

**OPTIMASI RUTE SEORANG LOPER KORAN DI FIDI AGENCY
MENGUNAKAN ALGORITMA GENETIKA METODE
SELEKSI RANKING**

SKRIPSI

**Diajukan kepada Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Negeri Yogyakarta
untuk Memenuhi Sebagian Persyaratan
guna Memperoleh Gelar Sarjana Sains**



**Oleh
Antonius Yuni Setiyawan
09305141007**

**PROGRAM STUDI MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI YOGYAKARTA
2014**

HALAMAN PERSETUJUAN

Skripsi

OPTIMASI RUTE SEORANG LOPER KORAN DI FIDI AGENCY MENGUNAKAN ALGORITMA GENETIKA METODE SELEKSI RANKING

Yang disusun oleh :

Nama : Antonius Yuni Setiawan

NIM : 09305141007

Prodi : Matematika

Telah disetujui untuk diujikan di depan dewan penguji skripsi

Program Studi Matematika

Jurusan Pendidikan Matematika

Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Negeri Yogyakarta

Yogyakarta, 18 November 2014

• Menyetujui •

Pembimbing



Fitriana Yuli S, M. Si

NIP. 19840707 200801 2 003

PENGESAHAN

SKRIPSI DENGAN JUDUL “OPTIMASI RUTE SEORANG LOPER KORAN DI FIDI AGENCY MENGUNAKAN ALGORITMA GENETIKA METODE SELEKSI RANKING”

Yang disusun oleh:
Nama : Antonius Yuni Setiyawan
NIM : 09305141007
Prodi : Matematika

Skripsi ini telah diuji di depan Dewan Penguji Skripsi pada tanggal 10
Desember 2014 dan dinyatakan LULUS.

DEWAN PENGUJI			
Nama	Jabatan	Tanda Tangan	Tanggal
<u>Fitriana Yuli S, M. Si.</u> NIP. 19840707200801 2 003	Ketua Penguji		2 Jan 2015
<u>Dwi Lestari, M. Sc.</u> NIP. 19850513201012 2 006	Sekretaris Penguji		28 Des 2014
<u>Sahid, M. Sc.</u> NIP. 19650905199110 1 100	Penguji Utama		23 Des 2014
<u>Nur Insani, M. Sc.</u> NIP. 19810106200501 2 005	Penguji Pendamping		31 Des 2014

Yogyakarta, 2-1-15
Fakultas Matematika dan Ilmu
Pengetahuan Alam
Dekan,



Dr. Hartono
NIP. 19620329 198702 1 002

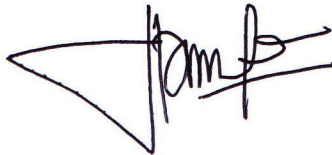
SURAT PERNYATAAN

Dengan ini saya Antonius Yuni Setiyawan menyatakan bahwa skripsi ini benar-benar karya saya sendiri. Sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang ditulis atau diterbitkan orang lain kecuali sebagai acuan atau kutipan dengan mengikuti tata penulisan karya ilmiah yang telah lazim.

Apabila terbukti pernyataan saya ini tidak benar, maka sepenuhnya menjadi tanggung jawab saya.

Yogyakarta, 18 November 2014

Yang menyatakan,

A handwritten signature in black ink, appearing to be 'Amf' with a large loop on the left and a horizontal line extending to the right.

Antonius Yuni Setiyawan

NIM. 09305141007

MOTTO

“Jawaban sebuah keberhasilan adalah terus belajar dan tak kenal lelah”

“Bukan kurangnya bakat atau tidak adanya modal yang menghalangi kita dari sukses, tapi tidak cukupnya keberanian”

“Dimana ada usaha disitu ada jalan, semua akan indah jika keberhasilan yang diraih diimbangi dengan sebuah proses kerja keras yang tak mengenal letih”

“Kegagalan hanya terjadi bila kita menyerah dengan cepat tanpa usaha yang keras”

“Bahagia itu sederhana, dimana kita bisa menikmati apa yang telah kita lakukan”

“Setiap pekerjaan dapat diselesaikan dengan mudah bila dikerjakan tanpa keengganan”

PERSEMBAHAN

“Tuhan Yang Maha Esa”

Terima kasih atas segala karunia yang telah Engkau anugerahkan padaku

“Ibu dan Bapak”

Doamu yang tiada terputus, kerja keras tiada henti, pengorbanan yang tak terbatas dan kasih sayang yang tak terbatas pula. Semuanya membuatku bangga memiliki kalian. Tiada kasih sayang yang seindah dan seabadi kasih sayangmu.

“Kakak-kakakku tercinta yang selalu mendukung dan memberi motivasi”

“Ibu Fitriana Yuli S, M. Si”

Terima kasih atas bimbingan dan masukan yang telah di berikan selama penulisan skripsi

“Teman-teman yang kusayangi, Yudha, Lulus, Chandra Hadi, Rudi, Falah, Aldi, Aran, Susi, Madha, Didi, Wachid. Kepada M. Thoyib yang selalu menemani, dan kepada Ani yang selalu memberi motivasi dan semangat”

“Rekan-rekan Matematika Subsidi 2009”

OPTIMASI RUTE SEORANG LOPER KORAN DI FIDI AGENCY MENGUNAKAN ALGORITMA GENETIKA METODE SELEKSI RANKING

Oleh
Antonius Yuni Setiyawan
NIM 09305141007

ABSTRAK

Algoritma genetika merupakan suatu metode pencarian berdasarkan pada mekanisme seleksi alam. Algoritma ini digunakan untuk mendapatkan solusi dalam masalah optimasi. Masalah optimasi yang akan dibahas adalah penentuan rute terpendek dalam pendistribusian koran. Tujuan penulisan skripsi ini adalah menjelaskan aplikasi algoritma genetika dengan metode seleksi ranking untuk mencari rute terpendek pendistribusian koran di Fidi Agency.

Langkah-langkah menentukan rute terpendek menggunakan algoritma genetika dengan metode seleksi ranking adalah mendefinisikan rute ke dalam individu dalam sebuah populasi, menghitung nilai *fitness* individu, menentukan induk dari individu dengan seleksi ranking, melakukan *order cross over* pada induk yang terpilih, menghasilkan individu baru dengan *swapping mutation*, menyusun populasi baru sampai memperoleh individu dengan nilai *fitness* optimum.

Hasil yang diperoleh dari perhitungan menggunakan algoritma genetika dengan seleksi ranking adalah rute pendistribusian koran di Fidi Agency sejauh 51,70 km. Rute ditetapkan setelah nilai *fitness* mengalami konvergen pada generasi ke-136. Nilai *fitness* yang mengalami konvergen adalah 0,0193.

Kata kunci: algoritma genetika, *fitness*, seleksi ranking, pendistribusian koran.

KATA PENGANTAR

Segala puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa yang telah memberikan rahmat dan karunia, sehingga penulis dapat menyelesaikan skripsi dengan judul “Optimasi Rute Seorang Loper Koran Di Fidi Agency Menggunakan Algoritma Genetika Metode Seleksi Ranking”.

Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bimbingan, arahan, bantuan serta motivasi dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Dr. Hartono, M. Si, selaku Dekan FMIPA UNY,
2. Bapak Dr. Sugiman, selaku Ketua Jurusan Pendidikan Matematika,
3. Bapak Dr. Agus Maman A, M. Si, selaku Koordinator Program Studi Matematika dan Dosen Pembimbing Akademik,
4. Ibu Fitriana Yuli S, M. Si, selaku Dosen Pembimbing Skripsi yang telah memberikan arahan, motivasi serta saran kepada penulis,
5. Bapak Sahid, M. Sc, selaku Dosen Penguji Utama yang telah memberikan berbagai masukan yang membangun,
6. Ibu Dwi Lestari, M. Sc, selaku Dosen Sekretaris Penguji yang telah memberikan berbagai masukan yang membangun,
7. Ibu Nur Insani, M. Sc, selaku Dosen Penguji Pendamping yang telah memberikan berbagai masukan yang membangun,
8. Seluruh Bapak/Ibu Dosen Jurusan Pendidikan Matematika yang telah memberikan banyak ilmu bermanfaat,

9. Teman-teman Matsub'09 yang telah memberikan bantuan dan semangat dalam menyelesaikan skripsi ini,

10. Seluruh pihak yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari adanya keterbatasan kemampuan, pengetahuan dan pengalaman. Oleh karena itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga skripsi ini dapat bermanfaat dan Tuhan dan Maha Esa memberikan balasan kebaikan kepada semua pihak yang telah membantu dalam penyelesaian skripsi ini.

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN.....	iv
HALAMAN MOTTO	v
HALAMAN PERSEMBAHAN	vi
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN.....	xiv
DAFTAR SIMBOL	xv

BAB I PENDAHULUAN

A. Latar Belakang Masalah	1
B. Pembatasan Masalah	3
C. Perumusan Masalah	3
D. Tujuan Penelitian	4
E. Manfaat Penelitian	4

BAB II KAJIAN TEORI

A. Graf.....	5
1. Definisi Graf.....	5
2. Graf tak-berarah (<i>undirected graph</i>).....	5
3. Graf Berbobot.....	6
B. <i>Travelling Salesman Problem</i> (TSP).....	7
C. Algoritma Genetika.....	10
1. Pengertian Algoritma Genetika.....	10

2.	Aplikasi Algoritma Genetika.....	11
3.	Karakteristik Algoritma Genetika.....	12
4.	Komponen Algoritma Genetika.....	15
a.	Teknik Pengkodean.....	15
b.	Membangkitkan Populasi Awal.....	16
c.	Fungsi <i>Fitness</i>	17
d.	Seleksi	18
e.	Pindah Silang (<i>Cross Over</i>).....	19
f.	Mutasi.....	22
g.	<i>Elitism</i>	23
BAB III PEMBAHASAN		
A.	Model Matematika TSP Pendistribusian Surat Kabar di Fidy Agency	25
B.	Penyelesaian Masalah TSP di Fidy Agency Menggunakan Algoritma Genetika.....	32
1.	Mendefinisikan Gen.....	32
2.	Mendefinisikan Individu.....	33
3.	Membangkitkan Populasi Awal.....	34
4.	Menentukan Nilai <i>Fitness</i>	36
5.	Seleksi Ranking.....	38
6.	Pindah Silang (<i>Cross Over</i>).....	40
7.	Mutasi.....	41
8.	Pembentukan Populasi Baru.....	43
BAB IV PENUTUP		
A.	Kesimpulan.....	50
B.	Saran.....	52
DAFTAR PUSTAKA		53
LAMPIRAN		55

DAFTAR TABEL

	Halaman
Tabel 1 Tabel Permasalahan TSP	10
Tabel 2 Alamat Kantor dan Pelanggan Setelah Direduksi	29
Tabel 3 Jarak Kantor Agen ke Rumah Pelanggan dan Antar Pelanggan	30
Tabel 4 Hasil Percobaan dengan <i>software Matlab</i>	47
Tabel 5 Alamat Kantor dan Pelanggan dan Pengelompokan	55
Tabel 6 Nilai Fitness Individu pada Generasi ke-116.....	72
Tabel 7 Probabilitas Individu pada Generasi ke-116	73

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Graf G_1	5
Gambar 2.2 Graf G_2	6
Gambar 2.3 Graf G_3	6
Gambar 2.4 Graf K Berbobot	7
Gambar 2.5 Skema Algoritma Genetika oleh David Goldberg (1989)	14
Gambar 2.6 Skema Algoritma Genetika oleh Michalewicz (1996)	15
Gambar 2.7 Sistematika proses <i>cross over</i>	20
Gambar 2.8 Sistematika Proses Mutasi.....	22
Gambar 3.1 Graf G_8	27
Gambar 3.2 Graf G_9	28
Gambar 3.3 Peta kantor dan rumah pelanggan.....	30
Gambar 3.4 Grafik pergerakan nilai <i>fitness</i>	48

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Data Alamat Kantor dan Pelanggan dan Pengelompokan.....	55
Lampiran 2 Hasil Wawancara dengan Salah Satu Loper Koran	58
Lampiran 3 Prosedur Algoritma Genetika Menggunakan <i>Software Matlab</i> dalam Penyelesaian Masalah Pendistribusian Koran pada Fidi Agency.....	60
Lampiran 4 Penghitungan Nilai <i>Fitness</i> dan Seleksi Ranking	71
Lampiran 5 Output <i>Software Matlab</i> Proses Pencarian Rute Terpendek Pendistribusian Koran di Fidi Agency	75

DAFTAR SIMBOL

V	himpunan titik
E	himpunan rusuk
v_i	titik ke- i
e_i	rusuk ke- i
n	banyaknya titik atau jumlah gen
s	banyaknya lintasan tertutup atau banyaknya semua rute
x_{ij}	merepresentasikan jarak dari rute dari i ke j
c_{ij}	merepresentasikan ada tidaknya perjalanan dari i ke j
f	nilai <i>fitness</i>
$R[i]$	Ranking inidividu ke- i
$P[i]$	Probabilitas individu ke- i

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Surat kabar atau yang lebih dikenal dengan koran, merupakan salah satu media informasi yang ada di masyarakat. Koran sudah dianggap sebagai media informasi yang efisien. Pembaca koran meliputi berbagai kalangan masyarakat. Dengan membaca koran, kita dapat terus mengikuti perkembangan-perkembangan aktual, baik dari dalam negeri maupun dari luar negeri. Pada intinya kita semua membutuhkan informasi. Informasi sudah dianggap sebagai kebutuhan pokok, yang tidak boleh dilewatkan (Muktamarudin, 2009: 3).

Pendistribusian koran dimulai dari gudang produksi yang menuju ke agen distributor. Agen akan mendistribusikan koran ke toko-toko dan pelanggan yang sudah ada melalui loper koran. Banyaknya permintaan dari pelanggan koran di suatu kota harus dipenuhi oleh agen distributor koran. Pendistribusian koran dari tempat agen distributor ke toko atau pelanggan haruslah tepat waktu dan biasanya dilakukan pada pagi-pagi benar karena pada pagi hari para konsumen ingin mengetahui kabar yang terjadi di hari sebelumnya ataupun agenda yang akan datang. Pelanggan yang bekerja sebagai pegawai kantor menginginkan pendistribusian koran sepagi mungkin. Berhentinya pelanggan dalam berlangganan sering diakibatkan oleh pendistribusian koran yang terlalu siang. Penentuan rute distribusi koran yang tepat dengan memaksimalkan pelanggan yang dilayani akan meminimumkan waktu tempuh dan biaya perjalanan. Seiring

dengan minimalnya waktu tempuh dan biaya perjalanan maka keuntungan yang diperoleh suatu perusahaan akan maksimal.

Fidi Agency merupakan salah satu agen distributor koran Tribun yang terletak di Jalan Kaliurang Km. 8,5, Sleman, Yogyakarta. Pengambilan data dilakukan dengan mewawancarai salah satu loper koran. Kendaraan yang dipakai seorang loper adalah kendaraan roda dua. Seorang loper mendistribusikan koran pada pukul 05.00 WIB, setelah koran didistribusikan dari gudang menuju agen.

Seorang loper diharuskan menyelesaikan pendistribusian koran sebelum pukul 07.30 WIB, dikarenakan banyak dari pelanggan adalah pekerja kantoran. Realita yang terjadi di lapangan adalah loper tersebut mendistribusikan koran dengan rute tempuh 72 km, kecepatan 50 km/jam dan menyelesaikan dengan waktu 2,5 jam. Permasalahan seorang loper koran yang harus mrngunjungi sejumlah pelanggan dengan rute terpendek tersebut dapat dibawa ke dalam permasalahan TSP.

Penelitian-penelitian yang telah dilakukan mengenai TSP dengan algoritma genetika antara lain penerapan algoritma genetika untuk *travelling salesmen problem* dengan menggunakan *order cross over* dan *insertion mutation* oleh Anwar Toni dan Yuliani Willi (2005), *travelling salesman problem* menggunakan algortima genetika via GPS berbasis android oleh Asmi Baharudin (2012), penyelesaian *travelling salesman problem* dengan algoritma genetika oleh Kusri (2008), penerapan algoritma genetika pada persoalan pedagang keliling oleh Aulia Fitrah dan Achmad Zaky (2006). Skripsi ini menggunakan algoritma

genetika metode seleksi ranking untuk menyelesaikan *travelling salesman problem* pendistribusian koran di Fidi Agency.

Algoritma genetika pertama kali dirintis oleh John Holland pada tahun 1960-an. Algoritma genetika telah dipelajari, diteliti, dan diaplikasikan secara luas pada berbagai bidang. Algoritma genetika banyak digunakan pada masalah praktis yang berfokus pada pencarian parameter-parameter optimal. Keuntungan penggunaan algoritma genetika sangat jelas terlihat dari kemudahan implementasi dan kemampuannya untuk menemukan solusi yang ‘bagus’ (bisa diterima) secara tepat untuk masalah-masalah berdimensi tinggi (Suyanto, 2005: 3). Suyanto menyelesaikan algoritma genetika menggunakan *software* Matlab. Pemilihan bahasa pemrograman Matlab lebih didasari pada kemudahan didalam mengimplementasikan komponen-komponen algoritma genetika yang banyak menggunakan operasi matriks (Suyanto, 2005: 17).

B. Pembatasan Masalah

Pembatasan masalah dalam skripsi ini adalah :

1. Agen distributor koran adalah Fidi Agency yang terletak di Jalan Kaliurang km 8,5 Yogyakarta
2. Permasalahan rute adalah dari salah satu loper koran di Fidi Agency
3. Metode seleksi yang digunakan dalam algoritma genetika adalah seleksi ranking

C. Perumusan Masalah

1. Bagaimana model matematika pendistribusian koran di Fidi Agency?

2. Bagaimana pencarian rute terpendek pendistribusian koran oleh salah satu loper di Fidi Agency dengan algoritma genetika metode seleksi ranking?

D. Tujuan Penelitian

Tujuan penulisan penelitian ini menurut rumusan masalah di atas adalah

1. Memperoleh model matematika pendistribusian koran di Fidi Agency
2. Memperoleh pencarian rute terpendek pendistribusian koran oleh salah satu loper di Fidi Agency dengan algoritma genetika metode seleksi ranking

E. Manfaat Penelitian

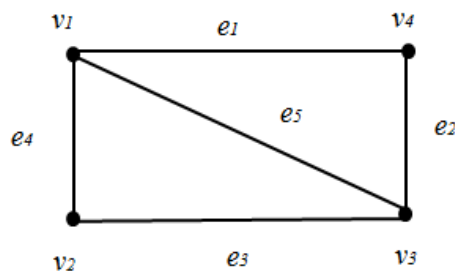
1. Memperdalam pemahaman tentang metode algoritma genetika terutama dalam mengoptimasi suatu rute distribusi
2. Memberi solusi permasalahan distribusi koran bagi Fidi Agency
3. Mengetahui rute terpendek sehingga meminimalisir waktu tempuh pendistribusian koran

BAB II KAJIAN TEORI

A. Graf

1. Pengertian Graf (Mardiyono, 1996: 1)

Graf adalah kumpulan simpul atau titik dan segmen garis yang menghubungkan dua titik yang disebut rusuk. Graf G dapat dilambangkan dengan (V, E) dengan himpunan $V \neq \emptyset$ yang unsur-unsurnya disebut titik dan himpunan E yang unsur-unsurnya disebut rusuk. Titik-titik pada graf dapat merupakan obyek sembarang seperti kota. Rusuk dapat menunjukkan hubungan (relasi) sembarang seperti ruas jalan raya penghubung antar dua kota. Gambar 2.1 adalah contoh graf $G_1(V, E)$ dengan $V = \{v_1, v_2, v_3, v_4\}$ dan $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$

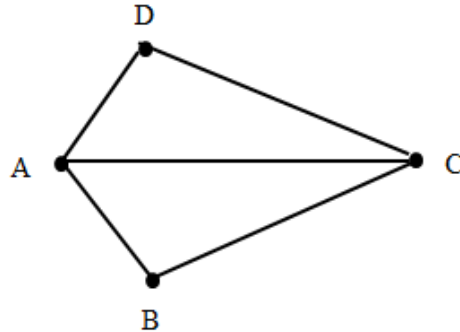


Gambar 2.1 Graf G_1

2. Graf Tak-Berarah (*Undirected Graph*)

Graf tak-berarah G didefinisikan sebagai suatu pasangan terurut (V, E) , dengan V suatu himpunan dan E suatu himpunan yang unsur-unsurnya berupa multi himpunan dengan dua unsur dari V . Suatu graf tak-berarah dapat direpresentasikan secara geometrik sebagai himpunan titik-titik V dengan suatu himpunan garis-garis E antara titik-titik tersebut.

Graf tak-berarah adalah graf yang tidak memiliki orientasi arah. Berikut contoh graf tak-berarah:

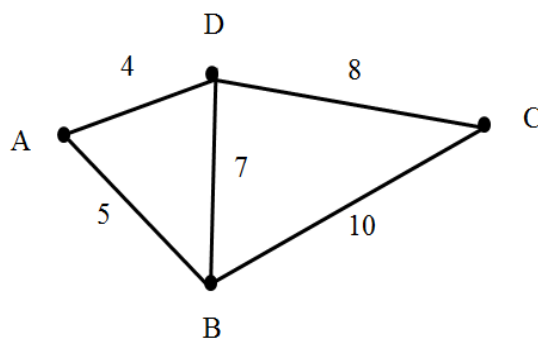


Gambar 2.2 Graf G_2

Urutan pasangan titik yang dihubungkan oleh rusuk pada graf G_2 tidak diperhatikan. Misalnya (A, C) dan (C, A) merupakan pasangan rusuk yang sama.

3. Graf Berbobot

Graf berbobot adalah graf yang setiap rusuknya diberi sebuah harga (bobot). Bobot pada setiap rusuk berbeda-beda bergantung pada masalah yang dimodelkan dengan graf. Bobot dapat menyatakan jarak antara dua buah kota, biaya perjalanan antara dua buah kota, waktu tempuh, ongkos produksi, dan sebagainya (Munir, 2009: 376). Berikut contoh graf berbobot:



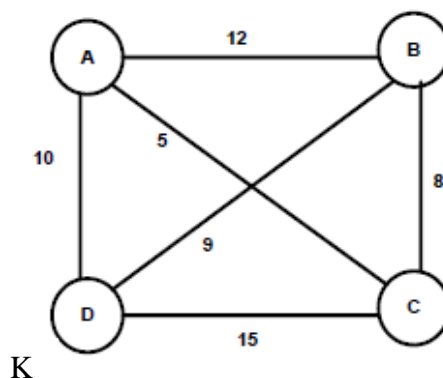
Gambar 2.3 Graf G_3

Urutan pasangan titik yang dihubungkan oleh rusuk pada graf G_3 memiliki bobot berupa jarak dalam satuan kilometer. Nilai bobot (A, B) adalah 5 km, dan nilai bobot (C, D) adalah 8 km.

B. *Travelling Salesman Problem (TSP)*

TSP adalah bagaimana menentukan rute optimal perjalanan *salesman* yang harus melalui semua kota tujuan tepat satu kali dan harus kembali ke kota awal (Edgar G. G dan Michael M. P, 1997: 377). Masalah tersebut dapat dimodelkan ke dalam graf berbobot dengan setiap kota tujuan digambarkan sebagai titik dan ruas jalan (rusuk) sebagai sisi berbobot mewakili panjang ruas jalan antara dua kota. Masalah perjalanan *salesman* tersebut dalam graf adalah mencari lintasan tertutup minimum yang memuat semua titik dalam suatu graf berbobot tersebut.

Permasalahan *traveling salesman problem* dapat dimodelkan sebagai graf tak-berarah dan berbobot. Berikut representasi banyaknya lintasan tertutup TSP dalam graf :



Gambar 2.4 Graf K Berbobot

Dari graf tersebut dicari banyaknya lintasan tertutup dari titik A kembali lagi ke titik A. Terdapat 6 lintasan tertutup pada Graf K yaitu, A-B-C-D-A, A-D-

C-B-A, A-C-D-B-A, A-B-D-C-A, A-D-B-C-A, dan A-C-B-D-A, sehingga banyaknya lintasan tertutup (s) dapat dicari dengan :

$$s = (n-1)! \quad (2.1)$$

Dalam graf K, rusuk-rusuknya tidak berarah sehingga $d(A,B) = d(B,A)$.
banyaknya lintasan menjadi :

$$s = \frac{(n-1)!}{2} \quad (2.2)$$

Karena sirkuit A-B-C-D-A = A-D-C-B-A, A-C-D-B-A = A-C-D-B-A, dan A-C-B-D-A = A-D-B-C-A. Jadi banyaknya semua kemungkinan lintasan tertutup ditentukan dengan rumus (2.2). Pada skripsi ini, lintasan tertutup dalam permasalahan TSP disebut dengan rute perjalanan.

Asumsi dasar dari model TSP adalah setiap titik hanya dilalui sebanyak satu kali, seorang harus kembali ke titik awal dia berangkat, dan jarak antar dua titik adalah jarak yang terdekat (Davendra, 2010: 7). Berikut dijelaskan formulasi model TSP untuk meminimumkan rute. Didefinisikan $G(N, A)$ adalah graf tak berarah, dengan $N = \{v_1, v_2, v_3, \dots, v_n\}$ merupakan himpunan titik yang merepresentasikan n pelanggan, dan v_i merepresentasikan kantor, sedangkan $A = \{(v_i, v_j) \mid v_i, v_j \in V, i \neq j\}$ adalah himpunan sisi yang menghubungkan antar titik, yang merepresentasikan ruas jalan penghubung antar titik.

Jarak perjalanan dari titik i ke titik j direpresentasikan oleh c_{ij} . Selanjutnya didefinisikan variabel keputusan x_{ij} yang merepresentasikan ada tidaknya perjalanan dari titik i ke j dalam suatu rute sebagai berikut :

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

Jika Z merupakan fungsi tujuan TSP, maka fungsi tujuan Z dirumuskan dengan meminimumkan

$$Z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}. \quad (2.3)$$

dengan kendala,

$$\sum_{i=1}^n x_{i1} = 1. \quad (2.4)$$

⋮

$$\sum_{i=1}^n x_{in} = 1.$$

dan

$$\sum_{j=1}^n x_{1j} = 1. \quad (2.5)$$

⋮

$$\sum_{j=1}^n x_{nj} = 1.$$

Permasalahan TSP jika disajikan dalam bentuk tabel adalah :

Tabel 1. Tabel Permasalahan TSP

	v_1	v_2	v_3	V_n	$\sum_{j=1}^n x_{ij}$
v_1	$C_{11}X_{11}$	$C_{12}X_{12}$	$C_{13}X_{13}$	$C_{1n}X_{1n}$	1
v_2	$C_{21}X_{21}$	$C_{22}X_{22}$	$C_{23}X_{23}$	$C_{2n}X_{2n}$	1
v_3	$C_{31}X_{31}$	$C_{32}X_{32}$	$C_{33}X_{33}$	$C_{3n}X_{3n}$	1
....
v_n	$C_{n1}X_{n1}$	$C_{n2}X_{n2}$	$C_{n3}X_{n3}$	$C_{nn}X_{nn}$	1
$\sum_{i=1}^n x_{ij}$	1	1	1	1	n

TSP dalam perkembangannya memiliki beberapa jenis, sesuai kendala yang ditambahkan dalam model (Takes, 2010: 13). *Travelling Salesman Problem with Times Windows (TSPTW)* adalah jenis dari TSP yang menambahkan kendala batasan waktu pelayanan (*times windows*) untuk setiap kota. Jenis TSP yang lain adalah *m-Travelling Salesman Problem* (m-TSP). Jenis TSP ini menambahkan kendala jumlah *salesman*, sehingga terdapat sejumlah *m salesman* untuk mengunjungi seluruh kota.

C. Algoritma Genetika

1. Pengertian Algoritma Genetika

Menurut Dwi Ana Ratna Wati (2011: 162), algoritma genetika merupakan sebuah metode untuk menyelesaikan masalah optimasi dengan meniru proses

seleksi alam, proses yang menyebabkan evolusi biologis. Algoritma genetika memodifikasi sebuah populasi individu yang merupakan kandidat solusi secara berulang-ulang. Algoritma genetika memilih individu-individu secara acak dari populasi untuk dijadikan induk-induk dan menggunakannya untuk menghasilkan keturunan pada generasi berikutnya. Setelah melalui beberapa generasi, populasi berevolusi menuju kondisi solusi optimal.

Algoritma genetika adalah algoritma yang memanfaatkan proses seleksi alamiah yang dikenal dengan proses evolusi (Goldberg, 1989: 15). Dalam proses evolusi, individu secara terus menerus mengalami perubahan gen untuk menyesuaikan diri dengan lingkungan hidupnya. "Hanya individu-individu yang kuat yang mampu bertahan". Proses seleksi alamiah ini melibatkan perubahan gen yang terjadi pada individu melalui proses perkembangbiakan. Dalam algoritma genetika, proses perkembangbiakan menjadi proses dasar yang menjadi perhatian utama, dengan dasar berpikir: "Bagaimana mendapatkan keturunan yang lebih baik".

Proses algoritma genetika secara umum untuk semua kasus adalah mendefinisikan individu, mendefinisikan nilai *fitness*, menentukan proses pembangkitan populasi awal, menentukan proses seleksi, menentukan proses perkawinan silang dan mutasi gen yang akan digunakan (Ahmad Basuki, 2003: 4).

2. Aplikasi Algoritma Genetika

Sejak pertama kali dirintis oleh John Holland pada tahun 1960-an, algoritma genetika telah dipelajari, diteliti dan diaplikasikan secara luas pada berbagai bidang. Algoritma genetika banyak digunakan pada masalah praktis yang berfokus pada pencarian parameter-parameter optimal. Hal ini membuat banyak

orang mengira bahwa algoritma genetika hanya bisa digunakan untuk masalah optimasi.

Keuntungan penggunaan algoritma genetika sangat jelas terlihat dari kemudahan implementasi dan kemampuannya untuk menemukan solusi yang ‘bagus’ (bisa diterima) secara cepat untuk masalah-masalah berdimensi tinggi. Algoritma genetika sangat berguna dan efisien untuk masalah dengan karakteristik sebagai berikut (Suyanto, 2005: 3):

- a. ruang masalah sangat besar, kompleks, dan sulit dipahami,
- b. kurang atau bahkan tidak ada pengetahuan yang memadai untuk mempresentasikan masalah ke dalam ruang pencarian yang lebih sempit,
- c. tidak tersedianya analisis matematika yang memadai,
- d. ketika metode-metode konvensional sudah tidak mampu menyelesaikan masalah yang dihadapi,
- e. solusi yang diharapkan tidak harus paling optimal, tetapi solusi yang cukup ‘bagus’ atau bisa diterima,
- f. terdapat batasan waktu, misalnya dalam *real systems* atau sistem waktu nyata.

Algoritma genetika telah banyak diaplikasikan untuk penyelesaian masalah dan pemodelan dalam bidang teknologi, bisnis, dan *entertainment*.

3. Karakteristik Algoritma Genetika (Satriyanto, 2009: 27)

a. Istilah dalam algoritma genetika

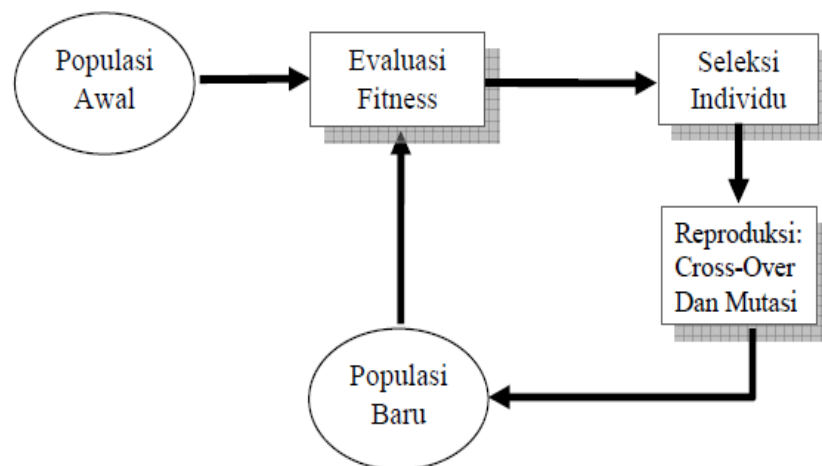
Beberapa definisi penting yang perlu diperhatikan dalam membangun penyelesaian masalah menggunakan algoritma genetika yakni :

- 1) Gen (*Genotype*) adalah sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom.
- 2) *Allele* yaitu nilai dari sebuah gen, bisa berupa bilangan biner, float, integer, karakter dan kombinatorial.
- 3) Kromosom adalah gabungan gen-gen yang membentuk nilai tertentu.
- 4) Individu merupakan suatu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.
- 5) Populasi merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi. Populasi terdiri dari sekumpulan kromosom.
- 6) Induk adalah kromosom yang akan dikenai operasi genetika (*cross over*)
- 7) *Cross over* adalah operasi genetika yang mewakili proses perkembangbiakan antar individu.
- 8) *Offspring* adalah kromosom yang merupakan hasil dari operasi genetika (*cross over*) dikenal keturunan atau sebagai anak.
- 9) Mutasi merupakan operasi genetika yang mewakili proses mutasi dalam perjalanan hidup individu. Mutasi berperan menghasilkan perubahan acak dalam populasi, yang berguna untuk menambah variasi dari kromosom-kromosom dalam sebuah populasi.
- 10) Proses seleksi merupakan proses yang mewakili proses seleksi alam (*natural selection*) dari teori Darwin. Proses ini dilakukan untuk menentukan induk dari operasi genetika (*cross over*) yang akan dilakukan untuk menghasilkan keturunan (*offspring*).

- 11) Nilai *fitness* merupakan penilaian yang menentukan bagus tidaknya sebuah kromosom.
- 12) Fungsi evaluasi adalah fungsi yang digunakan untuk menentukan nilai dari nilai *fitness*. Fungsi evaluasi ini merupakan sekumpulan kriteria-kriteria tertentu dari permasalahan yang ingin diselesaikan.
- 13) Generasi merupakan satuan dari populasi setelah mengalami operasi-operasi genetika, berkembang biak, dan menghasilkan keturunan. Pada akhir dari setiap generasi, untuk menjaga agar jumlah kromosom dalam populasi tetap konstan, kromosom-kromosom yang mempunyai nilai *fitness* yang rendah dan memiliki peringkat di bawah nilai minimal akan dihapus dari populasi.

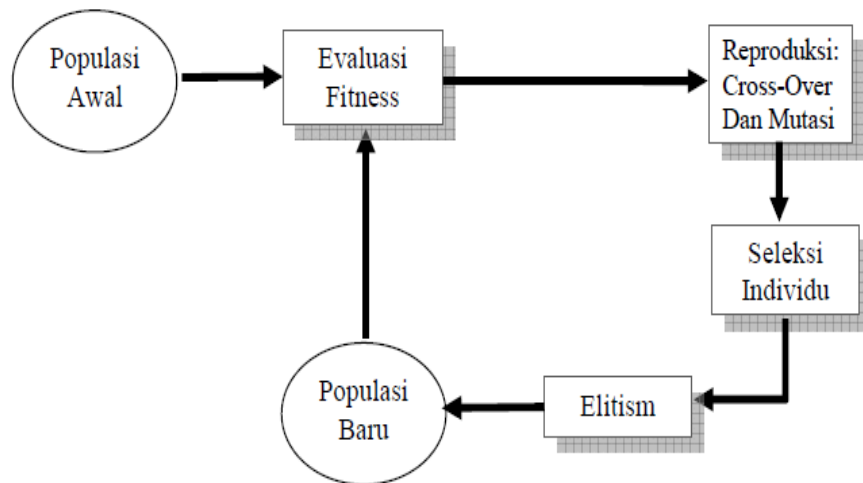
b. Skema alur algoritma genetika

Skema algoritma genetika pertama kali dikemukakan oleh David Goldberg (1989), dimana gambaran tersebut dapat dilihat pada gambar berikut:



Gambar 2.5 Skema Algoritma Genetika oleh David Goldberg (1989)

Siklus ini kemudian diperbaiki oleh beberapa ilmuwan yang mengembangkan algoritma genetika, yaitu Zbignew Michalewicz dengan menambahkan operator *elitism* dan membalik proses seleksi setelah proses reproduksi.



Gambar 2.6 Skema Algoritma Genetika oleh Michalewicz (1996)

4. Komponen Algoritma Genetika

Pada dasarnya, algoritma genetika memiliki enam komponen. Beberapa komponen algoritma genetika diantaranya yaitu :

a. Teknik Pengkodean (Kusumadewi, 2003: 39)

Teknik pengkodean adalah bagaimana mengkodekan gen dari kromosom, dimana gen merupakan bagian dari kromosom. Satu gen biasanya akan mewakili satu variabel. Agar dapat diproses melalui algoritma genetika, maka alternatif solusi tersebut harus dikodekan terlebih dahulu ke dalam bentuk kromosom. Masing-masing kromosom berisi sejumlah gen yang mengkodekan informasi yang disimpan di dalam kromosom.

Gen dapat direpresentasikan dalam bentuk: bit, bilangan real, daftar aturan, elemen permutasi, elemen program atau representasi lainnya yang dapat diimplementasikan untuk operator genetika. Dengan demikian kromosom dapat direpresentasikan dengan menggunakan:

- 1) String bit : 10011 dst.
- 2) Array bilangan real : 65.65, -67.98, 77.34 dst.
- 3) Elemen permutasi : E2, E10, E5 dst.
- 4) Daftar aturan : R1, R2, R3 dst.
- 5) Struktur lainnya

b. Membangkitkan Populasi Awal (Kusumadewi, 2003: 42)

Membangkitkan populasi awal adalah proses membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu. Ukuran untuk populasi tergantung pada masalah yang akan diselesaikan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian dilakukan pembangkitan populasi awal.

Teknik dalam pembangkitan populasi awal ini ada beberapa cara, diantaranya adalah *random generator*, pendekatan tertentu, dan permutasi gen. Pada skripsi ini digunakan teknik pembangkitan populasi berupa *random generator*. Inti dari cara ini adalah melibatkan pembangkitan bilangan random untuk nilai setiap gen sesuai dengan representasi kromosom yang digunakan.

Dalam TSP, populasi menyatakan sejumlah solusi (jalur) yang dicari secara acak. Individu menyatakan solusi (jalur) yang mungkin. Misalkan dalam

populasi terdapat 4 individu, maka contoh populasi awal TSP dengan 7 kota (kota 1 sampai 7) adalah sebagai berikut:

1 2 4 3 6 5 7

5 7 4 6 2 3 1

7 2 6 4 5 1 3

4 6 5 7 3 2 1

c. Fungsi *Fitness* (Ahmad Basuki, 2003: 6)

Suatu individu atau kromosom dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performasinya. Fungsi yang digunakan untuk mengukur nilai kecocokan atau derajat optimalitas suatu kromosom disebut dengan fungsi *fitness* (*fitness function*). Nilai yang dihasilkan dari fungsi tersebut menandakan seberapa optimal solusi yang diperoleh. Algoritma genetika bertujuan mencari individu dengan nilai *fitness* yang paling tinggi.

Permasalahan TSP bertujuan meminimalkan jarak, maka nilai *fitness* adalah inversi dari total jarak dari jalur yang didapatkan. Cara melakukan inversi yaitu menggunakan rumus :

$$\text{Nilai } fitness = 1/x \quad (2.6)$$

atau

$$\text{Nilai } fitness = 100000 - x \quad (2.7)$$

dimana x adalah total jarak dari jalur yang didapatkan.

d. Seleksi

Seleksi merupakan proses pemilihan individu terbaik untuk menjadi calon orang tua yang akan dilakukan proses selanjutnya. Proses pemilihan tersebut biasanya dipilih berdasarkan probabilitas dari individu yang terbaik dalam populasi. Individu yang terbaik ditentukan berdasarkan nilai *fitness* masing-masing dari tiap-tiap individu (Csuhendra, 2012: 57).

Terdapat beberapa metode seleksi menurut Kusumadewi (2003: 43), yaitu seleksi ranking (*rank-based fitness assignment*), seleksi roulette wheel (*roulette wheel selection*), *stochastic universal sampling*, seleksi lokal (*local selection*), seleksi dengan pemotongan (*truncation selection*), dan seleksi dengan turnamen (*tournament selection*). Skripsi ini menggunakan metode seleksi ranking (*rank-based fitness assignment*). Cara kerja metode seleksi ranking adalah sebagai berikut :

- a) Dihitung nilai *fitness* dari masing-masing individu (f_i , dimana i adalah individu ke-1 s/d ke- n)
- b) Nilai *fitness* diurutkan dari nilai yang terkecil hingga paling besar
- c) Setelah diurutkan, kromosom terburuk diberi nilai *fitness* baru sebesar 1, kromosom kedua terburuk diberi nilai 2, dan seterusnya. Kromosom terbaik diberi nilai *fitness* baru sebesar n dimana n adalah banyak kromosom dalam suatu populasi.
- d) Dihitung total *fitness* semua individu
- e) Dihitung probabilitas masing-masing individu

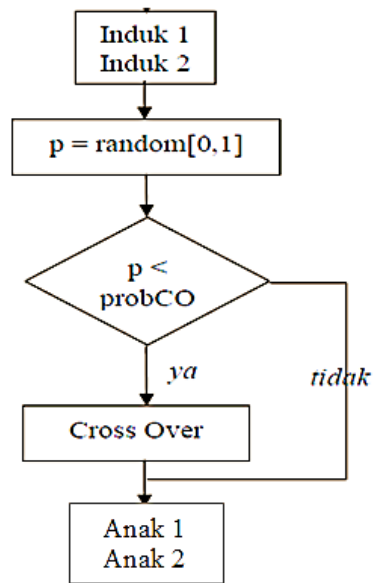
- f) Dari probabilitas tersebut, dihitung jatah masing-masing individu pada angka 1 sampai 100
- g) Dibangkitkan bilangan random antara 1 sampai 100
- h) Dari bilangan random yang dihasilkan, ditentukan individu mana yang terpilih dalam proses seleksi

Seleksi ranking memperbaiki proses seleksi dengan *roulette wheel*. Seleksi *roulette wheel* memungkinkan satu kromosom mempunyai nilai *fitness* yang mendominasi hingga 90%, sehingga nilai *fitness* yang lain akan mempunyai kemungkinan yang sangat kecil untuk terpilih.

e. Cross Over (Pindah Silang)

Pindah silang (*Cross over*) adalah operator dari algoritma genetika yang melibatkan dua induk untuk membentuk kromosom baru. Pindah silang menghasilkan keturunan baru dalam ruang pencarian yang siap diuji. Operasi ini tidak selalu dilakukan pada setiap individu yang ada. Individu dipilih secara acak untuk dilakukan *crossing* dengan P_c (*Probabilitas cross over*) antara 0,6 s/d 0,95. Jika pindah silang tidak dilakukan, maka nilai dari induk akan diturunkan kepada keturunan (Michalewicz, 1996: 35).

Prinsip dari pindah silang ini adalah melakukan operasi (pertukaran aritmatika) pada gen yang bersesuaian dari dua induk untuk menghasilkan individu baru. Proses *cross over* dilakukan pada setiap individu dengan probabilitas *cross over* yang ditentukan. Secara skematis proses *cross over* dapat digambarkan sebagai berikut:



Gambar 2.7 Sistematika proses *cross over*

Teknik *cross over* yang digunakan adalah teknik *order cross over*. *Order cross over* (OX) diperkenalkan oleh Davis (W.S.E. Tanjung, 2010: 47). Teknik ini dijelaskan dalam contoh 2.1.

Contoh 2.1 *order cross over*

Dari 2 induk diketahui:

$$p_1 = (1 \ 2 \ 3 \ | \ \mathbf{4 \ 5 \ 6 \ 7} \ | \ 8 \ 9)$$

$$p_2 = (4 \ 5 \ 2 \ | \ \mathbf{1 \ 8 \ 7 \ 6} \ | \ 9 \ 3)$$

Dibangkitkan 2 bilangan acak sebelum gen induk-1 dan setelah gen induk-1. Hal yang sama juga dilakukan untuk induk-2. Didapatkan keturunan dengan gen yang sama:

$$o_1 = (x \ x \ x \ | \ \mathbf{4 \ 5 \ 6 \ 7} \ | \ x \ x)$$

$$o_2 = (x \ x \ x \ | \ \mathbf{1 \ 8 \ 7 \ 6} \ | \ x \ x)$$

Langkah berikutnya untuk mendapatkan keturunan pertama adalah mengurutkan gen yang berada pada induk kedua dengan urutan gen yang berada pada posisi setelah bilangan acak kedua diikuti dengan gen yang berada pada posisi sebelum bilangan acak pertama dan diakhiri dengan gen yang berada pada posisi diantara kedua bilangan acak.

9-3-4-5-2-1-8-7-6

Kemudian gen yang telah diurutkan tersebut dibandingkan dengan keturunan pertama. Apabila gen tersebut ada pada keturunan kedua maka abaikan gen tersebut dari urutan itu.

Kemudian masukkan urutan yang baru saja didapat pada keturunan dengan cara memasukkan urutan gen pada posisi setelah bilangan acak kedua terlebih dahulu dan sisanya dimasukkan pada posisi sebelum bilangan acak pertama. Begitu juga untuk menghasilkan keturunan kedua. Keturunan 1 diperoleh:

$$o_1 = (x \ x \ x \mid 4 \ 5 \ 6 \ 7 \mid x \ x)$$

$$o_1 = (2 \ 1 \ 8 \mid 4 \ 5 \ 6 \ 7 \mid 9 \ 3)$$

dengan jalan yang sama buat o_2 sehingga :

$$o_2 = (x \ x \ x \mid 1 \ 8 \ 7 \ 6 \mid x \ x)$$

$$o_2 = (3 \ 4 \ 5 \mid 1 \ 8 \ 7 \ 6 \mid 9 \ 2)$$

Keterangan:

p_1 = Induk 1

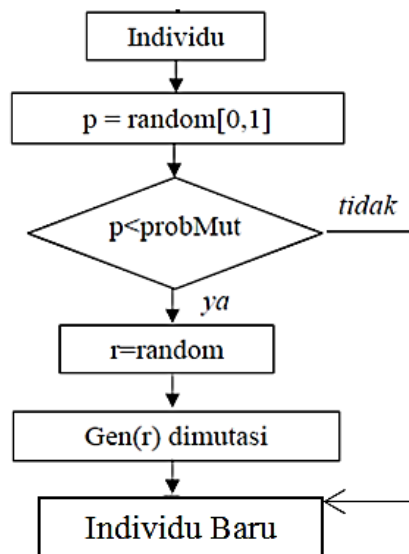
p_2 = Induk 2

o_1 = Keturunan 1 (anak ke-1)

o_2 = Keturunan 2 (anak ke-2)

f. Mutasi

Mutasi merupakan proses untuk mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Operasi mutasi yang dilakukan pada kromosom dengan tujuan untuk memperoleh kromosom-kromosom baru sebagai kandidat solusi pada generasi mendatang dengan *fitness* yang lebih baik, dan lama-kelamaan menuju solusi optimum yang diinginkan. Penekanan *selektif* memegang peranan yang penting. Jika dalam proses pemilihan kromosom-kromosom cenderung terus pada kromosom yang memiliki *fitness* yang tinggi saja, konvergensi prematur akan sangat mudah terjadi (N. Murniati, 2009: 62). Secara skematis proses mutasi dapat digambarkan sebagai berikut:



Gambar 2.8 Sistematika Proses Mutasi

Teknik mutasi yang digunakan dalam skripsi ini adalah teknik *swapping mutation*. Teknik ini diawali dengan memilih dua bilangan acak kemudian gen

yang berada pada posisi bilangan acak pertama ditukar dengan gen yang berada pada bilangan acak kedua dalam probabilitas tertentu (Suyanto, 2005: 57).

Contoh 2.2 *swapping mutation*:

Individu = (1 2 3 4 5 6 8 9 7)

Memindahkan 8 ke 2, sehingga didapatkan individu baru:

Individu = (1 8 3 4 5 6 2 9 7)

g. Elitism

Elitism merupakan proses untuk menjaga agar individu bernilai *fitness* tertinggi tersebut tidak hilang selama evolusi (Kusumadewi, 2003). Proses seleksi dilakukan secara random sehingga tidak ada jaminan bahwa suatu individu yang bernilai *fitness* tertinggi akan selalu terpilih. Walaupun individu bernilai *fitness* tertinggi terpilih, mungkin saja individu tersebut akan rusak (nilai *fitness* menurun) karena proses pindah silang. Oleh karena itu, untuk menjaga agar individu bernilai *fitness* tertinggi tersebut tidak hilang selama evolusi, maka perlu dibuat satu atau beberapa *copy*.

Komponen-komponen di atas akan mengevaluasi setiap populasi dengan menghitung nilai *fitness* setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti. Bila kriteria berhenti belum terpenuhi maka akan dibentuk lagi generasi baru dengan mengulangi langkah kedua. Beberapa kriteria berhenti menurut Budi Sukmawan (2003: 25) yang sering digunakan antara lain :

- a. berhenti pada generasi tertentu
- b. berhenti setelah dalam beberapa generasi berturut-turut didapatkan nilai *fitness* tertinggi yang tidak berubah

- c. berhenti bila dalam n generasi berikutnya tidak didapatkan nilai *fitness* yang lebih optimal.

BAB III

PEMBAHASAN

Berikut pembahasan mengenai penyelesaian TSP dengan algoritma genetika, yang selanjutnya mengambil studi kasus optimasi rute distribusi koran di Fidi Agency Yogyakarta.

A. Model Matematika TSP pada Pendistribusian Surat Kabar di Fidi Agency

Fidy Agency adalah salah satu perusahaan yang bergerak dalam bidang pendistribusian surat kabar. Surat kabar yang didistribusikan adalah koran Tribun Jogja. Fidy Agency terletak di Jalan Kaliurang Km. 8,5 Sleman Yogyakarta. Sasaran pendistribusian koran adalah kecamatan-kecamatan di Kabupaten Sleman yang terletak di sekitar Jalan Kaliurang, diantaranya adalah Kecamatan Ngemplak, Ngaglik, Pakem, Depok, Mlati, dan Kalasan. Banyak dari pelanggan koran adalah mereka yang bertempat tinggal di perumahan.

Menurut wawancara langsung dengan salah satu karyawan di Fidi Agency, alur distribusi koran dimulai dari gudang percetakan koran yang selanjutnya dikirim ke agen-agen yang tersebar. Dari agen akan didistribusikan lagi kepada toko-toko dan pelanggan oleh seseorang yang dinamakan loper koran. Fidi Agency mempunyai 6 loper koran. Setiap loper memiliki kendaraan masing-masing berupa sepeda motor. Kendaraan berupa sepeda motor akan memudahkan untuk menghindari kemacetan dan untuk mencari jalan pintas sehingga lebih cepat mencapai tujuan dan meminimalkan waktu tempuh.

Sepeda motor dapat mengangkut hingga 70 eksemplar koran. Setiap loper mempunyai wilayah distribusi yang sudah ditetapkan oleh kantor, yang sewaktu-waktu bisa berubah-ubah karena naik turunnya jumlah pelanggan. Waktu pendistribusian koran adalah pagi hari yaitu pukul 05.00 WIB yang dimulai dari kantor perusahaan dan kembali menuju kantor lagi. Sebagian besar pelanggan koran Tribun adalah pekerja kantoran, sehingga pelanggan menginginkan koran untuk didistribusikan sepagi mungkin.

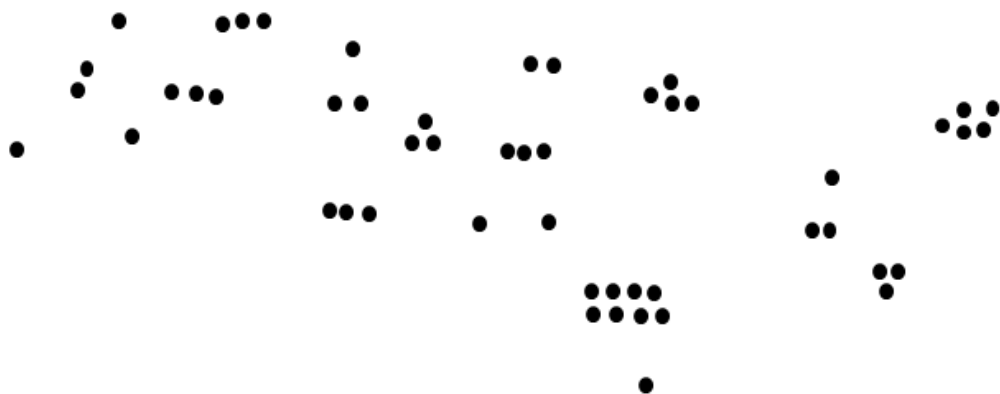
Pekerja kantor pada umumnya masuk kerja jam 07.30 WIB sehingga hanya ada sedikit waktu untuk membaca koran di pagi harinya. Keterlambatan dalam pendistribusian akan mengakibatkan keluhan dari pelanggan, berhentinya berlangganan atau pindah ke agen lain yang lebih pagi dalam pendistribusian. Pelanggan menginginkan koran didistribusikan sebelum jam 07.00 WIB. Masalah pendistribusian koran di Fidi Agency dapat dianggap sebagai *travelling salesman problem*.

Asumsi TSP dalam permasalahan pendistribusian koran ini adalah :

- Tiap rumah pelanggan hanya dikunjungi satu kali, dan rumah pelanggan diasumsikan sebagai titik
- Loper koran harus kembali ke kantor atau tempat semula, dikarenakan ia harus memberi laporan kepada kantor
- Karena jalan yang dilalui tidak ada yang merupakan jalan satu arah, maka jarak antar titik adalah simetris, artinya jarak dari titik A ke B sama dengan jarak dari titik B ke A

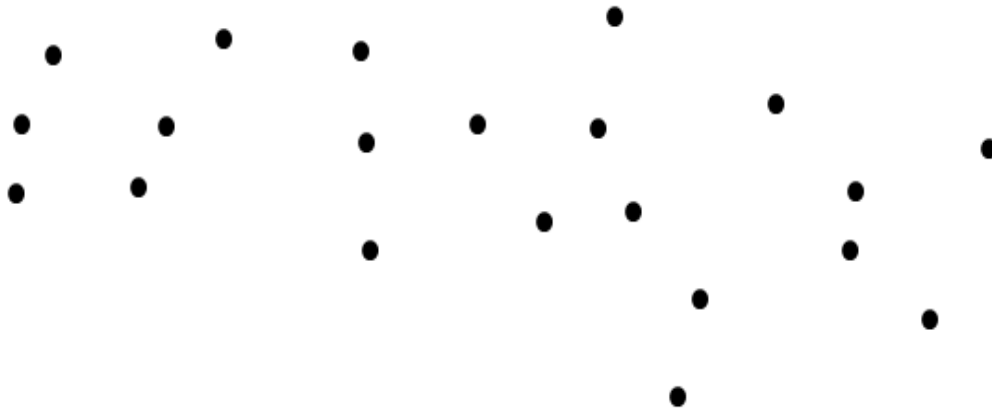
- Jarak antar titik harus jarak yang paling dekat, pencarian jarak terdekat menggunakan aplikasi Navitel. Aplikasi Navitel adalah aplikasi pada *hand phone android* yang dapat menentukan jarak terdekat antara dua titik pada peta.
- Jumlah pelanggan yang dikunjungi adalah 50 pelanggan (alamat pelanggan dapat dilihat pada lampiran 1). Pelanggan yang berdekatan dianggap satu titik, dengan kriteria pelanggan itu memiliki jarak kurang dari 50 meter, yang kebanyakan berada di kawasan perumahan.

Proses ini dinamakan proses reduksi. Proses reduksi adalah menjadikan beberapa pelanggan yang letaknya berdekatan menjadi satu saja. Misalnya di perumahan A terdapat 3 pelanggan yang rumahnya berdekatan yaitu kurang dari 50 m, maka 3 pelanggan tersebut direduksi menjadi 1 pelanggan. Berikut disajikan graf G_8 , graf yang memuat seluruh titik (rumah pelanggan) yang beralamatkan di Kecamatan Ngaglik, Ngeemplak, dan Kalasan (Gambar 3.1).



Gambar 3.1 Graf G_8

Graf G_8 menunjukkan graf kosong yang merupakan representasi rumah pelanggan di Fidy Agency, yang selanjutnya titik-titik yang berdekatan direduksi menjadi graf kosong yang ditunjukkan pada Graf G_9 berikut ini:



Gambar 3.2 Graf G_9

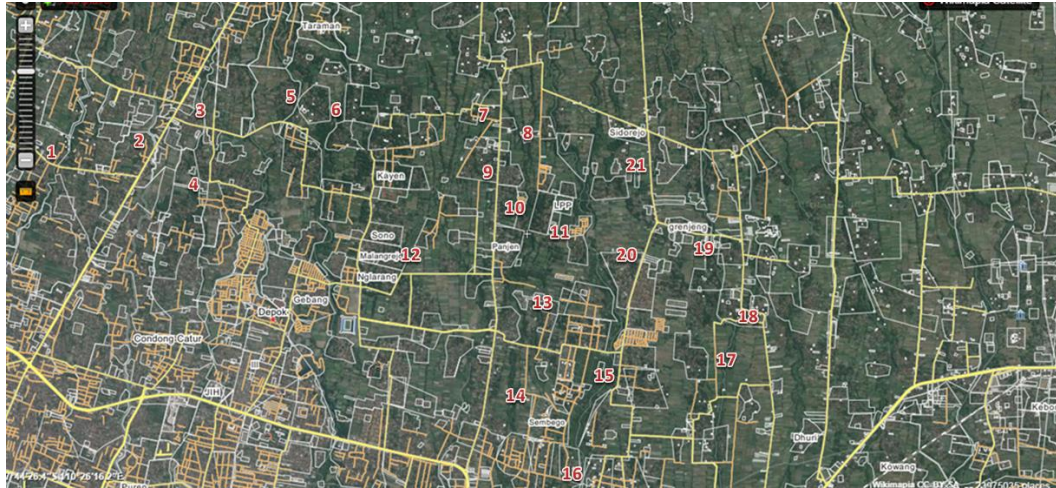
TSP adalah bagaimana menentukan rute optimal perjalanan *salesman* yang harus melalui semua kota tujuan tepat satu kali dan harus kembali ke kota awal. Masalah tersebut dapat dimodelkan kedalam graf berbobot dengan setiap kota tujuan digambarkan sebagai titik dan ruas jalan sebagai sisi berbobot mewakili panjang ruas jalan antara dua kota. Masalah perjalanan *salesman* tersebut dalam graf adalah mencari lintasan tertutup minimum yang memuat semua titik dalam suatu graf berbobot tersebut.

Selanjutnya adalah menentukan jarak kantor agen ke rumah pelanggan dan antar rumah pelanggan. Jarak yang dicari adalah jarak terpendek yang tidak melalui rumah pelanggan yang lain. Sebelum itu, akan disajikan tabel alamat kantor dan rumah pelanggan yang sudah direduksi menjadi 21 titik.

Tabel 2. Alamat Kantor dan Pelanggan Setelah Direduksi

Titik	Alamat Pelanggan
1	kantor <i>Fidi Agency</i>
2	wilayah Jl Kaliurang km.9
3	wilayah Gandok
4	wilayah Perum Fortuna
5	wilayah Palgading 1
6	wilayah Palgading 2
7	wilayah klinik Assyifa
8	wilayah Perum UNY
9	wilayah Perum Pokoh
10	wilayah Dolo
11	wilayah Kepuh
12	wilayah Keniten
13	wilayah Pucang Anom
14	wilayah Banjeng
15	wilayah Perum Sukoasri
16	wilayah Pesona Maguwo
17	wilayah Kadirojo
18	wilayah Sambisari
19	wilayah Perum Citra Ringin
20	wilayah Purwomartani
21	wilayah Perum Pertamina

Selanjutnya akan disajikan letak geografis dalam peta dari pelanggan-pelanggan di atas (Gambar 3.3). Letak pelanggan tersebar di Kecamatan Ngemplak, Ngaglik, dan Kalasan.



Gambar 3.3 Peta kantor dan rumah pelanggan

Selanjutnya, akan diperoleh tabel jarak kantor agen ke rumah pelanggan dan antar rumah pelanggan oleh salah satu loper koran di Fidy Agency (Tabel 3). Pencarian jarak terdekat antar titik menggunakan aplikasi Navitel.

Tabel 3. Jarak Kantor Agen ke Rumah Pelanggan dan Antar Rumah Pelanggan (dalam satuan Km)

	1	2	3	4	5	6	7	8	9	10	11
1	0	1,6	1,5	1000	1000	1000	1000	1000	1000	1000	12
2	1,6	0	0,28	1000	1000	1000	8,3	8,8	1000	1000	1000
3	1,5	0,28	0	0,4	1,2	1	1000	1000	1000	1000	13
4	1000	1000	0,4	0	1,3	1,2	1000	1000	1000	1000	13
5	1000	1000	1,2	1,3	0	0,62	1000	1000	1000	1000	14
6	1000	1000	1	1,2	0,62	0	1000	1000	1000	1000	14
7	1000	8,3	1000	1000	1000	1000	0	1,9	0,56	0,44	0,74
8	1000	8,8	1000	1000	1000	1000	1,9	0	2,4	2,3	2,6
9	1000	1000	1000	1000	1000	1000	0,56	2,4	0	0,2	0,5
10	1000	1000	1000	1000	1000	1000	0,44	2,3	0,2	0	0,4
11	12	1000	13	13	14	14	0,74	2,6	0,5	0,4	0
12	10	11	11	11	4,3	3,7	1000	1000	1000	2,6	2,3
13	13	13	13	14	14	14	1000	0,84	1000	1000	0,54
14	1000	1000	1000	1000	1000	1000	1000	0,92	1000	1000	1
15	1000	1000	1000	1000	1000	1000	1000	1,3	1000	1000	1
16	10	10	10	11	12	11	1000	1000	1000	1000	4
17	12	12	12	12	13	13	1000	1000	1000	1000	3
18	16	16	16	17	17	17	1000	5,4	1000	1000	4,3
19	1000	15	13	15	16	16	4,4	3,5	4,9	4,8	5,1
20	1000	14	13	14	15	14	1000	4,6	1000	1000	4,3
21	1000	10	1000	1000	1000	1000	3,3	2,3	3,7	3,6	3,9

	12	13	14	15	16	17	18	19	20	21
1	10	13	1000	1000	10	12	16	1000	1000	1000
2	11	13	1000	1000	10	12	16	15	14	10
3	11	13	1000	1000	10	12	16	13	13	1000
4	11	14	1000	1000	11	12	17	15	14	1000
5	4,3	14	1000	1000	12	13	17	16	15	1000
6	3,7	14	1000	1000	11	13	17	16	14	1000
7	1000	1000	1000	1000	1000	1000	1000	4,4	1000	3,3
8	1000	0,84	0,92	1,3	1000	1000	5,4	3,5	4,6	2,3
9	1000	1000	1000	1000	1000	1000	1000	4,9	1000	3,7
10	2,6	1000	1000	1000	1000	1000	1000	4,8	1000	3,6
11	2,3	0,54	1	1	4	3	4,3	5,1	4,3	3,9
12	0	2,7	3,2	3,1	4,2	3,1	4,5	5,8	4,5	6
13	2,7	0	0,5	0,48	4,4	3,4	4,7	1000	4,7	1000
14	3,2	0,5	0	0,44	4,9	3,9	5,2	4,3	5,2	3,2
15	3,1	0,48	0,44	0	4,9	3,8	5,2	3,6	5,2	1000
16	4,2	4,4	4,9	4,9	0	2,2	3,5	4,8	3,5	5
17	3,1	3,4	3,9	3,8	2,2	0	1,4	2,7	1,4	3
18	4,5	4,7	5,2	5,2	3,5	1,4	0	2,8	1,5	3,1
19	5,8	1000	4,3	3,6	4,8	2,7	2,8	0	2,1	1,1
20	4,5	4,7	5,2	5,2	3,5	1,4	1,5	2,1	0	2,3
21	6	1000	3,2	1000	5	3	3,1	1,1	2,3	0

Jarak antara titik yang sama selalu nol. Syarat TSP yaitu setiap titik harus dikunjungi satu kali, maka jika ada jalur terdekat dari 2 titik yang harus melewati titik lain maka jalur tersebut dianggap tidak ada. Jalur yang demikian diberi dengan simbol M. Simbol M menyatakan bilangan relatif besar agar nantinya tidak terpilih dalam pencarian rute terpendek. Karena kepentingan komputasi dengan Matlab, nilai M akan diganti dengan bilangan besar yaitu 1000.

Data jarak pada tabel 3 akan dibawa ke dalam pemodelan TSP. Pemodelan permasalahan TSP merujuk pada rumus 2.3 dengan n sebanyak 21. Kendala permasalahan TSP di Fidi Agency merujuk pada rumus 2.4 dan 2.5. Jarak antar pelanggan dalam tabel 3 merepresentasikan *c_{ij}*. Selanjutnya pencarian rute terpendek akan diselesaikan menggunakan algoritma genetika dengan bantuan *software* Matlab.

B. Penyelesaian Masalah TSP di Fidy Agency Menggunakan Algoritma Genetika

Salah satu metode heuristik untuk menyelesaikan masalah optimasi adalah algoritma genetika. Algoritma genetika menggunakan teknik optimasi yang didasarkan pada genetika alami. Untuk menghasilkan suatu solusi optimal, algoritma genetika melakukan proses pencarian di antara sejumlah alternatif titik optimal berdasarkan fungsi probabilistik. Algoritma genetika merupakan salah satu metode untuk menyelesaikan masalah *travelling salesman problem*.

Urutan langkah-langkah yang digunakan untuk menyelesaikan masalah TSP menggunakan algoritma genetika adalah :

1. Mendefinisikan Gen

Gen dalam hal ini merupakan representasi dari kantor agen yang merupakan tempat awal pendistribusian dan rumah pelanggan yang merupakan tempat yang harus dikunjungi oleh seorang loper koran, dengan kata lain gen adalah titik suatu graf. Representasi gen adalah sebagai berikut :

Gen 1 =	kantor Fidy Agency	= Lokasi 1
Gen 2 =	wilayah Jl Kaliurang km.9	= Lokasi 2
Gen 3 =	wilayah Gandok	= Lokasi 3
Gen 4 =	wilayah Perum Fortuna	= Lokasi 4
Gen 5 =	wilayah Palgading 1	= Lokasi 5
Gen 6 =	wilayah Palgading 2	= Lokasi 6
Gen 7 =	wilayah klinik Assyifa	= Lokasi 7
Gen 8 =	wilayah Perum UNY	= Lokasi 8
Gen 9 =	wilayah Perum Pokoh	= Lokasi 9

Gen 10 =	wilayah Dolo	= Lokasi 10
Gen 11 =	wilayah Kepuh	= Lokasi 11
Gen 12 =	wilayah Keniten	= Lokasi 12
Gen 13 =	wilayah Pucang Anom	= Lokasi 13
Gen 14 =	wilayah Banjeng	= Lokasi 14
Gen 15 =	wilayah Perum Sukoasri	= Lokasi 15
Gen 16 =	wilayah Pesona Maguwo	= Lokasi 16
Gen 17 =	wilayah Kadirojo	= Lokasi 17
Gen 18 =	wilayah Sambisari	= Lokasi 18
Gen 19 =	wilayah Perum Citra Ringin	= Lokasi 19
Gen 20 =	wilayah Purwomartani	= Lokasi 20
Gen 21 =	wilayah Perum Pertamina	= Lokasi 21

Dalam satu kromosom terdapat 21 gen. Dalam kromosom itu berisi gen dari 1 sampai 21 yang membentuk suatu rute perjalanan yang berawal dan berakhir pada titik yang sama.

2. Mendefinisikan Individu

Individu dalam hal ini merupakan satu kali rute perjalanan yang melewati kantor agen dan semua rumah pelanggan dan kembali ke tempat semula. Representasi individu dalam algoritma genetika adalah sebuah kromosom yang berisi susunan gen-gen secara acak dari 1 sampai 21.

Contoh 3.1:

Individu 1 = 1 5 9 8 10 12 11 13 16 19 18 21 2
3 4 7 15 14 17 6 20

Individu 2 = 2 3 4 6 8 10 13 9 12 16 17 20 21

	18	11	19	5	14	15	1	7					
Individu 3 =	5	6	8	3	1	4	9	11	14	16	19	20	12
	7	2	10	17	15	13	18	21					

Banyaknya individu atau rute yang mungkin dilalui oleh seorang loper adalah dengan menggunakan rumus 2.2. Semua rute yang mungkin dilalui dari kantor agen ke semua pelanggan dan kembali ke kantor agen yaitu sebanyak:

$$s = \frac{(n-1)!}{2}$$

$$s = \frac{(21-1)!}{2} = 1.216.451.004.088.320.000$$

Untuk mencari rute terpendek, harus diketahui jarak antara kantor agen dengan masing-masing rumah pelanggan, begitu juga jarak antar rumah pelanggan, yang selanjutnya akan didapatkan jarak pada satu rute perjalanan. Jarak antar kantor agen ke rumah pelanggan dan antar rumah pelanggan dapat dilihat pada Tabel 3.

3. Membangkitkan Populasi Awal

Selanjutnya pencarian rute terpendek pendistribusian koran di Fidi Agency adalah membangkitkan populasi awal. Dalam membangkitkan populasi awal digunakan teknik berupa *random generator*. Inti dari cara ini adalah melibatkan pembangkitan bilangan random untuk nilai setiap gen sesuai dengan representasi kromosom yang digunakan. Sebagai contoh, sebuah kromosom untuk 9 kota bisa direpresentasikan:

[0.23 0.83 0.45 0.74 0.87 0.11 0.56 0.69 0.78]

Posisi i dalam list menunjukkan kota i . Nilai acak dalam posisi i menentukan urutan didatanginya kota i dalam lintasan TSP. Dengan kunci-kunci random di atas, kita dapat menentukan bahwa nilai 0.11 adalah yang paling kecil, sehingga kota ke- i menempati urutan kedua dst. Sehingga dengan demikian, dari kunci-kunci random di atas kita dapat menentukan lintasan:

6 – 1 – 3 – 7 – 8 – 4 – 9 – 2 – 5

Dengan bantuan *software* Matlab, diambil beberapa rute secara acak. (prosedur dapat dilihat di lampiran 3). Hasil pengambilan secara acak rute perjalanan yang membentuk populasi pada generasi awal adalah sebagai berikut:

Individu 1 = 13 8 9 10 11 16 5 6 12 4 3 1 2 7 21
20 18 17 19 15 14

Individu 2 = 18 8 9 10 11 16 5 6 12 4 3 14 2 7 21
20 17 13 1 19 15

Individu 3 = 14 13 1 2 7 21 20 18 17 19 15 8 10 11
16 9 5 6 12 4 3

Individu 4 = 18 17 19 15 14 16 5 13 8 9 10 11 6 12
4 3 1 2 7 21 20

Individu 5 = 18 17 19 15 14 16 5 13 8 9 10 11 6 12
4 3 1 2 7 21 20

Individu 6 = 12 4 3 1 2 7 21 20 18 19 15 14 17 13
8 9 10 11 16 5 6

Individu 7 = 19 15 14 9 10 11 16 6 5 12 4 13 8 3
1 2 7 17 21 18 20

Individu 8 = 13 1 2 7 21 18 8 9 10 11 16 5 6 17 4
3 12 19 15 14 20

Individu 9 = 6 12 4 3 18 21 19 15 14 20 13 1 2 7
17 8 9 10 11 16 5

Individu 10 = 12 4 3 1 2 7 21 20 18 19 15 14 17 13
8 9 10 11 16 5 6

Individu 11 = 6 12 4 3 10 17 19 15 14 20 13 1 2 7
21 8 9 18 11 16 5

Individu 12 = 9 10 11 16 5 15 14 6 12 4 3 1 2 7 21
20 18 17 19 13 8

Individu 13 = 9 10 11 16 5 20 14 6 12 4 3 1 2 7 21
15 18 17 19 13 8

Individu 14 = 5 6 12 4 3 7 21 20 18 19 15 14 17 13
8 1 2 9 10 11 16

Individu 15 = 14 17 13 5 6 7 21 19 15 18 8 9 10 11
16 12 4 3 1 2 20

Pembangkitan populasi awal dalam skripsi ini menghasilkan ukuran populasi sebanyak 15 individu. Individu-individu tersebut selanjutnya akan dihitung nilai masing-masing *fitness*nya.

2. Menentukan Nilai *Fitness*

Setelah pembangkitan populasi awal dilakukan, langkah selanjutnya adalah menentukan nilai *fitness* dari masing-masing individu. Setiap individu dihitung jarak totalnya. Kemudian dihitung nilai *fitness* dengan menentukan inversi total jarak dari rute yang didapatkan. Cara melakukan inversi ditentukan dengan rumus (2.3) pada bab sebelumnya. Dengan bantuan *software* Matlab,

ditentukan nilai *fitness* dari individu (prosedurnya dan perhitungannya terdapat di lampiran 4). Nilai *fitness* yang didapatkan sebagai berikut.

Nilai fitness individu 1 = 0.0159

Nilai fitness individu 2 = 0.0014

Nilai fitness individu 3 = 0.0048

Nilai fitness individu 4 = 0.0115

Nilai fitness individu 5 = 0.0126

Nilai fitness individu 6 = 0.0137

Nilai fitness individu 7 = 0.0037

Nilai fitness individu 8 = 0.0104

Nilai fitness individu 9 = 0.0081

Nilai fitness individu 10 = 0.0148

Nilai fitness individu 11 = 0.0003

Nilai fitness individu 12 = 0.0059

Nilai fitness individu 13 = 0.0026

Nilai fitness individu 14 = 0.0070

Nilai fitness individu 15 = 0.0093

Setelah dihitung nilai *fitness* dari setiap individu, maka didapatkan nilai *fitness* terbaik dari populasi di atas yaitu pada individu ke-1 dengan nilai *fitness* sebesar 0,0159. Individu dengan nilai *fitness* terbaik dari populasi generasi ini akan dipertahankan dan dibawa ke generasi selanjutnya. Langkah selanjutnya adalah melakukan seleksi untuk menentukan individu sebagai induk.

3. Seleksi Ranking

Fungsi seleksi ranking ini adalah memilih secara acak individu dari populasi untuk dijadikan sebagai induk. Induk tersebut akan di lakukan proses pindah silang dengan individu lain yang terpilih. Metode ini menirukan permainan roda dimana masing-masing kromosom menempati lingkaran pada roda sesuai dengan proporsi ranking nilai *fitness*nya.

Dengan bantuan *software* Matlab didapatkan induk-induk yang terpilih secara acak (prosedur seleksi ranking terdapat di lampiran 3, dan proses perhitungannya terdapat di lampiran 4). Berikut hasil individu yang terpilih sebagai induk.

1) Induk 1 = Individu 12 = 9 10 11 16 5 15 14 6 12 4 3 1
2 7 21 20 18 17 19 13 8

Induk 2 = Individu 3 = 14 13 1 2 7 21 20 18 17 19 15
8 10 11 16 9 5 6 12 4 3

2) Induk 1 = Individu 1 = 13 8 9 10 11 16 5 6 12 4 3 1
2 7 21 20 18 17 19 15 14

Induk 2 = Individu 3 = 14 13 1 2 7 21 20 18 17 19 15
8 10 11 16 9 5 6 12 4 3

3) Induk 1 = Individu 6 = 12 4 3 1 2 7 21 20 18 19 15
14 17 13 8 9 10 11 16 5 6

Induk 2 = Individu 2 = 18 8 9 10 11 16 5 6 12 4 3
14 2 7 21 20 17 13 1 19 15

4) Induk 1 = Individu 1 = 13 8 9 10 11 16 5 6 12 4 3 1
2 7 21 20 18 17 19 15 14

Induk 2 = Individu 4 = 18 17 19 15 14 16 5 13 8 9 10
11 6 12 4 3 1 2 7 21 20

5) Induk 1 = Individu 12 = 9 10 11 16 5 15 14 6 12 4 3 1
2 7 21 20 18 17 19 13 8

Induk 2 = Individu 4 = 18 17 19 15 14 16 5 13 8 9 10
11 6 12 4 3 1 2 7 21 20

6) Induk 1 = Individu 10 = 12 4 3 1 2 7 21 20 18 19 15
14 17 13 8 9 10 11 16 5 6

Induk 2 = Individu 10 = 12 4 3 1 2 7 21 20 18 19 15
14 17 13 8 9 10 11 16 5 6

7) Induk 1 = Individu 12 = 9 10 11 16 5 15 14 6 12 4 3 1
2 7 21 20 18 17 19 13 8

Induk 2 = Individu 1 = 13 8 9 10 11 16 5 6 12 4 3 1
2 7 21 20 18 17 19 15 14

Individu-individu di atas terpilih sebagai induk dengan melakukan proses seleksi ranking sebanyak 7 kali. Seleksi ranking sebanyak 7 kali akan menghasilkan 14 induk. Selanjutnya 14 induk akan diseleksi dan didapatkan 15 populasi baru dengan 1 populasi tambahan yaitu *copy* dari individu dengan nilai *fitness* terbaik pada generasi sebelumnya. Induk pertama dan induk kedua selanjutnya akan dilakukan pindah silang guna mendapatkan anak atau keturunan baru.

4. Pindah Silang (*Cross Over*)

Setelah terpilih induk-induk dari proses seleksi ranking, selanjutnya induk-induk tersebut akan dilakukan proses pindah silang. Pindah silang menghasilkan individu baru hasil dari 2 induk yang disebut anak. Pindah silang ini diimplementasikan dengan skema *order cross over*.

Setiap induk menghasilkan 1 anak agar proses seleksi pada generasi selanjutnya mendapatkan jumlah populasi yang sama. Dengan bantuan *software* Matlab didapatkan keturunan (prosedur pindah silang terdapat di lampiran 3). Berikut hasil keturunan yang diperoleh.

1) Anak 1 = 4 3 1 2 7 21 20 18 17 19 13 8 10 11
16 9 5 6 15 14 12

Anak 2 = 19 15 8 10 11 16 9 5 6 12 4 3 2 7 21
20 18 17 14 13 1

2) Anak 1 = 3 1 2 7 21 20 18 17 19 14 15 8 10 11
16 9 5 6 13 12 4

Anak 2 = 15 8 10 11 16 9 5 6 12 4 3 1 2 7 21
20 18 17 14 13 19

3) Anak 1 = 13 8 9 10 11 16 5 6 12 4 3 1 2 7 21
20 18 19 15 14 17

Anak 2 = 2 7 21 17 13 1 15 20 18 19 8 9 10 11
16 5 6 12 4 3 14

4) Anak 1 = 11 16 5 6 2 7 21 20 18 17 19 15 14 12
4 3 1 13 8 9 10

Anak 2 = 16 5 13 8 9 10 11 6 12 4 3 1 2 7 21
20 18 17 19 15 14

5) Anak 1 = 5 15 14 20 18 17 19 13 8 9 10 11 6 12
4 3 1 2 7 21 16

Anak 2 = 14 16 5 8 9 10 11 6 12 4 3 1 2 7 21
20 18 17 19 13 15

6) Anak 1 = 9 10 11 16 5 6 21 20 18 19 15 14 12 4
3 1 2 7 17 13 8

Anak 2 = 9 10 11 16 5 6 21 20 18 19 15 14 12 4
3 1 2 7 17 13 8

7) Anak 1 = 13 8 9 10 11 16 5 6 12 4 3 1 2 7 21
20 18 17 19 15 14

Anak 2 = 9 10 11 16 5 15 14 6 12 4 3 1 2 7 21
20 18 17 19 13 8

Anak yang dihasilkan dari proses pindah silang di atas, selanjutnya akan dilakukan proses mutasi. Proses mutasi dilakukan pada anak hasil pindah silang dengan tujuan untuk memperoleh individu baru sebagai kandidat solusi pada generasi mendatang dengan *fitness* yang lebih baik, dan lama-kelamaan menuju solusi optimum yang diinginkan.

5. Mutasi

Setelah dilakukan operator pindah silang didapatkan keturunan-keturunan yang selanjutnya akan di proses mutasi. Skema mutasi yang digunakan adalah

swapping mutation. Untuk semua gen yang ada, jika bilangan random yang dibangkitkan [0,1) kurang dari probabilitas mutasi yang ditentukan, maka nilai gen tersebut akan ditukarkan dengan nilai gen yang lain yang dipilih secara acak.

Dengan bantuan *software* Matlab didapatkan individu hasil mutasi. Prosedur mutasi terdapat di lampiran 3. Berikut hasil mutasi yang diperoleh.

1) Individu baru anak 1 = 4 3 1 2 7 21 20 18 17 19 13 8
 10 11 16 9 5 6 15 14 12

Individu baru anak 2 = 19 15 8 10 11 16 9 5 6 12 4
 3 2 7 21 20 18 17 14 13 1

2) Individu baru anak 1 = 3 1 2 7 21 **17** 18 **20** 19 14 15
 8 10 11 16 9 5 6 13 12 4

Individu baru anak 2 = 15 8 10 11 16 9 5 6 12 4 3 1
 2 7 21 20 18 17 14 13 19

3) Individu baru anak 1 = 13 8 9 10 11 16 5 6 12 4 3 1
 2 7 21 20 18 19 15 14 17

Individu baru anak 2 = 2 7 21 17 13 1 15 20 18 19 8
 9 10 11 16 5 6 12 4 3 14

4) Individu baru anak 1 = 11 16 5 6 2 7 21 20 18 17 19
 15 14 12 4 3 1 13 8 9 10

Individu baru anak 2 = 16 5 13 8 9 10 11 6 12 4 3 1
 2 7 21 20 18 17 19 15 14

5) Individu baru anak 1 = 5 15 14 20 18 17 19 13 8 9 10
 11 6 12 4 3 1 2 7 21 16

Individu baru anak 2 = 14 16 5 8 9 10 11 6 12 4 3 1
 2 7 21 20 18 17 19 13 15

6) Individu baru anak 1 = 9 10 11 16 5 6 21 20 18 19 15
 14 12 4 3 1 2 7 17 13 8

Individu baru anak 2 = 9 10 11 16 5 6 21 20 18 19 15
 14 12 4 3 1 2 7 17 13 8

7) Individu baru anak 1 = 13 8 9 10 11 16 5 6 12 4 3 1
 2 7 21 20 18 17 19 15 14

Individu baru anak 2 = 9 10 11 16 5 15 14 6 12 4 3 1
 2 7 21 20 18 17 19 13 8

Pada individu 2, individu baru anak 1 adalah hasil *swapping mutation* dari kromosom ke-6 ditukar dengan kromosom ke-8. Individu baru yang dihasilkan akan digunakan untuk membentuk populasi baru pada generasi kedua.

6. Pembentukan Populasi Baru

Setelah langkah-langkah di atas dilakukan, maka dibentuk populasi selanjutnya di generasi kedua. Individu terbaik dengan nilai *fitness* tertinggi pada populasi awal dibawa ke populasi selanjutnya, proses ini dinamakan sebagai *Elitism*. Prosedur pembentukan populasi selanjutnya terdapat dalam lampiran 3 dengan bantuan *software* Matlab. Berikut merupakan hasil populasi baru di generasi selanjutnya.

Individu 1 = 13 8 9 10 11 16 5 6 12 4 3 1 2 7 21
 20 18 17 19 15 14

Individu 2 = 4 3 1 2 7 21 20 18 17 19 13 8 10 11
16 9 5 6 15 14 12

Individu 3 = 19 15 8 10 11 16 9 5 6 12 4 3 2 7 21
20 18 17 14 13 1

Individu 4 = 3 1 2 7 21 20 18 17 19 14 15 8 10 11
16 9 5 6 13 12 4

Individu 5 = 15 8 10 11 16 9 5 2 12 4 3 1 6 7 21
20 18 17 14 13 19

Individu 6 = 13 8 9 10 11 16 5 6 12 4 3 1 2 7 21
20 18 19 15 14 17

Individu 7 = 2 7 21 17 13 1 15 20 18 19 8 9 10 11
16 5 6 12 4 3 14

Individu 8 = 11 16 5 6 2 7 21 20 18 17 19 15 14 12
4 3 1 13 8 9 10

Individu 9 = 16 5 13 8 9 10 11 6 12 4 3 1 2 7 21
20 18 17 19 15 14

Individu 10 = 5 15 14 20 18 17 19 13 8 9 10 11 6 12
4 3 1 2 7 21 16

Individu 11 = 14 16 5 8 9 10 11 6 12 4 3 1 2 7 21
20 18 17 19 13 15

Individu 12 = 9 10 11 16 5 6 21 20 18 19 15 14 12 4
3 1 2 7 17 13 8

Individu 13 = 9 10 11 16 5 6 21 20 18 19 15 14 12 4
3 1 2 7 17 13 8

Individu 14 = 13 8 9 10 11 16 5 6 12 4 3 1 2 7 21
20 18 17 19 15 14

Individu 15 = 9 10 11 16 5 15 14 6 12 4 7 1 2 3 21
20 18 17 19 13 8

Prosedur-prosedur penentuan nilai *fitness*, seleksi, pindah silang dan mutasi dilakukan pada generasi kedua untuk menentukan populasi di generasi selanjutnya. Iterasi dilakukan hingga mendapatkan nilai *fitness* yang optimum dan *konvergen* di generasi tertentu. Hasil *fitness* paling optimum dan konvergen pada generasi ke-134. Populasi baru dari generasi ke-134 sebagai berikut.

Individu 1 = 1 2 7 21 20 18 17 19 15 14 13 8 9 10
11 12 16 5 6 4 3

Individu 2 = 17 11 3 1 14 16 2 7 21 20 18 19 12 13
8 9 10 5 15 6 4

Individu 3 = 4 12 13 8 9 10 11 5 15 14 6 3 1 16 2
7 21 20 18 17 19

Individu 4 = 21 20 18 17 19 15 5 6 4 12 13 8 9 10
11 16 3 1 14 2 7

Individu 5 = 8 9 10 11 16 6 4 3 1 20 18 17 19 15
12 5 2 7 21 14 13

Individu 6 = 11 16 17 19 15 14 5 6 12 4 3 1 2 7
21 20 13 18 8 9 10

Individu 7 = 20 18 17 19 15 14 5 6 12 4 3 13 8 9
10 11 16 1 2 7 21

Individu 8 = 7 21 17 19 15 14 5 6 12 4 3 13 8 9 10
20 18 1 2 11 16

Individu 9 = 7 21 19 12 3 1 2 17 13 14 15 8 9 10
11 20 18 5 6 4 16

Individu10 = 11 14 18 17 19 15 5 6 9 4 3 1 2 7 21
16 20 12 13 8 10

Individu 11 = 11 14 3 1 2 7 21 16 9 20 18 17 19 15
12 5 6 4 13 8 10

Individu 12 = 20 15 18 8 9 10 11 5 16 17 19 13 14 6
12 4 3 1 2 7 21

Individu 13 = 20 18 17 19 15 14 5 6 12 13 8 9 10 11
4 3 1 16 2 7 21

Individu 14 = 1 2 7 21 20 18 17 19 15 14 13 8 9 10
11 12 16 5 6 4 3

Individu 15 = 2 7 21 20 18 17 19 15 14 12 13 8 9 10
11 16 5 6 4 3 1

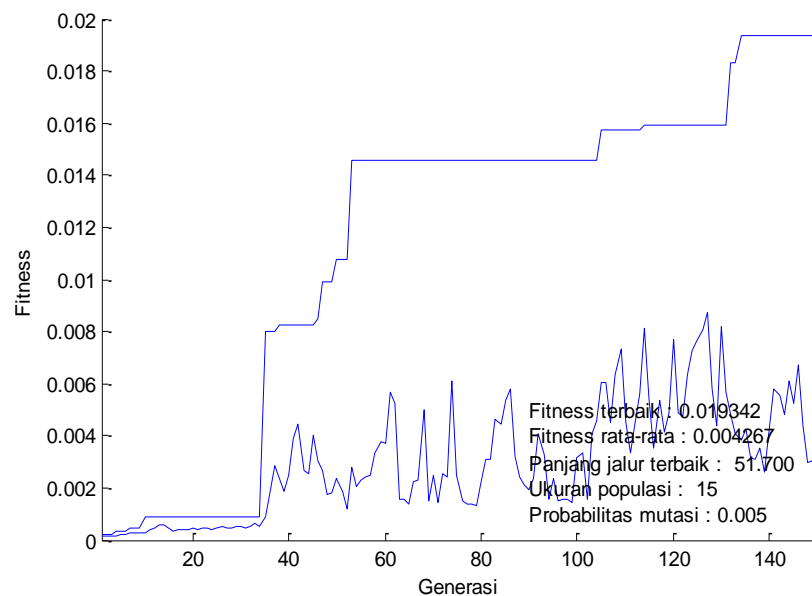
Individu 1 merupakan solusi yang didapatkan karena memiliki nilai *fitness* tertinggi yaitu 0,0193. Algoritma genetika bersifat *random generator*, sehingga setiap melakukan proses seleksi maka akan selalu menghasilkan solusi yang berbeda. Diperlukan beberapa kali percobaan dalam mengaplikasikan

algoritma genetika menggunakan *software* Matlab agar didapatkan solusi yang optimum, yaitu dengan mencoba beberapa nilai ukuran populasi dan jumlah generasi. Berikut tabel percobaan sehingga didapatkan rute terpendek solusi pendistribusian koran di Fidi Agency yaitu 51,70 km.

Tabel 4. Hasil Percobaan dengan *software* Matlab

Percobaan ke-	Ukuran Populasi	Jumlah Generasi	Konvergen Generasi ke-	Nilai Fitness	Panjang Rute (km)
1	15	50	45	0,0137	72,48
2	15	50	35	0,0131	76,20
3	15	50	38	0,0136	73,22
4	15	100	69	0,0161	62,00
5	15	100	88	0,0151	65,94
6	15	150	135	0,0193	51,70
7	15	150	125	0,0184	54,14
8	20	50	15	0,0144	69,38
9	20	50	25	0,0138	72,00
10	20	50	44	0,0136	73,30
11	20	100	60	0,0153	64,96
12	20	100	70	0,0159	62,72
13	20	150	110	0,0171	58,20
14	20	150	90	0,0159	62,84
15	25	50	38	0,0134	74,52
16	25	50	46	0,0120	82,92
17	25	50	40	0,0117	85,14
18	25	100	68	0,0130	76,68
19	25	100	82	0,0142	70,24
20	25	150	107	0,0163	61,00

Dari tabel didapatkan nilai *fitness* paling optimum adalah 0,0193. Berikut grafik pergerakan nilai *fitness* sampai kovergen ke nilai tertentu:



Gambar 3.4 Grafik pergerakan nilai *fitness*

Kurva yang berada di atas merupakan nilai *fitness* pada generasi ke-n. Kurva yang berada di bawah merupakan nilai *fitness* rata-rata dari n generasi. Jadi telah diketahui solusi dari permasalahan Fidi Agency untuk pendistribusian koran. Rute terpendek yang dapat ditempuh adalah berawal dari Kantor Fidi Agency – Jl Kaliurang km.9 – Klinik Assyifa – Perum Pertamina – Purwomartani – Sambisari – Kadirojo – Perum Citra Ringin – Perum Sukoasri – Banjeng – Pucang Anom – Perum UNY – Perum Pokoh – Dolo – Kepuh – Keniten – Pesona Maguwo – Palgading 1 – Palgading 2 – Perum Fortuna – Gandok – kantor Fidy Agency. Jarak rute tempuhnya adalah 51,70 km.

Realita yang terjadi di lapangan adalah looper tersebut mendistribusikan koran dengan rute tempuh 72 km, kecepatan motor adalah 50 km/jam dan menyelesaikan dengan waktu 2,5 jam. Jika dibandingkan dengan pencarian rute

menggunakan algoritma genetika, maka waktu yang diperlukan menjadi 1,034 jam atau 62 menit dengan kecepatan motor yang sama. Jika setiap pelanggan dilayani dalam 30 detik, maka waktu ditambah dengan 25 menit. Maka waktu yang diperlukan menjadi 87 menit atau 1,45 jam.

BAB IV

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian di atas mengenai penentuan rute terpendek pendistribusian koran di Fidy Agency dengan menggunakan algoritma genetika, dapat disimpulkan sebagai berikut.

1. Model matematika pendistribusian koran di Fidy Agency oleh salah satu loper adalah sebagai berikut.

Didefinisikan $G(N, A)$ adalah graf tak berarah, dengan $N = \{v_1, v_2, v_3, \dots, v_{21}\}$ merupakan himpunan titik yang merepresentasikan 21 pelanggan, dan v_1 merepresentasikan kantor, sedangkan $A = \{(v_i, v_j) \mid v_i, v_j \in V, i \neq j\}$ adalah himpunan sisi yang menghubungkan antar titik, yang merepresentasikan ruas jalan penghubung antar titik. Jarak perjalanan dari titik i ke titik j direpresentasikan oleh c_{ij} . Selanjutnya didefinisikan variabel keputusan x_{ij} yang merepresentasikan ada tidaknya perjalanan dari titik i ke j dalam suatu rute sebagai berikut :

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

Jika Z merupakan fungsi tujuan TSP, maka fungsi tujuan Z dirumuskan dengan meminimumkan:

$$Z = \sum_{i=1}^{21} \sum_{j=1}^{21} c_{ij} x_{ij}.$$

dengan kendala,

$$\sum_{i=1}^{21} x_{i1} = 1.$$

⋮

$$\sum_{i=1}^{21} x_{i,21} = 1.$$

dan

$$\sum_{j=1}^{21} x_{1j} = 1.$$

⋮

$$\sum_{j=1}^{21} x_{21,j} = 1.$$

2. Rute terpendek pendistribusian koran pada Fidy Agency telah didapatkan dengan mengikuti langkah-langkah di atas. Rute terpendek pendistribusian koran di Fidi Agency yaitu dari dari kantor Fidi Agency – Jl Kaliurang km.9 – Klinik Assyifa – Perum Pertamina – Purwomartani – Sambisari – Kadirojo – Perum Citra Ringin – Perum Sukoasri – Banjeng – Pucang Anom – Perum UNY – Perum Pokoh – Dolo – Kepuh – Keniten – Pesona Maguwo – Palgading 1 –Palgading 2 – Perum Fortuna – Gandok – Kantor Fidy Agency. Jarak rute yang dilalui adalah 51,70 km.

B. Saran

Dalam penelitian skripsi ini, penulis baru membahas mengenai algoritma genetika menggunakan seleksi ranking untuk optimalisasi jarak atau rute pendistribusian koran, penulis menyarankan agar membahas dengan metode seleksi lain yaitu *rank-based fitness assignment*, seleksi lokal (*local selection*), seleksi dengan pemotongan (*truncation selection*) dan seleksi dengan turnamen (*tournament selection*) untuk menyelesaikan masalah optimasi pendistribusian koran dalam kasus TSP.

DAFTAR PUSTAKA

- Ahmad Basuki. (2003). *Algoritma Genetika*. Surabaya: PENS-ITS.
- Anwar Toni & Yuliani Willi. (2005). Penerapan Algoritma Genetika untuk Travelling Salesmen Problem dengan Menggunakan Order Crossover dan Insertion Mutation. *Jurnal*. UII.
- Asmi Baharudin. (2012). Travelling Salesman Problem menggunakan Algoritma Genetika Berbasis Android. *Jurnal*. ITS.
- Aulia F, & Achmad Z. (2006). *Penerapan Algoritma Genetika pada Pedagang Keliling*. Diakses dari [http://www. Eprint.upnjatim.ac.id/4361/1/file/pdf](http://www.Eprint.upnjatim.ac.id/4361/1/file/pdf) pada tanggal 5 Juni 2014.
- Budi Sukmawan. (2003). *Sekilas Tentang Algoritma Genetika dan Aplikasinya Pada Optimasi Jaringan Pipa Air Bersih*. Diakses pada 8 Juni 2014 dari [www. Bdg.centrin.net.id/~budskman/ga](http://www.Bdg.centrin.net.id/~budskman/ga)
- Csuhendra. (2012). *Metode Seleksi pada Algoritma Genetika Menggunakan Roulette Wheel dan Rank-Based*. Diakses pada 10 Juni 2014 dari <http://csuhendra.wordpress.com/2012/09/06/metode-seleksi-pada-algoritma-genetik-menggunakan-roulette-wheel-dan-rank-based/>
- Davendra, D. (2010). *Traveling Salesman Problem, Theory and Applications*. Croatia : Janeza Trdine.
- Goldberg, David. (1999). *An Introduction to Genetic Algorithms for Scientists and Engineers*. Singapore: Usa-Print.

- Kusrini K. (2008). *Penyelesaian Travelling Salesman Problem dengan Algoritma Genetika*. Diakses dari <http://www.puslit.petra.ac.id/journals/request.php>. pada tanggal 5 Juni 2014.
- Michalewicz, Z. (1996). *Genetic Algorithm and Data Structures*. Evolution Programs, 3rd, revised and extended edition. Springer-Verlag.
- Munir, Rinaldi. (2010). *Matematika Diskrit*. Bandung: Informatika Bandung.
- N, Murniati. (2009). *Penerapan Algoritma Genetika pada DNA Sequencing By Hibbridization*. Depok: Departemen Matematika UI.
- Ratna Wati D. A. (2011). *Sistem Kendali Cerdas*. Yogyakarta: Graha Ilmu.
- Satriyanto. (2009). *Algoritma Genetika*. Diakses pada 18 Juni 2014 dari <http://lectures.eepisits.edu/~kangedi/materi%20kuliah/kecerdasan%20Buatan/bab%207%20Algoritma%20Genetika.pdf>.
- Sri Kusumadewi. (2003). *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- Sugeng Mardiyono. (1996). *Matematika Diskret*. Yogyakarta: IKIP Yogyakarta.
- Suyanto. (2005). *Algoritma Genetika dalam MATLAB*. Yogyakarta: Andi Offset.
- Takes, Frank. (2010). *Applying Monte Carlo Techniques to the Capacitated Vehicle Routing Problem*. Master Thesis: Leiden University.
- W. S. E. Tanjung (2010). *Kajian Algoritma Genetika*. Skripsi. Universitas Sumatra Utara.

LAMPIRAN 1

Tabel 5 Alamat Kantor dan Pelanggan dan Pengelompokan

No	Nama Kantor dan Pelanggan	Alamat	Kelompok
1	Kantor Fidi Agency	Jl. Kaliurang Km 8,5	1
2	Warung Sisha Awan Mbengi	Jl. Kaliurang Km 9,3 Klabanan	2
3	Bpk Ahmad	Jl. Kaliurang Km.9	
4	Bpk Sudrajat	Gandok Tambakan No.5	3
5	Ibu Rossa	Perum Fortuna A-3	4
6	Bpk Ari Budi	Jl. Krapyak Palgading	5
7	Warung Lotek	Jl. Krapyak Palgading	
8	Bpk Suyono	Dusun Palgading	
9	Bpk Kristiawan	Dusun Krapyak	6
10	Bpk Lutfi	Krapyak Rt.05 Rw.55 Wedomartani	
11	Bpk Ahmad M.	Krapyak, Wedomartani	
12	Klinik Assyifa	Rejosari Wedomartani Ngemplak	7
13	Bpk Basuki	Perum UNY	8
14	Bpk Triyoko	Perum UNY	
15	Bpk St.Darmadi	Perum Pokoh Baru Jl. Menur No.7	9
16	Bpk Pujiyanto	Jl.Raflesia No.10 Pokoh Baru,	

		Wedomartani	
17	Ibu Rini	Dolo Rt.2 Wedomartani Ngemplak	10
18	Bpk Krisna W	Dolo No.240 Rt.03 Rw.26	
19	Ibu Sri A	Dolo Rt.2 Rw.26	
20	Bpk Johan	Kepuh Rt 2 Wedomartani	11
21	Bpk Sunu	Kepuh Wedomartani	
22	Bpk Edy J.	Kepuh Wedomartani	
23	Bpk Lutfi Arip	Dusun Nomporejo Wedomartani Rt.4 Rw.27	12
24	Bapak Riyanto	Perum Puri Niten Asri No.41	
25	Bpk Pujiono	Perum Puri Niten Asri No.35	
26	Bpk Siswanto	Pucanganom Rt.8 Rw.27 Wedomartani	13
27	Bpk Ngadirin S	Bokoharjo Banjeng Rt.6 Rw.36	14
28	Bpk Misnar	Perum Soka Asri Permai L-3	15
29	Bpk Sutedjo	Perum Soka Asri Permai L-4	
30	Bpk Toni	Perum Soka Asri Permai M-14	
31	Bpk Soetopo	Perum Soka Asri Permai M-2	
32	Ibu Susono	Perum Soka Asri Permai M-3	
33	Bpk H. Edy P	Perum Soka Asri Permai F-13	
34	Bpk A.Bachtiar	Perum Soka Asri Permai F-15	
35	Bpk Thomas	Perum Soka Asri Permai L-6	

36	Bpk Michael S	Corongan Rt 5/23 Maguwoharjo	16
37	Bpk Arif	Kadirojo 1 Rt.6 Rw.2 No.30 Kalasan	17
38	Bpk Narto	Jl.Candi Sambu Sari Rt.7 Rw.3	18
39	Ibu Yahya	Sambu Sari No.45 Rt.3 Rw.2	
40	Bpk Boedi W	Perum Citra Ringin Mas B-26 Purwomartani	19
41	Bpk Rahmadi W	Perum Citra Ringin Mas C-17	
42	Bpk Eko Agus S	Perum Citra Ringin Mas C-3	
43	Bpk Didit	Perum Purwomartani A-1	20
44	Bpk Hari	Perum Purwomartani A-4	
45	Bpk Dicky	Perum Purwomartani Baru A-8	
46	Bpk Rc. Hidayat	Perum Purwomartani Baru A-27	
47	Bpk Sumardiono	Perum Purwomartani Jl.Bimo No.6	
48	Ibu Wigati	Perum Pertamina S-2	21
49	Bpk Budi	Perum Pertamina F-5	
50	Bpk Sudarso	Perum Pertamina H-2	
51	Bpk Sigit	Perum Pertamina D-5	

Lampiran 2

Hasil Wawancara dengan salah satu loper koran di Fidy Agency

1. Siapa nama Bapak?

Saya Pak Suyono

2. Bagaimana alur perjalanan koran hingga sampai ke tangan pembaca?

Pertama-tama koran didistribusikan dari gudang percetakan ke agen-agen menggunakan mobil box pada pagi-pagi benar, setelah itu koran didistribusikan ke toko-toko dan ke rumah pelanggan melalui seorang loper koran.

2. Berapa jumlah loper yang terdapat di Fidy Agency?

Fidy Agency memiliki 6 orang loper.

3. Berada di daerah mana saja Bapak mengantar koran dan berapa jumlah pelanggannya?

Kalau saya mengantar di daerah Kecamatan Ngaglik, Ngemplak dan Kalasan. Kebanyakan mereka adalah yang tinggal di daerah perumahan. Jumlah pelanggan koran milik saya adalah 50 pelanggan.

4. Dengan apa dan berapa lama waktu yang dibutuhkan untuk mengantar koran?

Saya mengantar koran dengan motor bebek. Saya berangkat dari kantor jam 5 pagi, dan sampai di kantor lagi jam setengah 8 pagi. Kecepatan motor ya paling sekitar 50 km/jam.

5. Bagaimana rute yang ditempuh Bapak untuk mengantarkan koran? Apakah ada aturan resmi dari kantor ataukah hanya semau Bapak saja?

Kalau dari kantor tidak ada aturan untuk rute yang ditempuh. Rute dalam mengantar koran hanya semau saya saja, hanya dengan kira-kira saja. Untuk setiap hari biasanya rata-rata jarak yang saya tempuh adalah 72 km.

5. Apa saja hambatan atau masalah yang ditemui dalam pendistribusian koran ini?

Seringkali para pelanggan banyak yang komplain karena keterlambatan dalam mengantar koran. Mereka ingin koran didistribusikan lebih pagi, Apalagi mereka yang pekerja kantor, biasanya meminta untuk didistribusikan sebelum jam 7 pagi.

Lampiran 3

Prosedur algoritma genetika menggunakan *software* Matlab dalam penyelesaian masalah pendistribusian koran pada Fidi Agency. Prosedur algoritma genetika ini dimodifikasi dari Suyanto (2005: 86).

1. Membangkitkan Populasi Awal

Membangkitkan populasi awal diimplementasikan menggunakan baris-baris perintah pada fungsi inisialisasipopulasi.m berikut ini.

```
function p =InisialisasiPopulasi (up,g)
for b= 1:up,
    [x,y]= sort(rand(1,g));
    p (b,:)=y;
end
```

Perintah `rand (1,g)` menyatakan pembangkitan matriks berukuran 1 x g (jumlah gen) yang berisi bilangan random dalam interval [0,1). Pada perintah `[x,y]=sort(rand(1,g))`, x menyatakan bilangan-bilangan random dalam interval [0,1) hasil pengurutan secara *ascending*. Sedangkan y merupakan indeks dari bilangan-bilangan yang dibangkitkan random. Indeks y merupakan nomor urut pelanggan koran yang dibangkitkan. Iterasi di atas dilakukan sebanyak up (banyaknya individu dalam populasi), sehingga didapatkan populasi awal.

2. Menghitung nilai *Fitness*

Perhitungan nilai *fitness* dari populasi awal yang sudah terbentuk diimplementasikan dalam perintah pada fungsi `tspindividu.m` berikut ini.

```
function f = tspindividu(k,g,jk)

tb = 0;

for b=1:g-1,

    tb= tb + jk(k(b+1),k(b));

end

tb=tb + jk(k(g), k(1));

f = 1/tb;
```

Variabel pada fungsi `tspindividu.m` adalah `k` (individu dari populasi), `g` (jumlah gen) dan `jk` (jarak lokasi rumah pelanggan dan gudang). Sedangkan `tb` merupakan total jarak dari individu (nilai dari individu) sedangkan nilai *fitness* suatu individu dinyatakan dalam $f=1/tb$.

```
function lfr=linearfitnessranking(up,f,maxf,minf)

[s,h] = sort(f);

for rr=1:up,

    Lfr(h(up-rr+1))=maxf-(maxf-minf)*((rr-1)/(up-1));

end
```

3. Seleksi Ranking

Operator seleksi dari populasi awal yang sudah terbentuk diimplementasikan dalam perintah pada fungsi `seleksiranking.m` berikut ini.

```

function p=seleksiranking(up,lf)

jf= sum(lf);

kf = 0;

rn = rand;

b = 1;

while b<=up,

kf = kf + lf(b);

if (kf/jf)>rn,

    p =b;

    break;

end

b=b+1;

end

```

Variabel pada fungsi seleksiranking.m adalah up (banyaknya individu dalam populasi) dan lf (nilai *fitness*). Sedangkan $jf = \text{sum}(lf)$ merupakan jumlah keseluruhan nilai *fitness* dalam populasi. Arti kf adalah nilai *fitness* kumulatif dari individu. Jika kumulatif *fitness* dibagi jumlah *fitness* lebih dari bilangan random yang dibangkitkan maka iterasi berhenti, sehingga didapatkan p (indeks) dari individu yang terpilih.sebagai induk.

4. Operator Pindah Silang (*Order Cross Over*)

Operator pindah silang dari populasi awal yang sudah terbentuk diimplementasikan dalam perintah pada fungsi pindahsilang.m berikut ini.

```

function a= pindahsilang(bpk,ibu,g)

cp1=1 + fix(rand*(g-1));

cp2=1 + fix(rand*(g-1));

while cp2==cp1,

    cp2 = 1+fix(rand*(g-1));

end

if cp1 < cp2,

    cps=cp1;

    cpd=cp2;

else

    cps = cp2;

    cpd = cp1;

end

a(1,cps+1:cpd) = ibu(cps+1:cpd);

a(2,cps+1:cpd) = bpk(cps+1:cpd);

Sbpk = [];

Sibu = [];

for b=1:g,

    if ~ismember(bpk(b),a (1,:)),

        Sbpk = [Sbpk bpk(b)];

    end

    if ~ismember(ibu(b),a (2,:)),

        Sibu = [Sibu ibu (b)];

    end

end

a(1,cpd+1:g)=Sbpk(1:g-cpd);

```

```

a(1,1:cps)= Sbpk(1+g-cpd:length(Sbpk));
a(2,cpd+1:g)=Sibu(1:g-cpd);
a(2,1:cps)=Sibu(1+g-cpd:length(Sibu));

```

Variabel pada fungsi pindahsilang.m adalah bpk (induk ke-1 yang terpilih), ibu (induk ke-2 yang terpilih) dan g (jumlah gen). Mula-mula 2 buah bilangan dibangkitkan secara acak untuk menentukan titik potong kedua induk. Kemudian dua kromosom anak (a) mendapatkan gen-gen dari kromosom bpk dan ibu. Posisi-posisi gen yang masih kosong pada a pertama diisi dengan gen dari bpk yang belum ada pada a pertama, dan a kedua diisi dengan gen dari ibu yang belum ada pada a kedua. Hasil dari fungsi ini berupa kromosom baru yang membawa sifat dari induknya.

5. Mutasi dengan *Swapping Mutation*

Operator pindah silang dari populasi awal yang sudah terbentuk diimplementasikan dalam perintah pada fungsi mutasi.m berikut ini.

```

function mk = mutasi(k,g,pm)
mk = k;

for b=1:g,
    if rand < pm,
        TM2=1+fix(rand*g);
        while TM2==b,
            TM2 = 1 + fix(rand*g);

```

```

        end

        tp = mk(b) ;

        mk(b) = mk(TM2) ;

        mk(TM2) = tp ;

    end

end

```

Variabel pada fungsi mutasi.m di atas adalah k (anak hasil pindah silang), g (jumlah gen) dan pm (probabilitas mutasi). Mula-mula membangkitkan bilangan random dalam interval [0,1). Jika bilangan random yang dibangkitkan kurang dari pm maka ditentukan posisi gen dalam kromosom yang akan ditukar. Kemudian menukar nilai gen yang terpilih dalam kromosom sehingga didapatkan kromosom baru hasil mutasi (mk).

6. Program Utama

Sebagai program utama, fungsi ini memanggil semua fungsi-fungsi di atas. Pada program utama berikut, memasukkan variabel-variabel jk (jarak antar lokasi), g (jumlah gen), up (banyaknya individu dalam populasi), ps (probabilitas pindah silang), pm (probabilitas mutasi). Program di bawah ini juga ditambahkan perintah untuk menampilkan grafik. Pada akhir program variabel jt menyatakan rute terpendek yang didapatkan.

```

clc

clear all

% untuk pendistribusian pada Fidi Agency

```

```

jk = [0      1.6    1.5    1000  1000  1000  1000  1000  1000  1000
      12     10     13     1000  1000  10    12     16     1000  1000
      1000;
      1.6     0      0.28   1000  1000  1000  8.3    8.8    1000  1000
      1000  11     13     1000  1000  10    12     16     15     14
      10;
      1.5    0.28   0       0.4    1.2    1      1000  1000  1000  1000
      13     11     13     1000  1000  10    12     16     13     13
      1000;
      1000  1000  0.4     0       1.3    1.2    1000  1000  1000  1000
      13     11     14     1000  1000  11    12     17     15     14
      1000;
      1000  1000  1.2     1.3     0       0.62  1000  1000  1000  1000
      14     4.3    14     1000  1000  12    13     17     16     15
      1000;
      1000  1000  1       1.2     0.62   0      1000  1000  1000  1000
      14     3.7    14     1000  1000  11    13     17     16     14
      1000;
      1000  8.3    1000  1000  1000  1000  0       1.9    0.56  0.44
      0.74  1000  1000  1000  1000  1000  1000  1000  4.4    1000
      3.3;
      1000  8.8    1000  1000  1000  1000  1.9    0       2.4    2.3
      2.6   1000  0.84  0.92  1.3    1000  1000  5.4    3.5    4.6
      2.3;
      1000  1000  1000  1000  1000  1000  0.56  2.4    0       0.2
      0.5   1000  1000  1000  1000  1000  1000  1000  4.9    1000
      3.7;
      1000  1000  1000  1000  1000  1000  0.44  2.3    0.2    0
      0.4   2.6   1000  1000  1000  1000  1000  1000  4.8    1000
      3.6;
      12     1000  13     13     14     14     0.74  2.6    0.5    0.4
      0      2.3    0.54  1      1      4      3      4.3    5.1    4.3
      3.9;
      10     11     11     11     4.3    3.7    1000  1000  1000  2.6
      2.3    0      2.7    3.2    3.1    4.2    3.1    4.5    5.8    4.5
      6;
      13     13     13     14     14     14     1000  0.84  1000  1000
      0.54  2.7    0      0.5    0.48  4.4    3.4    4.7    1000  4.7
      1000;
      1000  1000  1000  1000  1000  1000  1000  0.92  1000  1000
      1      3.2    0.5    0      0.44  4.9    3.9    5.2    4.3    5.2
      3.2;
      1000  1000  1000  1000  1000  1000  1000  1.3    1000  1000
      1      3.1    0.48  0.44  0      4.9    3.8    5.2    3.6    5.2
      1000;
      10     10     10     11     12     11     1000  1000  1000  1000
      4      4.2    4.4    4.9    4.9    0      2.2    3.5    4.8    3.5
      5;
      12     12     12     12     13     13     1000  1000  1000  1000
      3      3.1    3.4    3.9    3.8    2.2    0      1.4    2.7    1.4
      3;
      16     16     16     17     17     17     1000  5.4    1000  1000
      4.3    4.5    4.7    5.2    5.2    3.5    1.4    0      2.8    1.5
      3.1;

```

```

1000 15 13 15 16 16 4.4 3.5 4.9 4.8
5.1 5.8 1000 4.3 3.6 4.8 2.7 2.8 0 2.1
1.1;
1000 14 13 14 15 14 1000 4.6 1000 1000
4.3 4.5 4.7 5.2 5.2 3.5 1.4 1.5 2.1 0
2.3;
1000 10 1000 1000 1000 1000 3.3 2.3 3.7 3.6
3.9 6 1000 3.2 1000 5 3 3.1 1.1 2.3
0]

g = length(jk(:,1))

up = 15

ps = 0.8

pm = 0.005

maxG = 150

pjh = 30

fth = 1/pjh

bg = fth

%Inisialisasi grafis

hfig =figure;

hold on

set(hfig, 'position', [50,50,600,400]);

set(hfig, 'DoubleBuffer', 'on');

axis([1 maxG 0 bg]);

hbestplot1 = plot(1:maxG,zeros(1,maxG));

hbestplot2 = plot(1:maxG,zeros(1,maxG));

htext1 = text(0.6*maxG, 0.25*bg, sprintf('Fitness terbaik: %7.6f',
0.0));

htext2 = text(0.6*maxG, 0.20*bg, sprintf('Fitness rata-rata:
%7.6f', 0.0));

htext3 = text(0.6*maxG, 0.15*bg, sprintf('Panjang Jalur Terbaik:
%7.3f', 0.0));

htext4 = text(0.6*maxG, 0.10*bg, sprintf('Ukuran Populasi: %3.0f',
0.0));

```

```

htext5 = text(0.6*maxG, 0.05*bg, sprintf('Probabilitas: %4.3f',
0.0));

xlabel('Generasi');

ylabel('Fitness');

hold off

drawnow;

%Inisialisasi Populasi

p = inisialisasipopulasi(up,g)

for generasi = 1:maxG,

    maxf = tspindividu(p(1,:),g,jk);

    minf=maxf;

    iit = 1;

    for b=1:up,

        f(b)= tspindividu(p (b,:), g,jk);

        if (f(b)> maxf),

            maxf=f(b)

            iit = b;

            jt = p (b,:);

        end

        if (f(b)<=minf),

            minf = f(b)

        end

    end

    fr = mean(f);

    plotvektor1=get(hbestplot1, 'YData');

    plotvektor1(generasi)= maxf;

    set(hbestplot1,'YData',plotvektor1);

```

```

plotvektor2=get(hbestplot2, 'YData');

plotvektor2(generasi)= fr;

set(hbestplot2,'YData',plotvektor2);

set(htext1,'String',sprintf('Fitness terbaik : %7.6f', maxf));

set(htext2,'String',sprintf('Fitness rata-rata : %7.6f',fr));

set(htext3,'String',sprintf('Panjang jalur terbaik : %7.3f',
1/maxf));

set(htext4,'String',sprintf('Ukuran populasi : %3.0f', up));

set(htext5,'String',sprintf('Probabilitas mutasi : %4.3f',
pm));

drawnow

if maxf > fth,

    break;

end

tp = p;

%Elitism

if mod(up,2)==0;          %Ukuran Populasi Genap

    IterasiMulai = 3;

    tp(1,:)= p(iit,:);

    tp(2,:)= p(iit,:);

else                      %Ukuran Populasi Ganjil

    IterasiMulai = 2;

    tp(1,:)=p(iit,:);

end

lf = linearfitnessranking(up,f,maxf,minf)

% Seleksi Ranking dan Pindah silang

for d=IterasiMulai:2:up,

    IP1=seleksiranking(up,lf)

```

```

IP2=seleksiranking(up,lf)

if (rand < ps),

    a = pindahsilang(p(IP1,:),p (IP2,:),g)

    tp(d,:)= a(1,:);

    tp(d+1,:)=a(2,:);

else

    tp(d,:) = p(IP1,:);

    tp(d+1,:)= p(IP2,:);

end

end

%Mutasi dilakukan pada semua kromosom

for e=IterasiMulai:up,

    tp(e,:)=mutasi(tp(e,:),g,pm);

end

p = tp

end

jt

save jt.mat jt

```

Lampiran 4

Perhitungan nilai *fitness* dan seleksi ranking untuk pendistribusian koran di Fidy Agency. Dari pembentukan populasi pada generasi 116:

Individu 1= 13 8 9 10 11 16 5 6 12 4 3 1 2 7 21
20 18 17 19 15 14

Individu 2= 18 8 9 10 11 16 5 6 12 4 3 14 2 7 21
20 17 13 1 19 15

Individu 3= 14 13 1 2 7 21 20 18 17 19 15 8 10 11
16 9 5 6 12 4 3

Individu 4= 18 17 19 15 14 16 5 13 8 9 10 11 6 12
4 3 1 2 7 21 20

Individu 5= 18 17 19 15 14 16 5 13 8 9 10 11 6 12
4 3 1 2 7 21 20

Individu 6= 12 4 3 1 2 7 21 20 18 19 15 14 17 13
8 9 10 11 16 5 6

Individu 7= 19 15 14 9 10 11 16 6 5 12 4 13 8 3
1 2 7 17 21 18 20

Individu 8= 13 1 2 7 21 18 8 9 10 11 16 5 6 17 4
3 12 19 15 14 20

Individu 9= 6 12 4 3 18 21 19 15 14 20 13 1 2 7
17 8 9 10 11 16 5

Individu 10= 12 4 3 1 2 7 21 20 18 19 15 14 17 13
8 9 10 11 16 5 6

Individu 11= 6 12 4 3 10 17 19 15 14 20 13 1 2 7
21 8 9 18 11 16 5

Individu 12= 9 10 11 16 5 15 14 6 12 4 3 1 2 7 21
20 18 17 19 13 8

Individu 13= 9 10 11 16 5 20 14 6 12 4 3 1 2 7 21
15 18 17 19 13 8

Individu 14= 5 6 12 4 3 7 21 20 18 19 15 14 17 13
8 1 2 9 10 11 16

Individu 15= 14 17 13 5 6 7 21 19 15 18 8 9 10 11
 16 12 4 3 1 2 20

Didapatkan nilai fitness dalam tabel 6 berikut.

Tabel 6. Nilai *Fitness* Individu pada Generasi ke-116

Individu ke- i	F(i)=Fitness ke-i (1/x)	R(i)=Ran king
1	0.0159	15
2	0.0014	2
3	0.0048	5
4	0.0115	11
5	0.0126	12
6	0.0137	13
7	0.0037	4
8	0.0104	10
9	0.0081	8
10	0.0148	14
11	0.0003	1
12	0.0059	6
13	0.0026	3
14	0.0070	7
15	0.0093	9

Kemudian memilih individu yang akan diseleksi dengan seleksi ranking dengan langkah-langkah sebagai berikut.

Menghitung nilai total ranking.

$$\text{Total ranking} = \sum R(i) = 1+2+3+4+\dots+15 = 120$$

Menghitung probabilitas individu dan probabilitas kumulatif individu diberikan dalam tabel 7 berikut.

Tabel 7. Probabilitas Individu pada Generasi ke-116

Individu ke-i	Probabilitas Individu $P[i] = R(i)/\sum R(i)$	Probabilitas Kumulatif	Individu ke-i	Probabilitas Individu $P[i] = f(i)/\sum f(i)$	Probabilitas Kumulatif
1	0.125	0,125	9	0.0667	0,6667
2	0.0167	0,1417	10	0.1167	0,7833
3	0.0417	0,1833	11	0.0083	0,7917
4	0.0917	0,275	12	0.05	0,8417
5	0.1	0,375	13	0.025	0,8667
6	0.1083	0,4833	14	0.0583	0,925
7	0.0333	0,5166	15	0.075	1
8	0.0833	0,6			

Membangkitkan 2 bilangan acak (dengan *software* Matlab) antara [0,1] untuk menentukan individu yang terpilih sebagai induk.

$$r_1 = \begin{matrix} 0.8123 \\ 0.1720, \end{matrix} \quad r_5 = \begin{matrix} 0.8251 \\ 0.2144 \end{matrix}$$

$r_2 =$	0.0075 0.1745,	$r_6 =$	0.7295 0.,6920
$r_3 =$	0.4292 0.1355,	$r_7 =$	0.8285 0.,0032
$r_4 =$	0.0095 0.1974,		

Berdasarkan bilangan acak yang dibangkitkan terhadap probabilitas kumulatif individunya, maka individu yang terpilih sebagai induk adalah sebagai berikut.

seleksi ke-1=	individu 12 individu 3,	seleksi ke-5=	individu 12 individu 4,
seleksi ke-2=	individu 1 individu 3,	seleksi ke-6=	individu 10 individu 10,
seleksi ke-3=	individu 6 individu 2,	seleksi ke-7=	individu 12 individu 1,
seleksi ke-4=	individu 1 individu 4,		

Lampiran 5

Output *Software* Matlab proses pencarian rute terpendek pendistribusian koran di Fidi Agency dengan menggunakan algoritma genetika metode seleksi ranking.

GENERASI 116

Populasi =

```
13 8 9 10 11 16 5 6 12 4 3 1 2 7 21 20 18 17 19 15
14
18 8 9 10 11 16 5 6 12 4 3 14 2 7 21 20 17 13 1 19
15
14 13 1 2 7 21 20 18 17 19 15 8 10 11 16 9 5 6 12
4 3
18 17 19 15 14 16 5 13 8 9 10 11 6 12 4 3 1 2 7 21
20
18 17 19 15 14 16 5 13 8 9 10 11 6 12 4 3 1 2 7 21
20
12 4 3 1 2 7 21 20 18 19 15 14 17 13 8 9 10 11 16
5 6
19 15 14 9 10 11 16 6 5 12 4 13 8 3 1 2 7 17 21 18
20
13 1 2 7 21 18 8 9 10 11 16 5 6 17 4 3 12 19 15 14
20
6 12 4 3 18 21 19 15 14 20 13 1 2 7 17 8 9 10 11
16 5
12 4 3 1 2 7 21 20 18 19 15 14 17 13 8 9 10 11 16
5 6
6 12 4 3 10 17 19 15 14 20 13 1 2 7 21 8 9 18 11
16 5
9 10 11 16 5 15 14 6 12 4 3 1 2 7 21 20 18 17 19
13 8
9 10 11 16 5 20 14 6 12 4 3 1 2 7 21 15 18 17 19
13 8
5 6 12 4 3 7 21 20 18 19 15 14 17 13 8 1 2 9 10 11
16
```

14 17 13 5 6 7 21 19 15 18 8 9 10 11 16 12 4 3 1 2
20

MinF =
0.0159

MinF =
3.2461e-004

MinF =
3.2430e-004

LinearFitness =
0.0159 0.0014 0.0048 0.0115 0.0126 0.0137 0.0037 0.0104 0.0081
0.0148 0.0003 0.0059 0.0026 0.0070 0.0093

IP1 =
12
IP2 =
3

Anak =
4 3 1 2 7 21 20 18 17 19 13 8 10 11 16 9 5 6 15 14
12
19 15 8 10 11 16 9 5 6 12 4 3 2 7 21 20 18 17 14
13 1

IP1 =
1
IP2 =
3

Anak =
3 1 2 7 21 20 18 17 19 14 15 8 10 11 16 9 5 6 13
12 4
15 8 10 11 16 9 5 6 12 4 3 1 2 7 21 20 18 17 14 13
19

IP1 =
6

IP2 =

2

Anak =

13 8 9 10 11 16 5 6 12 4 3 1 2 7 21 20 18 19 15 14
17
2 7 21 17 13 1 15 20 18 19 8 9 10 11 16 5 6 12 4 3
14

IP1 =

1

IP2 =

4

Anak =

11 16 5 6 2 7 21 20 18 17 19 15 14 12 4 3 1 13 8 9
10
16 5 13 8 9 10 11 6 12 4 3 1 2 7 21 20 18 17 19 15
14

IP1 =

12

IP2 =

4

Anak =

5 15 14 20 18 17 19 13 8 9 10 11 6 12 4 3 1 2 7 21
16
14 16 5 8 9 10 11 6 12 4 3 1 2 7 21 20 18 17 19 13
15

IP1 =

10

IP2 =

10

Anak =

9 10 11 16 5 6 21 20 18 19 15 14 12 4 3 1 2 7 17
13 8

9 10 11 16 5 6 21 20 18 19 15 14 12 4 3 1 2 7 17
 13 8
 IP1 =
 12
 IP2 =
 1

Anak =
 13 8 9 10 11 16 5 6 12 4 3 1 2 7 21 20 18 17 19 15
 14
 9 10 11 16 5 15 14 6 12 4 3 1 2 7 21 20 18 17 19
 13 8

GENERASI 117

Populasi =
 13 8 9 10 11 16 5 6 12 4 3 1 2 7 21 20 18 17 19 15
 14
 4 3 1 2 7 21 20 18 17 19 13 8 10 11 16 9 5 6 15 14
 12
 19 15 8 10 11 16 9 5 6 12 4 3 2 7 21 20 18 17 14
 13 1
 3 1 2 7 21 20 18 17 19 14 15 8 10 11 16 9 5 6 13
 12 4
 15 8 10 11 16 9 5 2 12 4 3 1 6 7 21 20 18 17 14 13
 19
 13 8 9 10 11 16 5 6 12 4 3 1 2 7 21 20 18 19 15 14
 17
 2 7 21 17 13 1 15 20 18 19 8 9 10 11 16 5 6 12 4 3
 14
 11 16 5 6 2 7 21 20 18 17 19 15 14 12 4 3 1 13 8 9
 10
 16 5 13 8 9 10 11 6 12 4 3 1 2 7 21 20 18 17 19 15
 14
 5 15 14 20 18 17 19 13 8 9 10 11 6 12 4 3 1 2 7 21
 16
 14 16 5 8 9 10 11 6 12 4 3 1 2 7 21 20 18 17 19 13
 15
 9 10 11 16 5 6 21 20 18 19 15 14 12 4 3 1 2 7 17
 13 8

9 10 11 16 5 6 21 20 18 19 15 14 12 4 3 1 2 7 17
 13 8
 13 8 9 10 11 16 5 6 12 4 3 1 2 7 21 20 18 17 19 15
 14
 9 10 11 16 5 15 14 6 12 4 7 1 2 3 21 20 18 17 19
 13 8

MinF =
 0.0159

MinF =
 2.4717e-004

MinF =
 1.6533e-004

LinearFitness =
 0.0148 0.0024 0.0047 0.0081 0.0002 0.0137 0.0035 0.0114 0.0126
 0.0058 0.0069 0.0092 0.0103 0.0159 0.0013

IP1 =
 11
 IP2 =
 1

Anak =
 17 19 13 15 11 16 5 6 12 14 8 9 10 4 3 1 2 7 21 20
 18
 17 19 15 14 9 10 11 6 12 13 8 16 5 4 3 1 2 7 21 20
 18

IP1 =
 6
 IP2 =
 6

Anak =
 10 11 16 5 6 12 4 3 1 2 7 15 14 17 21 20 18 19 13
 8 9

10 11 16 5 6 12 4 3 1 2 7 15 14 17 21 20 18 19 13
8 9

IP1 =

14

IP2 =

12

Anak =

17 10 11 16 5 6 21 20 18 19 15 14 12 4 13 8 9 3 1
2 7
13 8 9 10 11 16 5 6 12 4 3 1 2 7 21 20 18 19 15 14
17

IP1 =

9

IP2 =

1

IP1 =

8

IP2 =

13

Anak =

1 13 8 9 10 6 11 16 5 2 7 21 20 18 17 19 15 14 12
4 3
1 2 17 13 8 7 9 10 11 16 5 6 21 20 18 19 15 14 12
4 3

IP1 =

3

IP2 =

12

Anak =

9 17 13 16 5 6 21 20 18 19 15 14 12 4 3 1 2 7 8 10
11
1 13 8 10 11 16 9 5 6 12 4 3 2 7 21 20 18 17 19 15
14

IP1 =

4

IP2 =

9

Begitu seterusnya hingga ke generasi 133 dan konvergen pada generasi ke 134.

GENERASI 133

Populasi =

```
2 7 21 20 18 17 19 15 14 12 13 8 9 10 11 16 5 6 4
3 1
20 18 17 19 15 14 5 6 12 4 3 13 8 9 10 11 16 1 2 7
21
20 18 19 12 3 1 2 17 13 14 15 8 9 10 11 4 16 5 6 7
21
1 20 19 15 14 17 18 13 8 9 10 11 16 5 6 12 4 3 2 7
21
7 8 9 10 11 16 5 21 20 19 15 14 17 18 12 4 3 13 6
1 2
9 10 3 11 18 17 12 19 15 14 16 5 6 13 4 1 2 7 21
20 8
1 2 7 21 20 18 17 19 15 14 13 8 9 10 11 12 16 5 6
4 3
19 12 13 8 9 10 11 5 15 14 6 4 3 1 16 2 7 21 20 18
17
10 11 4 16 5 6 12 3 21 20 1 2 7 18 17 19 15 14 13
8 9
7 21 20 18 19 14 5 6 12 4 3 13 8 9 10 11 16 17 15
1 2
11 16 17 19 15 14 5 6 12 4 3 1 2 7 21 20 13 18 8 9
10
20 18 19 15 14 17 5 6 12 4 3 13 8 9 10 11 16 1 2 7
21
12 4 3 1 2 13 7 21 20 18 19 8 9 10 11 16 5 6 17 15
14
10 11 17 19 15 14 5 6 18 4 3 1 2 7 21 16 20 12 13
8 9
```

10 11 3 1 14 16 2 7 21 20 18 17 19 15 12 5 6 4 13
8 9

MinF =
0.0183

MinF =
9.3310e-004

MinF =
9.1991e-004

MinF =
4.8459e-004

MinF =
4.7906e-004

MinF =
4.7622e-004

MaxF =
0.0193

MinF =
3.2510e-004

LinearFitness =
0.0180 0.0139 0.0085 0.0071 0.0030 0.0017 0.0193 0.0044 0.0003
0.0058 0.0153 0.0166 0.0098 0.0126 0.0112

IP1 =
8

IP2 =
15

Anak =
17 11 3 1 14 16 2 7 21 20 18 19 12 13 8 9 10 5 15
6 4
4 12 13 8 9 10 11 5 15 14 6 3 1 16 2 7 21 20 18 17
19

IP1 =

15

IP2 =

1

Anak =

21 20 18 17 19 15 5 6 4 12 13 8 9 10 11 16 3 1 14
2 7
8 9 10 11 16 6 4 3 1 20 18 17 19 15 12 5 2 7 21 14
13

IP1 =

11

IP2 =

2

IP1 =

3

IP2 =

2

Anak =

7 21 17 19 15 14 5 6 12 4 3 13 8 9 10 20 18 1 2 11
16
7 21 19 12 3 1 2 17 13 14 15 8 9 10 11 20 18 5 6 4
16

IP1 =

15

IP2 =

14

Anak =

11 14 18 17 19 15 5 6 9 4 3 1 2 7 21 16 20 12 13 8
10
11 14 3 1 2 7 21 16 9 20 18 17 19 15 12 5 6 4 13 8
10

IP1 =

11

IP2 =

8

Anak =

20 13 18 8 9 10 11 5 16 17 19 15 14 6 12 4 3 1 2 7
21

20 18 17 19 15 14 5 6 12 13 8 9 10 11 4 3 1 16 2 7
21

IP1 =

7

IP2 =

1

GENERASI 134 KONVERGEN

Populasi =

1 2 7 21 20 18 17 19 15 14 13 8 9 10 11 12 16 5 6
4 3
17 11 3 1 14 16 2 7 21 20 18 19 12 13 8 9 10 5 15
6 4
4 12 13 8 9 10 11 5 15 14 6 3 1 16 2 7 21 20 18 17
19
21 20 18 17 19 15 5 6 4 12 13 8 9 10 11 16 3 1 14
2 7
8 9 10 11 16 6 4 3 1 20 18 17 19 15 12 5 2 7 21 14
13
11 16 17 19 15 14 5 6 12 4 3 1 2 7 21 20 13 18 8 9
10
20 18 17 19 15 14 5 6 12 4 3 13 8 9 10 11 16 1 2 7
21
7 21 17 19 15 14 5 6 12 4 3 13 8 9 10 20 18 1 2 11
16
7 21 19 12 3 1 2 17 13 14 15 8 9 10 11 20 18 5 6 4
16
11 14 18 17 19 15 5 6 9 4 3 1 2 7 21 16 20 12 13 8
10
11 14 3 1 2 7 21 16 9 20 18 17 19 15 12 5 6 4 13 8
10

20 15 18 8 9 10 11 5 16 17 19 13 14 6 12 4 3 1 2 7
 21
 20 18 17 19 15 14 5 6 12 13 8 9 10 11 4 3 1 16 2 7
 21
 1 2 7 21 20 18 17 19 15 14 13 8 9 10 11 12 16 5 6
 4 3
 2 7 21 20 18 17 19 15 14 12 13 8 9 10 11 16 5 6 4
 3 1

MinF =
 0.0193

MinF =
 2.4535e-004

LinearFitness =
 0.0180 0.0002 0.0071 0.0030 0.0098 0.0153 0.0125 0.0016 0.0112
 0.0057 0.0043 0.0084 0.0139 0.0193 0.0166

IP1 =
 13
 IP2 =
 6

Anak =
 10 11 16 19 15 14 5 6 12 4 3 1 2 7 21 20 18 17 13
 8 9
 21 20 18 19 15 14 5 6 12 13 8 9 10 11 4 3 16 17 1
 2 7

IP1 =
 7
 IP2 =
 9

Anak =
 12 16 7 21 3 1 2 17 13 14 15 8 9 10 11 20 18 5 6 4
 19
 19 17 20 18 15 14 5 6 12 4 3 13 8 9 10 11 16 1 2 7
 21

IP1 =

13

IP2 =

4

Anak =

21 20 18 17 19 15 14 5 6 12 13 8 9 10 11 4 3 1 16
2 7
7 18 17 19 21 20 15 5 6 4 12 13 8 9 10 11 16 3 1
14 2

IP1 =

7

IP2 =

15

Anak =

19 4 3 1 2 7 21 15 14 12 13 8 9 10 11 16 5 6 20 18
17
20 18 17 19 15 14 5 6 12 4 3 13 8 9 10 11 16 1 2 7
21

IP1 =

10

IP2 =

6

Anak =

15 5 6 9 16 12 13 8 10 4 3 1 2 7 21 20 11 14 18 17
19
5 6 12 20 13 18 8 9 10 4 3 1 2 7 21 16 11 17 19 15
14

IP1 =

14

IP2 =

5

Anak =

8 9 10 11 16 6 4 3 1 20 18 17 19 15 12 5 2 7 21 14
13

20 18 17 19 5 2 7 21 15 14 13 8 9 10 11 12 16 6 4
3 1

IP1 =

1

IP2 =

15

GENERASI 135

Populasi =

1 2 7 21 20 18 17 19 15 14 13 8 9 10 11 12 16 5 6
4 3
10 11 16 19 15 14 5 6 12 4 3 1 2 7 21 20 18 17 13
8 9
21 20 18 19 15 14 5 6 12 13 8 9 10 11 4 3 16 17 1
2 7
12 16 7 21 3 1 2 17 13 14 15 8 9 10 11 20 18 5 6 4
19
19 17 20 18 15 14 5 6 12 4 3 13 8 9 10 11 16 1 2 7
21
21 20 18 17 19 15 14 5 6 12 13 8 9 10 11 4 3 1 16
2 7
7 18 17 19 21 20 15 5 6 4 12 13 8 9 10 11 16 3 1
14 2
19 4 3 1 2 7 21 15 14 12 13 8 9 10 11 16 5 6 20 18
17
20 18 17 19 15 14 5 6 12 4 3 13 8 9 10 11 16 1 2 7
21
15 5 6 9 16 12 13 8 10 4 3 1 2 7 21 20 11 14 18 17
19
5 6 12 20 13 18 8 9 10 4 3 1 2 7 21 16 11 17 19 15
14
8 9 10 11 16 6 4 3 1 20 18 17 19 15 12 5 2 7 21 14
13
20 18 17 19 5 2 7 21 15 14 13 8 9 10 11 12 16 6 4
3 1

1 2 7 21 20 18 17 19 15 14 13 8 9 10 11 12 16 5 6
4 3
2 7 21 20 18 17 19 15 14 12 13 8 9 10 11 16 1 6 4
3 5

Fitness maksimum = 0.0193

1 2 7 21 20 18 17 19 15 14 13 8 9 10 11 12 16 5 6
4 3

GENERASI 136

Populasi =

1 2 7 21 20 18 17 19 15 14 13 8 9 10 11 12 16 5 6
4 3
4 3 16 17 1 2 7 6 12 21 20 18 19 15 14 9 13 8 5 10
11
10 11 16 1 2 7 21 6 12 20 18 17 19 15 14 5 4 3 13
8 9
3 2 7 1 21 20 18 17 19 15 14 13 8 9 10 11 12 16 5
6 4
3 2 7 1 21 20 18 17 19 15 14 13 8 9 10 11 12 16 5
6 4
6 12 13 8 4 3 16 17 1 2 7 9 10 11 21 20 18 19 15
14 5
6 12 13 8 4 3 16 17 1 2 7 9 10 11 21 20 18 19 15
14 5
19 2 14 10 11 16 5 6 12 4 3 13 8 9 1 15 7 21 20 18
17
11 16 1 2 7 21 17 19 15 14 13 8 9 10 20 18 5 6 12
4 3
8 9 10 11 16 14 5 6 12 4 3 1 2 7 21 20 18 17 19 15
13
16 1 2 7 21 18 17 19 15 14 13 20 5 6 12 4 3 8 9 10
11
19 17 20 18 15 14 5 6 12 4 3 13 8 9 10 11 16 1 2 7
21
1 2 7 21 20 18 17 19 15 14 13 8 9 10 11 12 16 5 6
4 3

11 4 3 1 16 2 7 21 15 14 20 18 17 19 5 6 12 13 8 9
 10
 9 10 11 16 4 3 1 5 6 12 20 18 17 19 2 7 21 15 14
 13 8

Fitness maksimum = 0.0193

1 2 7 21 20 18 17 19 15 14 13 8 9 10 11 12 16 5 6 4
 3

Untuk generasi selanjutnya tetap didapatkan nilai *fitness* maksimum adalah 0,0193. Dapat diartikan bahwa nilai *fitness* telah konstan, sehingga didapatkan solusi optimum rute terpendek dengan rute

1 2 7 21 20 18 17 19 15 14 13 8 9 10 11 12 16 5 6 4
 3

Untuk rute diawali dari gudang perusahaan dan kembali ke gudang perusahaan maka rute distribusi menjadi

**1 2 7 21 20 18 17 19 15 14 13 8 9 10 11 12 16 5 6 4
 3**

Atau

1 3 4 6 5 16 12 11 10 9 8 13 14 15 19 17 18 20 21 7 2