2) – Anjie Aristianty (Jl. Sido Asih 5 No. 15) – Elisabet Yania (Jl. Sido Asih 4/79) – Toko Pro Atk (Jl. Tlogosari Raya I/69) – Januar Wahyu (Jl. Parang Kusuma XI/13) – Nouva Alesia (Jl. Parang Kusumo VII/7) – Adi (Bank BRI KCP Tlogosari) – Vani (Jl. Parang Kesit) – Esti (Jl. Bugen) – Maulita (Jl. Grinsing) – Yudi Ucil (Jl. Tlogosari Raya 2 Blok H2) – Putri (Jl. Seruni 7/28) – Michelle Buison (Jl. Soekarno Hatta No.28) – Ida Hanifah (Jl. Malangsari Cluster III) – Optik Audhifa (Jl. Parang Kembang Raya No.11) – Rendy Risk (Jl. Parang kembang X/30) – Awan Djati (Jl. Parang Baris 6 No. 10) – Ningrum (Jl. Tlogosari Raya 2/ 30) – Elisa Amalia (Jl. Lintang Trenggono V/2) – Zerlin (Jl. Tirto Mukti 3 No. 1022) – Agung (Jl. Wahyu temurun 2 No. 15) – Ferdinand (Jl. Tlogosari

Raya 2 No. 62) – Jalur Nugraha Ekakurir Semarang (Jl. Sultan Agung)” dengan jarak yang ditempuh 35,180 Km.

2. Hasil pencarian jarak minimum berdasarkan analisis perhitungan masalah jaringan TSP pada pengiriman barang PT. Jalur Nugraha Ekakurir Semarang dengan algoritma genetika menggunakan populasi 100 dan generasi 1000 menghasilkan solusi optimal yaitu 35,180 Km. Hasil tersebut lebih baik dari 6 variasi populasi dan generasi lainnya.

5.2. Saran

1. Dari hasil penelitian ini diharapkan dapat memberikan sumbangan kepada PT. Jalur Nugraha Ekakurir Semarang, bahwa dalam penentuan rute awal pengiriman barang hendaknya mengaplikasikan algoritma fuzzy evolusi sehingga biaya pengiriman barang yang dikeluarkan minimal.
2. Diharapkan pada penelitian selanjutnya dapat dikaji menggunakan variasi populasi dan generasi yang lain dengan algoritma fuzzy evolusi untuk menyelesaikan permasalahan TSP pada pemodelan jaringan.
3. Diharapkan pada penelitian selanjutnya dapat menghubungkan plot rute optimal dengan map.

Daftar Pustaka

- Basuki, A. 2003a. *Algoritma Genetika Suatu Alternatif Penyelesaian Permasalahan Searching, Optimasi dan Machine Learning*. Online. Tersedia di <http://lecturer.eepis-its.edu/~basuki/lecture/AlgoritmaGenetika.pdf> [diakses 21-3-2013].
- Basuki, A, 2003b. *Strategi Menggunakan Algoritma Genetika*. <http://lecturer.eepis-its.edu/~basuki/lecture/StrategiAlgoritmaGenetika.pdf> [21-3-2013].
- Bindu & P. Tanwar. 2012. Fuzzy Inspired Hybrid Genetic Approach to Optimize Travelling Salesman Problem. *International Journal of Computer Science & Communication Network*, 2(3): 416-420. Tersedia di [http://www.ijcscn.com/Documents/Volumes/vol2issue3/ijcscn2012020322 .pdf](http://www.ijcscn.com/Documents/Volumes/vol2issue3/ijcscn2012020322.pdf) [diakses 21-3-2013].
- Desiani, A. & M. Arhani. 2006. *Konsep Kecerdasan Buatan*. Yogyakarta: Andi Offset.
- Entin, 2006. *Kecerdasan Buatan*. Online. Tersedia di <http://lecturer.eepis-its.edu/~entin/kecerdasanbuatanbukuBab07AlgoritmaGenetika.pdf> [diakses 21-3-2013].
- Firmansyah, A. 2007. *Dasar-dasar Pemrograman MATLAB*. Tersedia di www.ilmukomputer.com [diakses 21-3-2013].
- Kusumadewi, S. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- Moon, C., J. Kim., G. Choi. & Y. Seo. 2002. An Efficient Genetic Algorithm for The Traveling Salesman Problem with Precedence Constraints. *European Journal of Operational Research*, 140:606-617. Tersedia di <http://www.ceet.niu.edu/faculty/ghrayeb/IENG576s04/papers/GA/genetic%20algorithm%20for%20the%20traveling%20salesman.pdf> [diakses 21-3-2013].
- Munir, R. 2005. *Matematika Diskrit*. Bandung: CV Informatika.
- Muzid, S. 2008. Pemanfaatan Algoritma Fuzzy Evolusi Untuk Penyelesaian Kasus *Travelling Salesman Problem*. *Seminar Nasional Aplikasi Teknologi Informasi*. Yogyakarta: Universitas Islam Indonesia. Online. Tersedia di <http://journal.uui.ac.id/index.php/Snati/article/view/556/480> [diakses 13-4-2013].

- Muzid, S & S. Kusumadewi. 2007. Membangun Toolbox Algoritma Evolusi Fuzzy untuk Matlab. *Seminar Nasional Aplikasi Teknologi Informasi*. Yogyakarta: Universitas Islam Indonesia. Online. Tersedia di <http://journal.uui.ac.id/index.php/Snati/article/viewFile/1633/1408> [diakses 13-4-2013].
- Rosen, K.H. 2003. *Discrete Mathematics and Its Applications. Fifth Edition*. New York: McGraw-Hill.
- Siang, J. J., 2002. *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer*. Yogyakarta: ANDI.
- Siswanto. 2007. *Operations Research. Jilid 1*. Jakarta: Erlangga.
- Sutarno, H. 2003. *Matematika Diskrit*. Malang: UM PRESS.
- Wibowo, M.A. 2009. *Aplikasi Algoritma Genetika Untuk Penjadwalan Mata Kuliah*. Semarang: Jurusan Matematika Fakultas MIPA UNDIP.

Lampiran 1

**Nama, Alamat dan Kode Lokasi Penerima Barang dari
PT. Jalur Nugraha Ekakurir Semarang**

No	Nama Penerima	Alamat	Kode Lokasi
1	JNE	Jl. Sultan Agung	1
2	Zerlin	Jl. Tirto Mukti 3 No. 1022	2
3	Rendy Risk	Jl. Parang Kembang X No. 30	3
4	Agung	Jl. Wahyu Temurun 2 No. 15	4
5	Teta	Jl. Satrio Manah II No. 2	5
6	Michelle Buison	Jl. Soekarno Hatta No. 28	6
7	Anjie Aristianty	Jl. Sido Asih 5 No. 15	7
8	Betty Ekowati	Jl. Parang Kusuma 1	8
9	Elisa Amalia	Jl. Lintang Trenggono V/ 2	9
10	Nouva Alesia	Jl. Parang Kusuma VII/ 7	10
11	Putri	Jl. Seruni 7 No. 28	11
12	Ferdinad	Jl. Tlogosari Raya 2 No. 62	12
13	Indah Putri	Jl. Gusti Putri No. 17	13
14	Maulita	Jl. Grinsing	14
15	Desy Nourma	Jl. Seruni 4 No. 1	15
16	Awan Djati	Jl Parang Baris 6 No. 10	16
17	Januar Wahyu	Jl. Parang Kusuma XI No. 13	17
18	Ida Hanifah	Jl. Malang Sari Cluster III	18
19	Fresma	Jl. Bledok Kantil 1 No. 5	19
20	Prasetyo Kentjono	Jl. Parang Barong Raya No. 10	20
21	Toko Pro Atk	Jl. Tlogosari Raya I No. 69	21
22	Vani	Jl. Parang Kesit	22
23	Ningrum	Jl. Tlogosari Raya 2/ 30	23
24	Yudi Ucil	Jl. Tlogosari Raya 2 Blok H2	24
25	Adi	Bank BRI KCP Tlogosari	25
26	Esti	Jl. Bugen	26
27	Optik Audhifa	Jl. Parang Kembang Raya No. 11	27
28	Elisabet Yania Adriani	Jl. Sido Asih 4/ 79	28

Lampiran 2

**Kode Lokasi, Koordinat X, Koordinat Y pada alamat penerima barang dari PT.
Jalur Nugraha Ekakurir.**

No.	Kode Lokasi	Koordinat X	Koordinat Y
1.	1	-7.02133989	110.41987414
2.	2	-6.989936464	110.46410464
3.	3	-6.98708183	110.46040320
4.	4	-6.987886509	110.46185159
5.	5	-6.98175722	110.45717382
6.	6	-6.978742071	110.45029177
7.	7	-6.973360855	110.46054267
8.	8	-6.98775805	110.45915865
9.	9	-6.987215613	110.46101474
10.	10	-6.98630976	110.45793556
11.	11	-6.97944632	110.45527481
12.	12	-6.984183857	110.45969023
13.	13	-6.95812238	110.45724892
14.	14	-6.976425231	110.46533947
15.	15	-6.97955281	110.45647644
16.	16	-6.98321084	110.46211981

No.	Kode Lokasi	Koordinat X	Koordinat Y
17.	17	-6.98575601	110.45819306
18.	18	-6.98180581	110.45368694
19.	19	-6.98433434	110.45642280
20.	20	-6.98218851	110.45939469
21.	21	-6.983949574	110.45950784
22.	22	-6.98515965	110.45838617
23.	23	-6.984695020	110.45955076
24.	24	-6.977932720	110.46136393
25.	25	-6.985546956	110.45926108
26.	26	-6.981693993	110.46678149
27.	27	-6.988259229	110.45885288
28.	28	-6.973595144	110.46042465

Lampiran 3

Jarak Antartitik Koordinat

Lokasi	1	2	3	4	5	6	7	8	9	10
1	0	13.7	12.8	13	12.4	10.2	13	8.7	12.3	12.2
2	13.7	0	0.65	0.45	1.7	2.5	2.6	0.95	0.9	1.2
3	12.8	0.65	0	0.24	1	1.8	1.9	0.5	0.21	0.7
4	13	0.45	0.24	0	1	2.1	1.9	0.9	0.29	1
5	12.4	1.7	1	1	0	0.95	1.5	1	1	0.75
6	10.2	2.5	1.8	2.1	0.95	0	2.2	1.7	1.9	1.5
7	13	2.6	1.9	1.9	1.5	2.2	0	1.9	2	1.7
8	8.7	0.95	0.5	0.9	1	1.7	1.9	0	0.9	0.29
9	12.3	0.9	0.21	0.29	1	1.9	2	0.9	0	0.85
10	12.2	1.2	0.7	1	0.75	1.5	1.7	0.29	0.85	0
11	12.7	2.3	1.4	1.4	0.4	0.75	1.5	1.4	1.3	1.8
12	9.7	1.3	0.6	0.65	0.65	1.4	1.4	0.75	0.45	1
13	11.8	1.4	1	1.1	0.75	1.3	1.7	0.45	0.95	0.35
14	13.2	2.1	1.7	1.7	1.6	2.3	1.2	2	1.5	2
15	12.5	2.1	1.5	1.5	0.5	0.95	1.3	1.4	1.2	1.7
16	12.1	1.3	0.6	0.45	0.8	1.6	1.5	1.3	0.4	0.85
17	11.6	1.2	0.75	0.9	0.7	1.6	1.6	0.35	0.75	0.14
18	12.4	2.1	1.3	1.3	0.45	0.6	1.9	1.2	1.1	1
19	11.8	1.5	1.1	1.1	0.65	1.2	1.7	0.6	0.9	0.55
20	11.9	1.5	0.7	0.75	0.35	1.2	1.2	0.7	0.55	0.5
21	11.6	1.3	0.9	1.1	0.9	1.7	1.8	0.65	1	0.4
22	11.7	1.3	0.8	0.85	0.5	1.5	1.5	0.4	0.6	0.21
23	9.7	1.4	0.75	0.75	0.65	1.5	1.5	0.9	0.6	1.1
24	12.4	2.1	1.3	1.3	1	1.8	0.7	1.5	1.2	1.7
25	11.5	1.2	1	0.8	0.7	1.6	1.6	0.45	0.65	0.23
26	12.8	1.3	1.3	1.7	1	2.2	1.6	1.8	1.3	1.6
27	12.3	0.75	0.29	0.65	1	1.9	1.9	0.75	0.85	0.4
28	13	2.6	1.9	1.8	1.4	2.2	0.1	1.9	1.7	1.6

Lokasi	10	11	12	13	14	15	16	17	18	19	20
1	12.2	12.7	9.7	11.8	13.2	12.5	12.1	11.6	12.4	11.8	11.9
2	1.2	2.3	1.3	1.4	2.1	2.1	1.3	1.2	2.1	1.5	1.5
3	0.7	1.4	0.6	1	1.7	1.5	0.6	0.75	1.3	1.1	0.7
4	1	1.4	0.65	1.1	1.7	1.5	0.45	0.9	1.3	1.1	0.75
5	0.75	0.4	0.65	0.75	1.6	0.5	0.8	0.7	0.45	0.65	0.35
6	1.5	0.75	1.4	1.3	2.3	0.95	1.6	1.6	0.6	1.2	1.2
7	1.7	1.5	1.4	1.7	1.2	1.3	1.5	1.6	1.9	1.7	1.2
8	0.29	1.4	0.75	0.45	2	1.4	1.3	0.35	1.2	0.6	0.7
9	0.85	1.3	0.45	0.95	1.5	1.2	0.4	0.75	1.1	0.9	0.55
10	0	1.8	1	0.35	2	1.7	0.85	0.14	1	0.55	0.5
11	1.8	0	1	0.85	1.6	0.2	1.2	1.1	0.45	0.75	0.85
12	1	1	0	0.5	2	1.3	0.9	0.29	1.1	0.55	0.6
13	0.35	0.85	0.5	0	1.8	0.85	0.9	0.21	0.7	0.11	0.6
14	2	1.6	2	1.8	0	1.4	1.3	1.7	2.1	1.8	1.3
15	1.7	0.2	1.3	0.85	1.4	0	1.1	1.3	0.6	0.75	0.65
16	0.85	1.2	0.9	0.9	1.3	1.1	0	0.75	1.1	0.9	0.55
17	0.14	1.1	0.29	0.21	1.7	1.3	0.75	0	0.9	0.3	0.45
18	1	0.45	1.1	0.7	2.1	0.6	1.1	0.9	0	0.55	0.7
19	0.55	0.75	0.55	0.11	1.8	0.75	0.9	0.3	0.55	0	0.55
20	0.5	0.85	0.6	0.6	1.3	0.65	0.55	0.45	0.7	0.55	0
21	0.4	1.3	1	0.55	1.9	1.3	1	0.35	1.2	0.6	0.65
22	0.21	0.95	0.28	0.24	1.6	0.9	0.65	0.12	0.85	0.26	0.35
23	1.1	1.1	0.11	0.75	1.5	1.1	0.55	0.6	1	0.75	0.4
24	1.7	1.1	1.3	1.3	0.6	0.9	1	1.2	1.5	1.4	0.7
25	0.23	1.2	0.17	0.35	1.6	1.1	0.6	0.17	1	0.4	0.45
26	1.6	1.5	1.8	1.6	0.8	1.3	0.8	1.5	1.8	1.7	1.1
27	0.4	1.4	0.45	0.65	2.1	1.4	0.9	0.45	1.3	0.7	0.75
28	1.6	1.5	1.7	1.7	2.1	1.3	1.5	1.6	1.9	1.8	1.2

Lokasi	21	22	23	24	25	26	27	28
1	11.6	11.7	9.7	12.4	11.5	12.8	12.3	13
2	1.3	1.3	1.4	2.1	1.2	1.3	0.75	2.6
3	0.9	0.8	0.75	1.3	1	1.3	0.29	1.9
4	1.1	0.85	0.75	1.3	0.8	1.7	0.65	1.8
5	0.9	0.5	0.65	1	0.7	1	1	1.4
6	1.7	1.5	1.5	1.8	1.6	2.2	1.9	2.2
7	1.8	1.5	1.5	0.7	1.6	1.6	1.9	0.1
8	0.65	0.4	0.9	1.5	0.45	1.8	0.75	1.9
9	1	0.6	0.6	1.2	0.65	1.3	0.85	1.7
10	0.4	0.21	1.1	1.7	0.23	1.6	0.4	1.6
11	1.3	0.95	1.1	1.1	1.2	1.5	1.4	1.5
12	1	0.28	0.11	1.3	0.17	1.8	0.45	1.7
13	0.55	0.24	0.75	1.3	0.35	1.6	0.65	1.7
14	1.9	1.6	1.5	0.6	1.6	0.8	2.1	2.1
15	1.3	0.9	1.1	0.9	1.1	1.3	1.4	1.3
16	1	0.65	0.55	1	0.6	0.8	0.9	1.5
17	0.35	0.12	0.6	1.2	0.17	1.5	0.45	1.6
18	1.2	0.85	1	1.5	1	1.8	1.3	1.9
19	0.6	0.26	0.75	1.4	0.4	1.7	0.7	1.8
20	0.65	0.35	0.4	0.7	0.45	1.1	0.75	1.2
21	0	0.35	0.8	0.8	0.45	1.2	0.6	1.3
22	0.35	0	0.5	1.1	0.16	1.5	0.45	1.5
23	0.8	0.5	0	1.3	0.089	1.8	0.35	1.7
24	0.8	1.1	1.3	0	0.9	0.9	1.2	1.1
25	0.45	0.16	0.089	0.9	0	2.2	0.7	2.2
26	1.2	1.5	1.8	0.9	2.2	0	2	3.6
27	0.6	0.45	0.35	1.2	0.7	2	0	2
28	1.3	1.5	1.7	1.1	2.2	3.6	2	0

Lampiran 4

Tampilan Simulasi Matlab

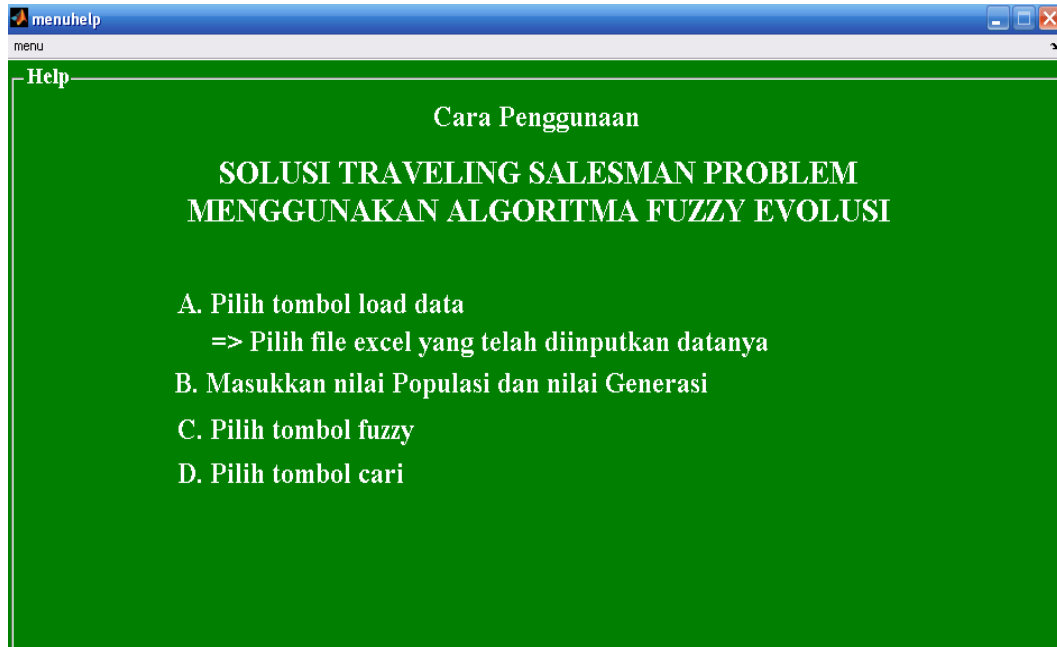
1. Tampilan Menu Utama



2. Tampilan Menu About



3. Tampilan Menu Help



4. Tampilan Menu TSP Fuzzy



Lampiran 5

Coding Pada Matlab

➤ TSPInisialisasi.m:

```
function Populasi = TSPInisiasiPopulasi(UkPop, JumGen)
for ii=1:UkPop,
    [Xval, Ind] = sort(rand(1, JumGen));
    Populasi(ii, :) = Ind;
end
```

➤ TSPEvaluasiIndividu.m:

```
function fitness =
TSPEvaluasiIndividu(Kromosom, JumGen, XYkota)
TB = 0;
load jr
for ii=1:JumGen-1
    a=jr(Kromosom(ii), Kromosom(ii+1));
    TB = TB + a;
end;
% jalur harus kembali ke kota asal
TB = TB + jr(Kromosom(JumGen), Kromosom(1));
fitness = 1/TB;
```

➤ LinearFitnessRanking.m:

```
function LFR =
LinearFitnessRanking(UkPop, Fitness, MaxF, MinF)
[SF, IndF] =sort(Fitness);
for rr=1:UkPop
    LFR(IndF(UkPop-rr+1)) = MaxF- (MaxF-MinF) * ((rr-
1)/(UkPop-1));
end
```

➤ Roulettewheel.m:

```
function Pindex = RouletteWheel(UkPop, LinearFitness)
JumFitness=sum(LinearFitness);
KumulatifFitness=0;
RN =rand;
ii=1;
```



```

while ii <= UkPop
    KumulatifFitness = KumulatifFitness +
LinearFitness(ii);
    if (KumulatifFitness/JumFitness) > RN
        Pindex = ii;
        break;
    end
    ii = ii+1;
end

```

➤ TSPPindahsilang.m:

```

function Anak = TSPPindahsilang(Bapak,Ibu,JumGen)
cp1 = 1 + fix(rand*(JumGen-1));
cp2 = 1 + fix(rand*(JumGen-1));
while cp2==cp1,
    cp2 = 1 + fix(rand*(JumGen-1));
end
if cp1 < cp2,
    cps = cp1;
    cpd = cp2;
else
    cps = cp2;
    cpd = cp1;
% else
%     cps = cp2;
%     cpd = cp1;
end
Anak(1,cps+1:cpd) = Ibu(cps+1:cpd);
Anak(2,cps+1:cpd) = Bapak(cps+1:cpd);

SisaGenbapak = [];
SisaGenIbu = [];
for ii=1:JumGen
    if ~ismember(Bapak(ii),Anak(1,:))
        SisaGenbapak = [SisaGenbapak Bapak(ii)];
    end
    if ~ismember(Ibu(ii),Anak(2,:))
        SisaGenIbu = [SisaGenIbu Ibu(ii)];
    end
end
Anak(1,cpd+1:JumGen) = SisaGenbapak(1:JumGen-cpd);

```

```
Anak(1,1:cps)=SisaGenbapak(1+JumGen-
cpd:length(SisaGenbapak));
Anak(2,cpd+1:JumGen) = SisaGenIbu(1:JumGen-cpd);
Anak(2,1:cps) = SisaGenIbu(1+JumGen-
cpd:length(SisaGenIbu));
```

➤ TSPMutasi.m:

```
function MutKrom = TSPMutasi(Kromosom,JumGen,Pmutasi)
MutKrom = Kromosom;
for ii=1:JumGen,
    if rand<Pmutasi,
        TM2 = 1+ fix(rand*JumGen);
        while TM2==ii,
            TM2=1+fix(rand*JumGen);
        end
        temp = MutKrom(ii);
        MutKrom(ii) = MutKrom(TM2);
        MutKrom(TM2)=temp;
    end
end
```

➤ TSPFuzzy.m:

```
function varargout = TSPFuzzy(varargin)
% TSPFUZZY M-file for TSPFuzzy.fig
%     TSPFUZZY, by itself, creates a new TSPFUZZY or
%     raises the existing
%     singleton*.
%
%     H = TSPFUZZY returns the handle to a new TSPFUZZY
%     or the handle to
%     the existing singleton*.
%
%     TSPFUZZY('CALLBACK',hObject,eventData,handles,...)
%     calls the local
%     function named CALLBACK in TSPFUZZY.M with the
%     given input arguments.
%
%     TSPFUZZY('Property','Value',...) creates a new
%     TSPFUZZY or raises the
%     existing singleton*. Starting from the left,
%     property value pairs are
```

```

%      applied to the GUI before TSPFuzzy_OpeningFcn gets
called.  An
%      unrecognized property name or invalid value makes
property application
%      stop.  All inputs are passed to
TSPFuzzy_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose
"GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
TSPFuzzy

% Last Modified by GUIDE v2.5 26-Aug-2013 17:07:10

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @TSPFuzzy_OpeningFcn, ...
                  'gui_OutputFcn',  @TSPFuzzy_OutputFcn,
                  ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before TSPFuzzy is made visible.

```

```

function TSPFuzzy_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
% varargin   command line arguments to TSPFuzzy (see
VARARGIN)

% Choose default command line output for TSPFuzzy
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes TSPFuzzy wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the
command line.
function varargout = TSPFuzzy_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see
VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in tmbcari.
function tmbcari_Callback(hObject, eventdata, handles)
% hObject    handle to tmbcari (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB

```

```

% handles      structure with handles and user data (see
GUIDATA)
tic
axes(handles.axes1)
cla reset
global XYkota
XYkota
whos XYkota
jr=xlsread('jaraktitik.xlsx',1,'B2:AC29');
save jr jr
clc;
JumGen = length(XYkota(:,1));
UkPop = str2num(get(handles.UkPop,'string'));
Psilang = str2num(get(handles.Psilang,'string'));
Pmutasi = str2num(get(handles.Pmutasi,'string'));
MaxG = str2num(get(handles.MaxG,'string'));

Populasi = TSPInisialisasiPopulasi(UkPop,JumGen);
MaxF= TSPEvaluasiIndividu(Populasi(1,:),JumGen,XYkota);
panjangh=(1/MaxF)/2;
Fthreshold = 1/panjangh;
Bgraf = Fthreshold;
hold on
axis([1 MaxG+20 0 Bgraf]);
hbestplot1 = plot(1:MaxG+20,zeros(1,MaxG+20),'r');
hbestplot2 = plot(1:MaxG+20,zeros(1,MaxG+20),'b');
htext1=text(0.6*MaxG,0.30*Bgraf,sprintf('Fitness terbaik:
%7.6f',0.0));
htext2=text(0.6*MaxG,0.25*Bgraf,sprintf('Fitness rata-
rata: %7.6f',0.0));
htext3=text(0.6*MaxG,0.20*Bgraf,sprintf('Panjang Jalur
terbaik: %7.3f',0.0));
htext4=text(0.6*MaxG,0.15*Bgraf,sprintf('Probabilitas
Mutasi: %4.3f',0.0));
htext5=text(0.6*MaxG,0.10*Bgraf,sprintf('Probabilitas
Crossover: %4.3f',0.0));
htext6=text(0.6*MaxG,0.05*Bgraf,sprintf('Waktu Eksekusi:
%4.3f',0.0));

xlabel('Generasi');
ylabel('Fitness');
legend('fitness terbaik','fitness rata-rata')

```

```

hold off
axes(handles.axes1)
drawnow;

Populasi = TSPInisialisasiPopulasi(UkPop, JumGen);

for generasi=1:MaxG
    MaxF =
    TSPEvaluasiIndividu(Populasi(1,:), JumGen, XYkota);
    MinF=MaxF;
    IndeksIndividuTerbaik = 1;
    for ii=1:UkPop
        Fitness(ii) =
    TSPEvaluasiIndividu(Populasi(ii,:), JumGen, XYkota);
        if (Fitness(ii) >MaxF),
            MaxF = Fitness(ii);
            IndeksIndividuTerbaik=ii;
            JalurTerbaik=Populasi(ii,:);
        end
    end
    axes(handles.axes1)
    FitnessRataRata=mean(Fitness);
    plotvector1=get(hbestplot1, 'YData');
    plotvector1(generasi)=MaxF;
    set(hbestplot1, 'YData', plotvector1);
    plotvector2=get(hbestplot2, 'YData');
    plotvector2(generasi)=FitnessRataRata;
    set(hbestplot2, 'YData', plotvector2);
    set(htext1, 'string', sprintf('Fitness terbaik:
%7.6f', MaxF));
    set(htext2, 'string', sprintf('Fitness rata-rata:
%7.6f', FitnessRataRata));
    set(htext3, 'string', sprintf('Panjang jalur terbaik:
%7.3f Km', 1/MaxF));
    set(htext4, 'String', sprintf('Probabilitas Mutasi:
%4.3f', Pmutasi));
    set(htext5, 'String', sprintf('Probabilitas Crossover:
%4.3f', Psilang));

    drawnow
    if MaxF > Fthreshold,
        break;

```

```

end
TempPopulasi = Populasi;

if mod(UkPop,2)==0,
    IterasiMulai=3;

TempPopulasi(1,:)=Populasi(IndeksIndividuTerbaik,:);

TempPopulasi(1,:)=Populasi(IndeksIndividuTerbaik,:);
else
    IterasiMulai=2;
    TempPopulasi(1,:) =
Populasi(IndeksIndividuTerbaik,:);
end
    LinearFitness =
LinearFitnessRanking(UkPop,Fitness,MaxF,MinF);
    for jj=IterasiMulai:2:UkPop
        IP1=RouletteWheel(UkPop,LinearFitness);
        IP2=RouletteWheel(UkPop,LinearFitness);
        if (rand<Psilang)
            Anak =
TSPPindahsilang(Populasi(IP1,:),Populasi(IP2,:),JumGen);
            TempPopulasi(jj,:) = Anak(1,:);
            TempPopulasi(jj+1,:)=Anak(2,:);
        else
            TempPopulasi(jj,:)=Populasi(IP1,:);
            TempPopulasi(jj+1,:)=Populasi(IP2,:);
        end
    end
end

for kk=IterasiMulai:UkPop,

TempPopulasi(kk,:)=(TSPMutasi(TempPopulasi(kk,:),JumGen,Pmu
tasi));
end
    Populasi=TempPopulasi;
end
%Tanpa tanda ';' berarti menampilkan nilai dari variabel
'JalurTerbaik'
JalurTerbaik1=num2str(JalurTerbaik);
waktu=toc;
set(htext6,'String',sprintf('Waktu Eksekusi: %4.3f
detik',waktu));

```

```

%simpan variabel 'JalurTerbaik' ke dalam file
JalurTerbaik.mat
save JalurTerbaik.mat JalurTerbaik
set(handles.jater,'string',JalurTerbaik1);
save XYkota XYkota

function XYkota_Callback(hObject, eventdata, handles)
% hObject      handle to XYkota (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of XYkota
as text
%           str2double(get(hObject,'String')) returns
contents of XYkota as a double

% --- Executes during object creation, after setting all
properties.
function XYkota_CreateFcn(hObject, eventdata, handles)
% hObject      handle to XYkota (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function UkPop_Callback(hObject, eventdata, handles)
% hObject      handle to UkPop (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB

```



```

% handles      structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of UkPop
as text
%           str2double(get(hObject,'String')) returns
contents of UkPop as a double

% --- Executes during object creation, after setting all
properties.
function UkPop_CreateFcn(hObject, eventdata, handles)
% hObject      handle to UkPop (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Psilang_Callback(hObject, eventdata, handles)
% hObject      handle to Psilang (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
Psilang as text
%           str2double(get(hObject,'String')) returns
contents of Psilang as a double

% --- Executes during object creation, after setting all
properties.

```

```

function Psilang_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Psilang (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function Pmutasi_Callback(hObject, eventdata, handles)
% hObject    handle to Pmutasi (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
Pmutasi as text
%         str2double(get(hObject,'String')) returns
contents of Pmutasi as a double

```

```

% --- Executes during object creation, after setting all
properties.
function Pmutasi_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Pmutasi (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function MaxG_Callback(hObject, eventdata, handles)
% hObject      handle to MaxG (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of MaxG
as text
%          str2double(get(hObject,'String')) returns
contents of MaxG as a double

```

```

% --- Executes during object creation, after setting all
properties.

```

```

function MaxG_CreateFcn(hObject, eventdata, handles)
% hObject      handle to MaxG (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      empty - handles not created until after all
CreateFcns called

```

```

% Hint: edit controls usually have a white background on
Windows.

```

```

%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function PanjJalHarp_Callback(hObject, eventdata,
handles)
% hObject      handle to PanjJalHarp (see GCBO)

```

```

% eventdata reserved - to be defined in a future version
of MATLAB
% handles structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
PanjJalHarp as text
% str2double(get(hObject,'String')) returns
contents of PanjJalHarp as a double

% --- Executes during object creation, after setting all
properties.
function PanjJalHarp_CreateFcn(hObject, eventdata,
handles)
% hObject handle to PanjJalHarp (see GCBO)
% eventdata reserved - to be defined in a future version
of MATLAB
% handles empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function Fthreshold_Callback(hObject, eventdata, handles)
% hObject handle to Fthreshold (see GCBO)
% eventdata reserved - to be defined in a future version
of MATLAB
% handles structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
Fthreshold as text
% str2double(get(hObject,'String')) returns
contents of Fthreshold as a double

```

```

% --- Executes during object creation, after setting all
properties.
function Fthreshold_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to Fthreshold (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata,
handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
set(handles.axes1,'plot','');

function jater_Callback(hObject, eventdata, handles)
% hObject    handle to jater (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of jater
as text
%         str2double(get(hObject,'String')) returns
contents of jater as a double

```

```

% --- Executes during object creation, after setting all
properties.
function jater_CreateFcn(hObject, eventdata, handles)
% hObject    handle to jater (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in menu.
function menu_Callback(hObject, eventdata, handles)
% hObject    handle to menu (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
pos_size=get(handles.figure1,'position');
user_response=tanya_kembali_utama('Menu','Konfirmasi
Kembali ke Menu Utama');
switch user_response
    case ('No')
    case ('Yes')
        delete(handles.figure1);
        haldepan
end

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata,
handles)
% hObject    handle to pushbutton5 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version
of MATLAB
% handles structure with handles and user data (see
GUIDATA)

pop=str2num(get(handles.UkPop, 'string'));
gen=str2num(get(handles.MaxG, 'string'));

a=readfis('evolusi');
hs=evalfis ([pop gen],a);
set(handles.Psilang, 'string', hs(:,1));
set(handles.Pmutasi, 'string', hs(:,2));

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata,
handles)
% hObject handle to pushbutton6 (see GCBO)
% eventdata reserved - to be defined in a future version
of MATLAB
% handles structure with handles and user data (see
GUIDATA)
[nama_file1,nama_path1]=uigetfile({'*.xlsx'; '*.xls'}, 'Buk
a File Excel');
if isequal(nama_file1,0)
    return;
end
global XYkota
[num1, txt1] = xlsread(nama_file1, 1, 'A1:B1000');
XYkota =num1
whos XYkota

% Tampilkan Nilai Training
t = uitable(handles.uitable1);
set(t, 'Data', XYkota);

% --- Executes on button press in pbjlr.
function pbjlr_Callback(hObject, eventdata, handles)
% hObject handle to pbjlr (see GCBO)
% eventdata reserved - to be defined in a future version
of MATLAB

```

```

% handles      structure with handles and user data (see
GUIDATA)

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata,
handles)
% hObject      handle to pushbutton8 (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)
load JalurTerbaik
load XYkota

figure(1)
h=XYkota;
urt=JalurTerbaik;
x1=[]
y2=[]
for nn=urt ;
    p=h(nn,:)
    x=p(1,1)
    y=p(1,2)
    plot(x,y,'*r')
    hold on
    pause(0.2)
    x1=[x1 x]
    y2=[y2 y]
    text(x,y,[' \leftarrow', num2str(nn)] , 'FontSize',9)
end
hold on
x1=[x1,x1(1)]
y2=[y2,y2(1)]

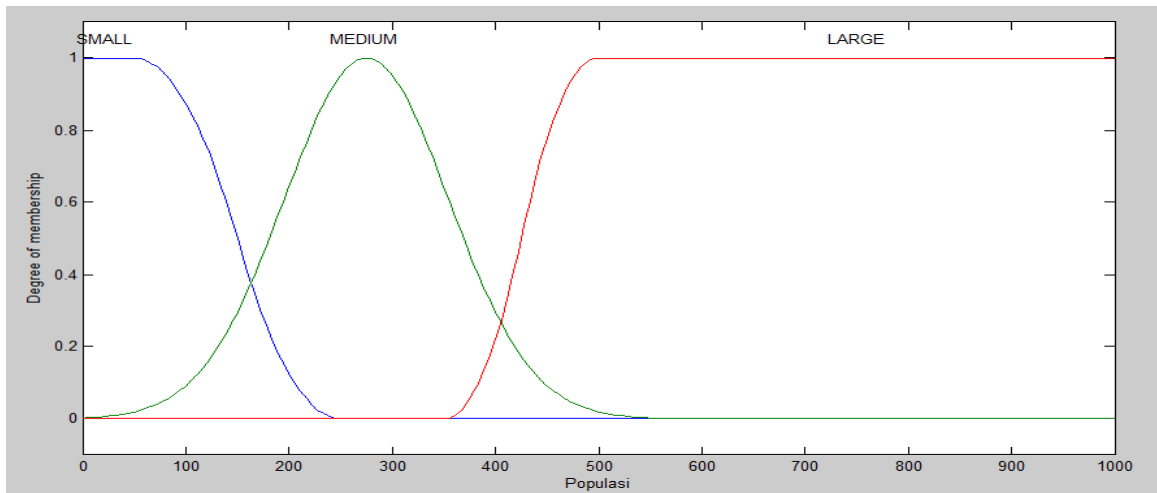
figure(1)
plot(x1,y2,'-r')
xlabel('koordinat x')
ylabel('koordinat y')
title('PLOT KOORDINAT')

```


Lampiran 6

Semesta Pembicaraan, Domain, Fungsi Keanggotaan dan Aturan Fuzzy

1. Semesta Pembicaraan, Domain dan Fungsi Keanggotaan Populasi



➤ Semesta pembicaraan: $[0, 1000]$

➤ Domain SMALL: $[50, 250]$

Fungsi Keanggotaan:

$$S(x; 50, 150, 250) = \begin{cases} 1; & x \leq 50 \\ 1 - 2 \left(\frac{x - 50}{250 - 50} \right)^2; & 50 \leq x \leq 150 \\ 2 \left(\frac{250 - x}{250 - 50} \right)^2; & 150 \leq x \leq 250 \\ 0; & x \geq 250 \end{cases}$$

➤ Domain MEDIUM: $[80, 275]$

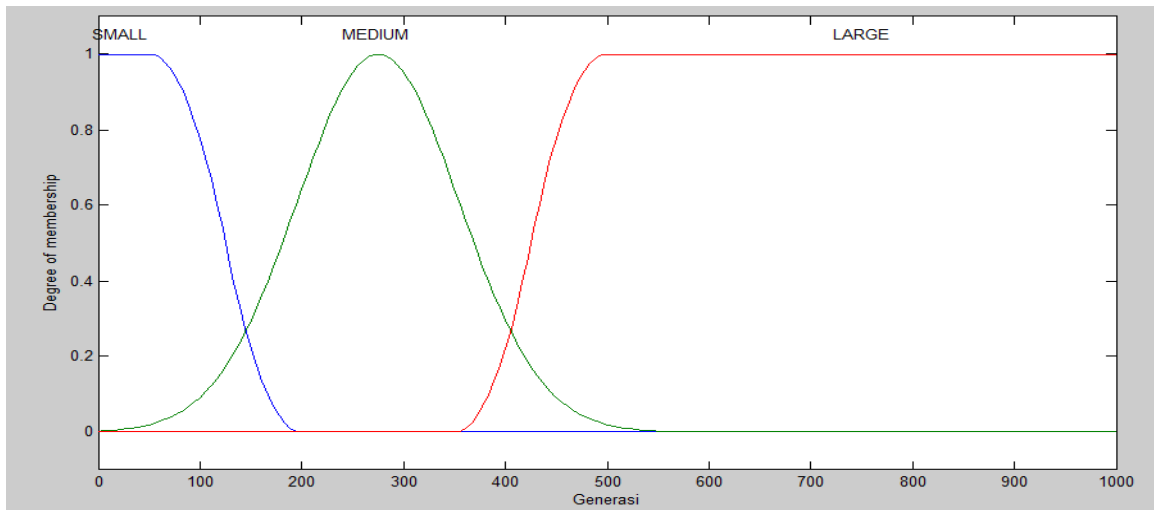
Fungsi Keanggotaan: $S(x; 80, 275) = e^{-(x-275)^2/2(80)^2}$

➤ Domain LARGE: [350, 500]

Fungsi Keanggotaan:

$$S(x; 350, 425, 500) = \begin{cases} 0; & x \leq 350 \\ 2 \left(\frac{x - 350}{500 - 350} \right)^2; & 350 \leq x \leq 425 \\ 1 - 2 \left(\frac{500 - x}{500 - 350} \right)^2; & 425 \leq x \leq 500 \\ 1; & x \geq 500 \end{cases}$$

2. Semesta Pembicaraan, Domain dan Fungsi Keanggotaan Generasi



➤ Semesta pembicaraan: [0, 1000]

➤ Domain SHORT: [50, 200]

Fungsi Keanggotaan:

$$S(x; 50, 125, 200) = \begin{cases} 1; & x \leq 50 \\ 1 - 2 \left(\frac{x - 50}{200 - 50} \right)^2; & 50 \leq x \leq 125 \\ 2 \left(\frac{200 - x}{200 - 50} \right)^2; & 125 \leq x \leq 200 \\ 0; & x \geq 200 \end{cases}$$

- Domain MEDIUM: [80, 275]

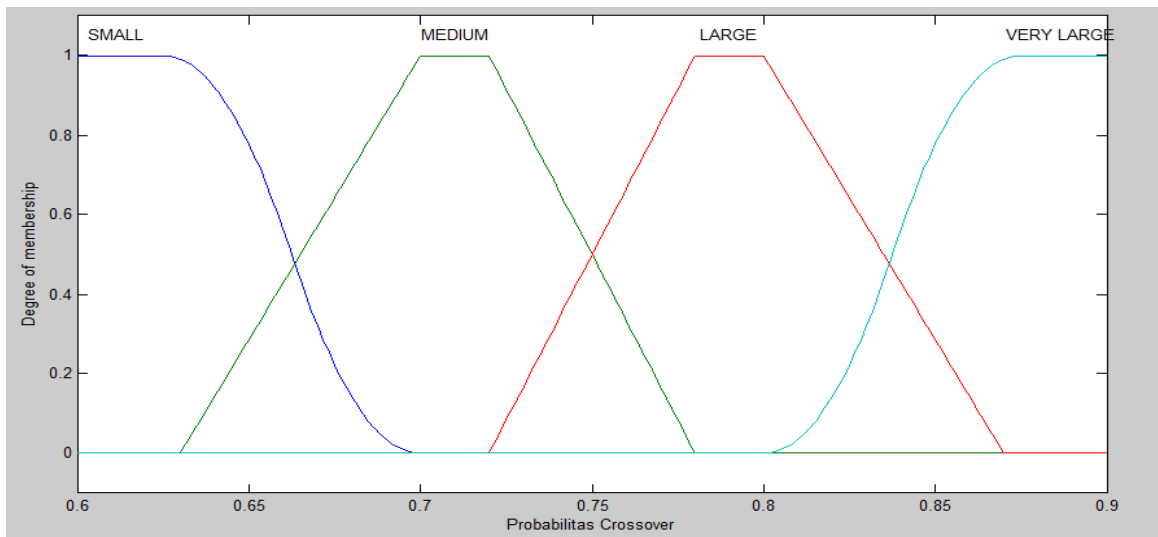
Fungsi Keanggotaan: $S(x; 80, 275) = e^{-(x-275)^2/2(80)^2}$

- Domain LONG: [350, 500]

Fungsi Keanggotaan:

$$S(x; 350, 425, 500) = \begin{cases} 0; x \leq 350 \\ 2 \left(\frac{x - 350}{500 - 350} \right)^2; 350 \leq x \leq 425 \\ 1 - 2 \left(\frac{500 - x}{500 - 350} \right)^2; 425 \leq x \leq 500 \\ 1; x \geq 500 \end{cases}$$

3. Semesta Pembicaraan, Domain dan Fungsi Keanggotaan Crossover



- Semesta pembicaraan: [0.6, 0.9]

➤ Domain SMALL: [0.625, 0.7]

Fungsi Keanggotaan:

$$S(x; 0.625, 0.6625, 0.7) = \begin{cases} 1; x \leq 0,625 \\ 1 - 2 \left(\frac{x - 0,625}{0,7 - 0,625} \right)^2; 0,625 \leq x \leq 0,6625 \\ 2 \left(\frac{0,7 - x}{0,7 - 0,625} \right)^2; 0,6625 \leq x \leq 0,7 \\ 0; x \geq 0,7 \end{cases}$$

➤ Domain MEDIUM: [0.63, 0.7, 0.72, 0.78]

Fungsi Keanggotaan:

$$S(x; 0.63, 0.7, 0.72, 0.78) = \begin{cases} 0; x \leq 0,63 \text{ atau } x \geq 0,78 \\ \frac{x - 0,63}{0,7 - 0,63}; 0,63 \leq x \leq 0,7 \\ 1; 0,7 \leq x \leq 0,72 \\ \frac{0,78 - x}{0,78 - 0,72}; 0,72 \leq x \leq 0,78 \end{cases}$$

➤ Domain LARGE: [0.72, 0.78, 0.8, 0.87]

Fungsi Keanggotaan:

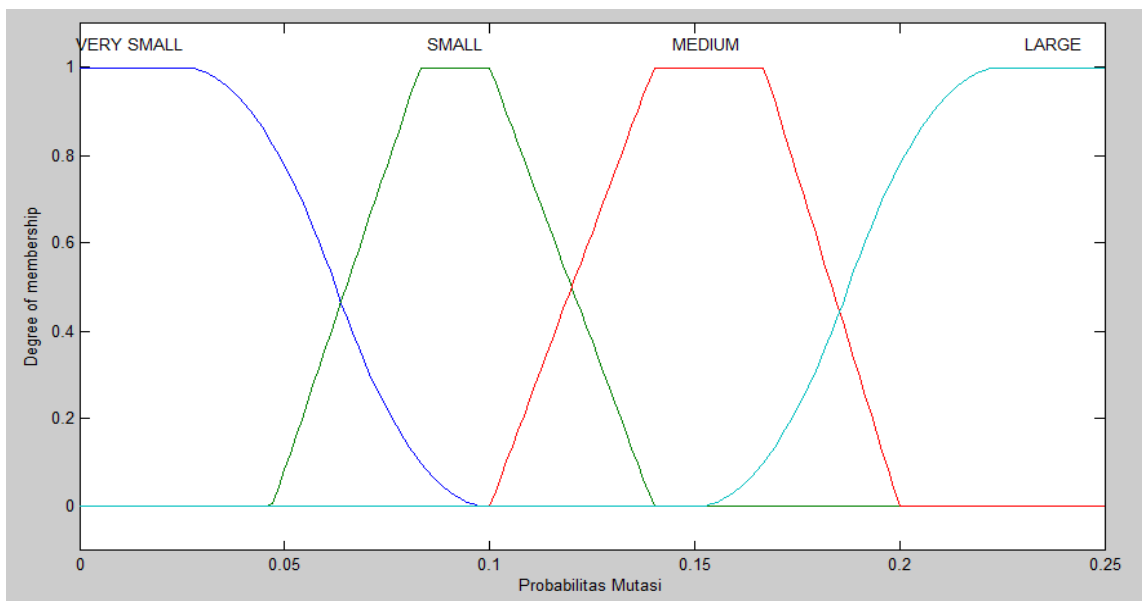
$$S(x; 0.72, 0.78, 0.8, 0.87) = \begin{cases} 0; x \leq 0,72 \text{ atau } x \geq 0,87 \\ \frac{x - 0,72}{0,78 - 0,72}; 0,72 \leq x \leq 0,78 \\ 1; 0,78 \leq x \leq 0,8 \\ \frac{0,87 - x}{0,87 - 0,8}; 0,8 \leq x \leq 0,87 \end{cases}$$

➤ Domain VERY LARGE: [0.8, 0.875]

Fungsi Keanggotaan:

$$S(x; 0.8, 0.8375, 0.875) = \begin{cases} 0; & x \leq 0,8 \\ 2 \left(\frac{x - 0,8}{0,875 - 0,8} \right)^2; & 0,8 \leq x \leq 0,8375 \\ 1 - 2 \left(\frac{0,875 - x}{0,875 - 0,8} \right)^2; & 0,8375 \leq x \leq 0,875 \\ 1; & x \geq 0,875 \end{cases}$$

4. Semesta Pembicaraan, Domain dan Fungsi Keanggotaan Mutasi



➤ Semesta pembicaraan: [0, 0.25]

➤ Domain VERY SMALL: [0.025, 0.1]

Fungsi Keanggotaan:

$$S(x; 0.025, 0.0625, 0.1) = \begin{cases} 1; & x \leq 0,025 \\ 1 - 2 \left(\frac{x - 0,025}{0,025 - 0,1} \right)^2; & 0,025 \leq x \leq 0,0625 \\ 2 \left(\frac{0,1 - x}{0,1 - 0,025} \right)^2; & 0,0625 \leq x \leq 0,1 \\ 0; & x \geq 0,1 \end{cases}$$

➤ Domain SMALL: [0.047, 0.083, 0.1, 0.14]

Fungsi Keanggotaan:

$$S(x; 0.047, 0.083, 0.1, 0.14) = \begin{cases} 0; x \leq 0,047 \text{ atau } x \geq 0,14 \\ \frac{x - 0,047}{0,083 - 0,047}; 0,047 \leq x \leq 0,083 \\ 1; 0,083 \leq x \leq 0,1 \\ \frac{0,14 - x}{0,14 - 0,1}; 0,1 \leq x \leq 0,14 \end{cases}$$

➤ Domain MEDIUM: [0.1, 0.14, 0.167, 0.2]

Fungsi Keanggotaan:

$$S(x; 0.1, 0.14, 0.167, 0.2) = \begin{cases} 0; x \leq 0,1 \text{ atau } x \geq 0,2 \\ \frac{x - 0,1}{0,14 - 0,1}; 0,1 \leq x \leq 0,14 \\ 1; 0,14 \leq x \leq 0,167 \\ \frac{0,2 - x}{0,2 - 0,167}; 0,167 \leq x \leq 0,2 \end{cases}$$

➤ Domain LARGE: [0.15, 0.225]

Fungsi Keanggotaan:

$$S(x; 0.15, 0.1875, 0.225) = \begin{cases} 0; x \leq 0,15 \\ 2 \left(\frac{x - 0,15}{0,225 - 0,15} \right)^2; 0,15 \leq x \leq 0,1875 \\ 1 - 2 \left(\frac{0,225 - x}{0,225 - 0,15} \right)^2; 0,1875 \leq x \leq 0,225 \\ 1; x \geq 0,225 \end{cases}$$

5. Aturan Fuzzy

- *IF (Populasi is SMALL) AND (Generasi is SHORT)*

THEN (ProbCrossover is MEDIUM) AND (ProbMutasi is LARGE).

- *IF (Populasi is MEDIUM) AND (Generasi is SHORT)*

THEN (ProbCrossover is SMALL) AND (ProbMutasi is MEDIUM).

- *IF (Populasi is LARGE) AND (Generasi is SHORT)*

THEN (ProbCrossover is SMALL) AND (ProbMutasi is SMALL).

- *IF (Populasi is SMALL) AND (Generasi is MEDIUM)*

THEN (ProbCrossover is LARGE) AND (ProbMutasi is MEDIUM).

- *IF (Populasi is MEDIUM) AND (Generasi is MEDIUM)*

THEN (ProbCrossover is LARGE) AND (ProbMutasi is SMALL).

- *IF (Populasi is LARGE) AND (Generasi is MEDIUM)*

THEN (ProbCrossover is MEDIUM) AND (ProbMutasi is VERYSMALL).

- *IF (Populasi is SMALL) AND (Generasi is LONG)*

THEN (ProbCrossover is VERYLARGE) AND (ProbMutasi is SMALL).

- *IF (Populasi is MEDIUM) AND (Generasi is LONG)*

THEN (ProbCrossover is VERYLARGE) AND (ProbMutasi is VERYSMALL).

- *IF (Populasi is LARGE) AND (Generasi is LONG)*

THEN (ProbCrossover is LARGE) AND (ProbMutasi is VERYSMALL).