

VEHICLE ROUTING PROBLEM DENGAN ALGORITMA
GENETIKA BERBASIS *FUZZY LOGIC CONTROLLER*

Skripsi

Disusun untuk melengkapi syarat-syarat
guna memperoleh gelar Sarjana Sains



DEWANTI KUMALA SARI

3125120199

PROGRAM STUDI MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI JAKARTA

2017

LEMBAR PERSETUJUAN HASIL SIDANG SKRIPSI

VEHICLE ROUTING PROBLEM DENGAN ALGORITMA *GENETIKA* BERBASIS *FUZZY LOGIC CONTROLLER*

Nama : Dewanti Kumala Sari

No. Registrasi : 3125120199

	Nama	Tanda Tangan	Tanggal
Penanggung Jawab			
Dekan	: Prof. Dr. Suyono, M.Si. NIP. 19671218 199303 1 005
Wakil Penanggung Jawab			
Wakil Dekan I	: Dr. Muktiningsih, M.Si. NIP. 19640511 198903 2 001
Ketua	: Dr. Lukita Ambarwati, S.Pd., M.Si. NIP. 19721026 200112 2 001
Sekretaris	: Ibnu Hadi, M.Si. NIP. 19810718 200801 1 017
Penguji Ahli	: Drs. Mulyono, M.Kom. NIP. 19660517 199403 1 003
Pembimbing I	: Ratna Widyati, S.Si., M.Kom. NIP. 19750925 200212 2 002
Pembimbing II	: Med Irzal, M. Kom. NIP. 19770615 200312 1 001

Dinyatakan lulus ujian skripsi tanggal: 14 Agustus 2017

ABSTRACT

DEWANTI KUMALA SARI, 3125120199. Vehicle Routing Problem With Genetic Algorithm Based on Fuzzy Logic Controller. Thesis. Faculty of Mathematics and Natural Science Jakarta State University. 2017.

Vehicle routing problem (VRP) has a key role in logistics management that plays a role in designing the optimal route of a number of vehicles at a customer service depot. Genetic algorithm is an algorithm Often used to solve the problem of optimal route search. The genetic algorithm will be combined with a fuzzy logic controller to control the probability value of the crossover and the probability of the mutation. From the analysis of Genetic algorithm based on fuzzy logic controller for the case discussed obtained the result that the optimal solution using the input population size 100 and generation 100 is better than the solutions obtained with the population and generation inputs are (100 and 100), (100 and 500), (100 And 1000), (500 and 100), and (1000 and 100). The best route is A-X-M-K-J-L-C-O-R-A for vehicle 1, A-F-T-B-G-H-D-Y-E-A for vehicle 2 and A-U-V-S-Q-W-N-I-P-A for vehicle 3 with the best total path length is 672.4 km by entering population 100 and generation 100.

Keywords : *vehicle routing problem*, genetic algorithm, fuzzy logic controller, probability of crossover, probability of mutation.

ABSTRAK

DEWANTI KUMALA SARI, 3125120199. *Vehicle Routing Problem* Dengan Algoritma Genetika Berbasis *Fuzzy Logic Controller*. Skripsi. Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Jakarta. 2017.

Vehicle routing problem (VRP) memiliki peranan pokok dalam manajemen logistik yang berperan dalam merancang rute optimal sejumlah kendaraan pada depot pelayanan pelanggan. Algoritma genetika merupakan algoritma yang sering digunakan untuk menyelesaikan masalah pencarian rute optimal. Algoritma genetika akan dikombinasikan dengan *fuzzy logic controller* untuk menentukan nilai probabilitas *crossover* dan probabilitas mutasi. Dari hasil analisis algoritma genetika berbasis *fuzzy logic controller* untuk kasus yang dibahas diperoleh hasil bahwa solusi optimal menggunakan *input* ukuran populasi 100 dan generasi 100 lebih baik dari solusi yang didapatkan dengan input populasi dan generasinya berturut-turut adalah (100 dan 100), (100 dan 500), (100 dan 1000), (500 dan 100), dan (1000 dan 100). Kemudian didapatkan rute terbaiknya adalah A-X-M-K-J-L-C-O-R-A untuk kendaraan 1, A-F-T-B-G-H-D-Y-E-A untuk kendaraan 2, dan A-U-V-S-Q-W-N-I-P-A untuk kendaraan 3 dengan panjang total jalur terbaiknya adalah 672.4 km dengan memasukkan populasi 100 dan generasi 100.

Kata kunci : *vehicle routing problem*, algoritma genetika, *fuzzy logic controller*, probabilitas *crossover*, probabilitas mutasi.

PERSEMBAHANKU...

” Karena sesungguhnya sesudah kesulitan itu ada kemudahan. Sesungguhnya sesudah kesulitan itu ada kemudahan. ”

-Q.S Al-Insyirah : 5-6

” Learn from yesterday, live for today, hope for tomorrow. The important thing is not to stop questioning ”

-Albert Einstein

Skripsi ini kupersembahkan untuk Ibu, Bapak, Bulik Susi.
”Terima kasih atas dukungan, do’a, serta kasih sayang kalian”.

KATA PENGANTAR

Puji syukur kepada Tuhan YME atas pengetahuan dan kemampuan sehingga penulis dapat menyelesaikan skripsi yang berjudul ” *Vehicle Routing Problem* dengan Algoritma Genetika berbasis *Fuzzy Logic Controller*” yang merupakan salah satu syarat dalam memperoleh gelar Sarjana Jurusan Matematika Universitas Negeri Jakarta.

Skripsi ini berhasil diselesaikan tidak terlepas dari adanya bantuan dari berbagai pihak. Oleh karena itu, dalam kesempatan ini penulis ingin menyampaikan terima kasih terutama kepada:

1. Ibu Ratna Widyati, S.Si., M.Kom. selaku Dosen Pembimbing I dan Bapak Med Irzal, M.Kom. selaku Dosen Pembimbing II, yang telah meluangkan waktunya dalam memberikan bimbingan, saran, nasehat serta arahan sehingga skripsi ini dapat menjadi lebih baik dan terarah.
2. Ibu Dr. Lukita Ambarwati, S. Pd, M. Si., selaku Koordinator Prodi Matematika FMIPA UNJ yang telah banyak membantu penulis.
3. Ibu Ratna Widyati, S. Si, M. Kom., selaku Pembimbing Akademik atas segala bimbingan dan kerja sama Ibu selama perkuliahan, dan seluruh Bapak/Ibu dosen atas pengajarannya yang telah diberikan, serta karyawan/karyawati FMIPA UNJ yang telah memberikan informasi yang penulis butuhkan dalam menyelesaikan skripsi.
4. Keluargaku tercinta : Ibu, Bapak, Ardi dan Lina yang meskipun jauh selalu mendukung, memberi motivasi, dan setia membantu penulis dengan penuh cinta dan kasih sayang yang tulus.

5. Bulik Susi dan keluarga yang selalu memberikan dukungan selama penulis menempuh pendidikan dari SMA sampai kuliah.
6. Cantik (Aan, Atika, Almira, Alphien, Bety, Dwi, Fatmah, Yohana, Zuhairini), selaku teman dekat masa perkuliahan yang selalu memberikan semangat dan seluruh mahasiswa matematika angkatan 2012 dan 2011, selaku teman dan kakak tingkat penulis yang selalu memberikan masukan ketika penulis menyusun skripsi ini.
7. Bobby Reynaldo, selaku teman penulis yang telah banyak memberikan masukan dalam proses penyusunan skripsi ini.
8. Fitria dan Arofah, sahabat yang selalu memotivasi dan menyemangati penulis.
9. Bebi, Jimmy dan Gembul, kucing-kucing kesayangan penulis yang selalu menemani di kala mengerjakan skripsi.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Masukan dan kritikan akan sangat berarti. Semoga skripsi ini dapat bermanfaat bagi pembaca sekalian.

Jakarta, Agustus 2017

Dewanti Kumala Sari

DAFTAR ISI

ABSTRACT	i
ABSTRAK	ii
KATA PENGANTAR	iv
DAFTAR ISI	viii
DAFTAR SIMBOL	ix
DAFTAR TABEL	ix
DAFTAR GAMBAR	xi
I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah	3
1.3 Pembatasan Masalah	3
1.4 Tujuan Penulisan	4
1.5 Manfaat Penulisan	4
1.6 Metode Penelitian	4
II LANDASAN TEORI	5
2.1 Teori Graf	5
2.1.1 Definisi Graf	5
2.1.2 Graf Terhubung	6
2.1.3 Graf Berarah	7
2.1.4 Graf Berbobot	7

2.1.5	Lintasan Hamilton	8
2.2	<i>Vehicle Routing Problem</i>	9
2.2.1	Formulasi Matematika VRP	11
2.3	Algoritma Genetika	17
2.3.1	Parameter Algoritma Genetika	17
2.3.2	Komponen Utama Algoritma Genetika	19
2.4	Logika Fuzzy	24
2.4.1	Himpunan Fuzzy	25
2.4.2	Fungsi Keanggotaan Fuzzy	29
2.4.3	Sistem Inferensi Fuzzy	32
2.4.4	<i>Fuzzy Logic Controller</i>	34
2.5	Algoritma Genetika berbasis <i>Fuzzy Logic Controller</i>	36
2.6	Implementasi Manual	37

III PEMBAHASAN **51**

3.1	Parameter Algoritma Genetika berbasis <i>Fuzzy Logic Controller</i> .	51
3.2	Langkah Penyelesaian	52
3.2.1	<i>Fuzzy Logic Controller</i>	52
3.2.2	Algoritma genetika dengan <i>fuzzy logic controller</i>	53
3.3	Studi Kasus	54
3.4	Hasil dan Pembahasan	56
3.4.1	Perhitungan menggunakan populasi 100 dan generasi 100	56
3.4.2	Perhitungan menggunakan populasi 100 dan generasi 500	58
3.4.3	Perhitungan menggunakan populasi 100 dan generasi 1000	59
3.4.4	Perhitungan menggunakan populasi 500 dan generasi 100	61
3.4.5	Perhitungan menggunakan populasi 1000 dan generasi 100	62

3.5 Analisis Penyelesaian <i>Vehicle Routing Problem</i> dengan Algoritma Genetika berbasis <i>Fuzzy Logic Controller</i>	64
IV PENUTUP	66
4.1 Kesimpulan	66
4.2 Saran	68
DAFTAR PUSTAKA	69
LAMPIRAN-LAMPIRAN	71

DAFTAR TABEL

2.1	Tabel Simetris dari Biaya Antar Lokasi	13
2.2	<i>Savings</i> setelah diurutkan	14
2.3	Jumlah Permintaan	16
2.4	Rute Hasil Perhitungan	16
3.1	Pembagian rute tempuh VRP	55
3.2	Hasil perhitungan pada kendaraan 1	57
3.3	Hasil perhitungan pada kendaraan 2	57
3.4	Hasil perhitungan pada kendaraan 3	58
3.5	Hasil perhitungan pada kendaraan 1	58
3.6	Hasil perhitungan pada kendaraan 2	59
3.7	Hasil perhitungan pada kendaraan 3	59
3.8	Hasil perhitungan pada kendaraan 1	60
3.9	Hasil perhitungan pada kendaraan 2	60
3.10	Hasil perhitungan pada kendaraan 3	60
3.11	Hasil perhitungan pada kendaraan 1	61
3.12	Hasil perhitungan pada kendaraan 2	61
3.13	Hasil perhitungan pada kendaraan 3	62
3.14	Hasil perhitungan pada kendaraan 1	62
3.15	Hasil perhitungan pada kendaraan 2	63
3.16	Hasil perhitungan pada kendaraan 3	63
3.17	Tabel Hasil Panjang Jalur Terbaik	64
3.18	Tabel Hasil Panjang Jalur Terbaik dan Rute Terbaik	65
3.19	Tabel Hasil Probabilitas <i>Crossover</i> dan Probabilitas Mutasi	65

DAFTAR GAMBAR

2.1	Contoh Graf G	6
2.2	(a) Graf Terhubung, (b) Graf Tak Terhubung	7
2.3	(a) Graf Berarah, (b) Graf Tak Berarah	7
2.4	Graf Berbobot	8
2.5	Hamiltonian pada graf G	9
2.6	Visualisasi <i>Vehicle Routing Problem</i>	10
2.7	Contoh soal penyelesaian VRP dengan metode <i>Savings</i>	13
2.8	Graf VRP yang terbentuk	16
2.9	Diagram Alir Proses Crossover	21
2.10	Penentuan <i>mapping section</i>	22
2.11	Pembentukan 2 kromosom anak	22
2.12	Pemetaan pada kromosom anak	23
2.13	Perbaikan kromosom	23
2.14	Diagram Alir Proses Mutasi	24
2.15	Logika Fuzzy sebagai <i>Black Box</i>	25
2.16	Himpunan fuzzy untuk variabel umur	27
2.17	Himpunan fuzzy untuk variabel temperatur	27
2.18	Representasi Linear Naik	29
2.19	Representasi Linear Turun	30
2.20	Representasi Fungsi Sigmoid	31
2.21	Representasi Fungsi Trapesium	32
2.22	Proses Defuzzifikasi	34
2.23	Lintasan VRP yang akan dilalui	37
3.1	<i>Fuzzy Logic Controller</i> dengan Sistem Inferensi Fuzzy Mamdani	52

3.2	Diagram Alir Algoritma Genetika berbasis <i>Fuzzy Logic Controller</i>	54
3.3	<i>Scatter Plot</i> Letak Lokasi Customer	55
3.4	<i>Output</i> pencarian probabilitas <i>crossover</i> dan mutasi	56
3.5	Plot Rute Terbaik	64

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Transportasi merupakan salah satu komponen yang sangat penting dalam sistem manajemen logistik. Peningkatan efisiensi dari sistem transportasi dapat dilakukan dengan memaksimalkan utilitas dari alat transportasi yang ada. Untuk mengurangi biaya transportasi dan juga untuk meningkatkan pelayanan terhadap para *customer*, perlu dicari jalur atau rute terbaik yang dapat meminimalkan jarak dan waktu.

Permasalahan yang bertujuan untuk membuat suatu rute yang optimal, untuk beberapa kendaraan, agar dapat melayani sejumlah konsumen, disebut sebagai *Vehicle Routing Problem* (VRP). VRP merupakan suatu permasalahan penting yang terdapat pada sistem transportasi yang bertujuan meminimalkan total jarak tempuh. VRP termasuk ke dalam kelas *Nondeterministic Polynomial-Hard* (NPH) artinya parameter yang mempengaruhi sangat kompleks dan penyelesaiannya tidak dapat diselesaikan dengan sebuah algoritma linier saja, karena membutuhkan waktu yang sangat lama, yang pada umumnya menggunakan pendekatan heuristik untuk mencari solusinya.

Terdapat beberapa penelitian tentang VRP yang dilakukan sebelumnya yaitu (Hermansyah,2011) yang menyelesaikan VRP dengan Algoritma Genetika standar. Penelitian Surekha dan Sumathi (2011) membahas penyelesaian VRP *with Time Window* (VRPTW) menggunakan Algoritma Genetika dengan kendala *multi-depot*. Pada kedua penelitian tersebut parameter pada

Algoritma Genetika bersifat tetap. Hal ini dapat menyebabkan Algoritma Genetika menjadi lebih rentan terjebak dalam optimum lokal. Untuk itu Algoritma Genetika akan dikombinasikan dengan *Fuzzy Logic Controller*. *Fuzzy logic controller* digunakan untuk membuat parameter Algoritma Genetika menjadi adaptif. Beberapa peneliti melaporkan penggunaan *fuzzy logic controller* untuk menentukan nilai parameter Algoritma Genetika secara otomatis (*automatic fine-tuning*) berdasarkan informasi yang diperoleh dari populasi sebelumnya. Kombinasi Algoritma Genetika dengan *Fuzzy Logic Controller* ditunjukkan mampu memberikan kualitas solusi yang lebih baik dibandingkan dengan Algoritma Genetika standar (Admi Syarif,2014). Selain itu juga dapat menghindari terjadinya konvergensi prematur yang sering terjadi pada Algoritma Genetika standar. Konvergensi prematur adalah keadaan pada saat solusi yang dihasilkan terjebak pada optimum lokal mulai dari generasi awal sampai akhir.

Logika fuzzy sebagai komponen utama pembangun *soft computing*, terbukti telah memiliki kinerja yang sangat baik untuk menyelesaikan masalah-masalah yang mengandung ketidakjelasan. Algoritma genetika dapat menghasilkan nilai fitness rata-rata yang tertinggi ketika dikombinasikan dengan himpunan fuzzy untuk membentuk suatu fungsi evaluasi. Himpunan fuzzy dapat mendefinisikan baik atau buruk suatu solusi yang dihasilkan berdasarkan derajat keanggotaannya dalam suatu himpunan fuzzy.

Berdasarkan permasalahan di atas, maka penulis menggunakan Algoritma Genetika berbasis *Fuzzy Logic Controller* untuk menyelesaikan kasus VRP. Untuk itu pada penulisan skripsi ini penulis mengangkat judul ” *Vehicle Routing Problem* dengan Algoritma Genetika Berbasis *Fuzzy Logic Controller*”.

1.2 Perumusan Masalah

Rumusan masalah yang dikaji berdasarkan latar belakang di atas adalah sebagai berikut:

1. Bagaimana hasil pencarian solusi optimal pada *Vehicle Routing Problem* dengan Algoritma Genetika berbasis *Fuzzy Logic Controller*?
2. Bagaimana rute terbaik hasil pencarian solusi optimal *Vehicle Routing Problem*?

1.3 Pembatasan Masalah

Pembatasan masalah dalam penulisan ini adalah:

1. Permasalahan yang akan diteliti hanya menggunakan depot tunggal.
2. Kendaraan selalu berangkat dan kembali di depot.
3. Pada permasalahan ini hanya mencari jarak terpendek dengan permintaan *customer* dan kapasitas kendaraan diabaikan.
4. Bobot antara titik yang ditentukan hanyalah bobot jarak dengan mengabaikan bobot-bobot lainnya, sehingga rute terbaik ditentukan berdasarkan jarak terpendek antara titik dari awal perjalanan sampai kembali lagi.
5. Setiap kendaraan akan mengunjungi masing-masing 8 kota dan pembagian kota ditentukan berdasarkan koordinat letak kota.

1.4 Tujuan Penulisan

Tujuan dalam penulisan skripsi ini adalah untuk memperoleh solusi optimal dari *Vehicle Routing Problem* dengan Algoritma Genetika berbasis *Fuzzy Logic Controller*.

1.5 Manfaat Penulisan

Manfaat yang diharapkan dari penulisan skripsi ini adalah

1. Dapat menambah wawasan dan pengetahuan tentang pendistribusian dan transportasi suatu jaringan.
2. Dapat menerapkan Algoritma Genetika berbasis *Fuzzy Logic Controller* untuk memperoleh solusi optimal pada permasalahan *Vehicle Routing Problem*.

1.6 Metode Penelitian

Skripsi ini merupakan kajian pustaka dalam bidang matematika terapan yaitu komputasi yang didasarkan pada buku-buku dan jurnal-jurnal tentang teori logika fuzzy dan algoritma genetika.

BAB II

LANDASAN TEORI

Pada bab ini, akan dibahas mengenai pengertian teori graf, *Vehicle Routing Problem*, algoritma genetika, logika fuzzy, dan hubungan antara algoritma genetika dengan *fuzzy logic controller*.

2.1 Teori Graf

2.1.1 Definisi Graf

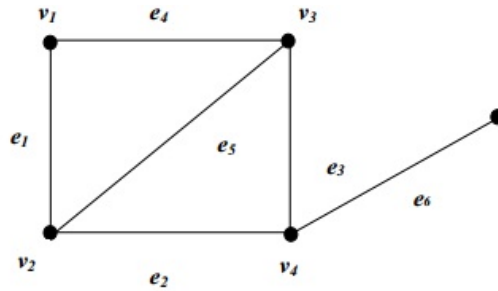
Sebuah graf G berisikan dua himpunan yaitu himpunan berhingga tak kosong $V(G)$ dari objek-objek yang disebut titik dan himpunan berhingga $E(G)$ yang elemen-elemennya disebut sisi sedemikian hingga setiap elemen e dalam $E(G)$ merupakan pasangan tak berurutan dari titik-titik di $V(G)$. Himpunan $V(G)$ disebut himpunan titik G dan himpunan $E(G)$ disebut himpunan sisi G . Misalkan u dan v dua titik di G dan $e = u, v$ (sering ditulis $e = uv$ adalah sebuah sisi G . Titik u dan titik v berhubungan langsung (*adjacent*) di G ; sisi e menghubungkan (*joining*) titik u dan titik v di G ; u dan v titik-titik akhir sisi e ; sisi e terkait (*incident*) dengan titik v dan titik u . (Budayasa, 2007: 1-2)

Definisi di atas menyatakan bahwa V tidak boleh kosong, sedangkan E boleh kosong. Jadi, sebuah graf dimungkinkan tidak mempunyai sisi satu buah pun, tetapi titiknya harus ada, minimal satu. Graf yang hanya mempunyai satu buah titik tanpa sebuah sisi pun dinamakan graf trivial (Munir, 2010: 356)

Contoh 2.1

Sebuah graf $G = (V, E)$ dengan $V = \{v_1, v_2, v_3, v_4, v_5\}$ dan $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$

. $e_1 = v_1v_2$; $e_2 = v_2v_4$; $e_3 = v_3v_4$; $e_4 = v_1v_3$; $e_5 = v_2v_3$; $e_6 = v_4v_5$



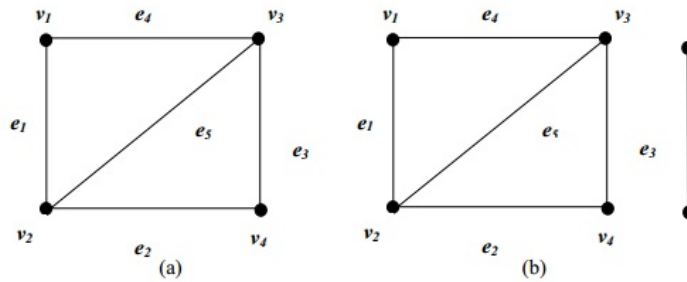
Gambar 2.1: Contoh Graf G

2.1.2 Graf Terhubung

Suatu graf G dikatakan terhubung (*connected*) jika untuk setiap dua titik G yang berbeda terdapat sebuah lintasan yang menghubungkan kedua titik tersebut. Sebaliknya graf G adalah sebuah graf bagian terhubung maksimal (titik dan sisi) dari G (Budayasa, 2007:8). Graf H dikatakan graf bagian terhubung maksimal dari graf G apabila tidak ada graf bagian lain dari G yang terhubung dan memuat H . Jadi setiap graf terhubung memiliki tepat satu komponen sedangkan graf tidak terhubung memiliki paling sedikit dua komponen.

Contoh 2.2

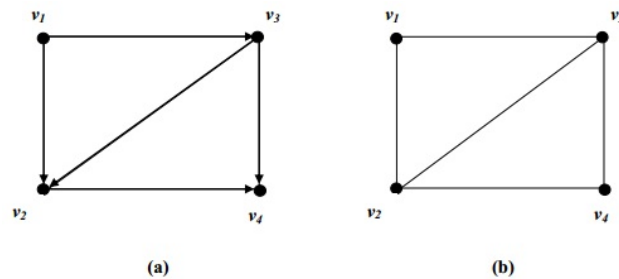
Diberikan sebuah graf seperti Gambar 2.2



Gambar 2.2: (a) Graf Terhubung, (b) Graf Tak Terhubung

2.1.3 Graf Berarah

Graf berarah adalah graf yang sisinya mempunyai orientasi arah. Pada graf tak berarah, urutan pasangan simpul yang dihubungkan oleh sisi diperhatikan. Jadi (u, v) dan (v, u) adalah sisi yang berbeda. (Munir, 2010: 358).

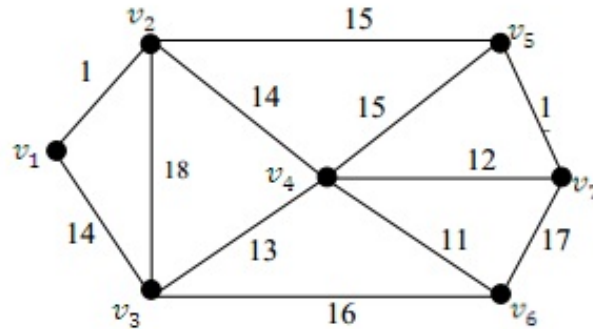


Gambar 2.3: (a) Graf Berarah, (b) Graf Tak Berarah

2.1.4 Graf Berbobot

Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot). Bobot pada setiap sisi dapat menyatakan jarak dua buah kota, biaya perjalanan antara dua buah kota, waktu tempuh pesan dari simpul ke komunikasi

lain, dan sebagainya (Munir, 2005: 357)



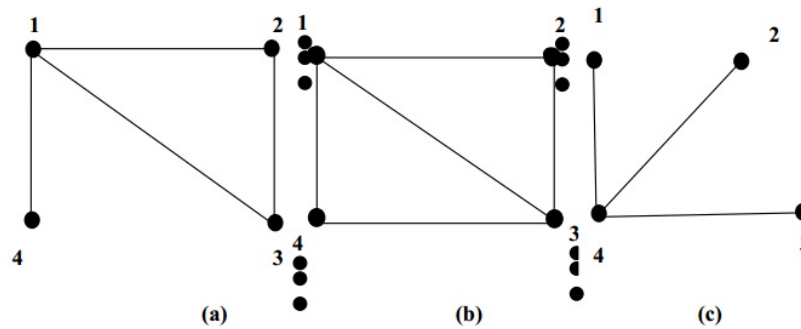
Gambar 2.4: Graf Berbobot

2.1.5 Lintasan Hamilton

Misalkan G sebuah graf. Sebuah lintasan G yang memuat semua titik di G disebut lintasan Hamilton. Graf non Hamilton yang memuat lintasan Hamilton disebut graf semi-Hamilton (Budayasa, 2007: 130). Sirkuit Hamilton adalah sirkuit yang melalui tiap titik di dalam graf tepat satu kali, kecuali titik asal (sekaligus titik akhir) yang dilalui dua kali. Graf Hamilton adalah graf yang memiliki sirkuit Hamilton (Munir, 2005: 232).

Keterangan gambar:

1. Graf G memiliki lintasan Hamilton misalnya 3-2-1-4
2. Graf G memiliki sirkuit Hamilton misalnya 1-2-3-4-1
3. Graf G tidak memiliki lintasan maupun sirkuit Hamilton

Gambar 2.5: Hamiltonian pada graf G

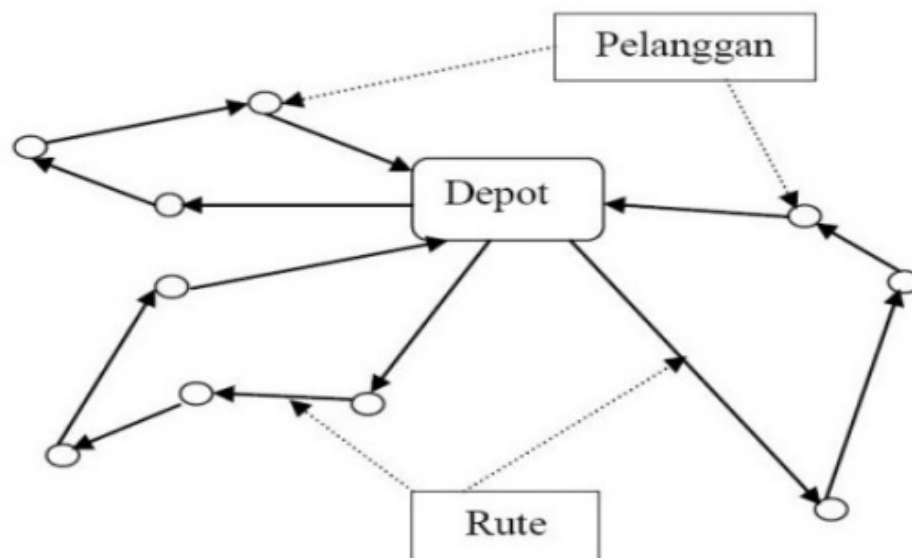
2.2 *Vehicle Routing Problem*

Kallehauge dkk. (2001) mendefinisikan permasalahan m -TSP sebagai salah satu variasi dari TSP, dimana terdapat m -*salesman* mengunjungi sejumlah kota dan tiap kota hanya dapat dikunjungi oleh tepat satu *salesman* saja. Tiap *salesman* berawal dari suatu depot dan pada akhir perjalanannya juga harus kembali ke depot tersebut. Permasalahan m -TSP sering disebut sebagai *vehicle routing problem* (VRP), dimana sebuah kota diasosiasikan sebagai sebuah *demand* atau konsumen, dan tiap kendaraan yang dipakai untuk perjalanan dianggap memiliki kapasitas tertentu. Total jumlah *demand* dalam suatu rute, tidak boleh melebihi kapasitas dari kendaraan yang ditugasi melewati rute tersebut. Hal ini membuat VRP kadang juga disebut sebagai *Capacitated Vehicle Routing Problem*. Sama seperti permasalahan TSP, dalam VRP juga terdapat suatu depot, dimana tiap kendaraan harus berangkat dan kembali ke depot itu. Dalam VRP, selain bertujuan untuk meminimalkan total jarak atau total biaya travel, dapat juga untuk meminimalkan jumlah kendaraan yang digunakan (K).

Toth dan Vigo menggambarkan *vehicle routing problem* sebagai suatu graf

lengkap $G = (V, A)$, di mana $V = (0, \dots, n)$ adalah himpunan titik dan A adalah himpunan busur. Node $i = 1, \dots, n$, menunjukkan pelanggan, sedangkan node 0 menunjukkan depot. Terkadang depot digambarkan juga dengan $n + 1$. Biaya *non negative*/ jarak tempuh (c_{ijk}), terkait dengan setiap busur $(i, j) \in A$ dan merupakan biaya travel yang dikeluarkan dalam perjalanan dari titik i ke titik j . Tujuan *vehicle routing problem* adalah untuk mengatur rute biaya terendah kendaraan sedemikian hingga:

1. Setiap rute dimulai dan diakhiri di depot.
2. Setiap pelanggan dikunjungi tepatnya sekali dengan satu kendaraan.
3. Jumlah permintaan dari rute kendaraan yang ada tidak melebihi kapasitas kendaraan.



Gambar 2.6: Visualisasi *Vehicle Routing Problem*

2.2.1 Formulasi Matematika VRP

Kallehauge dkk. memformulasikan sebuah model dari *vehicle routing problem* sebagai berikut:

$$\min C = \sum_{k=1}^K \sum_{i=0}^{N+1} \sum_{j=0}^{N+1} (c_{ij} x_{ijk}) \quad (2.1)$$

dengan kendala :

$$\sum_{k=1}^K \sum_{j=0}^{N+1} x_{ijk}; i = 1, 2, \dots, N \quad (2.2)$$

$$\sum_{i=1}^N d_i \sum_{j=0}^{N+1} x_{ijk} \leq v_k; k = 1, 2, \dots, K \quad (2.3)$$

$$\sum_{j=0}^{N+1} x_{0jk} = 1; k = 1, 2, \dots, K \quad (2.4)$$

$$\sum_{i=0}^{N+1} x_{ihk} - \sum_{j=0}^{N+1} x_{hjk} = 0; h = 1, 2, \dots, N; k = 1, 2, \dots, K \quad (2.5)$$

$$\sum_{i=0}^{N+1} x_{i, N+1, k} = 1; k = 1, 2, \dots, K \quad (2.6)$$

$$x_{ijk} \in \{0, 1\}; i = 0, 2, \dots, N+1; k = 1, 2, \dots, K \quad (2.7)$$

dengan

c_{ijk} = biaya perjalanan antara konsumen i dan j

K = nomor kendaraan

d_i = total permintaan kendaraan k sampai konsumen i .

N = nomor pelanggan (0 menunjukkan depot)

v_k = kapasitas maksimum kendaraan k

$$x_{ijk} = \begin{cases} 1, & \text{jika ada perjalanan kendaraan } k \text{ dari kota } i \text{ ke } j \\ 0, & \text{jika tidak ada perjalanan kendaraan } k \text{ dari kota } i \text{ ke } j \end{cases}$$

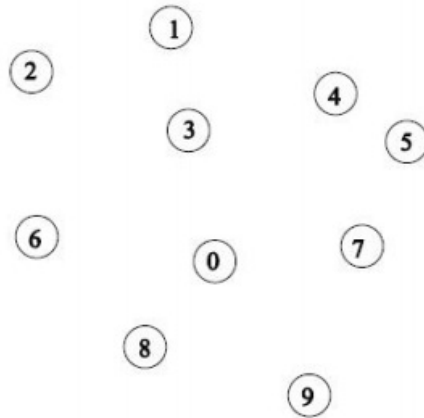
Persamaan (2.1) menunjukkan fungsi tujuan dari permasalahan ini, yaitu untuk meminimalkan total biaya perjalanan. Persamaan (2.2) menunjukkan bahwa tiap konsumen hanya dapat dilayani oleh satu kendaraan saja. Persamaan (2.3) digunakan untuk membatasi total jumlah permintaan yang dibawa oleh kendaraan k , tidak boleh melebihi kapasitas dari kendaraan tersebut. Persamaan (2.4)-(2.7) digunakan untuk memastikan bahwa tiap kendaraan berangkat dari depot 0, dan setelah selesai melayani seorang konsumen, kendaraan tersebut akan pergi, serta pada akhirnya, kendaraan tersebut akan kembali ke depot 0.

Salah satu metode sederhana yang digunakan untuk menyelesaikan VRP adalah Metode *Savings*. Tujuannya adalah untuk mengoptimalkan rute pengiriman sehingga semua barang dapat terkirim dengan biaya pengiriman minimal.

Langkah-langkah metode *Savings* adalah sebagai berikut:

1. Pertama, setiap n kendaraan pengangkut melayani satu pelanggan.
2. Hitung *savings* dengan menggabungkan siklus dengan sisi (i, j) menggunakan rumus $s_{ij} = c_{io} + c_{oj} - c_{ij}$ untuk setiap pasang simpul (i, j) .
3. Urutkan *savings* dari besar ke kecil.
4. Ambil sisi (i, j) dari bagian paling atas data *savings*. Gabungkan dua siklus yang terpisah dengan sisi (i, j) .
5. Ulangi data keempat hingga semua data pada daftar *savings* telah dikerjakan atau kapasitas sudah penuh.

Berikut adalah sebuah graf dengan pusat 0 dan akan mengirimkan barang ke 9 pelanggan dengan kapasitas kendaraan pengangkut $K=40$



Gambar 2.7: Contoh soal penyelesaian VRP dengan metode *Savings*

Dari Gambar 2.7 terdapat 9 pelanggan yang harus dikunjungi dan akan dibentuk rute *vehicle routing problem*. Kemudian diketahui biaya antar lokasi yang disajikan pada Tabel 2.1

Tabel 2.1: Tabel Simetris dari Biaya Antar Lokasi

c_{ij}	0	1	2	3	4	5	6	7	8	9
0	-	12	11	7	10	10	9	8	6	12
1	12	-	8	5	9	12	14	16	17	22
2	11	8	-	9	15	17	8	18	14	22
3	7	5	9	-	7	9	11	12	12	17
4	10	9	15	7	-	3	17	7	15	18
5	10	12	17	9	3	-	18	6	15	15
6	9	14	8	11	17	18	-	16	8	16
7	8	16	18	12	7	6	16	-	11	11
8	6	17	14	12	15	15	8	11	-	10
9	12	22	22	17	18	15	16	11	10	-

Penyelesaian:

1. Dengan menghitung $s_{ij} = c_{io} + c_{oj} - c_{ij}$, maka diperoleh tabel *savings* pada tabel

Tabel 2.2: *Savings* setelah diurutkan

s_{ij}	1	2	3	4	5	6	7	8	9
1		15	14	13	10	7	4	1	2
2			9	6	4	12	1	3	1
3				10	8	5	3	1	2
4					17	2	11	1	4
5						1	12	1	7
6							1	7	5
7								3	9
8									8

2. Urutkan *savings* dari besar ke kecil

$$(4, 5) = 10 + 10 - 3 = 17$$

$$(1, 2) = 12 + 11 + 8 = 15$$

...(1,3),(1,4),(2,6),(5,7),(4,7), (1,5),(3,4),(2,3),(7,9),(3,5), (8,9),(1,6), dan seterusnya

3. Cari rutenya

Solusi inisialisasi : siklus $0 - 1 - 0$, $0 - 2 - 0$, ..., $0 - 9 - 0$

Sisi (4,5) : gabungkan siklus $0-4-0$ dan $0-5-0$, lalu didapat hasil = $0-4-5-0$, maka simpan $d_4 + d_5 = 20 < K$

Sisi (1,2) : gabungkan siklus $0-1-0$ dan $0-2-0$, lalu didapat hasil = $0-1-2-0$, maka simpan $d_1 + d_2 = 25 < K$

Sisi (1,3) : gabungkan siklus $0-1-2-0$ dan $0-3-0$, tetapi ternyata melebihi batas kapasitas, karena $d_1 + d_2 + d_3 = 43 > K$

Sisi (1,4) : gabungkan siklus 0-1-2-0 dan 0-4-0, tetapi ternyata melebihi batas kapasitas, karena $d_1 + d_2 + d_4 = 42 > K$

Sisi (2,6) : simpul 2 sudah termasuk dalam siklus 0-1-2-0, maka gabungkan siklus 0-1-2-0 dan 0-6-0. Lalu didapat hasil = 0-1-2-6-0, maka simpan $d_1 + d_2 + d_6 = 30 < K$

Sisi (5,7) : simpul 5 sudah termasuk dalam siklus 0-4-5-0, maka gabungkan siklus 0-4-5-0 dan 0-7-0. Lalu didapat hasil = 0-4-5-7-0, maka simpan $d_4 + d_5 + d_7 = 29 < K$

Sisi (4,7) : termasuk siklus 0-4-5-7-0

Sisi (1,5) : 0-4-5-7-0 simpul 5 merupakan bagian dari siklus 0-4-5-7-0, sehingga tidak mungkin bercabang di lokasi lain

Sisi (3,4) : gabungan siklus 0-3-0 dan 0-4-5-7-0, tetapi melebihi batas kapasitas : $d_3 + d_4 + d_5 + d_7 = 47 < K$

Sisi (2,3) : simpul 2 merupakan bagian dalam siklus 0-1-2-6-0

Sisi (7,9) : simpul 7 sudah termasuk dalam siklus 0-4-5-7-0, maka gabungkan siklus 0-4-5-7-0 dan 0-9-0. Lalu didapat hasil = 0-4-5-7-9-0, maka simpan $d_4 + d_5 + d_7 + d_9 = 35 < K$

Sisi (3,5) : simpul 2 merupakan bagian dalam siklus 0-4-5-7-9-0

Sisi (8,9) : simpul 9 sudah termasuk dalam siklus 0-4-5-7-9-0, maka gabungkan siklus 0-4-5-7-9-0 dan 0-8-0. Lalu didapat hasil = 0-4-5-7-8-9-0, maka simpan $d_4 + d_5 + d_7 + d_8 + d_9 = 39 < K$

Sisi (1,6) : simpul 1 dan 6 sudah masuk ke dalam siklus

Sisi (5,9) : simpul 5 dan 9 sudah masuk ke dalam siklus

Sisi (6,8) : gabungkan siklus 0-1-2-6-0 dan 0-4-5-7-8-9-0, tetapi ternyata melebihi kapasitas karena $(d_1 + d_2 + d_6) + (d_4 + d_5 + d_7 + d_8 + d_9) = 69 > K$

Tabel 2.3: Jumlah Permintaan

i	0	1	2	3	4	5	6	7	8	9
d_i	10	15	18	3	5	9	4	6	6	12

4. Diperoleh rute :

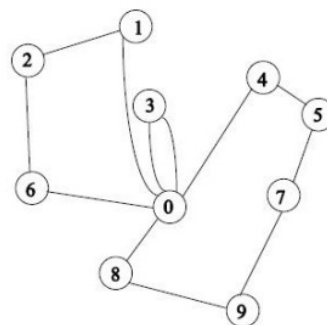
Tabel 2.4: Rute Hasil Perhitungan

Rute	Muatan	Biaya
0-3-0	18	14
0-1-2-6-0	30	37
0-4-5-7-8-9-0	39	46

Dari tabel diatas maka kita dapatkan total biaya perjalanan = $14+37+46$
 $= 97$

5. Graf yang terbentuk

Dari hasil perhitungan dengan metode *Savings* maka diperoleh graf sebagai berikut



Gambar 2.8: Graf VRP yang terbentuk

2.3 Algoritma Genetika

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. 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, integer maupun karakter.
2. Kromosom adalah gabungan gen-gen yang membentuk nilai tertentu.
3. Individu menyatakan satu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.
4. Populasi merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi.
5. Generasi menyatakan satu satuan siklus proses evolusi.
6. Nilai fitness menyatakan seberapa baik nilai dari suatu individu atau solusi yang didapatkan.

2.3.1 Parameter Algoritma Genetika

1. Probabilitas *Crossover* (P_c)

Parameter P_c mempunyai nilai 0-1. Nilai ini menggambarkan seberapa

sering *crossover* akan dilakukan. Mudah dipahami bahwa jika nilai P_c adalah 0, berarti tidak dilakukan proses *crossover* atau dengan kata lain seluruh kromosom pada generasi yang baru dibuat dengan kromosom pada generasi sebelumnya. Sebaliknya jika bernilai satu, berarti semua kromosom berpeluang akan melakukan *crossover*.

2. Probabilitas Mutasi (p_m)

Seperti halnya P_c , parameter P_m mempunyai nilai 0-1. Nilai ini menggambarkan seberapa sering mutasi akan dilakukan. Jika nilai parameter ini adalah 0, berarti tidak dilakukan proses mutasi terhadap kromosom. Kromosom anak (*offspring*) hanya diperoleh melalui proses *crossover*. Jika p_m bernilai satu, berarti semua kromosom berpeluang melakukan proses mutasi.

3. Ukuran Populasi (*Population size*)

Nilai parameter *PopSize* menunjukkan jumlah kromosom pada populasi. Apabila jumlah kromosom dalam suatu populasi terlalu kecil maka akan semakin sedikit kromosom yang melakukan *crossover* dan mutasi. Hal ini akan mempengaruhi kualitas solusi yang diperoleh.

4. Maksimum Generasi (*Maximum generation*)

Parameter *MaxGen* memberikan jumlah maksimum generasi yang dilakukan pada proses Algoritma Genetika. Dengan kata lain, parameter ini merupakan parameter yang dipakai sebagai kriteria pemberhentian proses Algoritma Genetika.

2.3.2 Komponen Utama Algoritma Genetika

Menurut S.Kusumadewi (2003) komponen utama Algoritma Genetika meliputi teknik pengkodean, inisialisasi, evaluasi fitness, seleksi, *crossover* dan mutasi.

1. Teknik Pengkodean (*encoding*)

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. Gen dapat direpresentasikan dalam bentuk *string*, pohon (*tree*), *array*, bilangan real, daftar aturan, elemen permutasi, elemen program, atau representasi lainnya yang dapat diimplementasikan untuk operator genetika.

2. Inisialisasi

Inisialisasi dilakukan untuk membangkitkan himpunan solusi baru secara acak yang terdiri atas sejumlah kromosom di dalam sebuah populasi. 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 objektifnya dievaluasi, kemudian mengubah fungsi objektif tersebut ke dalam fungsi *fitness*, un-

tuk maksimasi problem, fitness sama dengan fungsi objektifnya. Output dari fungsi *fitness* dipergunakan sebagai dasar untuk menyeleksi individu pada generasi berikutnya. Untuk permasalahan minimalisasi, nilai fitness adalah inversi dari nilai maksimal yang diharapkan. Proses inversi dapat dilakukan dengan rumusan $fitness = \frac{1}{f(x)}$ dengan x merupakan kromosom (Basuki, 2003:17).

$$fitness = A - f(x) \text{ atau } fitness = \frac{A}{f(f(x)+\epsilon)}$$

Keterangan :

A = konstanta yang ditentukan

x = individu (kromosom)

$f(x)$ = nilai fungsi kromosom

ϵ = bilangan kecil yang ditentukan untuk menghindari pembagi nol.

4. Seleksi

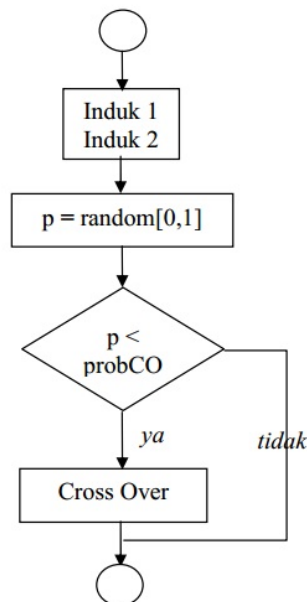
Seleksi ini bertujuan memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling baik. Metode seleksi yang digunakan dalam pembahasan skripsi ini adalah dengan *Roulette Wheel Selection*. Cara kerja metode ini adalah sebagai berikut.

- (a) Menghitung total fitness pada populasi awal.
- (b) Menghitung probabilitas masing-masing kromosom untuk terpilih ke generasi berikutnya (P_k).
- (c) Menghitung probabilitas kumulatif Q_k untuk masing-masing kromosom.
- (d) Menentukan kromosom mana yang terpilih dengan membangkitkan bilangan acak pada *range* $[0,1]$ sebanyak kromosom.

Apabila $Q_k - 1 < R_k < Q_k$ maka kromosom terpilih ke generasi berikutnya. (Admi Syarif, 2014, 23)

5. *Crossover*(Pindah Silang)

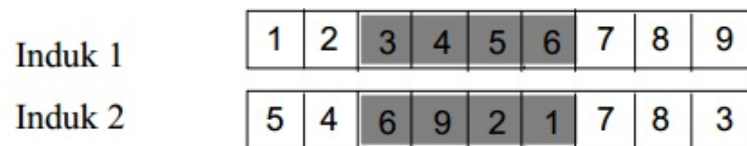
Crossover merupakan suatu proses pembentukan kromosom turunan (*offspring*) dengan menggabungkan elemen dari kromosom induk yang terpilih. Proses ini dilakukan dalam upaya mendapatkan kromosom baru dengan solusi yang lebih baik. *Crossover* tidak selalu dilakukan pada semua kromosom. Kromosom dipilih secara acak dengan pembangkitan bilangan acak sebanyak kromosom. Masing-masing bilangan acak merepresentasikan kromosom, sehingga jika bilangan acak yang dihasilkan kurang dari P_c maka kromosom tersebut terpilih untuk dipindah silang. Pada Gambar 2.9 diilustrasikan diagram alir penggunaan P_c pada proses *crossover*.



Gambar 2.9: Diagram Alir Proses Crossover

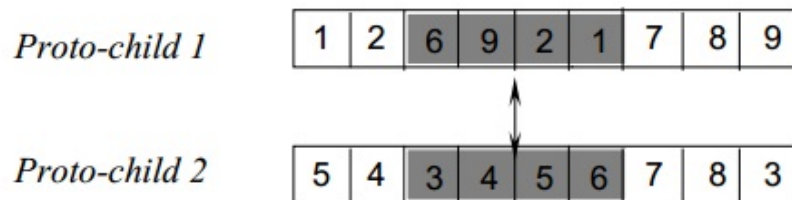
Teknik *crossover* yang digunakan adalah PMX (*Partially Mapped Crossover*). PMX pertama kali diperkenalkan oleh Goldberg dan Lingle (1985). Metode ini biasa digunakan untuk melakukan *crossover* pada kasus yang menggunakan representasi permutasi. Metode ini dapat dilihat sebagai suatu pengembangan dari metode *crossover* dua titik untuk kromosom biner dan selanjutnya dilakukan perbaikan. (Admi Syarif, 2014:14) Berikut adalah langkah-langkah *crossover* dengan metode PMX :

- (a) Menentukan *mapping section* pada sepasang kromosom secara acak.



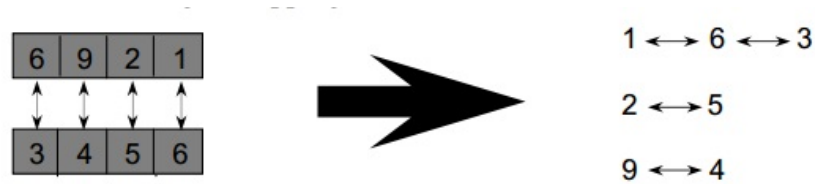
Gambar 2.10: Penentuan *mapping section*

- (b) Menukar kedua buah *mapping section* untuk membentuk 2 kromosom anak



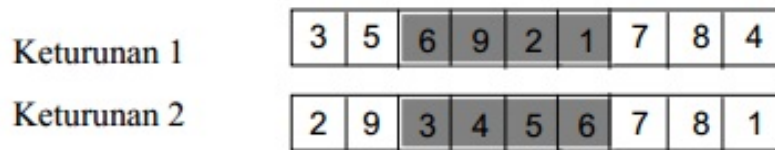
Gambar 2.11: Pembentukan 2 kromosom anak

(c) Menentukan pemetaan masing-masing gen pada kromosom anak



Gambar 2.12: Pemetaan pada kromosom anak

(d) Memperbaiki kromosom dengan menggunakan informasi yang diperoleh pada langkah 3



Gambar 2.13: Perbaikan kromosom

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.

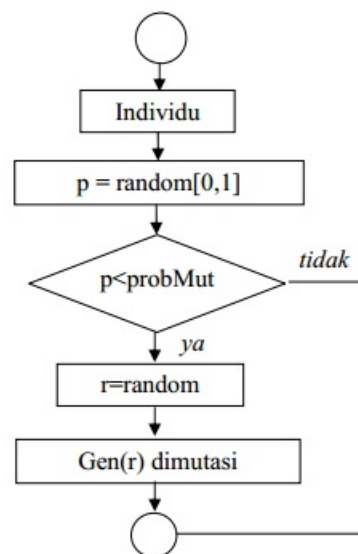
Teknik mutasi yang digunakan adalah *Swapping Mutation*. Proses mutasi dengan cara ini dilakukan dengan menentukan jumlah kromosom yang akan mengalami mutasi dalam satu populasi melalui parameter *mutation rate* (P_m). Proses mutasi dilakukan dengan cara menukar gen yang telah

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 dengan rumus :

$$[\text{panjang total gen} = \text{gen dalam 1 kromosom} \times \text{jumlah kromosom}]$$

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)

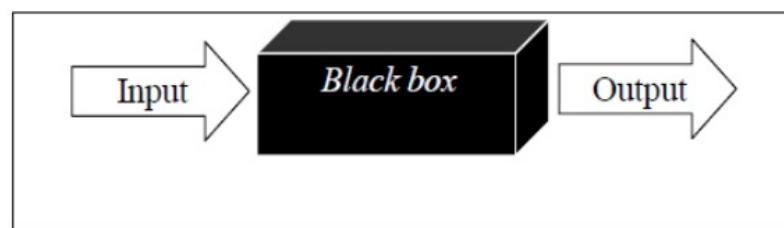


Gambar 2.14: Diagram Alir Proses Mutasi

2.4 Logika Fuzzy

Konsep logika fuzzy pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh dari Universitas California, pada bulan Juni 1965. Logika fuzzy merupakan

alat matematika untuk menangani ketidakpastian. Secara umum, logika fuzzy memberikan struktur kesimpulan yang memungkinkan kemampuan sesuai penalaran manusia. Teori logika fuzzy didasarkan pada konsep derajat keanggotaan yang relatif (Sivanandam, Sumathi, dan Deepa, 2007). Pengertian lainnya, logika fuzzy merupakan suatu metode pengambilan keputusan berbasis aturan yang digunakan untuk memecahkan keabu-abuan masalah pada sistem yang sulit dimodelkan atau memiliki ambiguitas. Dasar logika fuzzy adalah teori himpunan fuzzy. Logika fuzzy adalah suatu cara yang tepat untuk memetakan suatu ruang input ke dalam suatu ruang *output* (Kusumadewi dan Purnomo, 2010).



Gambar 2.15: Logika Fuzzy sebagai *Black Box*

Berdasarkan Gambar 2.15, logika fuzzy dapat dianggap sebagai kotak hitam yang berhubungan antara ruang input menuju ruang output. Kotak hitam yang dimaksudkan adalah metode yang dapat digunakan untuk mengolah data input menjadi output dalam bentuk informasi yang baik (Kusumadewi, 2010).

2.4.1 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 pengelompokkan objek dalam batas yang samar. Himpunan fuzzy merupakan sebuah generalisasi dari himpunan *crisp*. Jika pada himpunan *crisp*, nilai keanggotaan hanya ada 2 kemungkinan, yaitu 0 atau 1 maka himpunan fuzzy didasarkan pada gagasan untuk memperluas jangkauan fungsi karakteristik sedemikian hingga fungsi tersebut akan mencakup bilangan real pada interval $[0,1]$.

Pada himpunan fuzzy terdapat 2 komponen, 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.

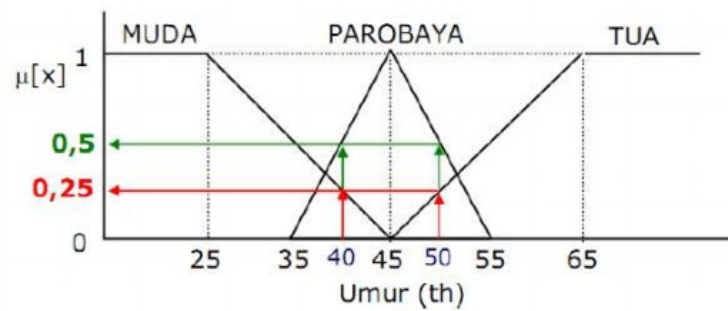
Ada beberapa hal yang perlu diketahui dalam memahami sistem fuzzy, yaitu:

1. Variabel fuzzy

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

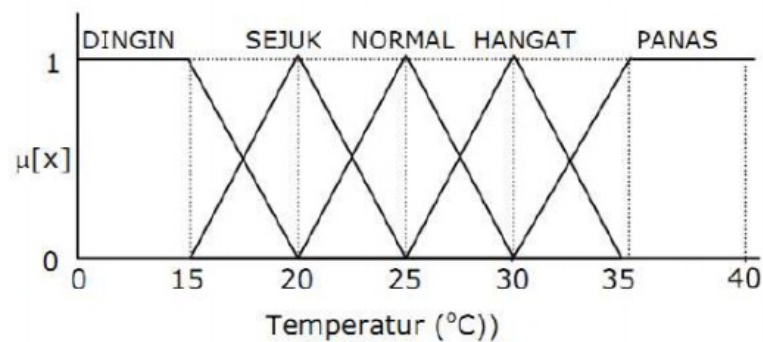
2. Himpunan fuzzy

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



Gambar 2.16: Himpunan fuzzy untuk variabel umur

- (a) Variabel umur, terbagi menjadi 3 himpunan fuzzy, yaitu : MUDA, PAROBAYA, dan TUA
- (b) Variabel temperatur, terbagi menjadi 5 himpunan fuzzy, yaitu : DINGIN, SEJUK, NORMAL, HANGAT, dan PANAS



Gambar 2.17: Himpunan fuzzy untuk 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 :

- (a) Semesta pembicaraan untuk variabel umur : $[0, +\infty]$
- (b) 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:

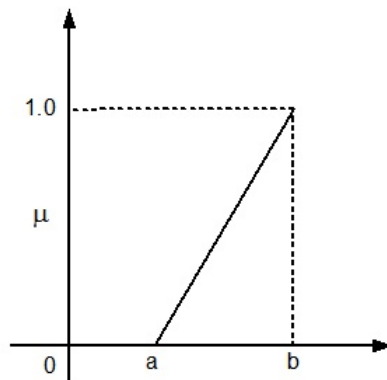
- (a) Muda = $[0 \ 45]$.
- (b) Parobaya = $[35 \ 55]$.
- (c) Tua = $[45 \ +]$.
- (d) Dingin = $[0 \ 20]$.
- (e) Sejuk = $[15 \ 25]$.
- (f) Normal = $[20 \ 30]$.
- (g) Hangat = $[25 \ 35]$.
- (h) Panas = $[30 \ 40]$

2.4.2 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 :

1. Representasi Linear Naik

Kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak kekanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi. (Kusumadewi dkk, 2006: 40)



Gambar 2.18: Representasi Linear Naik

Fungsi keanggotaan:

$$\mu(x) = f(x) = \begin{cases} 0, & x \leq a \\ \frac{(x-a)}{(b-a)}, & a \leq x \leq b \\ 1, & x \geq b \end{cases} \quad (2.8)$$

Keterangan :

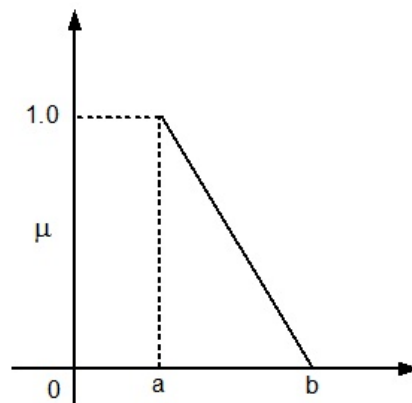
a = nilai domain yang mempunyai derajat keanggotaan nol

b = nilai domain yang mempunyai derajat keanggotaan satu

x = nilai input yang akan di ubah ke dalam bilangan fuzzy

2. Representasi Linear Turun

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



Gambar 2.19: Representasi Linear Turun

Fungsi keanggotaan:

$$\mu(x) = f(x) = \begin{cases} \frac{(b-x)}{(b-a)}, & a \leq x \leq b \\ 0, & x \geq b \end{cases} \quad (2.9)$$

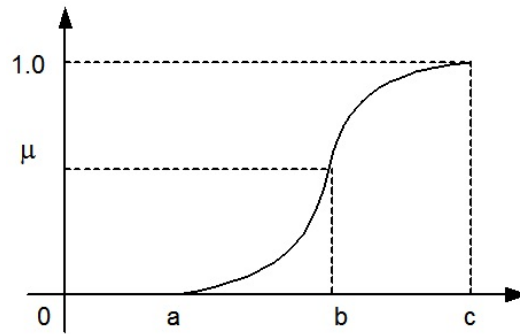
3. Representasi Fungsi Sigmoid

Fungsi segitiga pada dasarnya merupakan gabungan antara dua garis (linear), seperti terlihat pada Gambar 2.20. fungsi ini mempunyai tiga

parameter. Bilangan fuzzy yang mempunyai tiga parameter dan dapat dipresentasikan dengan fungsi segitiga, disebut bilangan fuzzy segitiga.

Fungsi keanggotaan:

$$\mu(x) = f(x) = \begin{cases} 0, & x \leq a \\ 2\left(\frac{x-a}{c-a}\right)^2, & a < x \leq b \\ 1 - 2\left(\frac{c-x}{c-b}\right)^2, & b \leq x \leq c \\ 1, & x \geq c \end{cases} \quad (2.10)$$



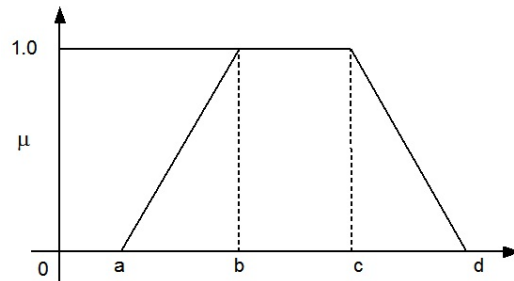
Gambar 2.20: Representasi Fungsi Sigmoid

4. Representasi Fungsi Trapesium

Fungsi trapesium pada dasarnya seperti bentuk fungsi segitiga, hanya saja ada beberapa nilai yang memiliki nilai keanggotaan 1. Fungsi ini mempunyai empat parameter, yaitu a , b , c , d . Sejumlah bilangan fuzzy adalah bilangan fuzzy trapesium yang dinotasikan dengan (a, b, c, d) dimana a , b , c , dan d adalah bilangan riil.

Fungsi keanggotaan :

$$\mu(x) = f(x) = \begin{cases} 0, & x \leq a \\ \frac{(x-a)}{(b-a)}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{(b-x)}{(c-b)}, & b \leq x \leq c \\ 0, & x \geq d \end{cases} \quad (2.11)$$



Gambar 2.21: Representasi Fungsi Trapesium

Keterangan:

a = nilai domain terkecil yang mempunyai derajat keanggotaan nol

b = nilai domain terkecil yang mempunyai derajat keanggotaan satu

c = nilai domain terbesar yang mempunyai derajat keanggotaan satu

d = nilai domain terbesar yang mempunyai derajat keanggotaan nol

x = nilai input yang akan di ubah ke dalam bilangan fuzzy

2.4.3 Sistem Inferensi Fuzzy

Terdapat 3 metode pada sistem inferensi fuzzy yaitu: metode Tsukamoto, metode Mamdani dan metode Sugeno. Dalam skripsi ini metode yang dipakai dan yang akan dibahas adalah metode Mamdani. Tahapan-tahapan dalam metode Mamdani adalah sebagai berikut : (Widodo,2012:125)

1. Pembentukan himpunan Fuzzy

Pada tahapan ini variabel input (*crisp*) dari sistem fuzzy ditransfer ke dalam himpunan fuzzy untuk dapat digunakan dalam perhitungan nilai kebenaran dari premis pada setiap aturan dalam basis pengetahuan .

2. Aplikasi 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 sebagai berikut:

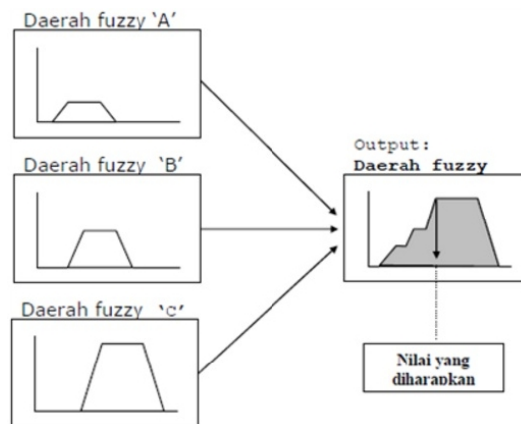
$$\text{IF } x \text{ is } A \text{ THEN } y \text{ is } B$$

3. Komposisi aturan

Tidak seperti penalaran monoton, apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan korelasi antar aturan.

4. Defuzzifikasi

Input dari proses defuzzifikasi adalah suatu himpunan fuzzy yang diperoleh dari komposisi aturan-aturan fuzzy, sedangkan *output* sistem tidak berupa himpunan fuzzy, melainkan berupa konstanta atau persamaan linear. Sehingga jika diberikan suatu himpunan fuzzy dalam *range* tertentu, maka harus dapat diambil suatu nilai konstanta atau persamaan linier tertentu sebagai *output*.



Gambar 2.22: Proses Defuzzifikasi

2.4.4 *Fuzzy Logic Controller*

Pemikiran utama teori logika fuzzy adalah memetakan sebuah ruang input ke dalam ruang output dengan menggunakan aturan *IF-THEN*. Pemetaan dilakukan dalam suatu *fuzzy inference system* (FIS). Urutan aturan bisa sembarangan. FIS mengevaluasi aturan secara simultan untuk menghasilkan kesimpulan. Oleh karena itu, semua aturan harus lebih didefinisikan lebih dahulu sebelum kita membangun sebuah FIS. FIS merupakan sebuah metode yang dapat menginterpretasikan harga-harga dalam vektor input. Menarik kesimpulan berdasarkan sekumpulan aturan IF-THEN yang diberikan dan kemudian menghasilkan vektor *output*. (Widodo, 2012: 126)

1. Fungsi Keanggotaan

Fungsi keanggotaan mendefinisikan bagaimana tiap titik dalam ruang input dipetakan menjadi bobot atau derajat keanggotaan antara 0 dan 1. Dalam teori himpunan, ruang input juga dikenal sebagai *universe of discourse*. Secara umum, beberapa kesimpulan tentang himpunan dan fungsi keanggotaan fuzzy diberikan dibawah ini :

- (a) Himpunan fuzzy menekankan konsep variabel samar (*vague or fuzzy variable*) seperti variabel hari akhir minggu, suhu panas, pelari cepat, dll.
- (b) Himpunan fuzzy mengizinkan keanggotaan parsial dari suatu himpunan seperti hari jumat yang dianggap sebagai hari akhir minggu namun dengan derajat dibawah satu.
- (c) Derajat keanggotaan dalam himpunan fuzzy berkisar antara 0 sampai 1.
- (d) Tiap fungsi keanggotaan μ berasosiasi dengan sebuah himpunan fuzzy tertentu dan memetakan suatu nilai input ke nilai derajat keanggotaan yang sesuai. Misalnya dalam kasus himpunan fuzzy orang berbadan tinggi mempunyai fungsi keanggotaan sendiri, yaitu μ tinggi yang berbeda dengan fungsi keanggotaan dari himpunan fuzzy orang berbadan rendah, yaitu μ rendah.

2. *IF-THEN Rule*

Menginterpretasikan sebuah *IF-THEN rule* meliputi dua bagian. Pertama, mengevaluasi antecedent, yaitu melakukan fuzzifikasi pada input dan menerapkan operasi-operasi logika fuzzy dengan operator-operator fuzzy. Kedua, proses implikasi yaitu menerapkan hasil operasi logika fuzzy pada bagian antecedent untuk mengambil keputusan / kesimpulan dengan mengisi fuzzy set keluaran ke variabel keluaran.

2.5 Algoritma Genetika berbasis *Fuzzy Logic Controller*

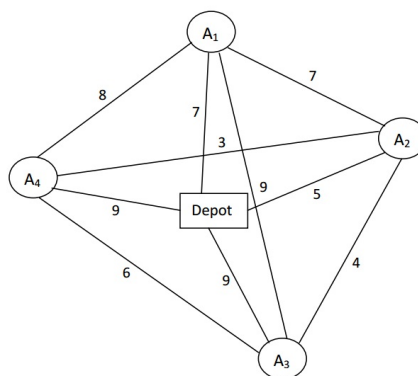
Metode ini hampir sama dengan metode Algoritma Genetika, namun parameter - parameter yang dipakai dihasilkan dari sebuah sistem fuzzy. Dalam Algoritma Genetika dengan *Fuzzy Logic Controller*, proses yang terjadi atau alur proses sama seperti dengan Algoritma Genetika, yang dikenalkan oleh John Holland 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. Sistem yang diusulkan adalah untuk mengoptimalkan proses hasil dari Algoritma Genetika. Dalam Algoritma Genetika dengan *Fuzzy Logic Controller* terdapat 8 tahap utama, yaitu:

1. Inisialisasi kromosom.
2. Fungsi evaluasi.
3. Seleksi.
4. Penentuan operator genetika, meliputi operator rekombinasi (*crossover*) dan mutasi.
5. Penentuan parameter kontrol Algoritma Genetika, yaitu ukuran populasi (*popsize*), probabilitas *crossover* (P_c), dan probabilitas mutasi (p_m).
6. *Crossover*
7. Mutasi

Dalam penentuan parameter ini dilakukan proses sistem fuzzy untuk mendapatkan nilai yang akan digunakan sebagai parameter. (Muzid, 2008:30)

2.6 Implementasi Manual

Berikut ini adalah contoh permasalahan VRP dengan depot tunggal dan sebuah kendaraan yang akan mengantar barang ke beberapa pelanggan. Jarak antar tempat pelanggan digambarkan dengan graf berikut. Tentukan rute terbaik yang dilalui oleh kendaraan.



Gambar 2.23: Lintasan VRP yang akan dilalui

Penyelesaian:

Ada 4 titik tujuan (*customer*) yang akan menjadi gen dalam kromosom yaitu titik kota selain depot asal.

Langkah 1 : Inisialisasi

Misalkan kita menggunakan 6 buah populasi dalam satu generasi, yaitu:

$$\text{Kromosom}[1] = [A_1 \ A_3 \ A_4 \ A_2]$$

$$\text{Kromosom}[2] = [A_3 \ A_1 \ A_4 \ A_2]$$

$$\text{Kromosom}[3] = [A_2 \ A_1 \ A_3 \ A_4]$$

$$\text{Kromosom}[4] = [A_4 \ A_1 \ A_2 \ A_3]$$

$$\text{Kromosom}[5] = [A_4 \ A_2 \ A_1 \ A_3]$$

$$\text{Kromosom}[6] = [A_2 \ A_3 \ A_4 \ A_1]$$

Langkah 2 : Evaluasi kromosom

Kita akan menghitung nilai fitness dari tiap kromosom yang telah dibagikan:

$$\text{Fitness}[1] = AA_1 + A_1A_3 + A_3A_4 + A_4A_2 + A_2A = 7 + 2 + 6 + 3 + 5 = 23$$

$$\text{Fitness}[2] = AA_3 + A_3A_1 + A_1A_4 + A_4A_2 + A_2A = 9 + 2 + 8 + 3 + 5 = 27$$

$$\text{Fitness}[3] = AA_2 + A_2A_1 + A_1A_3 + A_3A_4 + A_4A = 5 + 7 + 2 + 6 + 9 = 29$$

$$\text{Fitness}[4] = AA_4 + A_4A_1 + A_1A_2 + A_2A_3 + A_3A = 9 + 8 + 7 + 4 + 9 = 37$$

$$\text{Fitness}[5] = AA_4 + A_4A_2 + A_2A_1 + A_1A_3 + A_3A = 9 + 3 + 7 + 2 + 9 = 30$$

$$\text{Fitness}[6] = AA_2 + A_2A_3 + A_3A_4 + A_4A_1 + A_1A = 5 + 4 + 6 + 8 + 7 = 30$$

Langkah 3 : Seleksi Kromosom

Oleh karena pada persoalan VRP yang diinginkan yaitu kromosom dengan fitness yang lebih kecil akan mempunyai probabilitas untuk terpilih kembali lebih besar maka digunakan invers.

$$Q[i] = 1/\text{Fitness}[i]$$

$$Q[1] = 1/23 = 0,043$$

$$Q[2] = 1/27 = 0,037$$

$$Q[3] = 1/29 = 0,034$$

$$Q[4] = 1/37 = 0,027$$

$$Q[5] = 1/30 = 0,033$$

$$Q[6] = 1/30 = 0,033$$

$$\text{Total} = 0,043 + 0,037 + 0,034 + 0,027 + 0,033 + 0,033 = 0,207$$

Kemudian untuk mencari probabilitas kita menggunakan rumus berikut:

$$P[i] = Q[i] / \text{Total}$$

$$P[1] = 0,043 / 0,207 = 0,208$$

$$P[2] = 0,037 / 0,207 = 0,179$$

$$P[3] = 0,034 / 0,207 = 0,164$$

$$P[4] = 0,027 / 0,207 = 0,130$$

$$P[5] = 0,033 / 0,207 = 0,159$$

$$P[6] = 0,033 / 0,207 = 0,159$$

Dari probabilitas di atas dapat terlihat bahwa kromosom ke-1 mempunyai fitness paling kecil sehingga mempunyai probabilitas untuk terpilih pada generasi selanjutnya lebih besar dari kromosom lainnya. Untuk proses seleksi kita menggunakan metode *roulette wheel*, untuk itu kita terlebih dahulu menghitung probabilitas dari masing-masing kromosom.

$$C[1] = 0,208$$

$$C[2] = 0,208 + 0,179 = 0,387$$

$$C[3] = 0,387 + 0,164 = 0,551$$

$$C[4] = 0,551 + 0,130 = 0,681$$

$$C[5] = 0,681 + 0,159 = 0,840$$

$$C[6] = 0,840 + 0,159 = 1$$

Metode *roulette-wheel* adalah membangkitkan nilai acak R antara 0-1. Jika $R[k] < Qk[k]$ maka kromosom ke- k sebagai induk, selain itu pilih kromosom ke- k sebagai induk dengan syarat $Qk[k - 1] < R[k] < Qk[k]$. Kita putar *roulette-wheel* sebanyak jumlah kromosom yaitu 6 kali (membangkitkan bilangan acak R). Bila bilangan random R [i] yang dihasilkan sebagai berikut :

$$R[1] = 0,314$$

$$R[2] = 0,111$$

$$R[3] = 0,392$$

$$R[4] = 0,743$$

$$R[5] = 0,521$$

$$R[6] = 0,561$$

Induk yang dipilih dengan $Qk[k - 1] < R[k] < Qk[k]$ menghasilkan: $C[2]$, $C[1]$, $C[3]$, $C[5]$, $C[3]$, $C[4]$.

Setelah itu populasi baru akan terbentuk, yaitu:

$$\text{Kromosom}[1] = [2] = [A_3 \ A_1 \ A_4 \ A_2]$$

$$\text{Kromosom}[2] = [1] = [A_1 \ A_3 \ A_4 \ A_2]$$

$$\text{Kromosom}[3] = [3] = [A_2 \ A_1 \ A_3 \ A_4]$$

$$\text{Kromosom}[4] = [5] = [A_4 \ A_2 \ A_1 \ A_3]$$

$$\text{Kromosom}[5] = [3] = [A_2 \ A_1 \ A_3 \ A_4]$$

$$\text{Kromosom}[6] = [4] = [A_4 \ A_1 \ A_2 \ A_3]$$

Langkah 4 : Penentuan nilai parameter

Sebelum melakukan tahap *crossover* terlebih dulu akan dihitung nilai parameter probabilitas *crossover* yang diperoleh dari perhitungan *Fuzzy Logic Controller* dengan menginputkan populasi = 6 dan generasi = 100.

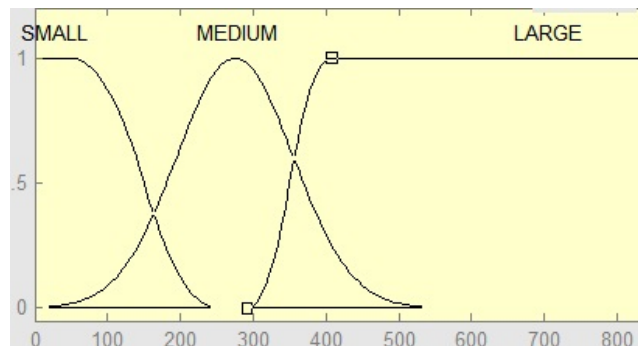
1. Pada semesta pembicaraan dan domain untuk variabel populasi, aturan yang digunakan adalah sebagai berikut:

Semesta pembicaraan: [0 1000]

Domain SMALL: [50 250]

Domain MEDIUM: [80 275]

Domain LARGE: [350 500]



Gambar 2.16 Semesta pembicaraan dan domain untuk variabel populasi
(Muzid, 2008)

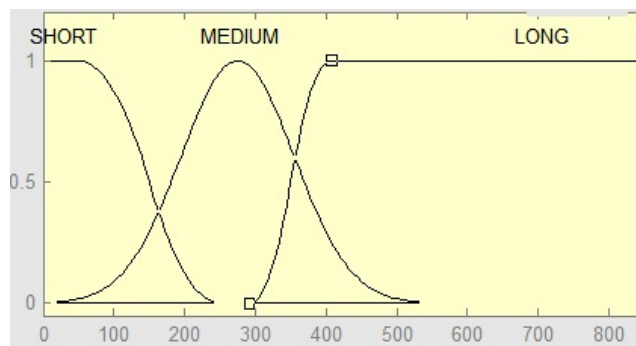
2. Pada semesta pembicaraan dan domain untuk variabel generasi, aturan nilai yang digunakan adalah sebagai berikut:

Semesta pembicaraan: [0 1000]

Domain SHORT: [50 200]

Domain MEDIUM: [80 275]

Domain LONG: [350 500]



Gambar 2.17 Semesta pembicaraan dan domain untuk variabel generasi (Muzid, 2008)

3. Pada semesta pembicaraan dan domain untuk hasil output yaitu nilai probabilitas *crossover*, aturan nilai yang adalah sebagai berikut:

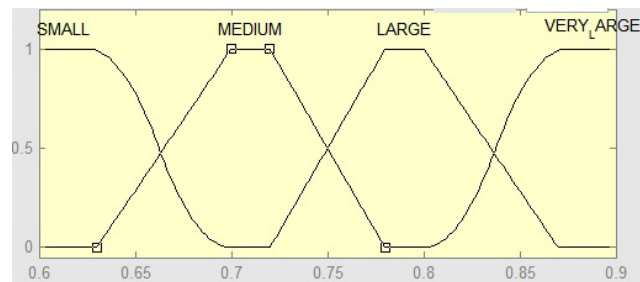
Semesta pembicaraan: [0.6 0.9]

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 VERY LARGE: [0.8 0.875]



Gambar 2.18 Semesta pembicaraan dan domain untuk variabel probabilitas *crossover* (Muzid, 2008)

4. Pada semesta pembicaraan dan domain untuk nilai probabilitas mutasi, aturan nilai yang digunakan adalah sebagai berikut:

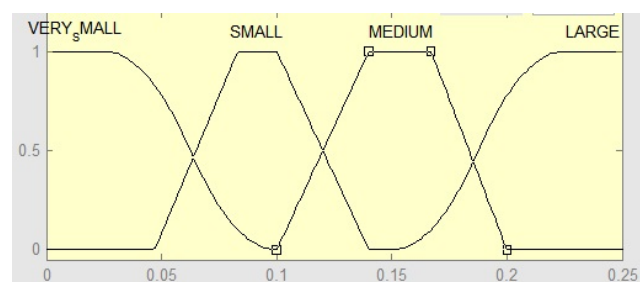
Semesta pembicaraan: [0 0.25]

Domain VERY SMALL: [0.025 0.1]

Domain SMALL: [0.47 0.83 0.1 0.14]

Domain MEDIUM: [0.1 0.14 0.167 0.2]

Domain LARGE: [0.15 0.225]



Gambar 2.19 Semesta pembicaraan dan domain untuk variabel probabilitas mutasi (Muzid, 2008)

Aturan Fuzzy: (Muzid, 2008)

1. IF (Populasi is SMALL) AND (Generasi is SHORT) THEN (Probabilitas Crossover is MEDIUM) AND (Probabilitas Mutasi is LARGE).
2. IF (Populasi is MEDIUM) AND (Generasi is SHORT) THEN (Probabilitas Crossover is SMALL) AND (Probabilitas Mutasi is MEDIUM).
3. IF (Populasi is LARGE) AND (Generasi is SHORT) THEN (Probabilitas Crossover is SMALL) AND (Probabilitas Mutasi is SMALL).
4. IF (Populasi is SMALL) AND (Generasi is MEDIUM) THEN (Probabilitas Crossover is LARGE) AND (Probabilitas Mutasi is MEDIUM).
5. IF (Populasi is MEDIUM) AND (Generasi is MEDIUM) THEN (Probabilitas Crossover is LARGE) AND (Probabilitas Mutasi is SMALL).
6. IF (Populasi is LARGE) AND (Generasi is MEDIUM) THEN (Probabilitas Crossover is MEDIUM) AND (Probabilitas Mutasi is VERY SMALL).
7. IF (Populasi is SMALL) AND (Generasi is LONG) THEN (Probabilitas Crossover is VERY LARGE) AND (Probabilitas Mutasi is SMALL).
8. IF (Populasi is MEDIUM) AND (Generasi is LONG) THEN (Probabilitas Crossover is VERY LARGE) AND (Probabilitas Mutasi is VERY SMALL).
9. IF (Populasi is LARGE) AND (Generasi is LONG) THEN (Probabilitas Crossover is LARGE) AND (Probabilitas Mutasi is VERY SMALL).

Penyelesaian:

Ada 4 variabel fuzzy yang akan dimodelkan, yaitu:

1. Populasi dengan nilai keanggotaan sebagai berikut:

$$\mu_{popSMALL}[6]=1$$

$$\mu_{popMEDIUM}[6]=0$$

$$\mu_{popLARGE}[6]=0$$

2. Generasi dengan nilai keanggotaan sebagai berikut:

$$\mu_{genSHORT}[100] = 1 - 2\left(\frac{100-50}{200-50}\right)^2 = 0.78$$

$$\mu_{genMEDIUM}[100] = e^{-(100-275)^2/2(80)^2} = 0.09$$

$$\mu_{genLONG}[100] = 0$$

3. Probabilitas Crossover dengan nilai keanggotaan sebagai berikut:

$$\mu_{crossSMALL}[x] = \begin{cases} 1; x \leq 0.625 \\ 1 - 2\left(\frac{x-0.625}{0.7-0.625}\right)^2; 0.625 \leq x \leq 0.6625 \\ 2\left(\frac{0.7-x}{0.7-0.625}\right)^2; 0.6625 \leq x \leq 0.7 \\ 0; x \geq 0.7 \end{cases}$$

$$\mu_{crossMEDIUM}[x] = \begin{cases} 0; x \leq 0.63 \text{ atau } x \geq 0.78 \\ \frac{x-0.63}{0.7-0.63}; 0.63 \leq x \leq 0.7 \\ 1; 0.7 \leq x \leq 0.72 \\ \frac{0.78-x}{0.78-0.72}; 0.72 \leq x \leq 0.78 \end{cases}$$

$$\mu_{crossLARGE}[x] = \begin{cases} 0; x \leq 0.72 \text{ atau } x \geq 0.87 \\ \frac{x-0.72}{0.78-0.72}; 0.72 \leq x \leq 0.78 \\ 1; 0.78 \leq x \leq 0.8 \\ \frac{0.87-x}{0.87-0.8}; 0.8 \leq x \leq 0.87 \end{cases}$$

$$\mu_{crossVERYLARGE}[x] = \begin{cases} 0; x \leq 0.8 \\ 2\left(\frac{x-0.8}{0.875-0.8}\right)^2; 0.8 \leq x \leq 0.8375 \\ 1 - 2\left(\frac{0.875-x}{0.875-0.8}\right)^2; 0.8375 \leq x \leq 0.875 \\ 1; x \geq 0.875 \end{cases}$$

4. Probabilitas Mutasi dengan nilai keanggotaan sebagai berikut:

$$\mu_{mutVERYSMALL}[x] = \begin{cases} 1; x \leq 0.025 \\ 1 - 2\left(\frac{x-0.025}{0.025-0.1}\right)^2; 0.025 \leq x \leq 0.0625 \\ 2\left(\frac{0.1-x}{0.1-0.025}\right)^2; 0.0625 \leq x \leq 0.1 \\ 0; x \geq 0.1 \end{cases}$$

$$\mu_{mutSMALL}[x] = \begin{cases} 0; x \leq 0.047 \text{ atau } x \geq 0.14 \\ \frac{x-0.047}{0.083-0.047}; 0.047 \leq x \leq 0.083 \\ 1; 0.083 \leq 0.1 \\ \frac{0.14-x}{0.14-0.1}; 0.1 \leq x \leq 0.14 \end{cases}$$

$$\mu_{mutMEDIUM}[x] = \begin{cases} 0; x \leq 0.1 \text{ atau } x \geq 0.2 \\ \frac{x-0.1}{0.14-0.1}; 0.1 \leq x \leq 0.14 \\ 1; 0.14 \geq 0.167 \\ \frac{0.2-x}{0.2-0.167}; 0.167 \leq x \leq 0.2 \end{cases}$$

$$\mu_{mutLARGE}[x] = \begin{cases} 0; x \leq 0.15 \\ 2\left(\frac{x-0.1}{0.14-0.1}\right)^2; 0.15 \leq x \leq 0.1875 \\ 1 - 2\left(\frac{0.225-x}{0.225-0.15}\right)^2; 0.1875 \leq 0.225 \\ 1; x \geq 0.225 \end{cases}$$

Sekarang kita cari nilai z untuk setiap aturan dengan menggunakan fungsi MIN pada aplikasi fungsi implikasinya:

1. IF (Populasi is SMALL) AND (Generasi is SHORT) THEN (Probabilitas Crossover is MEDIUM) AND (Probabilitas Mutasi is LARGE);

$$\begin{aligned} \alpha - \text{predikat}_1 &= \mu_{popSMALL} \cap \mu_{genSHORT} \\ &= \min(\mu_{popSMALL}[6] \cap \mu_{genSHORT}[100]) \\ &= \min(1; 0.78) \\ &= 0.78 \end{aligned}$$

Lihat himpunan Probabilitas Crossover MEDIUM,

$$\frac{x-0.63}{0.7-0.63} = 0.78 \rightarrow x_1 = 0.68$$

Lihat himpunan Probabilitas Mutasi LARGE,

$$1 - 2\left(\frac{0.225-X}{0.225-0.15}\right)^2 = 0.78 \rightarrow y_1 = 0.20$$

2. IF (Populasi is MEDIUM) AND (Generasi is SHORT) THEN (Probabilitas Crossover is SMALL) AND (Probabilitas Mutasi is MEDIUM).

$$\begin{aligned} \alpha - predikat_2 &= \mu_{popMEDIUM} \cap \mu_{genSHORT} \\ &= \min(\mu_{popMEDIUM}[6] \cap \mu_{genSHORT}[100]) \\ &= \min(0; 0.78) \\ &= 0 \end{aligned}$$

3. IF (Populasi is LARGE) AND (Generasi is SHORT) THEN (Probabilitas Crossover is SMALL) AND (Probabilitas Mutasi is SMALL).

$$\begin{aligned} \alpha - predikat_3 &= \mu_{popLARGE} \cap \mu_{genSHORT} \\ &= \min(\mu_{popLARGE}[6] \cap \mu_{genSHORT}[100]) \\ &= \min(0; 0.78) \\ &= 0 \end{aligned}$$

4. IF (Populasi is SMALL) AND (Generasi is MEDIUM) THEN (Probabi-

litas Crossover is LARGE) AND (Probabilitas Mutasi is MEDIUM).

$$\begin{aligned}
 \alpha - predikat_4 &= \mu_{popSMALL} \cap \mu_{genMEDIUM} \\
 &= \min(\mu_{popSMALL}[6] \cap \mu_{genMEDIUM}[100]) \\
 &= \min(1; 0.09) \\
 &= 0.09
 \end{aligned}$$

Lihat himpunan Probabilitas Crossover LARGE,

$$\frac{x-0.72}{0.78-0.72} = 0.09 \rightarrow x_4 = 0.7254$$

Lihat himpunan ProbMutasi MEDIUM,

$$\frac{x-0.1}{0.14-0.1} = 0.09 \rightarrow y_4 = 0.1054$$

5. IF (Populasi is MEDIUM) AND (Generasi is MEDIUM) THEN (Probabilitas Crossover is LARGE) AND (Probabilitas Mutasi is SMALL).

$$\begin{aligned}
 \alpha - predikat_5 &= \mu_{popMEDIUM} \cap \mu_{genMEDIUM} \\
 &= \min(\mu_{popMEDIUM}[6] \cap \mu_{genMEDIUM}[100]) \\
 &= \min(0; 0.09) \\
 &= 0
 \end{aligned}$$

6. IF (Populasi is LARGE) AND (Generasi is MEDIUM) THEN (Probabilitas Crossover is MEDIUM) AND (Prob Mutasi is VERY SMALL).

$$\begin{aligned}
 \alpha - predikat_6 &= \mu_{popLARGE} \cap \mu_{genMEDIUM} \\
 &= \min(\mu_{popLARGE}[6] \cap \mu_{genMEDIUM}[100]) \\
 &= \min(0; 0.09) \\
 &= 0
 \end{aligned}$$

7. IF (Populasi is SMALL) AND (Generasi is LONG) THEN (Probabilitas Crossover is VERY LARGE) AND (Probabilitas Mutasi is SMALL).

$$\begin{aligned}
 \alpha - predikat_7 &= \mu_{popSMALL} \cap \mu_{genLONG} \\
 &= \min(\mu_{popSMALL}[6] \cap \mu_{genLONG}[100]) \\
 &= \min(1; 0) \\
 &= 0
 \end{aligned}$$

8. IF (Populasi is MEDIUM) AND (Generasi is LONG) THEN (Probabilitas Crossover is VERY LARGE) AND (Probabilitas Mutasi is VERY SMALL).

$$\begin{aligned}
 \alpha - predikat_8 &= \mu_{popMEDIUM} \cap \mu_{genLONG} \\
 &= \min(\mu_{popMEDIUM}[6] \cap \mu_{genLONG}[100]) \\
 &= \min(0; 0) \\
 &= 0
 \end{aligned}$$

9. IF (Populasi is LARGE) AND (Generasi is LONG) THEN (Probabilitas Crossover is LARGE) AND (Probabilitas Mutasi is VERY SMALL).

$$\begin{aligned}
 \alpha - predikat_9 &= \mu_{popLARGE} \cap \mu_{genLONG} \\
 &= \min(\mu_{popMEDIUM}[6] \cap \mu_{genLONG}[100]) \\
 &= \min(0; 0) \\
 &= 0
 \end{aligned}$$

Dari sini kita dapat mencari berapakah nilai x (probabilitas *crossover*) dan y (probabilitas mutasi), yaitu:

$$x = \frac{\alpha\text{-predikat}_1*x_1 + \alpha\text{-predikat}_2*x_2 + \alpha\text{-predikat}_3*x_3 + \dots + \alpha\text{-predikat}_9*x_9}{\alpha\text{-predikat}_1 + \alpha\text{-predikat}_2 + \alpha\text{-predikat}_3 + \dots + \alpha\text{-predikat}_9}$$

$$x = \frac{0.78*0.68 + 0 + 0 + (0.09*0.7254) + 0 + 0 + 0 + 0 + 0}{0.78 + 0 + 0 + 0.09 + 0 + 0 + 0 + 0 + 0} = 0.69$$

Sedangkan,

$$y = \frac{\alpha\text{-predikat}_1*y_1 + \alpha\text{-predikat}_2*y_2 + \alpha\text{-predikat}_3*y_3 + \dots + \alpha\text{-predikat}_9*y_9}{\alpha\text{-predikat}_1 + \alpha\text{-predikat}_2 + \alpha\text{-predikat}_3 + \dots + \alpha\text{-predikat}_9}$$

$$y = \frac{0.78*0.2 + 0 + 0 + (0.09*0.1054) + 0 + 0 + 0 + 0 + 0}{0.78 + 0 + 0 + 0.09 + 0 + 0 + 0 + 0 + 0} = 0.19$$

Jadi nilai probabilitas crossover = 0,69 dan nilai probabilitas mutasi 0,19.

Langkah kelima : Pindah Silang (*Crossover*)

Nilai $\rho_c = 0,69$ berarti jika bilangan random dihasilkan ≤ 0.69 maka akan dipilih menjadi induk baru. Hasil 6 bilangan random yang dihasilkan adalah: R[1] = 0.851, R[2] = 0.211, R[3] = 0.202, R[4] = 0.877, R[5] = 0.771, R[6] = 0.131.

Kromosom ke-k yang dipilih sebagai induk jika $R[k] < \rho_c$. Maka yang akan dijadikan induk adalah kromosom[2], kromosom[3], dan kromosom[6].

Proses Crossover :

Kromosom[2] x Kromosom[3]

= [A₁A₃A₄A₂] x [A₂A₁A₃A₄]

= [A₂A₁A₄A₃] [A₁A₃A₂A₄]

Kromosom[3] = Kromosom[3] x Kromosom[6]

= [A₂A₁A₃A₄] x [A₄A₁A₂A₃]

= [A₄A₁A₃A₂] [A₂A₁A₄A₃]

Kromosom[6] = Kromosom[6] x Kromosom[2]

= [A₄A₁A₂A₃] x [A₁A₃A₄A₂]

= [A₁A₃A₂A₄] [A₄A₁A₃A₂]

Hasil Crossover:

Kromosom [1] = [A₂A₁A₄A₃]

Kromosom [2] = [A₁A₃A₂A₄]

Kromosom [3] = [A₄A₁A₃A₂]

$$\text{Kromosom [4]} = [A_2A_1A_4A_3]$$

$$\text{Kromosom [5]} = [A_1A_3A_2A_4]$$

$$\text{Kromosom [6]} = [A_4A_1A_3A_2]$$

Langkah keenam : Mutasi

Panjang total gen = jumlah gen dalam 1 kromosom * jumlah Kromosom(3) = $4 * 6 = 24$

Probabilitas mutasi (m) = 0,19 maka jumlah gen yang akan dimutasi adalah $= 0,19 * 24 = 4,56 = 5$.

Posisi tersebut didapat dari pembangkitan 5 bilangan acak. Misalkan 5 buah posisi gen yang akan dimutasi adalah 3, 7, 10, 20, 24.

$$\text{Kromosom [1]} = [A_2A_1\underline{A_4}A_3] = [A_2A_1A_3A_4]$$

$$\text{Kromosom [2]} = [A_1A_3\underline{A_2}A_4] = [A_1A_3A_4A_2]$$

$$\text{Kromosom [3]} = [A_4\underline{A_1}A_3A_2] = [A_1A_4A_3A_2]$$

$$\text{Kromosom [4]} = [A_2A_1A_4A_3] = [A_2A_1A_4A_3]$$

$$\text{Kromosom [5]} = [A_1A_3A_2\underline{A_4}] = [A_4A_3A_2A_1]$$

$$\text{Kromosom [6]} = [A_4A_1A_3\underline{A_2}] = [A_2A_1A_3A_4]$$

Proses algoritma genetika berbasis sistem inferensi fuzzy untuk 1 generasi telah selesai. Maka nilai fitness setelah 1 generasi adalah:

$$\text{Fitness [1]} = AA_2 + A_2A_1 + A_1A_3 + A_3A_4 + A_4A = 5 + 7 + 2 + 6 + 9 = 29$$

$$\text{Fitness [2]} = AA_1 + A_1A_3 + A_3A_4 + A_4A_2 + A_2A = 7 + 2 + 6 + 3 + 5 = 23$$

$$\text{Fitness [3]} = AA_1 + A_1A_4 + A_4A_3 + A_3A_2 + A_2A = 7 + 8 + 6 + 4 + 5 = 30$$

$$\text{Fitness [4]} = AA_2 + A_2A_1 + A_1A_4 + A_4A_3 + A_3A = 5 + 7 + 8 + 6 + 9 = 35$$

$$\text{Fitness [5]} = AA_4 + A_4A_3 + A_3A_2 + A_2A_1 + A_1A = 9 + 6 + 4 + 7 + 7 = 33$$

$$\text{Fitness [6]} = AA_2 + A_2A_1 + A_1A_3 + A_3A_4 + A_4A = 5 + 7 + 2 + 6 + 9 = 29$$

Apabila setelah dalam beberapa generasi berturut-turut diperoleh nilai fitness terendah tidak berubah, maka didapatkan rute paling optimal untuk kendaraan I pada kasus VRP di atas adalah $A - A_1 - A_3 - A_4 - A_2 - A$.

BAB III

PEMBAHASAN

Permasalahan *vehicle routing problem* dapat digambarkan sebagai berikut, misalkan terdapat suatu jenis komoditi ditempatkan di sebuah depot $i = 0$ dengan K kendaraan (*vehicle*) yang berada di depot tersebut. Andaikan ada N pelanggan dinyatakan dengan $i = 1, 2, 3, \dots, N$ dan jarak antara dua lokasi i dan j diketahui sebesar c_{ij} , $0 \leq i \neq j \leq N$. Masalah utama dalam *vehicle routing problem* ini adalah bagaimana menentukan rute untuk K kendaraan tersebut sedemikian sehingga setiap pelanggan dilayani oleh tepat satu kendaraan.

Data disajikan dalam bentuk matriks data jarak yang merepresentasikan jarak antar lokasi *customer*. Gen dalam kromosom merepresentasikan *customer* sehingga kromosom tersebut merepresentasikan rute perjalanan yang ditempuh kendaraan.

3.1 Parameter Algoritma Genetika berbasis *Fuzzy Logic Controller*

Parameter-parameter yang diperlukan dalam perhitungan Algoritma Genetika berbasis *Fuzzy Logic Controller*, yaitu:

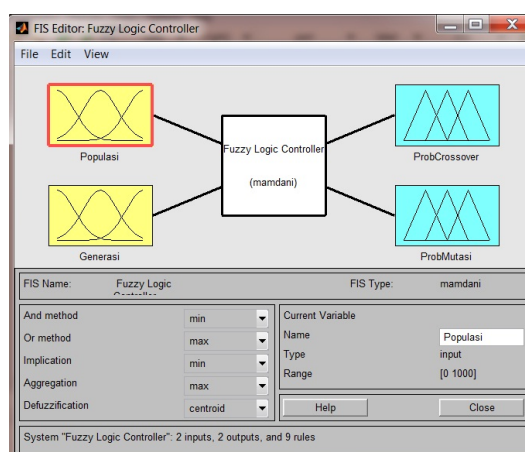
1. Populasi dan Generasi = (100 dan 100), (100 dan 500), (100 dan 1000), (500, 100), dan (1000,100).
2. Peluang *Crossover* (P_c) = dihitung menggunakan *fuzzy logic controller*
3. Peluang Mutasi (P_m) = dihitung menggunakan *fuzzy logic controller*

3.2 Langkah Penyelesaian

3.2.1 *Fuzzy Logic Controller*

Langkah-langkah yang dilakukan adalah sebagai berikut :

1. Pembentukan himpunan fuzzy (fuzzifikasi) 2 variabel *input* yaitu Populasi dan Generasi serta 2 variabel *output* yaitu Probabilitas *Crossover* dan Probabilitas Mutasi.
2. Menentukan masing-masing nilai keanggotaan fuzzy pada setiap variabel.
3. Melakukan aplikasi fungsi implikasi dengan menggunakan fungsi MIN.
4. Menentukan komposisi aturan fuzzy *IF-THEN rules*.
5. Melakukan proses defuzzifikasi dengan *input* populasi dan generasi untuk mendapatkan nilai *output* probabilitas *crossover* dan mutasi.
6. kembali pada loop Algoritma Genetika.



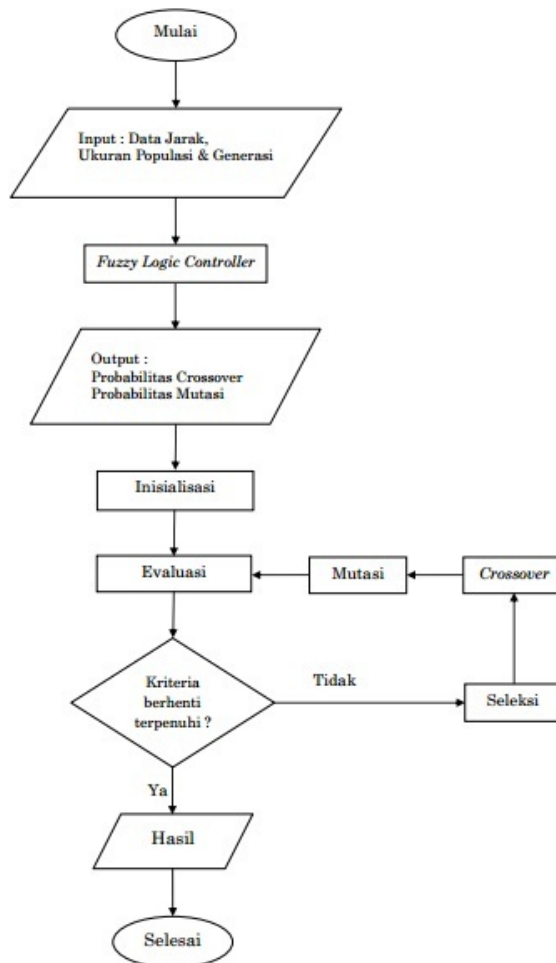
Gambar 3.1: *Fuzzy Logic Controller* dengan Sistem Inferensi Fuzzy Mamdani

Pada Gambar 3.1 dapat dilihat tampilan *Fuzzy Logic Controller* menggunakan Sistem Inferensi Fuzzy Mamdani. Dimulai dengan pembentukan himpunan fuzzy berupa 2 variabel input dan 2 variabel output, fungsi MIN pada aplikasi fungsi implikasi, metode MAX pada komposisi aturan dan metode centroid untuk defuzzifikasi.

3.2.2 Algoritma genetika dengan *fuzzy logic controller*

1. Masukkan input data jarak, ukuran populasi dan ukuran generasi.
2. Tahap evaluasi, menghitung nilai *fitness* dari $1/\text{total jarak}$, probabilitas *fitness* dari nilai *fitness* / total *fitness* dan nilai kumulatif *fitness*.
3. Tahap seleksi kromosom dengan proses *Roulette Wheel* dimulai dengan membangkitkan nilai random sebanyak kromosom.
4. Lakukan persilangan (*crossover*) pada kromosom yang induk untuk menghasilkan kromosom anak (*offspring*).
5. Lakukan proses mutasi pada kromosom *offspring*. Pada langkah 4 dan langkah 5, kedua probabilitas dari probabilitas *crossover* dan probabilitas mutasi akan diatur/ dikontrol dengan melakukan proses *fuzzy logic controller*.
6. kembali pada langkah 2 sampai pada iterasi maksimum sehingga memperoleh kromosom yang terbaik.
7. Selesai dengan kromosom terbaik.

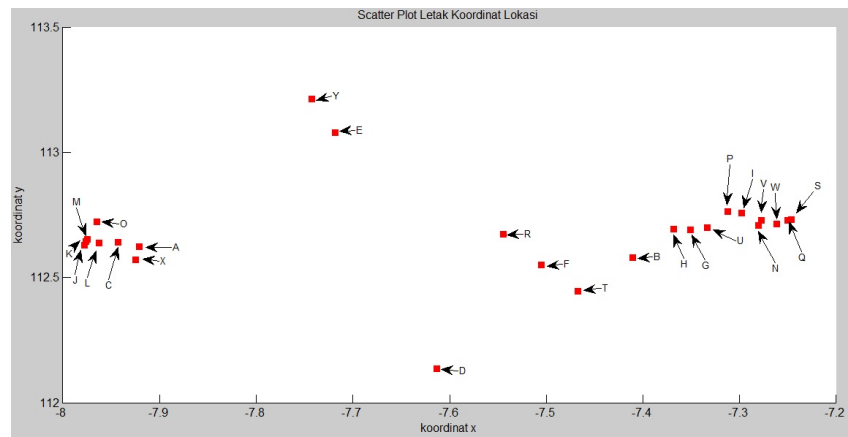
Sehingga pada langkah-langkah diatas dapat ditunjukkan dengan diagram alir adalah sebagai berikut :



Gambar 3.2: Diagram Alir Algoritma Genetika berbasis *Fuzzy Logic Controller*

3.3 Studi Kasus

Dalam penelitian ini, sebuah CV yang memproduksi pakan burung yaitu CV Ebod Jaya akan mendistribusikan produk ke sejumlah pelanggan. Dari data yang diperoleh dari alamat website Ebod Jaya terdapat 24 alamat yang akan dikunjungi menggunakan 3 kendaraan pengangkut. Pada Gambar 3.3 disajikan tampilan *scatter plot* ke-24 letak lokasi *customer*.



Gambar 3.3: *Scatter Plot* Letak Lokasi Customer

Jumlah alamat akan dibagi berdasarkan koordinat tempat menjadi 3 rute sehingga setiap kendaraan akan mengunjungi 8 alamat yang berbeda. Pembagian rute 3 kendaraan yang akan melayani customer adalah sebagai berikut:

Tabel 3.1: Pembagian rute tempuh VRP

Kendaraan	Rute
1	A-C-J-K-L-M-O-R-X-A
2	A-B-D-E-F-G-H-T-Y-A
3	A-I-N-P-Q-S-U-V-W-A

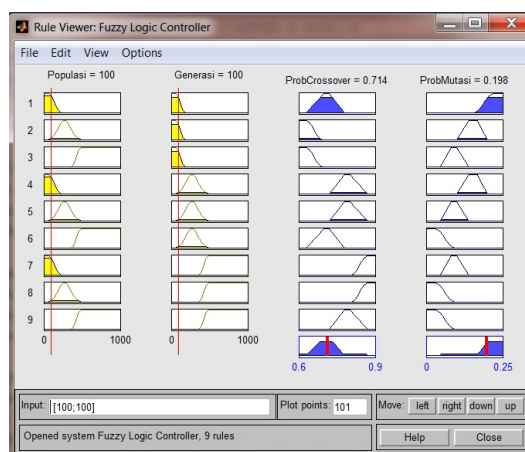
Data jarak antar lokasi diperoleh melalui proses pencarian jarak dengan *Google Maps*. Setelah jarak antar masing-masing lokasi diketahui maka selanjutnya akan dilakukan pencarian rute optimal dan jarak minimal yang dapat ditempuh dalam pengiriman produk dengan syarat semua alamat dilalui tepat satu kali kecuali titik asal yang sama dengan titik akhir yaitu depot. Data lengkap alamat dan jarak antar lokasi dapat dilihat pada Lampiran 1.

3.4 Hasil dan Pembahasan

Hasil simulasi program yang telah dirancang memerlukan pengujian data dengan melakukan proses pencarian rute terbaik dengan variasi ukuran populasi dan ukuran generasi yaitu: (100 dan 100), (100 dan 500), (100 dan 1000), (500 dan 100) dan (1000 dan 100). Kemudian dilakukan proses perhitungan sebanyak 5 kali dan diambil hasil terbaik yang paling minimum.

3.4.1 Perhitungan menggunakan populasi 100 dan generasi 100

Perhitungan dimulai dengan memasukkan data jarak untuk kendaraan 1 serta ukuran populasi= 100 dan generasi= 100. Nilai probabilitas *crossover* dan probabilitas mutasi dihasilkan melalui proses *fuzzy logic controller* dengan sistem fuzzy Mamdani dengan memasukkan populasi 100 dan generasi 100 dan menggunakan aturan fuzzy serta nilai fungsi keanggotaan fuzzy.



Gambar 3.4: *Output* pencarian probabilitas *crossover* dan mutasi

Kolom kuning menunjukkan variabel input yang dalam penelitian ini adalah ukuran populasi dan generasi. Kolom biru menunjukkan variabel output yang dalam penelitian ini adalah nilai probabilitas crossover dan nilai probabilitas mutasi.

Hasil perhitungan menggunakan populasi 100, generasi 100, probabilitas crossover 0,714, probabilitas mutasi 0,198 dan sebanyak 5 kali percobaan dapat dilihat pada Tabel 3.2, 3.3 dan 3.4.

Tabel 3.2: Hasil perhitungan pada kendaraan 1

No	Total jarak (km)	Jalur terbaik	Rute terbaik
1	159.9	1-8-7-6-4-5-2-3-9-1	A-R-O-M-K-L-C-J-X-A
2	162.7	1-9-3-6-4-5-2-8-7-1	A-X-J-M-K-L-C-R-O-A
3	159.8	1-8-7-6-4-3-5-2-9-1	A-R-O-M-K-J-L-C-X-A
4	158.9	1-9-6-4-3-5-2-7-8-1	A-X-M-K-J-L-C-O-R-A
5	165.4	1-2-4-6-5-3-9-7-8-1	A-C-K-M-L-J-X-O-R-A

Dari Tabel 3.2 di atas diperoleh jarak terpendek yang ditempuh oleh kendaraan 1 adalah 158.9 km dengan rute terbaiknya adalah A-X-M-K-J-L-C-O-R-A.

Tabel 3.3: Hasil perhitungan pada kendaraan 2

No	Total jarak (km)	Jalur terbaik	Rute terbaik
1	316.4	1-5-2-8-6-7-3-4-9-1	A-F-B-T-G-H-D-E-Y-A
2	305	1-3-7-6-5-8-2-9-4-1	A-D-H-G-F-T-B-Y-E-A
3	303.2	1-5-8-2-6-7-3-4-9-1	A-F-T-B-G-H-D-E-Y-A
4	304.1	1-3-7-6-8-5-2-9-4-1	A-D-H-G-T-F-B-Y-E-A
5	302.3	1-5-8-2-6-7-3-9-4-1	A-F-T-B-G-H-D-Y-E-A

Dari Tabel 3.3 di atas diperoleh jarak terpendek yang ditempuh oleh kendaraan 2 adalah 302.3 km dengan rute terbaiknya adalah A-F-T-B-G-H-D-Y-E-A.

Tabel 3.4: Hasil perhitungan pada kendaraan 3

No	Total jarak (km)	Jalur terbaik	Rute terbaik
1	222.1	1-7-3-8-5-6-4-2-9-1	A-U-N-V-Q-S-P-I-W-A
2	216.9	1-7-2-3-9-6-5-8-4-1	A-U-I-N-W-S-Q-V-P-A
3	211.2	1-7-8-6-5-9-3-2-4-1	A-U-V-S-Q-W-N-I-P-A
4	216.6	1-4-2-7-3-9-6-5-8-1	A-P-I-U-N-W-S-Q-V-A
5	218.4	1-7-4-2-3-9-6-5-8-1	A-U-P-I-N-W-S-Q-V-A

Dari Tabel 3.4 di atas diperoleh jarak terpendek yang ditempuh oleh kendaraan 3 adalah 211.2 km dengan rute terbaiknya adalah 1-7-8-6-5-9-3-2-4-1. Dengan demikian total jarak yang ditempuh oleh ketiga kendaraan tersebut adalah 672.4 km.

3.4.2 Perhitungan menggunakan populasi 100 dan generasi 500

Hasil perhitungan menggunakan populasi 100, generasi 500, probabilitas crossover 0,864, probabilitas mutasi 0,0889 dan sebanyak 5 kali percobaan dapat dilihat pada Tabel 3.5, 3.6 dan 3.7.

Tabel 3.5: Hasil perhitungan pada kendaraan 1

No	Total jarak (km)	Jalur terbaik	Rute terbaik
1	163.8	1-2-3-4-5-6-7-8-9-1	A-C-J-K-L-M-O-R-X-A
2	160.6	1-9-4-6-3-5-2-7-8-1	A-X-K-M-J-L-C-O-R-A
3	159.9	1-9-3-2-5-4-6-7-8-1	A-X-J-C-L-K-M-O-R-A
4	162.5	1-2-5-4-3-6-7-8-9-1	A-C-L-K-J-M-O-R-X-A
5	166.3	1-5-4-6-3-9-2-7-8-1	A-L-K-M-J-X-C-O-R-A

Dari Tabel 3.5 di atas diperoleh jarak terpendek yang ditempuh oleh kendaraan 1 adalah 159.9 km dengan rute terbaiknya adalah A-X-J-C-L-K-M-O-R-A.

Tabel 3.6: Hasil perhitungan pada kendaraan 2

No	Total jarak (km)	Jalur terbaik	Rute terbaik
1	318.4	1-3-5-8-7-6-2-9-4-1	A-F-T-B-G-H-D-E-Y-A
2	332.6	1-3-2-8-5-7-6-4-9-1	A-D-B-T-F-H-G-E-Y-A
3	345.5	1-9-4-3-2-5-6-7-8-1	A-Y-E-D-B-F-G-H-T-A
4	343.3	1-2-6-7-3-8-5-4-9-1	A-B-G-H-D-T-F-E-Y-A
5	466	1-3-8-2-5-4-6-7-9-1	A-D-T-B-F-E-G-H-Y-A

Dari Tabel 3.6 di atas diperoleh jarak terpendek yang ditempuh oleh kendaraan 2 adalah 318.4 km dengan rute terbaiknya adalah A-F-T-B-G-H-D-E-Y-A.

Tabel 3.7: Hasil perhitungan pada kendaraan 3

No	Total jarak (km)	Jalur terbaik	Rute terbaik
1	210.9	1-7-3-8-5-6-9-2-4-1	A-U-N-V-Q-S-P-I-W-A
2	212.6	1-7-3-5-9-6-8-2-4-1	A-U-I-N-W-S-Q-V-P-A
3	207.1	1-7-3-9-6-5-8-2-4-1	A-U-V-S-Q-W-N-I-P-A
4	221.9	1-7-3-8-5-6-2-4-9-1	A-P-I-U-N-W-S-Q-V-A
5	219.2	1-4-2-6-5-8-7-3-9-1	A-U-P-I-N-W-S-Q-V-A

Dari Tabel 3.7 di atas diperoleh jarak terpendek yang ditempuh oleh kendaraan 3 adalah 207.1 km dengan rute terbaiknya A-U-V-S-Q-W-N-I-P-A. Dengan demikian total jarak yang ditempuh oleh ketiga kendaraan tersebut adalah 685.4 km.

3.4.3 Perhitungan menggunakan populasi 100 dan generasi 1000

Hasil perhitungan menggunakan populasi 100, generasi 1000, probabilitas crossover 0,867, probabilitas mutasi 0,0872 dan sebanyak 5 kali percobaan dapat dilihat pada Tabel 3.8, 3.9 dan 3.10.

Tabel 3.8: Hasil perhitungan pada kendaraan 1

No	Total jarak (km)	Jalur terbaik	Rute terbaik
1	163.5	1-2-3-4-6-5-7-8-9-1	A-C-J-K-M-L-O-R-X-A
2	161.8	1-8-7-6-3-4-5-2-9-1	A-R-O-M-J-K-L-C-X-A
3	162.5	1-2-5-4-3-6-7-8-9-1	A-C-L-K-J-M-O-R-X-A
4	160.5	1-2-5-3-4-6-7-8-9-1	A-C-L-J-K-M-O-R-X-A
5	167.5	1-2-9-4-5-3-6-7-8-1	A-C-X-K-L-J-M-O-R-A

Dari Tabel 3.8 di atas diperoleh jarak terpendek yang ditempuh oleh kendaraan 1 adalah 160.5 km dengan rute terbaiknya adalah A-C-L-J-K-M-O-R-X-A

Tabel 3.9: Hasil perhitungan pada kendaraan 2

No	Total jarak (km)	Jalur Terbaik	Rute terbaik
1	303.2	1-5-8-2-6-7-3-4-9-1	A-F-T-B-G-H-D-E-Y-A
2	327.2	1-3-5-8-2-6-7-4-9-1	A-D-F-T-B-G-H-E-Y-A
3	302.3	1-5-8-2-6-7-3-9-4-1	A-F-T-B-G-H-D-Y-E-A
4	432.3	1-4-5-2-6-3-8-9-7-1	A-E-F-B-G-D-T-Y-H-A
5	330.2	1-6-7-2-5-8-3-4-9-1	A-G-H-B-F-T-D-E-Y-A

Dari Tabel 3.9 di atas diperoleh jarak terpendek yang ditempuh oleh kendaraan 2 adalah 302.3 km dengan rute terbaiknya adalah A-F-T-B-G-H-D-Y-E-A.

Tabel 3.10: Hasil perhitungan pada kendaraan 3

No	Total jarak (km)	Jalur terbaik	Rute terbaik
1	210.9	1-7-3-8-5-6-9-2-4-1	A-U-N-V-Q-S-W-I-P-A
2	218.6	1-7-3-9-8-2-5-6-4-1	A-U-N-W-V-I-Q-G-P-A
3	225.1	1-4-2-7-5-6-9-3-8-1	A-P-I-U-Q-S-W-N-V-A
4	216.7	1-4-2-7-3-9-5-6-8-1	A-P-I-U-N-W-Q-S-V-A
5	220.2	1-3-8-4-2-6-5-9-7-1	A-N-V-P-I-S-Q-W-U-A

Dari Tabel 3.10 di atas diperoleh jarak terpendek yang ditempuh oleh kendaraan 3 adalah 210.9 km dengan rute terbaiknya A-U-N-V-Q-S-W-I-P-A . Dengan demikian total jarak yang ditempuh oleh ketiga kendaraan tersebut adalah 673.7 km.

3.4.4 Perhitungan menggunakan populasi 500 dan generasi 100

Hasil perhitungan menggunakan populasi 500, generasi 100, probabilitas crossover 0,652, probabilitas mutasi 0,0887 dan sebanyak 5 kali percobaan dapat dilihat pada Tabel 3.11, 3.12 dan 3.13.

Tabel 3.11: Hasil perhitungan pada kendaraan 1

No	Total jarak (km)	Jalur terbaik	Rute terbaik
1	163.8	1-2-3-4-5-6-7-8-9-1	A-C-J-K-L-M-O-R-X-A
2	159.4	1-8-2-7-6-4-5-3-9-1	A-R-C-O-M-K-L-J-X-A
3	160.5	1-2-5-3-4-6-7-8-9-1	A-C-L-J-K-M-O-R-X-A
4	158	1-8-7-2-5-6-4-3-9-1	A-R-O-C-L-M-K-J-X-A
5	158.9	1-9-6-4-3-5-2-7-8-1	A-X-M-K-J-L-C-O-R-A

Dari Tabel 3.11 di atas diperoleh jarak terpendek yang ditempuh oleh kendaraan 1 adalah 158 km dengan rute terbaiknya adalah A-R-O-C-L-M-K-J-X-A.

Tabel 3.12: Hasil perhitungan pada kendaraan 2

No	Total jarak (km)	Jalur terbaik	Rute terbaik
1	324.4	1-3-7-6-8-2-5-4-9-1	A-D-H-G-T-B-F-E-Y-A
2	316.4	1-5-2-8-6-7-3-4-9-1	A-F-B-T-G-H-D-E-Y-A
3	310.1	1-3-6-7-5-8-2-9-4-1	A-D-G-H-F-T-B-Y-E-A
4	311.5	1-4-9-3-7-6-2-5-8-1	A-E-Y-D-H-G-B-F-T-A
5	333.7	1-4-9-2-5-3-7-6-8-1	A-E-Y-B-F-D-H-G-T-A

Dari Tabel 3.12 di atas diperoleh jarak terpendek yang ditempuh oleh kendaraan 2 adalah 310.1 km dengan rute terbaiknya adalah 1-3-6-7-5-8-2-9-4-1.

Tabel 3.13: Hasil perhitungan pada kendaraan 3

No	Total jarak (km)	Jalur Terbaik	Rute terbaik
1	210.1	1-7-3-6-5-9-8-2-4-1	A-U-N-S-Q-W-V-I-P-A
2	224.6	1-4-7-2-3-8-5-6-9-1	A-P-U-I-N-V-Q-S-W-A
3	207.1	1-7-3-9-6-5-8-2-4-1	A-U-N-W-S-Q-V-I-P-A
4	211.2	1-7-8-6-5-9-3-2-4-1	A-U-V-S-Q-W-N-I-P-A
5	228.4	1-2-3-7-4-6-5-8-9-1	A-I-N-U-P-S-Q-V-W-A

Dari Tabel 3.13 di atas diperoleh jarak terpendek yang ditempuh oleh kendaraan 3 adalah 207.1 km dengan rute terbaiknya A-U-N-W-S-Q-V-I-P-A. Dengan demikian total jarak yang ditempuh oleh ketiga kendaraan tersebut adalah 675.2 km.

3.4.5 Perhitungan menggunakan populasi 1000 dan generasi 100

Hasil perhitungan menggunakan populasi 500, generasi 100, probabilitas crossover 0,647, probabilitas mutasi 0,087 dan sebanyak 5 kali percobaan dapat dilihat pada Tabel 3.14, 3.15 dan 3.16.

Tabel 3.14: Hasil perhitungan pada kendaraan 1

No	Total jarak (km)	Jalur terbaik	Rute terbaik
1	160.6	1-9-4-6-3-5-2-7-8-1	A-X-K-M-J-L-C-O-R-A
2	159.4	1-9-3-5-4-6-7-2-8-1	A-X-J-L-K-M-O-C-R-A
3	163.8	1-2-3-4-5-6-7-8-9-1	A-C-J-K-L-M-O-R-X-A
4	161.8	1-9-3-4-6-5-2-8-7-1	A-X-J-K-M-L-C-R-O-A
5	159.9	1-9-5-2-3-4-6-7-8-1	A-X-L-C-J-K-M-O-R-A

Dari Tabel 3.14 di atas diperoleh jarak terpendek yang ditempuh oleh kendaraan 1 adalah 159.4 km dengan rute terbaiknya A-X-J-L-K-M-O-C-R-A.

Tabel 3.15: Hasil perhitungan pada kendaraan 2

No	Total jarak (km)	Jalur terbaik	Rute terbaik
1	339	1-4-9-3-5-2-6-7-8-1	A-E-Y-D-F-B-G-H-T-A
2	302.3	1-5-8-2-6-7-3-9-4-1	A-F-T-B-G-H-D-Y-E-A
3	340.6	1-4-9-2-8-5-6-3-7-1	A-E-Y-B-T-F-G-D-H-A
4	398.5	1-4-8-2-5-6-7-3-9-1	A-E-T-B-F-G-H-D-Y-A
5	316.6	1-3-6-7-2-8-5-4-9-1	A-D-G-H-B-T-F-E-Y-A

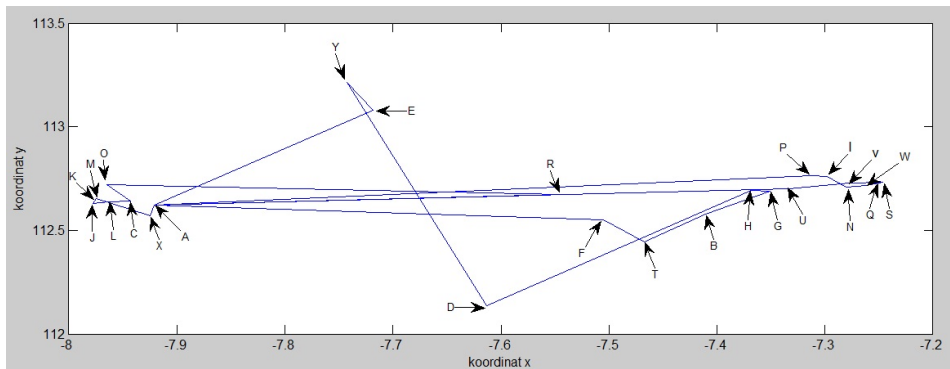
Dari Tabel 3.15 di atas diperoleh jarak terpendek yang ditempuh oleh kendaraan 2 adalah 302.3 km dengan rute terbaiknya A-F-T-B-G-H-D-Y-E-A.

Tabel 3.16: Hasil perhitungan pada kendaraan 3

No	Total jarak (km)	Jalur terbaik	Rute terbaik
1	223.1	1-7-3-6-5-9-4-2-8-1	A-U-N-S-Q-W-P-I-V-A
2	217	1-4-2-3-6-5-8-9-7-1	A-P-I-N-S-Q-V-W-U-A
3	220	1-4-2-6-5-9-3-7-8-1	A-P-I-S-Q-W-N-U-V-A
4	212.4	1-7-8-3-9-6-5-3-4-1	A-U-V-I-W-S-Q-N-P-A
5	232.4	1-7-5-2-9-6-8-3-4-1	A-U-Q-I-W-S-V-N-P-A

Dari Tabel 3.16 di atas diperoleh jarak terpendek yang ditempuh oleh kendaraan 3 adalah 212.4 km dengan rute terbaiknya A-U-V-I-W-S-Q-N-P-A . Dengan demikian total jarak yang ditempuh oleh ketiga kendaraan tersebut adalah 674.1 km.

Berikut adalah tampilan plot rute terbaik VRP pendistribusian produk pakan burung CV. Ebod Jaya :



Gambar 3.5: Plot Rute Terbaik

3.5 Analisis Penyelesaian *Vehicle Routing Problem* dengan Algoritma Genetika berbasis *Fuzzy Logic Controller*

Berdasarkan hasil penelitian diperoleh bahwa hasil panjang jalur terbaik masing-masing kendaraan pada jaringan VRP pendistribusian produk pakan burung CV. Ebod Jaya ke-24 alamat *customer* dengan menggunakan variasi ukuran populasi dan generasi yang berbeda dapat dijelaskan pada Tabel 3.17.

Tabel 3.17: Tabel Hasil Panjang Jalur Terbaik

Pop dan Gen	Rute K1(km)	Rute K2(km)	Rute K3(km)	Panjang rute(km)
100 dan 100	158.9	302.3	211.2	672.4
100 dan 500	159.9	318.4	207.1	685.4
100 dan 1000	160.5	302.3	210.9	673.7
500 dan 100	158	310.1	207.1	675.2
1000 dan 100	159.4	302.3	212.4	674.1

Berikut adalah rute yang terbentuk berdasarkan input kelima variasi populasi dan generasi:

Tabel 3.18: Tabel Hasil Panjang Jalur Terbaik dan Rute Terbaik

No	Rute terbaik K1	Rute terbaik K2	Rute terbaik K3
1	A-X-M-K-J-L-C-O-R-A	A-F-T-B-G-H-D-Y-E-A	A-U-V-S-Q-W-N-I-P-A
2	A-X-J-C-L-K-M-O-R-A	A-D-F-T-H-G-B-Y-E-A	A-U-N-W-S-Q-V-I-P-A
3	A-C-L-J-K-M-O-R-X-A	A-F-T-B-G-H-D-Y-E-A	A-U-N-V-Q-S-W-I-P-A
4	A-R-O-C-L-M-K-J-X-A	A-D-G-H-F-T-B-Y-E-A	A-U-N-W-S-Q-V-I-P-A
5	A-X-J-L-K-M-O-C-R-A	A-F-T-B-G-H-D-Y-E-A	A-U-V-N-W-S-Q-N-P-A

Dari kelima variasi populasi dan generasi pada Algoritma Genetika berbasis *Fuzzy Logic Controller* diperoleh bahwa dengan populasi 100 dan generasi 100 didapatkan panjang rute yang paling minimal dari rute yang lain dengan jalur terbaiknya adalah A-X-M-K-J-L-C-O-R-A untuk kendaraan 1, A-F-T-B-G-H-D-Y-E-A untuk kendaraan 2 dan A-U-V-S-Q-W-N-I-P-A untuk kendaraan 3.

Selanjutnya, analisa probabilitas crossover (p_c) dan probabilitas mutasi (p_m) dapat dijelaskan pada Tabel 3.18.

Tabel 3.19: Tabel Hasil Probabilitas *Crossover* dan Probabilitas Mutasi

No	Populasi	Generasi	p_c	p_m	Panjang rute terbaik(km)
1	100	100	0.714	0.198	672.4
2	100	500	0.864	0.0889	685.4
3	100	1000	0.867	0.0872	673.7
4	500	100	0.652	0.0887	675.2
5	1000	100	0.647	0.087	674.1

Pada Tabel dapat dilihat bahwa dari populasi 100 dan generasi 100 didapatkan nilai probabilitas mutasi yang paling besar dibanding dengan keempat variasi yang lain. Sedangkan probabilitas *crossover* yang dihasilkan oleh populasi 100 dan generasi 100 yaitu 0.714 menjadi *crossover* yang menghasilkan solusi terbaik.

BAB IV

PENUTUP

4.1 Kesimpulan

1. Proses pencarian solusi optimal *vehicle routing problem* dengan Algoritma Genetika berbasis *Fuzzy Logic Controller* meliputi langkah-langkah berikut: inialisasi, evaluasi, seleksi, pencarian nilai probabilitas crossover (p_c) dan probabilitas mutasi dengan *Fuzzy Logic Controller*, crossover dan mutasi. Berdasarkan analisis perhitungan masalah jaringan *vehicle routing problem* pada pendistribusian produk pakan burung CV.Ebod Jaya menggunakan algoritma genetika berbasis *Fuzzy Logic Controller* menggunakan populasi 100 dan generasi 100 menghasilkan solusi paling optimal yaitu 672.4 km. Hasil tersebut lebih baik dari 4 variasi populasi dan generasi lainnya.
2. Hasil perhitungan masalah solusi optimal dari permasalahan jaringan VRP menggunakan algoritma genetika berbasis *Fuzzy Logic Controller* pada penelitian ini menghasilkan rute terbaik pendistribusian produk pakan burung CV.Ebod Jaya ke seluruh pelanggan yaitu: **Ebod Jaya/Depot** (Jl. Kh. Yusuf, Tasikmadu, Kec. Lowokwaru, Kota Malang, Jawa Timur 65143) - Bpk. Sugih (Perum Puncak Sengkaling No. 12D Malang) - Bpk. Deni (Gg. IX, Sawojajar, Kedungkandang, Kota Malang) - Bpk. Dachman (Pakan Burung Ambonia, Jl. Terusan Kesatriaan No.1, Kesatrian, Blimbing, Malang) - Bpk. H. Budi (Pasar Burung Splindit Jl. Brawijaya No. 41 Malang) - Bpk.Effendy (Jl. Raden Tumenggung

Suryo No.35, Purwantoro, Blimbing, Kota Malang) - Sounic Master (Jl. Ahmad Yani, Gg. Satria Baru No. 31 Blimbing, Malang) - Bpk. Suhud (Jl. Raya Pakis No.73, Pakiskembar, Pakis, Malang) - Bpk. Sugeng / Toko 24 (Pakan Burung Abdul Majid, Jl. Brawijaya Kedong Solo, Porong, Kedungsolo, Sidoarjo) - **Depot** untuk kendaraan 1 ; **Depot** - Bpk. H. Selamat (Pakan Burung Raditya Jaya, Jl. Raden Wijaya No.53, Jotangan, Mojokerto) - Ibu Mustofa (Pasar Burung Empunala, Jl. Empu Nala, Balongsari, Magersari, Kota Mojokerto) - Bpk. Kholik (Pasar Burung Krajan Timur Kios No. 5 Krian, Surabaya) - Bpk. Jun (Town House Blok G-12 Sepanjang, Sidoarjo, Surabaya) - Gatot Sangkar (Taman Pondok Jati Blok M-9 Taman, Sidoarjo, Surabaya) - Bpk. Usman (Pasar Burung Sidoarjo, Jl. Raya Tenggulunan, Tenggulunan, Sidoarjo Regency) - Toko Haris (Jl. Panglima Sudirman, Jati, Mayangan, Kota Probolinggo) - Ibu Jimung (Pasar Sapi Tonggas, Jl. Tongas No.128, Tambakrejo, Tongas, Probolinggo) - **Depot** untuk kendaraan 2; dan **Depot** - Bpk. H. Kardi (Pakan Burung Garenk, Griya Kebraon 5, Kebraon, Karang Pilang, Surabaya) - Citrasari (Jl. Diponegoro No.196C Tegalsari, Surabaya) - Bpk. Heno (Jl. Tembaan No.79 Bubutan, Surabaya) - Ibu Yeni (Jl. Semarang No.142 Surabaya) - Ibu Suharsono (Jl. Pacuan Kuda No.154B Surabaya) - Toko Cahaya Baru (Ps. Dukuh Kupang, Jl. Dukuh Kupang Timur No.225, Dukuh Kupang, Dukuh Pakis, Surabaya) - Ibu Rohani (Pasar Burung Bratang, STAN Berdikari Los LLG No. 9, Surabaya) - Toko Seger Jaya (Jl. Panjang Jiwo Permai VI No.3, Panjang Jiwo, Tenggilis Mejoyo, Surabaya) - **Depot** untuk kendaraan 3 dengan total jarak yang ditempuh yaitu 672,4 km.

4.2 Saran

- Algoritma ini dapat dimanfaatkan pada variasi VRP yang lain seperti *Vehicle Routing Problem with Time Windows (VRPTW)* dengan tambahan kendala kapasitas kendaraan dan kendala batasan waktu.
- Pada penelitian selanjutnya dapat dikembangkan penyelesaian permasalahan VRP dengan algoritma dan *software* lain , agar didapat perbandingan dalam mengatasi solusi yang lebih baik dan pencarian lebih cepat.

DAFTAR PUSTAKA

- Budayasa, I.K. 2007. *Teori Graph dan Aplikasinya*. Surabaya: Unesa University Press.
- Desiani, Anita dan Arhami, Muhammad. 2006. *Konsep Kecerdasan Buatan*. Yogyakarta: Penerbit Andi.
- Hermansyah, Bambang. 2011. *Penyelesaian Vehicle Routing Problem(VRP) Menggunakan Algoritma Genetika. Skripsi*. Pekanbaru : Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau.
- Kallehauge, B., J. Larsen, dan O.B.G. Marsen. 2001. *Lagrangean Duality Applied on Vehicle Routing with Time Windows*. Technical Report, IMM, Technical University of Denmark.
- Kusumadewi, S.2003. *Artificial Intelligence: Teknik dan Aplikasinya*. Yogyakarta: Graha Ilmu.
- Kusumadewi, S.,dkk. 2006. *Fuzzy Multi Criteria Decision Making*, Yogyakarta: Graha Ilmu.
- Kusumadewi, S., Purnomo, H. 2010. *Aplikasi Logika Fuzzy untuk Pendukung Keputusan*. Yogyakarta: Graha Ilmu.
- 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*, hal.33-38. Universitas Islam Indonesia, Yogyakarta, 21 Juni 2008.
- Surekha, P. dan Sumathi, S. 2011. *Solution To Multi-Depot Vehicle Routing Problem Using Genetic Algorithms*. Department of EEE.PSG College of Technology. India.

- Toth, P. and Daniele Vigo. 2002. *An Overview of Vehicle Routing Problem*,
In Toth,P. and Vigo,D. The Vehicle Routing Problem. Philadelphia :
Society for Industrial and Applied Mathematics.
- Wibowo, M.A. 2009. *Aplikasi Algoritma Genetika Untuk Penjadwalan Mata
Kuliah*. Semarang: FMIPA UNDIP.
- Widodo, Prabowo. 2012. *Penerapan Soft Computing Dengan MATLAB*. Ban-
dung: Rekayasa Sains.

LAMPIRAN-LAMPIRAN

LAMPIRAN 1

Tabel Koordinat Lokasi Alamat (dalam Lintang Utara dan Lintang Selatan)

Kode Alamat	x	y
A	-7.9213037	112.6234373,17
B	-7.4108709	112.5779671,17
C	-7.9425434	112.6409541,17
D	-7.6131089	112.1344393,9
E	-7.7185538	113.0797367,17
F	-7.504883	112.5492138,17
G	-7.3504747	112.6902266,17
H	-7.3684785	112.6942589,17
I	-7.297809	112.7586173,17
J	-7.9780435	112.6293137,17
K	-7.975928	112.6422153,17
L	-7.962884	112.6383946,17
M	-7.9749151	112.6516035,17
N	-7.2805899	112.7088613,17
O	-7.9651853	112.7209297,17
P	-7.3125588	112.7626765,17
Q	-7.2505569	112.7283693,17
R	-7.5445748	112.6739899,17
S	-7.246713	112.7319247,17
T	-7.4670305	112.4453993,17
U	-7.333364	112.7003363,17
V	-7.2776279	112.7271493,17
W	-7.2619634	112.7145402,17
X	-7.9247504	112.5717908,17
Y	-7.742784	113.2138323,17

LAMPIRAN 2

Data Alamat Lokasi

No	Nama Pelanggan	Alamat	Kode Alamat
1	Ebod Jaya (Depot)	Jl. Kh. Yusuf, Tasikmadu, Kec. Lowokwaru, Kota Malang, Jawa Timur 65143	A
2	Bpk. Kholik	Pasar Burung Krajan Timur Kios No. 5 Krian, Surabaya	B
3	Sounic Master	Jl. Ahmad Yani, Gg. Satria Baru No. 31 Blimbing, Malang	C
4	Bpk. Usman	Pasar Burung Sidoarjo, Jl. Raya Tenggulunan, Sidoarjo Regency	D
5	Ibu Jimung	Pasar Sapi Tonggas, Jl. Tongas No.128, Tambakrejo, Probolinggo	E
6	Bpk. H. Selamat	Pakan Burung Raditya Jaya, Jl. Raden Wijaya No.53, Jotangan, Mojokerto	F
7	Bpk. Jun	Town House Blok G-12 Sepanjang, Sidoarjo - Surabaya	G
8	Gatot Sangkar	Taman Pondok Jati Blok M-9 Taman, Sidoarjo - Surabaya	H
9	Ibu Rohani	Pasar Burung Bratang, STAN Berdikari Los LGG No. 9 Surabaya	I
10	Bpk. H. Budi	Pasar Burung Splindit Jl. Brawijaya No. 41 Malang	J
11	Bpk. Dachman	Pakan Burung Ambonia, Jl. Terusan Kesatria No.1, Blimbing, Malang	K

No	Nama Pelanggan	Alamat	Kode Alamat
12	Bpk.Effendy	Jl. Raden Tumenggung Suryo No.35, Purwanto, Blimbing, Kota Malang	L
13	Bpk. Deni	Gg. IX, Sawojajar, Kedungkandang, Kota Malang	M
14	Toko Cahaya Baru	Ps. Dukuh Kupang, Jl. Dukuh Kupang Timur No.225, Surabaya	N
15	Bpk. Suhud	Jl. Raya Pakis No.73, Pakiskembar, Pakis, Malang	O
16	Toko Seger Jaya	Jl. Panjang Jiwo Permai VI No.3, Panjang Jiwo, Tenggilis Mejoyo, Surabaya	P
17	Ibu Yeni	Jl. Semarang No.142 Surabaya	Q
18	Bpk. Sugeng / Toko 24	Pakan Burung Abdul Majid, Jl. Brawijaya Kedong Solo, Porong, Sidoarjo	R
19	Bpk. Heno	Jl. Tembaan No.79 Bubutan, Surabaya	S
20	Ibu Mustofa	Pasar Burung Empunala, Jl. Empu Nala, Balongsari, Magersari, Mojokerto	T
21	Bpk. H. Kardi	Pakan Burung Garenk, Griya Kebraon 5, Karang Pilang, Surabaya	U
22	Citrasari	Jl. Diponegoro No.196C Tegalsari, Surabaya	V
23	Ibu Suharsono	Jl. Pacuan Kuda No.154B Surabaya	W
24	Bpk. Sugih	Perum Puncak Sengkaling No. 12D Malang	X
25	Toko Haris	Jl. Panglima Sudirman, Jati, Mayangan, Kota Probolinggo	Y

LAMPIRAN 3

Tabel Data Jarak

	A	B	C	D	E	F	G	H	I
A	0	80.2	5.1	65.1	70.5	70.5	82.6	83.5	88.7
B	80.2	0	79.4	19.2	78	12.7	16.3	17.3	27.7
C	5.1	79.4	0	65.3	70.9	70.9	83.6	84.3	88.6
D	65.1	19.2	65.3	0	58.8	25.9	20.2	15.8	22.7
E	70.7	78	70.9	58.8	0	72.2	77.2	78.1	82.9
F	70.7	12.7	70.9	25.9	72.2	0	28.6	29.3	40
G	82.6	16.3	83.6	20.2	77.2	28.6	0	3	12.7
H	83.5	17.3	84.3	15.8	78.1	29.3	3	0	14.2
I	88.7	27.7	88.6	22.7	82.9	40	12.7	14.2	0
J	8.4	83.8	4.5	68.4	73	78.1	85.9	87.5	92.7
K	8.8	83.6	4.8	66	73.1	78.2	86.1	87.6	92.9
L	6.7	81.6	2.7	66.2	70.7	75.8	83.7	85.2	90.5
M	9.6	84.1	5.3	69.1	73.6	78.7	86.6	88.1	93.4
N	88.3	29.5	88.6	25.9	82.9	41.7	14.4	15.9	8.3
O	13.5	87.6	10	72.3	76.9	82	89.9	91.4	96.7
P	88.2	27.8	88.6	22.7	82.9	40	12.7	14.3	4.6
Q	95.1	25.5	6.3	67.6	89.6	48.4	16.3	17.9	8.4
R	55.6	35.8	55.9	12.2	52.1	18.4	29.7	31.2	36.5
S	95.4	32	95.7	27	90	48.8	21.5	23.1	8.5
T	85.7	18.7	86	35.7	87.2	16.9	33.7	34.3	46.1
U	88.3	18.9	87	24.3	81.3	31.1	3.2	5.3	11.1
V	93.1	28.7	93.4	23.7	87.7	41	13.7	15.2	6.4
W	92	33.4	92.3	29.6	86.8	45.6	18.4	19.9	9.3
X	10.2	89.4	13.6	71.6	78.7	83.8	91.7	93.2	98.5
Y	87.3	70.5	86.8	74.7	15.9	87.7	92.6	94.1	99.4

	J	K	L	M	N	O	P	Q	R
A	8.4	8.8	6.7	9.6	88.3	13.5	88.2	95.1	55.6
B	83.8	83.6	81.6	84.1	29.5	87.6	27.8	25.5	35.8
C	4.5	4.8	2.7	5.3	88.6	10	88.6	6.3	55.9
D	68.4	66	66.2	69.1	25.9	72.3	22.7	67.6	12.2
E	73	73.1	70.7	73.6	82.9	76.9	82.9	89.6	52.1
F	78.1	78.2	75.8	78.7	41.7	82	40	48.4	18.4
G	85.9	86.1	83.7	86.6	14.4	89.9	12.7	16.3	29.7
H	87.5	87.6	85.2	88.1	15.9	91.4	14.3	17.9	31.2
I	92.7	92.9	90.5	93.4	8.3	96.7	4.6	8.4	36.5
J	0	2.1	2.6	3.4	91.9	12.8	91.8	98.7	59.2
K	2.1	0	2.5	1.3	91.5	11.2	91.6	98.3	58.9
L	2.6	2.5	0	2.9	89.3	10.8	89.3	96.1	56.6
M	3.4	1.3	2.9	0	92.2	9.9	92.4	99.1	59.6
N	91.9	91.5	89.3	92.2	0	95.5	10.7	6.4	35.3
O	12.8	11.2	10.8	9.9	95.5	0	94.5	101	61.8
P	91.8	91.6	89.3	92.4	10.7	94.5	0	93.9	35.7
Q	98.7	98.3	96.1	99.1	6.4	101	93.9	0	41.8
R	59.2	58.9	56.6	59.6	35.3	61.8	35.7	41.8	0
S	99	98.7	96.4	99.2	7	102	11.1	1.1	41.7
T	76.9	91.3	86.7	89.8	47.4	94.2	45.4	52.7	33.4
U	90.4	90	87.8	90.5	7.9	92.9	12.4	15.2	33
V	96.7	96.3	94.1	97.1	4.5	99.2	9.3	3.3	39.3
W	95.6	95.3	93	95.8	3.6	98.4	11.7	3	38.5
X	11.4	13	11.8	12.6	97.3	19.9	97.8	104	64.8
Y	90.1	89.8	87.6	90.5	98.2	92.7	98.6	105	68

	S	T	U	V	W	X	Y
A	95.4	85.7	88.3	93.1	92	10.2	87.3
B	32	18.7	18.9	28.7	33.4	89.4	70.5
C	95.7	86	87	93.4	92.3	13.6	86.8
D	27	35.7	24.3	23.7	29.6	71.6	74.7
E	90	87.2	81.3	87.7	86.8	78.7	15.9
F	48.8	16.9	31.1	41	45.6	83.8	87.7
G	21.5	33.7	3.2	13.7	18.4	91.7	92.6
H	23.1	34.3	5.3	15.2	19.9	93.2	94.1
I	8.5	46.1	11.1	6.4	9.3	98.5	99.4
J	99	76.9	90.4	96.7	95.6	11.4	90.1
K	98.7	91.3	90	96.3	95.3	13	89.8
L	96.4	86.7	87.8	94.1	93	11.8	87.6
M	99.2	89.8	90.5	97.1	95.8	12.6	90.5
N	7	47.4	7.9	4.5	3.6	97.3	98.2
O	102	94.2	92.9	99.2	98.4	19.9	92.7
P	11.1	45.4	12.4	9.3	11.7	97.8	98.6
Q	1.1	52.7	15.2	3.3	3	104	105
R	41.7	33.4	33	39.3	38.5	64.8	68
S	0	52.2	14.7	4.1	3.7	103	80.8
T	52.2	0	36.3	46.1	50.7	97.3	103
U	14.7	36.3	0	10	12.6	96.4	97.2
V	4.1	46.1	10	0	3.6	98	98.9
W	3.7	50.7	12.6	3.6	0	102	103
X	103	97.3	96.4	98	102	0	95.2
Y	80.8	103	97.2	98.9	103	95.2	0

LAMPIRAN 3

Tabel data jarak untuk kendaraan 1

	A	C	J	K	L	M	O	R	X
A	0	5.1	8.4	8.8	6.7	9.6	13.5	55.6	10.2
C	5.1	0	4.5	4.8	2.7	5.3	10	55.9	13.6
J	8.4	4.5	0	2.1	2.6	3.4	12.8	59.2	11.4
K	8.8	4.8	2.1	0	2.5	1.3	11.2	58.9	13
L	6.7	2.7	2.6	2.5	0	2.9	10.8	56.6	11.8
M	9.6	5.3	3.4	1.3	2.9	0	9.9	59.6	12.6
O	13.5	10	12.8	11.2	10.8	9.9	0	61.8	19.9
R	55.6	55.9	59.2	58.9	56.6	59.6	61.8	0	64.8
X	10.2	13.6	11.4	13	11.8	12.6	19.9	64.8	0

Tabel data jarak untuk kendaraan 2

	A	B	D	E	F	G	H	T	Y
A	0	80.2	65.1	70.5	70.5	82.6	83.5	85.7	87.3
B	80.2	0	19.2	78	12.7	16.3	17.3	18.7	70.5
D	65.1	19.2	0	58.8	25.9	20.2	15.8	35.7	74.7
E	70.5	78	58.8	0	72.2	77.2	78.1	87.2	15.9
F	70.5	12.7	25.9	72.2	0	28.6	29.3	16.9	87.7
G	82.6	16.3	20.2	77.2	28.6	0	3	33.7	92.6
H	83.5	17.3	15.8	78.1	29.3	3	0	34.3	94.1
T	85.7	18.7	35.7	87.2	16.9	33.7	34.3	0	103
Y	87.3	70.5	74.7	15.9	87.7	92.6	94.1	103	0

Tabel data jarak untuk kendaraan 3

	A	I	N	P	Q	S	U	V	W
A	0	88.7	88.3	88.2	95.1	95.4	88.3	93.1	92
I	88.7	0	8.3	4.6	8.4	8.5	11.1	6.4	9.3
N	88.3	8.3	0	10.7	6.4	7	7.9	4.5	3.6
P	88.2	4.6	10.7	0	93.9	11.1	12.4	9.3	11.7
Q	95.1	8.4	6.4	93.9	0	1.1	15.2	3.3	3
S	95.4	8.5	7	11.1	1.1	0	14.7	4.1	3.7
U	88.3	11.1	7.9	12.4	15.2	14.7	0	10	12.6
V	93.1	6.4	4.5	9.3	3.3	4.1	10	0	3.6
W	92	9.3	3.6	11.7	3	3.7	12.6	3.6	0

LAMPIRAN 4

Kode Program dengan Matlab

Kode program untuk fungsi **fungsikro.m**

```
function hasiljarak=fungsikro(kro,datajarak1)
[baris, kolom]=size(kro);
hasiljarak=[];
for i=1:baris
    jumlah=0;
    for j=1:kolom-1
        jumlah=jumlah+datajarak1(kro(i,j),kro(i,j+1));
    end
    hasiljarak=[hasiljarak;jumlah];
end
```

Kode program untuk Fuzzy Logic Controller.fis

```
[System]
Name='Fuzzy Logic Controller'
Type='mamdani'
Version=2.0
NumInputs=2
NumOutputs=2
NumRules=9
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='Populasi'
Range=[0 1000]
NumMFs=3
MF1='SMALL': 'zmf', [50 250]
MF2='MEDIUM': 'gausmf', [80 275]
MF3='LARGE': 'smf', [350 500]
```

```

[Input2]
Name='Generasi'
Range=[0 1000]
NumMFs=3
MF1='SHORT':'zmf',[50 200]
MF2='MEDIUM':'gausmf',[80 275]
MF3='LONG':'smf',[350 500]

[Output1]
Name='ProbCrossover'
Range=[0.6 0.9]
NumMFs=4
MF1='SMALL':'zmf',[0.625 0.7]
MF2='MEDIUM':'trapmf',[0.63 0.7 0.72 0.78]
MF3='LARGE':'trapmf',[0.72 0.78 0.8 0.87]
MF4='VERYLARGE':'smf',[0.8 0.875]

[Output2]
Name='ProbMutasi'
Range=[0 0.25]
NumMFs=4
MF1='VERYSMALL':'zmf',[0.025 0.1]
MF2='SMALL':'trapmf',[0.047 0.083 0.1 0.14]
MF3='MEDIUM':'trapmf',[0.1 0.14 0.167 0.2]
MF4='LARGE':'smf',[0.15 0.225]

[Rules]
1 1, 2 4 (1) : 1
2 1, 1 3 (1) : 1
3 1, 1 2 (1) : 1
1 2, 3 3 (1) : 1
2 2, 3 2 (1) : 1
3 2, 2 1 (1) : 1
1 3, 4 2 (1) : 1
2 3, 4 1 (1) : 1

```

Kode program untuk **fuzzygenetik.m**

```

%% TAHAP PENGIMPUMAN DATA
clear;clc;
disp('-----');
disp('----- Vehicle Routing Problem dengan Algoritma Genetik
berbasis Fuzzy Logic Controller -----');
disp('-----');
fprintf('\n\n');

% Memasukan data

clear;clc;
load datajarak3

% Memasukkan data
Pop = input('Masukkan ukuran Populasi : ');
Gen = input('Masukkan ukuran Generasi : ');

iterasi=1;
syarat=0;

clc;
disp('-----');
disp('----- Algoritma Genetik berbasis Fuzzy Logic
Controller -----');
disp('-----');
%% 1.TAHAP INISIALISASI

% INISIALISASI JALUR DENGAN MENGGUNAKAN 8 KROMOSOM SEBAGAI PROSES GENETIK
%inialisai bersifat random asalkan memenuhi syarat yaitu kota awal
%keberangkatan harus diinisialisasi dengan '1', dan tujuan akhir harus
%kembali ketitik awal keberangkatan.

%kro=kromosom
kro1=[1 2 3 4 5 6 7 8 9 1 ];
kro2=[1 3 4 5 2 7 9 6 8 1 ];
kro3=[1 5 7 2 6 3 4 8 9 1 ];
kro4=[1 2 3 4 5 9 7 6 8 1 ];
kro5=[1 8 3 7 5 4 6 2 9 1 ];
kro6=[1 7 5 2 9 6 8 3 4 1 ];
kro7=[1 2 5 4 6 3 8 9 7 1 ];
kro8=[1 5 4 6 3 9 2 7 8 1 ];
kro=[kro1;kro2;kro3;kro4;kro5;kro6;kro7;kro8];

% Bkro= banyaknya Baris kromosom ; Kkro= banyaknya Kolom kromosom
[bcro, kcro]=size(kro);
nilaikro=funksikro(kro,datajarak3);
fprintf('\nSetelah TAHAP INISIALISASI maka kromosom yang didapat
yaitu :\n\n');
for i=1:bcro
    fprintf('kromosom[%d] = %d %d %d %d %d %d %d %d %d %d ; nilai dari
kromosomnya adalah : %1.2f\n',i,kro(i,:),nilaikro(i));
end
fprintf('\nSelanjutnya kromosom akan diproses...\n\n');
disp('-----');
disp('-----');
awalan=randi(bcro,1);
koreksi=kro(awalan,:);
cek=nilaikro(awalan);

```

```

%% 2. TAHAP EVALUASI KROMOSOM
pengulangannilai=0;
keluar=false;
while keluar==false || pengulangannilai<=20
    fprintf('\nDari data diatas akan dilakukan proses Algoritma Genetik ke-
    %d\n',iterasi);
    disp('Didapat data sebagai berikut. ');
    syarat=0;keluar=false;
    nilaikro=funksikro(kro,datajarak3);

% PROSES MENCARI NILAI FITNESS
    for i=1:bcro
        fitness(i)=1/nilaikro(i);
    end
    fitness=fitness';
    totalq=sum(fitness);

%% 3. TAHAP SELEKSI KROMOSOM

% PROSES ROULETTE-WHEEL

% PROSES MENCARI NILAI PROBABILITAS
    for i=1:bcro
        p(i)=fitness(i)/totalq;
    end
    p=p';

% PROSES MENCARI NILAI KUMULATIF FITNESS
    c(1) = p(1);
    for i=2:bcro
        c(i)=c(i-1)+p(i);
    end
    c=c';

% MEMBANGKITAN 8 ANGKA RANDOM
    r=rand(bcro,1);

%crobaru=untuk menempatkan crommosom baru

% PROSES SELEKSI KROMOSOM
    crobaru=[];
    for i=1:bcro
        j=1;
        while c(j)<r(i)
            j=j+1;
        end
        crobaru=[crobaru;kro(j,:)];
    end
    kro=crobaru;
    nilaikro=funksikro(kro,datajarak3);
    fprintf('\n----- TAHAP SELEKSI -----
');
    fprintf('\ndidapat kromosom baru yaitu :\n\n');
    for i=1:bcro
        fprintf('kromosom[%d] = %d %d %d %d %d %d %d %d %d %d ; nilai dari
kromosomnya adalah : %1.2f\n',i,kro(i,:),nilaikro(i));
    end
    fprintf('\n');

```



```

%% 4. TAHAP PINDAH SILANG

%yang akan diubah tergantung ketentuan, saat ini sudah ditentukan bahwa
%akan ada paling banyak 3 kromosom yang di proses, dan setiap kromosom
%akan dipilih dari gen a ke 2 sampai ke 10.

% PENCARIAN NILAI PROBABILITAS CROSSOVER DENGAN FUZZY LOGIC CONTROLLER
    fis=readfis('Fuzzy Logic Controller');
    out=evalfis([Pop Gen],fis);
    pc=out(1);

% PROSES PEMILIHAN KROMOSOM UNTUK DIPINDAH SILANG
    randomselector=rand(bcro,1);
    ubah=[];berubah=[];
    for i=1:bcro
        if randomselector(i)<pc
            ubah=[ubah,i];
        end
    end
    l=length(ubah);
    berubah=randi(bcro,1,l);

%dengan melakukan metode Partial-mapped crossover(PMX), kelebihanya tidak
%perlu ada proses "perbaikan" karena yang di pindah silang sudah memiliki
%jenis perubahan yang sama.

% PROSES PINDAH SILANG
    krobaru=kro;
    for i=1:l
        krobaru(ubah(i),2:9)=kro(berubah(i),(2:9));
    end
    kro=krobaru;
    nilaikro=funksikro(kro,datajarak3);
    fprintf('\n----- TAHAP PINDAH SILANG -----');
    fprintf('\ndidapat kromosom baru yaitu :\n\n');
    for i=1:bcro
        fprintf('kromosom[%d] = %d %d %d %d %d %d %d %d %d ; nilai dari
kromosomnya adalah : %1.2f\n',i,kro(i,:),nilaikro(i));
    end
    fprintf('\n');

```

```

%% 5. TAHAP MUTASI

%yang akan diubah tergantung ketentuan, saat ini sudah ditentukan bahwa
%akan ada paling banyak 3 kromosom yang di proses, dan setiap kromosom
%akan dipilih 2 gen secara acak untuk dimutasi, gen yang dipilih hanya dari
%gen a ke 2 sampai ke 9.

% PENCARIAN NILAI PROBABILITAS MUTASI DENGAN FUZZY LOGIC CONTROLLER
    fis=readfis('Fuzzy Logic Controller');
    out=evalfis([Pop Gen],fis);
    pm=out(2);

% PROSES PEMILIHAN KROMOSOM UNTUK DIMUTASI
    randomselector=rand(bcro,1);
    ubah=[];berubah=[];
    for i=1:bcro
        if randomselector(i)<pm
            ubah=[ubah,i];
        end
    end
    l=length(ubah);
    berubah=randi(bcro,1,1);

% PROSES MUTASI
    l=length(ubah);
    if l>0
        r=1+randi(7,1,2);
    end
    for i=1:l
        swap=kro(ubah(i),r(i,1));
        kro(ubah(i),r(i,1))=kro(ubah(i),r(i,2));
        kro(ubah(i),r(i,2))=swap;
    end
    nilaikro=funksikro(kro,datajarak3);

%% 6. TAHAP EVALUASI

%tahap evaluasi adalah tahap penentuan dimana semua tahap harus diulang
%kembali atau berhenti. tahap ini akan menentukan apakah kromosom yang ada
%sudah memiliki gen yang optimum.

%TAHAP EVALUASI AKAN BERHENTI SAAT SEMUA KROMOSOM SUDAH MEMILIKI GEN YANG
%SAMA DAN GEN TERSEBUT SUDAH DIANGGAP MEMILIKI NILAI YANG OPTIMUM.
%JIKA BELUM MAKA SEMUA PROSES AKAN DIULANG DARI TAHAP SELEKSI.

    [hasil,indeksnya]=min(nilaikro);
    galat = 0.00001;
    for i=1:bcro
        if abs(nilaikro(i)-hasil) < galat
            syarat=syarat+1;
        end
    end
end

```

```

if cek>hasil
    cek=hasil;
    koreksi=kro(indeksnya,:);
    pengulangannilai=0;
elseif abs(cek-hasil) < galat
    pengulangannilai=pengulangannilai+1;
    if syarat>4
        keluar=true;
    end
else
    [~,indekmax]=max(nilaikro);
    kro(indekmax,:)=koreksi;
end
fprintf('\n----- TAHAP MUTASI -----');
fprintf('\ndidapat kromosom baru yaitu :\n\n');
for i=1:bcro
    fprintf('kromosom[%d] = %d %d %d %d %d %d %d %d %d %d ; nilai dari
kromosomnya adalah : %1.2f\n',i,kro(i,:),nilaikro(i));
end
fprintf('\n');
disp('-----');
disp('-----');
iterasi=iterasi+1;
end

```

```

%% MENAMPILKAN DATA YANG SUDAH SELESAI DIPROSES.
fprintf('\n
HASIL PROSES ALGORITMA GENETIK BERBASIS FUZZY
LOGIC CONTROLLER');
fprintf('\n\nsetelah proses ke-%d didapat jalur terpendek yaitu %1.1f\ndengan
track : ',iterasi-1,hasil);
disp(kro(indeksnya,:));
fprintf('\n');
disp('-----');
disp('-----');

```