

**IMPLEMENTASI ALGORITMA GENETIKA UNTUK
MINIMASI TOTAL PANJANG RUTE PENDISTRIBUSIAN
PRODUK PADA KASUS *TRAVELLING SALESMAN PROBLEM***

SKRIPSI

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Industri**



Oleh :

Nama : Kus Sri Wahyuni
Nomor Mahasiswa : 02522189
Jurusan : Teknik Industri

**JURUSAN TEKNIK INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2011

PENGAKUAN

Demi Allah, saya akui karya ini adalah hasil kerja saya sendiri kecuali nukilan dan ringkasan yang setiap satunya telah saya jelaskan sumbernya. Jika dikemudian hari ternyata terbukti pengakuan saya ini tidak benar dan melanggar peraturan yang sah dalam karya tulis dan hak intelektual maka saya bersedia ijazah yang telah saya terima untuk ditarik kembali oleh Universitas Islam Indonesia.

Yogyakarta, Mei 2011



Kus Sri Wahyuni

02522189

LEMBAR PENGESAHAN PEMBIMBING

**IMPLEMENTASI ALGORITMA GENETIKA UNTUK MINIMASI TOTAL
PANJANG RUTE PENDISTRIBUSIAN PRODUK PADA KASUS
*TRAVELLING SALESMAN PROBLEM***

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Industri**

Disusun Oleh:

Nama : KUS SRI WAHYUNI

No. Mahasiswa : 02 522 189

Yogyakarta, Mei 2011

Dosen Pembimbing



(Winda Nur Cahyo, ST. MT)

LEMBAR PENGESAHAN PENGUJI

**IMPLEMENTASI ALGORITMA GENETIKA UNTUK MINIMASI
TOTAL PANJANG RUTE PENDISTRIBUSIAN PRODUK PADA
KASUS TRAVELLING SALESMAN PROBLEM**

Oleh

Nama : Kus Sri Wahyuni

No Mahasiswa : 02 522 189

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Jurusan Teknik Industri

Fakultas Teknologi Industri

Universitas Islam Indonesia

Yogyakarta, Mei 2011

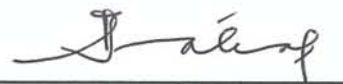
Tim Penguji

Tanda Tangan

Winda Nur Cahyo ST. MT
Ketua



Ir. Ali Parkhan, MT
Anggota I



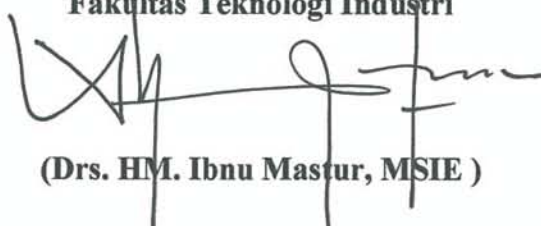
Yuli Agusti Rochman, ST., M. Eng
Anggota II



Mengetahui,

Ketua Jurusan Teknik Industri

Fakultas Teknologi Industri



(Drs. HM. Ibnu Mas'ur, MSIE)

24
6 2011

HALAMAN PERSEMBAHAN



Kupersembahkan karya ini untuk :
ALLAH SWT yang telah memberikan segala nikmat dan karunia-Nya.
Ibuku tersayang yang telah menjadi single parent terhebat untukku.
Alm. Bapakku tersayang yang belum pernah kulihat.
Serta kakak-kakakku terkasih.

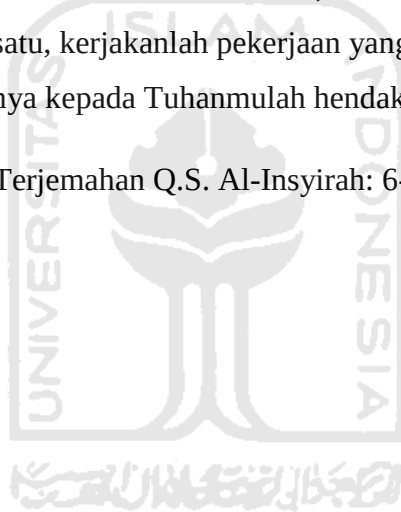
MOTTO

“Sesungguhnya orang yang beriman dan mengerjakan amal shaleh bagi mereka surga yang mengalir dibawahnya sungai-sungai : itulah keberuntungan yang besar”

(Terjemahan Q.S. Al Buruuj : 11)

“Sesungguhnya setelah kesulitan itu ada kemudahan, maka apabila engkau telah selesai dengan pekerjaan yang satu, kerjakanlah pekerjaan yang lain dengan sesungguhnya dan hanya kepada Tuhanmulah hendaknya kamu berharap”

(Terjemahan Q.S. Al-Insyirah: 6-8)



KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu'alaikum Wr.Wb

Segala puji bagi Allah SWT, yang telah memberikan rahmat dan hidayah-Nya serta yang telah memberikan penulis kehidupan, kesempatan, dan kesehatan sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Implementasi Algoritma Genetika Untuk Minimasi Total Panjang Rute Pendistribusian Produk Pada Kasus *Travelling Salesman Problem*” ini dengan baik.

Tugas akhir ini merupakan salah satu syarat untuk meraih gelar sarjana S1 pada jurusan Teknik Industri, Fakultas Teknologi Industri, Universitas Islam Indonesia, Yogyakarta.

Dalam mengerjakan Tugas Akhir ini penulis banyak mendapatkan dukungan dan semangat dari berbagai pihak. Baik secara langsung maupun tidak. Untuk itu penulis ingin mengucapkan terima kasih yang setulusnya kepada:

1. Bapak Gumbolo Hadi Susanto, Ir., Msc, selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
2. Bapak Drs. HM. Ibnu Mastur, MSIE selaku Ketua Jurusan Teknik Industri, Fakultas Teknologi Industri, Universitas Islam Indonesia.
3. Bapak Winda Nur Cahyo ST, MT, selaku dosen pembimbing Tugas Akhir yang banyak memberi masukan dan bimbingan selama tugas akhir ini.

4. Bapak Arya dan Bapak Ibnu Hadjar selaku pembimbing di PT. Tigaraksa Satria Tbk, Yogyakarta yang telah memberikan kemudahan bagi penulis dalam pengambilan data dan saran dalam penyusunan tugas akhir ini.
5. Kedua orang tua, Bapak Alm. Keling dan Ibu Soedjarmi. Kakak-kakakku : Kusdaryanto, Kushendarto, Kusbinarti, Kusprastyani. Serta Mas Lamon Simbangando yang selalu memberikan dukungan moril.
6. Semua pihak yang telah membantu dalam penyusunan tugas akhir ini yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa karya ini jauh dari sempurna. Hal ini karena keterbatasan penulis dalam hal wawasan, pengalaman, pengetahuan dan penguasaan ilmu yang penulis miliki. Oleh karena itu, kritik dan saran yang sifatnya membangun sangat penulis harapkan.

Akhir kata, dengan segala kerendahan hati penulis mengharapkan semoga karya yang sederhana ini dapat bermanfaat bagi pembaca sekalian. Amin.

Wassalamu'alaikum Wr.Wb

Yogyakarta, Mei 2011

Penulis

ABSTRAKSI

Ketatnya persaingan bisnis menyebabkan perusahaan juga harus mampu mempertahankan kualitas produk dan kualitas pelayanan produk, termasuk pendistribusian produk. Sehingga proses pendistribusian produk harus tetap dijaga dan diperhatikan. *Travelling Salesman Problem* (TSP) merupakan persoalan optimasi untuk mencari perjalanan terpendek bagi seorang salesman yang harus mendistribusikan produk ke beberapa agen dan kembali ke gudang perusahaan. Pada dasarnya persoalan ini adalah menentukan rute kendaraan yang mengangkut produk dari gudang perusahaan ke beberapa agen. Tujuannya adalah agar diperoleh rute distribusi dengan total jarak tempuh yang minimum dan mengetahui besarnya perbaikan jalur distribusi yang diperoleh. Pada penelitian ini TSP diimplementasikan menggunakan metode *heuristic* algoritma genetika (AG) untuk menentukan rute perjalanan yang hanya sekali melalui agen lainnya dan kembali ke gudang perusahaan. Dengan menggunakan metode AG diperoleh rute pendistribusian yang baru dengan total jarak tempuh 437,25 km yang semula 588,2 km sehingga mengalami pengurangan jarak tempuh sebesar 150,95 km atau dalam persentase sebesar 25,66%.

Kata kunci : distribusi produk, *Travelling Salesman Problem*, Algoritma Genetika



DAFTAR ISI

HALAMAN JUDUL	i
SURAT PERNYATAAN KEASLIAN TA	ii
SURAT KETERANGAN PENELITIAN	iii
LEMBAR PENGESAHAN PEMBIMBING	iv
LEMBAR PENGESAHAN PENGUJI ..	v
HALAMAN PERSEMBAHAN	vi
HALAMAN MOTTO..	vii
KATA PENGANTAR...	viii
ABSTRAKSI	x
DAFTAR ISI.....	xi
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
 BAB I PENDAHULUAN	
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan	4
 BAB II KAJIAN PUSTAKA	
2.1 Distribusi Produk	6
2.1.1 Konsep Distribusi Produk.	6
2.1.2 Jalur Terpendek.....	7
2.2 <i>Travelling Salesman Problem</i> (TSP)	9
2.2.1 Pengertian TSP.....	9
2.2.2. Sejarah dan Perkembangan Penyelesaian TSP	11

2.2.3 Konsep <i>Travelling Salesman Problem</i>	13
2.2.4 Aplikasi Permasalahan TSP	14
2.3 Algoritma Genetika (AG)	16
2.3.1 Struktur Umum Algoritma Genetika.....	16
2.3.2.Komponen-Komponen Utama Algoritma Genetika	17
2.3.3 Keuntungan Algoritma Genetika	20
2.3.4 Aplikasi Algoritma Genetika.	21
2.4 Algoritma Genetika dalam TSP	22
2.5 Penerapan Algoritma Genetika dalam Kasus TSP.....	23

BAB III METODOLOGI PENELITIAN

3.1 Obyek dan Tempat Penelitian	29
3.2 Kajian Literatur	29
3.3 Diagram Metodologi Penelitian	30
3.4 Pengumpulan Data	31
3.5 Pengolahan Data.....	32
3.6 Pembahasan.....	34
3.7 Kesimpulan dan Saran.....	35

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Profil Perusahaan	36
4.1.1 Sejarah Singkat PT. Tigaraksa Satria Tbk	36
4.1.2 Bisnis Utama	36
4.1.3 Kegiatan Usaha Lainnya	37
4.1.4 Visi PT. Tigaraksa Satria Tbk.....	37
4.2 Pengumpulan Data	38
4.2.1 Data Nama Agen, Alamat Lokasi Agen.....	38
4.2.2 Data Jarak dari Gudang ke Agen dan Jarak Antar Agen	40
4.2.3 Rute Pendistribusian	41
4.2.4 Jumlah dan Kapasitas Kendaraan	42
4.3 Pengolahan Data.....	43

4.3.1 Perhitungan Total Jarak Tempuh Rute Pendistribusian Awal	43
4.3.2 Membangkitkan Populasi Awal dengan Inisialisasi Populasi	45
4.3.3 Menentukan Evaluasi Fungsi	45
4.3.4 Tahap Seleksi Kromosom Induk (<i>Parent Selection</i>)	46
4.3.5 Tahap Rekombinasi	48
4.3.6 Mutasi	49
4.3.7 Elitisme	51
4.3.8 Implementasi TSP dalam Matlab	52
4.3.9 Menjalankan Evolusi per 250 Generasi	56
4.3.10 Menjalankan Evolusi dengan Beberapa Parameter	61
4.3.11 Rute Baru Pendistribusian Produk	65

BAB V PEMBAHASAN

5.1 Perbandingan Total Panjang Lintasan Urutan Rute Pendistribusian Rute Pendistribusian Produk Awal dan Setelah Diminimasi Menggunakan AG.....	67
5.2 Pemilihan Urutan Rute Pendistribusian Produk.....	68

BAB VI KESIMPULAN DAN SARAN

6.1 Kesimpulan	69
6.2 Saran	69

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

Tabel 4.1	Nama Agen/ Toko dan Alamat Lokasi.....	38
Tabel 4.2	Rute Pendistribusian Produk.....	42
Tabel 4.3	Jarak Tempuh Agen i ke Agen $i+1$	43
Tabel 4.4	Total Jarak Tempuh Agen i ke Agen $i+1$	44
Tabel 4.5	<i>Fitness</i> dan Panjang Jalur Terbaik per 250 Generasi	60
Tabel 4.6	Parameter Evolusi.....	61
Tabel 4.7	<i>Fitness</i> dan Panjang Jalur Terbaik per 250 Generasi Menggunakan 5 Parameter Evolusi.....	64
Tabel 4.8	Rute Baru Pendistribusian Produk.....	66



DAFTAR GAMBAR

Gambar 2.1	Graf ABCDEFG.....	8
Gambar 2.2	Model TSP.....	10
Gambar 2.3	Jarak Dua Kota Pada Koordinat Kartesius.....	11
Gambar 2.4	Mekanisme <i>Crossover</i>	27
Gambar 2.5	Mekanisme Mutasi.....	28
Gambar 3.1	Diagram Metodologi Penelitian.....	30
Gambar 3.2	Diagram Pengolahan Data Menggunakan AG.....	31
Gambar 4.1	Grafik Rute Pendistribusian Awal	41
Gambar 4.2	Grafik Evolusi dari Generasi 1 sampai 250.....	56
Gambar 4.3	Grafik Evolusi dari Generasi 1 sampai 500.....	56
Gambar 4.4	Grafik Evolusi dari Generasi 1 sampai 750.....	57
Gambar 4.5	Grafik Evolusi dari Generasi 1 sampai 1000.....	57
Gambar 4.6	Grafik Evolusi dari Generasi 1 sampai 1250.....	58
Gambar 4.7	Grafik Evolusi dari Generasi 1 sampai 1500.....	58
Gambar 4.8	Grafik Evolusi dari Generasi 1 sampai 1750.....	59
Gambar 4.9	Grafik Evolusi dari Generasi 1 sampai 2000.....	59
Gambar 4.10	Grafik Evolusi 2000 Generasi dengan Parameter 1.....	61
Gambar 4.11	Grafik Evolusi 2000 Generasi dengan Parameter 2.....	62
Gambar 4.12	Grafik Evolusi 2000 Generasi dengan Parameter 3.....	62
Gambar 4.13	Grafik Evolusi 2000 Generasi dengan Parameter 4.....	63
Gambar 4.14	Grafik Evolusi 2000 Generasi dengan Parameter 5.....	63
Gambar 4.15	Grafik Rute Pendistribusian Setelah Menggunakan AG.....	65

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Masalah pendistribusian produk pada lingkungan industri menjadi hal yang sangat penting pada situasi dan kondisi industri di Indonesia. Kondisi dimana kompetisi industri yang semakin ketat yang diakibatkan banyaknya kompetitor produk sejenis atau produk substitusi membuat konsumen dapat memilih produk dengan spesifikasi yang sama namun dengan harga serendah mungkin (Nudu J.H, 2007). Untuk itu dalam kompetisi yang ketat diperlukan penurunan biaya produksi harus dilakukan secara menyeluruh pada semua komponen biaya agar diperoleh keuntungan yang memadai, diantaranya adalah biaya distribusi.

Permasalahan TSP (*Traveling Salesman Problem*) adalah permasalahan dimana seorang *salesman* harus mengunjungi semua kota dimana tiap kota hanya dikunjungi sekali, dan dia harus mulai dari dan kembali ke kota asal (Amin A.R, Ikhsan M dan Wibisono L, 2008). Konsep dasar dari TSP adalah bagaimana menemukan rute terpendek yang menghubungkan satu lokasi ke lokasi berikutnya seperti beberapa kota yang dikunjungi oleh *salesman* dalam rute penjualannya. Seorang *salesman* menghabiskan waktunya untuk mengunjungi n kota dalam ruang lingkup yang tertutup. Dalam satu perjalanan, *salesman* hanya mengunjungi setiap kota satu kali saja, dan perjalanan itu akan berakhir pada kota pertama yang dikunjunginya. Tujuannya adalah menentukan rute dengan jarak total atau biaya yang paling minimum. TSP merupakan persoalan yang sulit bila dipandang dari sudut komputasinya. Beberapa metode telah digunakan untuk memecahkan persoalan

tersebut namun hingga saat ini belum ditemukan algoritma yang mampu untuk menyelesaikannya. Sebuah penelitian (Kurnianto, Fahrudin, dan Hardono, 2006) mengenai TSP menggunakan algoritma *brute force* memberikan hasil yang optimal untuk jumlah node yang sedikit tetapi membutuhkan waktu yang lama karena harus menelusuri seluruh kemungkinan rute yang ada. Cara termudah untuk menyelesaikan TSP yaitu dengan mencoba semua kemungkinan rute dan mencari rute yang terpendek. Namun, pada zaman yang serba praktis sekarang ini dibutuhkan algoritma yang dapat menyelesaikan TSP dengan cepat sehingga diperoleh solusi yang mendekati solusi optimal.

Oleh karena itu digunakan algoritma genetika untuk menentukan perjalanan terpendek yang melalui kota lainnya hanya sekali. Algoritma genetika yaitu algoritma pencarian dan optimasi yang terinspirasi oleh prinsip dari genetika dan seleksi alam (teori evolusi Darwin). Keuntungan penggunaan algoritma genetika sangat jelas terlihat dari kemudahan implementasi dan kemampuannya untuk menemukan solusi yang bagus (bisa diterima) secara cepat untuk masalah optimasi dan selain optimasi (Suyanto, 2005).

Penelitian ini dimaksudkan untuk mengatasi suatu permasalahan TSP dalam pendistribusian produk agar didapat total jarak terpendek. Penelitian ini menggunakan metode Algoritma Genetik (AG) menggunakan *software* Matlab 7.6 yang dapat diaplikasikan untuk mengatasi suatu permasalahan TSP. Algoritma ini sangat tepat digunakan untuk penyelesaian masalah optimasi yang kompleks dan sukar diselesaikan dengan metode konvensional.

1.2 Rumusan Masalah

1. Bagaimana rute pendistribusian produk yang harus ditempuh *salesman* untuk memperoleh total jarak tempuh yang minimum?
2. Berapa besar perubahan jalur distribusi terhadap jalur distribusi yang biasa ditempuh *salesman*?

1.3 Batasan Masalah

Untuk memudahkan peneliti agar permasalahan ini tidak semakin luas, maka peneliti menetapkan beberapa batasan masalah:

1. Penelitian ini dilakukan di PT. Tigaraksa Satria Yogyakarta.
2. Agen diwakili nomor.
3. Jarak perjalanan antara dua agen adalah simetris yaitu jarak dari agen A ke B adalah sama dengan jarak dari agen B ke A.
4. Panjang lintasan yang diperoleh dalam pendistribusian produk berdasarkan data yang ada diperusahaan.
5. Data yang diambil adalah data rute pendistribusian produk susu SGM.
6. Hambatan-hambatan pada saat pendistribusian produk diabaikan.
7. Metode yang digunakan adalah metode Algoritma Genetika dengan menggunakan bantuan *software* Matlab 7.6.

1.4 Tujuan Penelitian

Adapun tujuan penelitian ini adalah sebagai berikut :

1. Memperoleh rute distribusi dengan total jarak tempuh yang minimum.
2. Menentukan besarnya perbaikan jalur distribusi yang diperoleh.

1.5 Manfaat Penelitian

Dengan penelitian ini diharapkan akan dapat memberikan manfaat :

1. Jalur distribusi optimal yang diperoleh akan memberikan biaya operasional yang rendah bagi perusahaan.
2. Jalur distribusi yang lebih pendek akan memberikan peluang untuk melayani lebih banyak konsumen.
3. Jarak tempuh yang lebih pendek akan memberikan waktu untuk beristirahat yang lebih banyak kepada salesman.
4. Mengurangi biaya distribusi produk sehingga dapat dialihkan untuk sumber daya yang lain.

1.6 Sistematika Penulisan

Agar hasil penelitian dapat tersusun secara lebih baik maka dibuat susunan sistematika sebagai berikut:

BAB I PENDAHULUAN

Bab pendahuluan ini akan diuraikan secara singkat mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II KAJIAN PUSTAKA

Bab ini berisikan landasan teori-teori dasar tentang masalah penelitian, penjelasan mengenai konsep-konsep dasar mengenai permasalahan yang diangkat serta mendukung penelitian yang akan dilakukan

BAB III METODOLOGI PENELITIAN

Bab ini berisikan penjelasan mengenai obyek penelitian, tempat dan waktu penelitian, teknik pengumpulan data dan kerangka pemecahan masalah.

Tahap-tahap dalam pemecahan masalah dijelaskan dengan adanya *flow chart* pemecahan masalah.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Bab ini berisikan data–data yang diperlukan dalam penelitian, pengolahan data tersebut, baik secara langsung maupun tidak dengan bantuan *software*.

BAB V PEMBAHASAN

Bab ini membahas hasil penelitian berupa tabel hasil pengolahan data, grafik serta analisa yang menyangkut penjelasan teoritis secara kualitatif, kuantitatif maupun statistik dari hasil penelitian dan kajian untuk menjawab tujuan penelitian.

BAB VI KESIMPULAN DAN SARAN

Kesimpulan, memuat pernyataan singkat dan tepat yang dijabarkan dari hasil penelitian dan pembahasan untuk membuktikan atau menjawab permasalahan.

Saran, dibuat berdasarkan pengalaman dan pertimbangan penulis, ditujukan kepada para peneliti (perusahaan) dalam bidang yang sejenis, yang ingin melanjutkan, mengembangkan, atau menerapkan penelitian yang telah diselesaikan.

BAB II

KAJIAN PUSTAKA

2.1 Distribusi Produk

Keberhasilan persaingan dalam industri saat ini tidak hanya ditentukan oleh inovasi produk dan harga yang murah saja. Namun yang tidak kalah pentingnya adalah bagaimana kita membangun inovasi distribusi (Nudu J.H, 2007).

Semakin pesat pertumbuhan perusahaan, bertambah pula kerumitan masalah yang dihadapi oleh perusahaan sehingga mendorong perusahaan untuk merubah struktur manajemen yang sudah ada sesuai dengan pertumbuhan perusahaan. Fokus perhatian juga sudah meluas, saluran distribusi produk yang dulunya tidak terlalu diperhatikan sekarang telah berubah menjadi perhatian yang cukup berpengaruh. Distribusi produk berkaitan dengan penyampaian produk yang telah dipesan oleh konsumen dari perusahaan, pengiriman bahan baku ke lokasi produksi, pemindahan barang setengah jadi ke departemen selanjutnya supaya bisa diproses lebih lanjut, dan lain-lain (Novan, 2004).

2.1.1 Konsep Distribusi Produk

Pelaksanaan distribusi produk membutuhkan strategi yang efisien supaya menghindari kerugian-kerugian yang sebenarnya bisa dihindari. Strategi distribusi adalah strategi penyediaan barang-barang bagi para pelanggan potensial. Strategi ini meliputi pemilihan jalan atau saluran perantara pemasaran yang membentuk struktur distribusi. Dalam tahap awal strategi distribusi ini, produsen berusaha mengenali

saluran-saluran distribusi yang paling efektif untuk mencapai pasar yang hendak dilayani (Androvov, 2005).

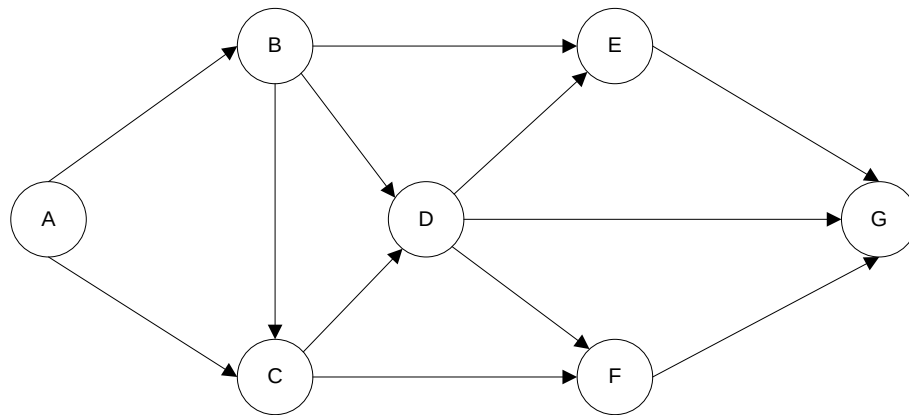
Salah satu strategi distribusi adalah menentukan jumlah perantara yang hendak dipakai pada masing-masing level dalam saluran ini yang optimal. Produsen harus mempertimbangkan manfaat yang dapat diperoleh dengan menggunakan banyak perantara dibandingkan dengan manfaat yang diperoleh dengan menggunakan sedikit perantara. Pada akhirnya harus diambil keputusan mengenai bagaimana produk tersebut akan didistribusikan secara fisik setelah dipilih saluran distribusi yang akan dipakai. Tingkat pelayanan terhadap konsumen harus ditentukan sesuai dengan strategi pemasaran secara menyeluruh dari perusahaan tersebut. Berbagai sistem distribusi dapat dianalisa untuk mendapatkan sistem distribusi yang memenuhi tingkat pelayanan yang efisien (Rewoldt dan Stewart, 1987).

2.1.2 Jalur Terpendek

Salah satu keputusan operasional yang sangat penting dalam manajemen distribusi adalah penentuan rute atau jalur pengiriman dari satu lokasi ke beberapa lokasi tujuan. Keputusan seperti ini sangat penting bagi mereka yang harus mengirimkan produk dari satu lokasi (misalnya gudang) ke berbagai toko yang tersebar di sebuah daerah. Setiap harinya produk harus didistribusikan dari gudang ke tempat penjualan untuk selanjutnya ke konsumen individu. Keputusan rute yang akan ditempuh oleh setiap kendaraan atau *salesman* akan sangat berpengaruh terhadap efisiensi waktu dan biaya pengiriman. Semakin pendek jarak tempuh setiap rute maka akan memberikan keuntungan kepada perusahaan.

Jalur terpendek adalah suatu jaringan pengarahan perjalanan dimana seseorang pengarah jalan ingin menentukan jalur terpendek antara dua kota, berdasarkan

beberapa jalur alternatif yang tersedia, dimana titik tujuan hanya satu (Saptono F dan Hidayat T, 2007). Gambar menunjukkan suatu graf ABCDEFG.



Gambar 2.1 Graf ABCDEFG

Pada gambar 2.1, misalkan kita dari kota A ingin menuju Kota G. Untuk menuju kota G, dapat dipilih beberapa jalur yang tersedia:

A – B – C – D – E – G

A – B – C – D – F – G

A – B – C – D – G

A – B – C – F – G

A – B – D – E – G

A – B – D – F – G

A – B – D – G

A – B – E – G

A – C – D – E – G

A – C – D – F – G

A – C – D – G

A – C – F – G

Berdasarkan data di atas, dapat dihitung jalur terpendek dengan mencari jarak antara jalur-jalur tersebut. Apabila jarak antar jalur belum diketahui, jarak dapat dihitung berdasarkan koordinat kota-kota tersebut. Setelah didapatkan hasil jarak antar kota, jalur terpendek dapat dihitung menggunakan metode yang ada.

2.2 *Tavelling Salesman Problem (TSP)*

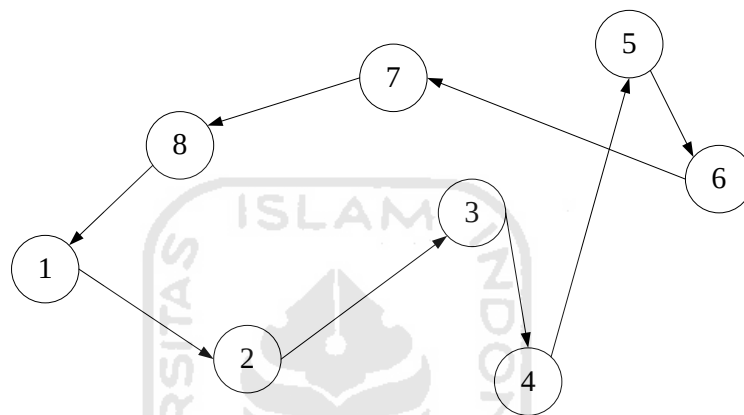
Selain masalah transportasi, efisiensi pengiriman barang atau distribusi produk juga ditentukan oleh lintasan yang diambil untuk mendistribusikan produk tersebut. Oleh karena itu solusi optimal dari permasalahan TSP ini akan sangat membantu perusahaan distribusi untuk mengefisiensikan proses pendistribusian produk baik dari segi waktu maupun biaya (Amin A.R, Ikhsan M dan Wibisono L, 2008).

2.2.1 **Pengertian TSP**

Permasalahan TSP (*Travelling Salesman Problem*) adalah permasalahan dimana seorang *salesman* harus mengunjungi semua kota dimana tiap kota hanya dikunjungi sekali, dan dia harus mulai dari dan kembali ke kota asal (Amin A.R, Ikhsan M dan Wibisono L, 2008). TSP melibatkan seorang *salesman* yang harus melakukan kunjungan ke sejumlah kota dalam menjajakan produknya.

Ningtyas, Vina, dan Ernastuti (2008) dalam penelitian terdahulunya menjelaskan bahwa TSP dapat diilustrasikan sebagai perjalanan yang harus melalui semua kota yang dituju dengan jarak yang terpendek menggunakan teknik pemrograman dinamik. Konsep TSP banyak diterapkan untuk penyelesaian berbagai masalah, antara lain untuk menentukan rute perjalanan dengan jarak semimum mungkin dengan melewati semua kota, menentukan *track* yang panjang dan memiliki

percabangan, membantu pengooptimalan proses perjalanan pesan (*message*) pada jaringan interkoneksi komputer, dan lain-lain. Persoalan yang timbul adalah menentukan rute dengan jarak tempuh terpendek supaya biaya pengangkutan menjadi minimum. Banyak formulasi persoalan telah dibuat yang mengindikasikan arah dengan perjalanan optimal. Secara umum TSP dapat diformulasikan (Paputungan, 2004) sebagai berikut :



Gambar 2.2 Model TSP

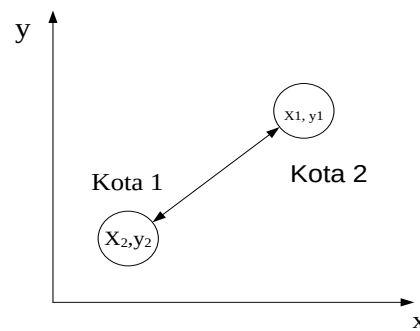
$$d_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2.1)$$

$$\text{Minimumkan} = \text{Min} \sum_{k=1}^n \sum_{m=1}^n c_{k,m} d \quad (2.2)$$

Dimana : c_{km} = Biaya perjalanan antara kota-k ke kota-m

d_{12} = panjang lintasan dari kota-1 ke kota-2

Biaya perjalanan antara dua kota ($c_{k,m}$) berbanding lurus (linier) dengan panjang lintasan dari satu kota ke kota berikutnya (d). Asumsi yang digunakan untuk mempermudah persoalan adalah bahwa letak kota dapat digambarkan dalam dua dimensi dalam koordinat kartesius.



Gambar 2.3 Jarak Dua Kota Pada Koordinat Kartesius

2.2.2 Sejarah dan Perkembangan Penyelesaian TSP

Permasalahan matematika tentang *Travelling Salesman Problem* (TSP) dikemukakan pada tahun 1800 oleh matematikawan India Wiliam Rowan Hamilton dan matematikawan Inggris Thomas Penyngton (Amin A.R, Ikhsan M dan Wibisono L, 2008). Kemudian TSP diperkenalkan oleh Rand pada tahun 1948, reputasi Rand membuat TSP dikenal dengan baik dan menjadi masalah yang populer (Puspitorini, 2008).

Dijelaskan oleh Amin A.R, Ikhsan M dan Wibisono L (2008) perkembangan penyelesaian atau pemecahan masalah TSP telah dicapai selama lebih dari 30 tahun kebelakang diantaranya adalah :

a. 49 Kota - *DANTZIG49*

DANTZIG49 adalah permasalahan yang diteliti oleh Dantzig, Fulkerson, dan Johnson yang dapat kita lihat dalam makalah mereka tahun 1954 tentang solusi dari permasalahan TSP yang terdiri dari satu kota tiap 48 negara bagian di Amerika Serikat ditambah kota Washington D.C. Para penulis bekerja dengan 49 kota tersebut. Jalur optimal dari 49 kota ini, menggunakan jalan pintas yang terdapat pada 7 kota selain 49 kota tersebut. Jalur yang dibuat berdasarkan jarak pada setiap jalur dalam kota diantara 49 kota tersebut. Para penulis

mengatakan bahwa mereka mendapat tabel jarak dari Bernice Brown karyawan perusahaan Rand.

b. 120 Kota - *GR120*

GR120 telah menjadi pengujian standar bagi permasalahan TSP sejak tahun 1977. GR120 ini mempunyai 120 titik yang terdiri dari jarak tempuh antara 120 kota yang terdapat di sekitar Jerman. Daftar kota-kota ini terdapat pada atlas umum negara Jerman tahun 1967/68.

c. 318 Kota - *LIN318*

LIN318 muncul pada tahun 1973 dalam makalah yang ditulis oleh S.Lin dan B.W. Kernighan. Data dari LIN318 terdiri dari 318 yang muncul dari hasil pengeboran, dimana boranya adalah sebuah sinar laser. Lin dan Kernighan menulis bahwa mereka mendapatkan datanya dari R.Haberman. Permasalahan ini pertama diselesaikan oleh H.Crowder dan M. W. Padberg pada tahun 1980.

d. 532 Kota - *ATT532*

ATT532 muncul pada tahun 1987 didalam makalah yang ditulis oleh M. Padberg dan G. Rinaldi. Dalam ATT532 terdapat data mengenai 532 kota yang berlokasi di benua Amerika Serikat, dan Padberg serta Rinaldi menulis bahwa mereka mendapatkan permasalahannya dari Shen Lin karyawan dari laboratorium AT&T Bell.

e. 666 Kota - *GR666*

GR666 pertama kali diselesaikan oleh O. Holland dan M. Groetschel, yang muncul dalam Tesis PhD Olaf Holland's pada tahun 1987. Data terdiri dari 666 kota menarik yang tersebar di seluruh dunia.

f. 2392 Kota - *PR2392*

PR2392 memiliki data sebanyak 2392 titik, hasil kontribusi dari M. Padberg dan G. Rinaldi. Layout dari titik-titik tersebut dibuat oleh perusahaan Tektronics.

g. 7397 Kota - *PLA7397*

PLA7397 adalah sebuah *programmed logic array application*, yang terdiri dari 7397 kota.

h. 15112 Kota - *D15112*

D15112 ini mempunyai data mengenai 15112 kota-kota yang terdapat di negara Jerman.

i. 24978 Kota – *Sweded24978*

24978 kota di Swedia, datanya didapat dari database nama-nama fitur geografi *National Imagery and Mapping Agency*.

2.2.3 Konsep Travelling Salesman Problem

Travelling Salesman Problem (TSP) dikenal sebagai salah satu permasalahan optimasi klasik yang berat untuk dipecahkan secara konvensional (Puspitorini, 2008). TSP dapat diilustrasikan sebagai perjalanan sebagai perjalanan seorang *salesman* yang harus melalui semua kota yang dituju dengan jarak terpendek, dimana setiap kota hanya boleh dilalui satu kali. Solusi dari TSP adalah jalur yang dilalui oleh *salesman* tersebut. Tentunya solusi terbaik atau optimal dari permasalahan ini ialah jalur dengan jarak terpendek atau dapat disebut juga dengan rute perjalanan minimum. Dalam kasus TSP simetris terdapat asumsi bahwa jarak 2 kota A dan B sama dengan jarak kota B ke A.

Konsep TSP banyak diterapkan untuk penyelesaian banyak diterapkan untuk penyelesaian berbagai masalah, antara lain untuk menentukan rute perjalanan seminimum mungkin dengan melewati semua kota, menentukan rute robot dalam menelusuri *track* yang panjang dan memiliki percabangan, membantu pengoptimalan proses perjalanan pesan (*message*) pada jaringan interkoneksi komputer dan lain-lain (Ningtyas, Vina dan Ernastuti, 2008). Metode optimasi yang lain telah dicoba untuk memecahkan TSP seperti *Nearest Neighbor*, *Algoritma Greedy*, *Nearest Insertion*, *Farthest Insertion*, *Double Spanning Tree*, *Trip*, *Space Filling Curve*, *Algoritma Karp*, *Like*, *Christofields* dan lain sebagainya (Amin A.R, Ikhsan M dan Wibisono L, 2008).

2.2.4 Aplikasi Permasalahan TSP

Ada beberapa aplikasi pemecahan permasalahan TSP (Amin A.R, Ikhsan M dan Wibisono L, 2008) yang telah dilakukan sebelumnya yaitu :

1. *Genome Sequencing*

Pusat penelitian di The National Institute of Health menggunakan *Concorde's TSP solver* untuk membangun peta radiasi hybrid sebagai bagian dari dari proyek *genome sequencing*. TSP akan mengusahakan jalan untuk menyatukan peta lokal kedalam sebuah peta radiasi hybrid untuk sebuah genome. Laporan mengenai proyek ini dapat dilihat dari makalah “*A Fast and Scalable Radiation Hybrid Map Construction and Integration Strategy*” yang ditulis oleh R. Agarwala, D.L. Applegate, D. Maglott, G.D. Schuler, and A.A. Schaffler.

Aplikasi TSP ini telah diadaptasikan oleh orang-orang Perancis dalam menuliskan Peta genome dari tikus. Proyek ini dideskripsikan pada “*A Radiation Hybrid Transcript Map of the Mouse Genome*”, *Nature Genetics* 29 (2001), pages 194—200.

2. *Starlight Interferometer Program*

Tim insinyur dari *Hernandez Engineering* di Houston dan dari Universitas Brigham Young, telah melakukan sebuah eksperimen menggunakan *chained linkernighan* untuk mengoptimasi penggambaran urutan dari objek-objek luar angkasa. Hal ini dapat kita lihat dalam proposal NASA *Starlight Space Interferometer Program*. Tujuan dari studi ini adalah meminimalisasikan penggunaan bahan bakar dalam pentargetan dan penggambaran dari manuver sepasang satelit yang diterjunkan dalam sebuah misi (kota-kota dalam TSP digambarkan sebagai objek-objek luar angkasa, dan biaya perjalanan antara satu kota dengan lainnya dianalogikan dengan kebutuhan bahan bakar yang dibutuhkan satelit). Laporan mengenai proyek ini dapat dibaca dalam makalah "*Fuel Saving Strategies for Separated Spacecraft Interferometry*".

3. *Scan Chain Optimization*

Sebuah manufaktur dalam semi konduktor menggunakan *concorde TSP solver* untuk diimplementasikan dalam *chained linkernighan heuristic* dalam eksperimen untuk mengoptimasi *scanning* rantai dari IC (*Integrated Circuit*). *Scan chains* adalah sebuah perutean yang bisa digunakan untuk tujuan pengetesan sebuah chip dan juga berguna untuk meminimalisasi baik dari segi waktu maupun tenaga.

2.3 Algoritma Genetika (AG)

Algoritma genetika (AG) merupakan salah satu algoritma pencarian terstruktur yang didasarkan pada analogi mekanisme seleksi dan informasi genetika alami. Dalam penggunaannya, algoritma genetika meniru beberapa proses yang ditemukan pada evolusi alamiah (Paputungan I.V, 2004).

Pada penelitian terdahulu (Nugraha I, 2008) telah mengaplikasikan AG untuk optimasi penjadwalan kegiatan belajar mengajar. Sejak Algoritma Genetika (AG) pertama kali dirintis oleh John Holland dari Universitas Michigan pada tahun 1960-an, AG telah diaplikasikan secara luas pada berbagai bidang. AG banyak digunakan untuk memecahkan masalah optimasi, walaupun pada kenyataannya juga memiliki kemampuan yang baik untuk masalah-masalah selain optimasi. John Holland menyatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun bukan) dapat diformulasikan dalam terminologi genetika. AG adalah simulasi dari proses evolusi Darwin dan operasi genetika atas kromosom (Sanjoyo, 2006).

2.3.1 Struktur Umum Algoritma Genetika

Pada AG, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang dikenal dengan populasi. Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan generasi. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi *fitness*. Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas dari kromosom dalam populasi tersebut. Generasi berikutnya dikenal dengan istilah anak (*offspring*) terbentuk dari gabungan

dua kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*). Selain operator penyilangan, suatu kromosom juga dapat dimodifikasi dengan menggunakan operator mutasi. Populasi generasi yang baru dibentuk dengan cara menyeleksi nilai *fitness* dari kromosom induk (*parent*) dan nilai *fitness* dari kromosom anak (*offspring*), serta menolak kromosom-kromosom yang lainnya sehingga ukuran populasi (jumlah kromosom dalam suatu populasi) konstan. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom yang terbaik (Sri Kusumadewi, 2003).

2.3.2 Komponen-Komponen Utama Algoritma Genetika

Ada 6 komponen utama algoritma genetika, yaitu (Sri Kusumadewi, 2003):

1. Teknik Penyandian

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

Demikian juga kromosom dapat direpresentasikan dengan menggunakan :

- *String bit* : 10011, 01101, 11101, dst.
- Pohon (*tree*) : $(*(-(ab))(* (CD)) / (EF)))$
- Bilangan *real* : 65.65, -67.98, 562.88, dst.
- Elemen permutasi : E2, E10, E5, dst.
- Daftar aturan : R1, R2, R3, dst.
- Elemen program : pemrograman genetika
- Struktur lainnya.

2. Prosedur Inisialisasi

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

3. Fungsi Evaluasi

Ada 2 hal yang harus dilakukan dalam melakukan evaluasi kromosom, yaitu: evaluasi fungsi objektif (fungsi tujuan) dan konversi fungsi objektif ke dalam fungsi *fitness*. Secara umum, fungsi *fitness* diturunkan dari fungsi objektif dengan nilai yang tidak negatif. Apabila ternyata fungsi objektif memiliki nilai negatif, maka perlu ditambahkan suatu konstanta C agar nilai *fitness* yang terbentuk menjadi tidak negatif.

4. Seleksi

Seleksi ini bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling *fit*. Ada beberapa metode seleksi dari induk, antara lain:

- *Roulette wheel selection.*
- *Rank-based fitness assignment.*
- *Stochastic universal sampling.*
- *Local selection.*
- *Truncation selection.*
- *Tournament selection.*

5. Operator Genetika

Operator untuk melakukan rekombinasi, yang terdiri dari:

- Rekombinasi bernilai *real*.
 - Rekombinasi diskret.
 - Rekombinasi *intermediate* (menengah).
 - Rekombinasi garis.
 - Rekombinasi garis yang diperluas.
- Rekombinasi bernilai biner (*crossover*).
 - *Crossover* satu titik.
 - *Crossover* banyak titik.
 - *Crossover* seragam.
- *Crossover* dengan permutasi.
 - mutasi.
 - Mutasi bernilai *real*.
 - Mutasi bernilai biner.

6. Penentuan Parameter

Yang disebut dengan parameter disini adalah parameter kontrol algoritma genetika, yaitu: ukuran populasi (*popsiz*e), peluang *crossover* (P_c), dan peluang mutasi (P_m).

Misalkan $P(\text{generasi})$ adalah populasi dari satu generasi, maka secara sederhana algoritma genetika terdiri dari langkah-langkah :

- Generasi = 0 (generasi awal).
- Inisialisasi populasi awal, $P(\text{generasi})$, secara acak.
- Evaluasi nilai *fitness* pada setiap individu dalam $P(\text{generasi})$.
- Kerjakan langkah-langkah berikut hingga generasi mencapai maksimum generasi:

- generasi = generasi + 1 (tambah generasi).
- Seleksi populasi tersebut untuk mendapatkan kandidat induk, $P'(\text{generasi})$.
- Lakukan *crossover* pada $P'(\text{generasi})$.
- Lakukan mutasi pada $P'(\text{generasi})$.
- Lakukan evaluasi *fitness* setiap individu pada $P'(\text{generasi})$.
- Bentuk populasi baru: $P(\text{generasi}) = \{P(\text{generasi}-1) \text{ yang survive, } P'(\text{generasi})\}$

2.3.3 Keuntungan Algoritma Genetika

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

1. Ruang masalah sangat besar, kompleks, dan sulit dipahami.
2. Kurang atau bahkan tidak ada pengetahuan yang memadai untuk merepresentasikan masalah ke dalam ruang pencarian yang lebih sempit.
3. Tidak tersedianya analisis matematika yang memadai.
4. Ketika metode-metode konvensional sudah tidak mampu menyelesaikan masalah yang dihadapi.
5. Solusi yang diharapkan tidak harus paling optimal, tetapi cukup bagus atau bisa diterima.
6. Terdapat batasan waktu, misalnya dalam *real time systems* atau sistem waktu nyata.

2.3.4 Aplikasi Algoritma Genetika

Sejak pertama kali dirintis oleh John Holland pada tahun 1960-an, AG telah dipelajari, diteliti dan diaplikasikan secara luas pada berbagai bidang. AG banyak digunakan pada masalah praktis yang berfokus pada pencarian parameter-parameter optimal. Hal ini membuat banyak orang mengira bahwa AG hanya bisa digunakan untuk masalah optimasi. Pada kenyataannya, AG juga memiliki performansi yang bagus untuk masalah-masalah selain optimasi.

AG telah banyak diaplikasikan untuk penyelesaian masalah dan pemodelan dalam bidang teknologi, bisnis, dan *entertainment*, seperti (suyanto, 2005) :

1. Optimasi

AG digunakan untuk optimasi numerik dan optimasi kombinatorial seperti *travelling salesman problem* (TSP), perancangan *integrated circuit* atau IC, *job shop scheduling*, optimasi video, dan suara.

2. Pemrograman Otomatis

AG telah digunakan untuk melalui proses evolusi terhadap program komputer untuk merancang struktur komputasional, seperti *cellular automata* dan *sorting networks*.

3. Machine Learning

AG telah berhasil diaplikasikan untuk memprediksi struktur protein. AG juga telah berhasil diaplikasikan dalam perancangan *neural networks* (jaringan saraf tiruan) untuk melakukan proses evolusi terhadap aturan-aturan pada *learning classifier systems* atau *symbolic production systems*. AG juga digunakan untuk mengontrol robot.

4. Model Ekonomi

AG telah digunakan untuk memodelkan proses-proses inovasi dan pembangunan *bidding strategies*.

5. Model Sistem Imunisasi

AG telah berhasil digunakan untuk memodelkan berbagai aspek pada sistem imunisasi alamiah, termasuk *somatic mutation* selama kehidupan individu dan menemukan keluarga dengan gen ganda (*multi-gene families*) sepanjang waktu evolusi.

6. Model Ekologis

AG telah berhasil digunakan untuk memodelkan fenomena ekologis seperti *host-parasite co-evolutions*, simbiosis dan aliran sumber daya dalam ekologi.

7. Interaksi antara Evolusi dan Belajar

AG telah digunakan untuk mempelajari bagaimana proses belajar suatu individu bisa mempengaruhi proses evaluasi suatu spesies dan sebaliknya.

2.4 Algoritma Genetika dalam TSP

Algoritma Genetik (AG) adalah algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis (Sri Kusumadewi, 2003). Metode ini diperkenalkan pertama oleh John Holland pada tahun 1975. AG merupakan salah satu bagian dari kecerdasan buatan.

Lalena M (1998) telah mengembangkan suatu metode penyelesaian TSP dengan menggunakan Algoritma Genetika (AG). Tujuan TSP adalah memperoleh total panjang lintasan terpendek untuk mengunjungi N kota hanya satu kali dan kembali ke kota awal. Algoritma genetika merupakan suatu model komputasi yang menurunkan kelakuannya dari beberapa mekanisme evolusi alam, seperti seleksi, rekombinasi,

mutasi dan reproduksi. Suatu populasi merupakan representasi abstrak dari kandidat solusi suatu masalah. Untuk menyelesaikan masalah dengan algoritma genetika, maka masalah tersebut harus dimodelkan dahulu melalui pengkodean yang dapat dilakukan komputasi. Jenis pengkodean tergantung masalahnya, dapat berupa barisan bit, permutasi riil, himpunan dan sebagainya. Pemodelan solusi merupakan proses konversi dari *phenotype* menjadi *genotype*. Suatu fungsi *fitness* berfungsi sebagai penentu kandidat solusi (*genome*) yang sedang diproses, sehingga menentukan kelayakan bertahannya *genome* tersebut. Operator-operator dalam algoritma genetika meliputi operasi seleksi, operasi rekombinasi (perkawinan silang), operasi mutasi, serta *update* generasi. Masing-masing operator dapat diterapkan dengan berbagai cara yang berbeda, sehingga algoritma genetika yang dihasilkan juga akan bervariasi (Purnomo I dan Oskar, 2005).

2.5 Penerapan Algoritma Genetika dalam Kasus TSP

TSP merupakan sebuah permasalahan optimasi yang dapat diterapkan pada berbagai kegiatan seperti routing dan penjadwalan produksi. Masalah optimasi TSP telah menjadi standar untuk mencoba algoritma komputational. Pokok masalah dari TSP adalah seorang *salesman* harus mengunjungi sejumlah kota. Semua kota yang ada harus dikunjungi oleh *salesman* tepat satu kali (Lukas, 2005).

Banyak metode dapat dipakai untuk menyelesaikan TSP. Metode yang juga dapat dipakai untuk menyelesaikan masalah TSP adalah algoritma genetika. Algoritma genetika adalah sebuah algoritma yang meniru cara kerja proses genetika pada makhluk hidup, dimana terdapat proses seleksi, rekombinasi dan mutasi untuk mendapatkan kromosom terbaik pada suatu generasi (Lukas, 2005). Algoritma genetika mengkombinasikan antara deretan struktur dengan pertukaran informasi acak

ke bentuk algoritma pencarian dengan beberapa perubahan. Pada setiap generasi, himpunan baru dari deretan individu dibuat berdasarkan kecocokan pada generasi sebelumnya. Satu siklus iterasi algoritma genetika (sering disebut generasi) terdapat dua proses, yakni proses seleksi dan rekombinasi. Proses seleksi adalah proses evaluasi kualitas setiap string di dalam populasi untuk memperoleh peringkat calon solusi. Berdasarkan hasil evaluasi, dipilih string-string yang akan mengalami proses rekombinasi. Proses rekombinasi meliputi proses genetika untuk memperoleh string baru dari pertukaran karakter dari calon-calon string yang didapat pada tahap seleksi. Proses rekombinasi akan menghasilkan string-string baru yang berbeda dibandingkan induknya dan dengan demikian diperoleh domain pencarian yang baru. Cara kerja algoritma genetika sangat sederhana, hanya mencakup proses penduplikasian string-string dan pertukaran bagian-bagian dari string. Meskipun cukup sederhana, tetapi mempunyai kemampuan untuk menyelesaikan persoalan optimasi.

Proses algoritma genetika terdiri dari beberapa langkah (Lukas, 2005), yaitu pengkodean (*encoding*), seleksi (*selection*), persilangan (*crossover*), mutasi (*mutation*), *decoding*. Pertama-tama, proses *encoding* adalah berbentuk *string* yang merupakan representasi dari kromosom pada satu generasi. Proses *selection* menentukan kromosom mana yang tetap tinggal pada generasi selanjutnya. Proses *crossover* akan menghasilkan kromosom baru yang merupakan pengganti dari kromosom yang hilang sehingga total kromosom pada satu generasi berjumlah tetap. Proses *mutation* memungkinkan terjadinya kromosom baru secara *unpredictable*. Proses terakhir adalah *decoding* yaitu mengambil makna dari hasil kromosom terbaik untuk menjawab permasalahannya.

a. Teknik *Encoding*

Proses *encoding* adalah salah satu proses yang sulit dalam algoritma genetika. Hal ini disebabkan karena proses *encoding* untuk setiap permasalahan berbeda-beda karena tidak semua teknik *encoding* cocok untuk semua permasalahan. Proses *encoding* menghasilkan *string* yang kemudian disebut kromosom. *String* terdiri dari sekumpulan bit. Bit ini dikenal dengan gen. Jadi satu kromosom terdiri dari sejumlah gen.

Ada bermacam-macam teknik *encoding* yang dapat dilakukan dalam algoritma genetika. Beberapa teknik-teknik *encoding* antara lain *binary encoding*, *permutation encoding*, *value encoding* serta *tree encoding*. Teknik *encoding* yang digunakan pada TSP adalah *permutation encoding*. Pada *permutation encoding*, kromosom-kromosom adalah kumpulan angka yang mewakili posisi dalam sebuah rangkaian. Pada TSP, kromosom mewakili urutan kota yang akan dikunjungi *salesman*.

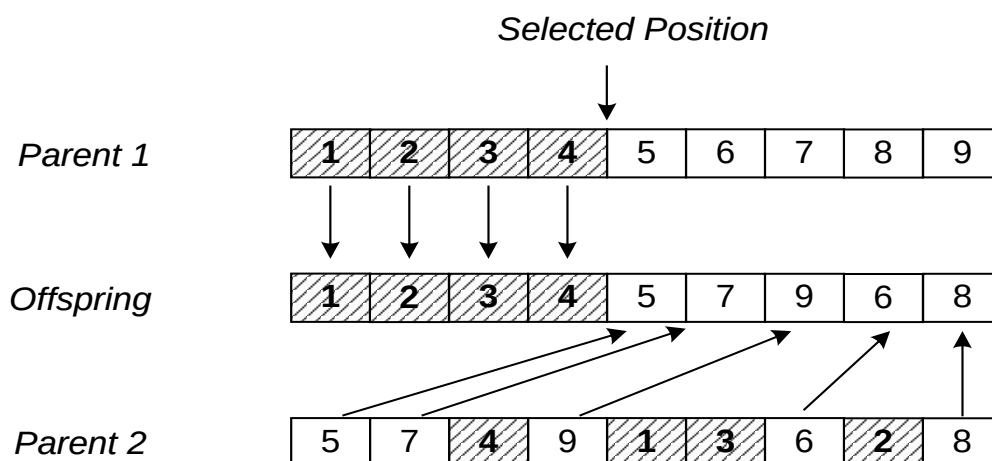
b. Proses Seleksi

Proses seleksi adalah proses yang memegang penting dalam algoritma genetika. Proses seleksi ini digunakan agar hanya kromosom-kromosom yang berkualitas yang dapat melanjutkan peranannya dalam proses algoritma genetika berikutnya. Ada bermacam-macam teknik untuk melakukan proses seleksi, diantaranya adalah *roulette wheel selection*, *rank based selection* dan *steady state selection*. Pada proses seleksi ini digunakan suatu parameter yang disebut kesesuaian atau *fitness*. *Fitness* digunakan untuk menentukan seberapa baik kromosom akan bertahan hidup. Semakin tinggi nilai *fitness* suatu kromosom maka makin baik kromosom itu akan bertahan hidup. Untuk mempertahankan jumlah kromosom tetap pada satu generasi maka perlu dibangkitkan kromosom baru yang merupakan hasil penyilangan dari kromosom hidup. Untuk dilakukan proses rekombinasi.

c. Proses Rekombinasi

Proses rekombinasi atau yang lebih dikenal dengan proses pindah silang (*crossover*) adalah menyilangkan dua kromosom sehingga membentuk kromosom baru yang harapannya lebih baik dari pada induknya. Tidak semua kromosom pada suatu populasi akan mengalami proses rekombinasi. Kemungkinan suatu kromosom mengalami proses rekombinasi didasarkan pada probabilitas *crossover*. Ada beberapa teknik rekombinasi yang dapat digunakan untuk menyelesaikan TSP, antara lain adalah *partially mapped crossover* (PMX), *order crossover* dan *cycle crossover*.

Teknik rekombinasi *order crossover* (OX) diawali dengan membangkitkan dua bilangan acak. Kemudian gen yang berada diantara kedua bilangan acak akan disalin ke *offspring* dengan posisi yang sama. Langkah berikutnya untuk mendapatkan *offspring* pertama adalah mengurutkan gen yang berada pada *parent* 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. Kemudian masukkan urutan yang baru saja didapat pada kromosom anak (*offspring*) dengan cara memasukkan urutan gen pada posisi setelah bilangan acak kedua terlebih dahulu kemudian sisanya dimasukkan pada posisi sebelum bilangan acak pertama. Begitu juga untuk menghasilkan *offspring* kedua.



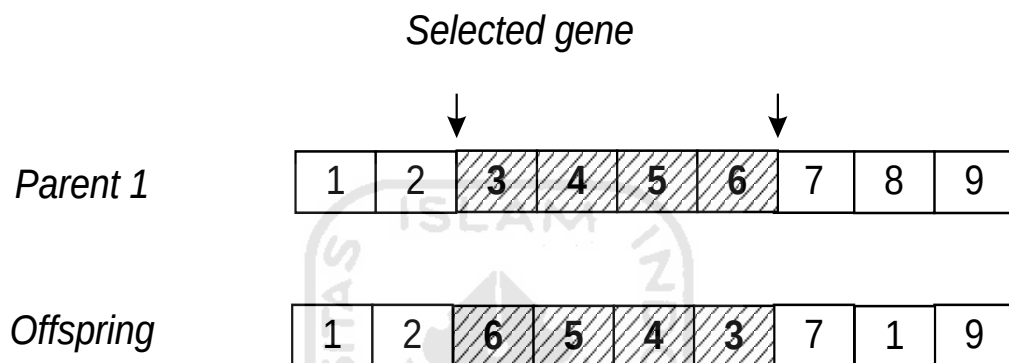
Gambar 2.4 Mekanisme Crossover

d. Proses Mutasi

Proses mutasi ini dilakukan setelah proses rekombinasi dengan cara memilih kromosom yang akan dimutasi secara acak, dan kemudian menentukan titik mutasi pada kromosom tersebut secara acak pula. Banyaknya kromosom yang akan mengalami mutasi dihitung berdasarkan probabilitas mutasi yang telah ditentukan terlebih dahulu. Apabila probabilitas mutasi adalah 100% maka semua kromosom yang ada pada populasi tersebut akan mengalami mutasi. Sebaliknya, jika probabilitas mutasi yang digunakan adalah 0% maka tidak ada kromosom yang mengalami mutasi pada populasi tersebut.

Ada bermacam-macam teknik mutasi yang dapat digunakan untuk menyelesaikan suatu masalah dengan algoritma genetika. Seperti pada teknik rekombinasi, teknik mutasi juga dirancang untuk digunakan untuk pada suatu masalah yang spesifik sehingga tidak setiap teknik mutasi dapat diterapkan pada suatu masalah yang akan diselesaikan. Selain itu, teknik mutasi yang digunakan juga harus sesuai dengan teknik *encoding* yang digunakan untuk menyelesaikan permasalahan tersebut. Beberapa teknik mutasi yang dapat digunakan dalam penyelesaian TSP adalah *invers mutation*,

insertion mutation, dan *reprocal mutation*. *Invers mutation* adalah dengan cara memilih dua posisi gen secara acak dari kromosom dan membalikkan urutan diantara posisi yang telah dipilih sebelumnya. *Insertion mutation* adalah dengan cara memilih nomor dua gen secara acak dan melakukan *swapping* (menukar posisi) secara acak pula ke kromosom tersebut. *Reprocal mutation* adalah dengan cara memilih dua gen secara random dan kemudian mempertukarkan posisinya.



Gambar 2.5 Mekanisme Mutasi

BAB III

METODOLOGI PENELITIAN

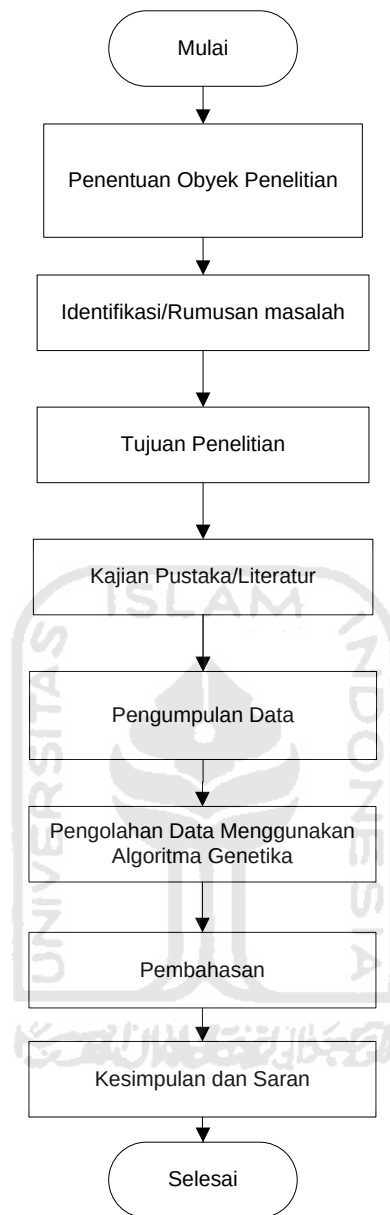
3.1 Obyek dan Tempat Penelitian

Urutan pendistribusian produk dari kota i ke kota $i+1$ dalam suatu rute pendistribusian produk menjadi hal yang sangat mempengaruhi total jarak tempuh. Penelitian dilakukan pada PT. Tigaraksa Satria Tbk, Yogyakarta. Perusahaan ini adalah perusahaan distribusi yang mendistribusikan produk berdasarkan permintaan konsumen (*Costumer Product*) sehingga jenis produk yang didistribusikan beragam disesuaikan dengan permintaan konsumen. Pada penelitian ini, yang menjadi objek penelitian adalah total panjang lintasan rute pendistribusian produk pada kasus *Travelling Salesman Problem*.

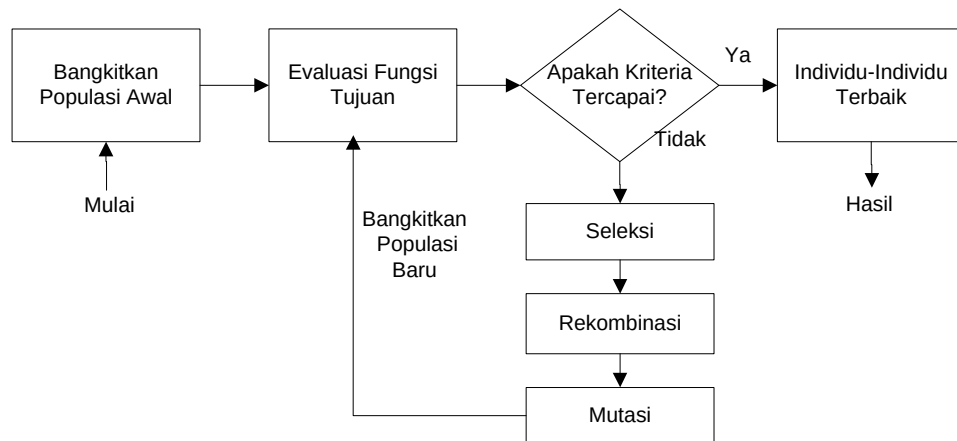
3.2 Kajian Literatur

Tahapan ini menegaskan bahwa untuk memperoleh kajian pustaka dalam memecahkan masalah yang ada, dan bekerja sebagai sistem dasar yang berhubungan dengan teori dan metodologi. Kajian pustaka merupakan penjelasan tentang teori ilmu pengetahuan untuk memecahkan masalah yang ada pada perusahaan, dan media untuk mendukung penelitian ini adalah buku-buku dan jurnal-jurnal.

3.3 Diagram Metodologi Penelitian



Gambar 3.1 Diagram Metodologi Penelitian



Input data : jumlah agen, titik koordinat posisi setiap agen, nilai-nilai parameter genetika

Output/ hasil : *fitness* terbaik, total panjang lintasan rute pendistribusian produk, rute baru

Gambar 3.2 Diagram Pengolahan Data Menggunakan Algoritma Genetika

3.4 Pengumpulan Data

Adapun data yang dibutuhkan dalam penelitian ini dapat dikumpulkan dengan beberapa cara, sebagai berikut :

1. Data Primer

Data primer adalah data-data yang dikumpulkan dan diperoleh melalui observasi langsung dengan karyawan yang ada pada PT. Tigaraksa Satria Tbk, wawancara guna menelaah lebih dalam tentang penelitian yang dilakukan untuk memenuhi data yang dibutuhkan dalam penelitian seperti :

- a. Data jumlah, nama dan alamat lokasi agen.
- b. Data nama dan alamat agen.
- c. Data jarak dari gudang ke agen dan jarak antar agen

- d. Jumlah armada/ kendaraan yang digunakan dalam penelitian.
- e. Jenis dan kapasitas kendaraan yang digunakan.

2. Data Sekunder

Data sekunder adalah data yang diperoleh langsung dari catatan/ berkas yang ada di perusahaan, seperti data sejarah perusahaan.

3.5 Pengolahan Data

Pada tahap ini menerangkan tentang tahapan-tahapan untuk menyelesaikan pokok permasalahan dalam penelitian. Setelah mengumpulkan informasi yang terkait, data diproses dengan menggunakan metode Algoritma Genetika. Langkah-langkah pengolahan data dengan algoritma genetika :

1. Membangkitkan Populasi Awal dengan Inisialisasi Populasi

Tahap ini dilakukan dengan membangkitkan kromosom sejumlah ukuran populasi (*popsize*) yang telah ditentukan dengan representasi kromosom model integer. Dalam kasus ini, nama-nama agen disimbolkan dengan angka yang berbasis pada urutan rute pendistribusian produk yang sudah ada.

2. Evaluasi Fungsi Tujuan

Setelah sejumlah kromosom dibangkitkan maka dapat dihitung nilai *fitness* dari tiap kromosom yang dibangkitkan. Dalam kasus ini, fungsi evaluasi setiap kromosom yang digunakan adalah :

$$F_k = 1/TB$$

$$k = 1, 2, 3, \dots, \text{pop size}$$

$$TB = \text{total panjang lintasan 1 rute}$$

3. Seleksi

Dalam tahap ini akan dilakukan seleksi pemilihan kromosom-kromosom induk yang akan mengalami proses genetika dalam tahap –tahap selanjutnya. Metode pemilihan kromosom induk yang digunakan dalam penelitian ini adalah metode pemilihan induk dengan seleksi *roulette wheel parent selection* dengan cara mencari *fitness* relatif dan *fitness* kumulatif setiap kromosom.

4. Rekombinasi

Melakukan operasi-operasi genetika untuk rekombinasi yang dikenakan pada kromosom-kromosom induk dengan operasi persilangan (*crossover*). Metode persilangan yang digunakan dalam penelitian ini adalah *order crossover*. Konsep dari metode ini adalah menentukan interval titik potong kromosom induk pertama secara random, kemudian meletakkan posisi gen yang ada pada interval tersebut pada kromosom anak seperti posisi induk pertama dan mengisi ruangan kosong dalam kromosom anak dengan gen-gen dari induk kedua yang belum ada pada ruangan gen anak yang sudah terisi dari induk pertama. Persilangan dilakukan dengan cara menentukan probabilitas persilangan dari *popsiz*e kromosom induk sebesar 0.5 yang artinya maksimal sejumlah $0,5 \times \text{popsize akan mengalami persilangan.$

5. Mutasi

Langkah pertama dalam operasi mutasi adalah menentukan jumlah kromosom induk yang akan terkena mutasi. Metode mutasi yang digunakan dalam penelitian ini adalah *invers mutation*. Mekanisme dari operator mutasi *invers mutation* adalah dengan cara memilih dua posisi gen secara acak dari kromosom dan membalikkan urutan diantara posisi yang telah dipilih sebelumnya. Penentuan jumlah kromosom induk yang diharapkan akan

mengalami mutasi pada gen-nya dilakukan dengan cara menentukan probabilitas mutasi dari *popsiz*e kromosom induk. Dalam penelitian ini, probabilitas mutasi ditentukan sebesar 0.01 yang artinya maksimal sejumlah $0.01 \times \text{popsiz}$ e akan mengalami mutasi.

6. Elitisme

Karena seleksi dilakukan secara random, maka tidak ada jaminan bahwa suatu individu bernilai *fitness* terbaik akan selalu terpilih. Walaupun individu bernilai *fitness* terbaik terpilih, mungkin saja individu tersebut akan rusak selama proses evolusi generasi. Untuk menjaga agar individu bernilai *fitness* terbaik tersebut tidak hilang selama evolusi generasi, maka perlu dibuat satu atau beberapa kopinya.

7. Jika aturan pemberhentian terpenuhi, proses berhenti dan keluarkan kromosom yang paling baik. Apabila aturan pemberhentian tidak terpenuhi maka kembali ke langkah 2.
8. Langkah-langkah AG tersebut diatas diimplementasikan dalam fungsi-fungsi pada *software matlab 7.6*.

3.6 Pembahasan

Dari hasil perhitungan dan pengolahan data yang dilakukan, akan dilakukan pembahasan terhadap proses pembentukan rute dan sistem distribusi baru yang didapat. Untuk selanjutnya dibandingkan dengan rute sebelumnya yang ada pada perusahaan apakah sudah optimal atau belum.

3.7 Kesimpulan dan Saran

Penarikan kesimpulan terhadap kasus yang diselesaikan dilakukan pada tahap akhir dalam penelitian ini setelah dilakukan pembahasan terhadap kasus yang dipecahkan. Penarikan kesimpulan bertujuan untuk menjawab tujuan penelitian yang sudah ditetapkan.

Saran-saran juga dikemukakan untuk memberikan masukan mengenai penyelesaian kasus yang dihadapi pada sistem yang diteliti.



BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Profil Perusahaan

4.1.1 Sejarah Singkat PT. Tigaraksa Satria Tbk

PT. Tigaraksa Satria Tbk adalah perusahaan yang bergerak di bidang distribusi produk. Sejarah bisnis distribusi Tigaraksa awalnya didirikan pada tahun 1919 sebagai perusahaan perdagangan yang dijalankan sendirian oleh Mr. Widjaja, secara bertahap bisnis keluarga berkembang dan beradaptasi dengan keadaan. Pada tahun 1960, tiga anak laki-laki Mr. Widjaja mengambil alih kendali dan mulai mengimpor produk konsumen sebagai tambahan bisnis utamanya, komoditas ekspor. Tonggak pertama yang menandai transformasi dari sebuah perusahaan perdagangan keluarga dengan berputar menjadi perusahaan penjualan dan bisnis distribusi yang terpisah mulai beroperasi pada tahun 1988. PT. Tigaraksa Satria Tbk yang kemudian pada April tahun 1990 menjadi perusahaan publik dan mencatatkan sahamnya di Bursa Efek Jakarta dan Surabaya di bawah simbol TGKA.

4.1.2 Bisnis Utama

Setelah sekitar 17 tahun pengalaman dalam bidang penjualan dan perusahaan distribusi, PT. Tigaraksa Satria Tbk dikenal sebagai salah satu dari perusahaan penjualan dan distribusi nasional terkemuka di Indonesia. Tigaraksa mendistribusikan berbagai macam produk konsumen yang bergerak cepat termasuk produk susu bayi, makanan dan minuman, perawatan tubuh dan produk perawatan rumah. PT. Tigaraksa Satria Tbk memiliki 15 kantor cabang dan 4 depot pasar terkemuka di seluruh

Indonesia, konsumennya lebih dari 50 merek produk terkenal dan memiliki sekitar 800 karyawan nasional.

4.1.3 Kegiatan Usaha lainnya

Selain bisnis utamanya yaitu penjualan dan distribusi. Tigaraksa juga memiliki satu Divisi *Direct Selling* TOP yang bergerak di bidang penjualan langsung buku-buku pendidikan anak-anak yang berkualitas tinggi. Perseroan juga telah mengembangkan bisnisnya dengan memproduksi berbagai produk merk sendiri yang berkualitas tinggi serta mendirikan beberapa anak perusahaan yang memproduksi, memasarkan dan mendistribusikan kompor gas serta menyediakan jasa isi ulang gas LPG melalui PT. Blue Gas Indonesia juga anak perusahaan yang bergerak di bidang garmen.

4.1.4 Visi PT. Tigaraksa Satria Tbk

“TO BECOME AN EXCELLENT AND SUCCESFUL MARKET-DRIVEN SALES ORGANIZATION”

PT. Tigaraksa Satria Tbk memiliki visi menjadi perusahaan yang terbaik dan sukses. Yaitu seluruh proses akan dilakukan dengan benar untuk mendapatkan hasil akhir yang melebihi standar. Hal tersebut dapat dilihat dari perbedaan yang kecil dalam suatu proses dan tidak ada kerusakan. Dan target perusahaan adalah memenuhi permintaan untuk menopang perusahaan. Dengan demikian juga akan memberikan keuntungan kepada karyawan, direktur dan para pemegang saham. Karena parameter sukses sebuah perusahaan ditentukan oleh pasar atau pelanggan, seluruh peraturan, kegiatan dan tujuan dari proses internal harus memenuhi kepuasan pelanggan, perusahaan memiliki motto nilai suatu penawaran adalah nilai suatu kepuasan.

4.2. Pengumpulan Data

Pengumpulan data dilakukan dengan observasi langsung ke PT. Tigaraksa Satria Tbk Yogyakarta mengenai sistem distribusi yang dilakukan perusahaan dalam mendistribusikan produk susu SGM ke agen-agen yang tersebar di beberapa wilayah sekitar.

4.2.1 Data Nama Agen dan Alamat Lokasi Agen

Data nama agen dan alamat lokasi merupakan data agen tetap yang menjadi tujuan pendistribusian produk. Dari data alamat lokasi agen maka didapat data titik koordinat setiap agen. Adapun jumlah agen yang dikunjungi ada 60 agen yang terletak di kota Yogyakarta. Berikut adalah tabel nama, alamat dan titik koordinat 60 agen.

Tabel 4.1 Nama Agen/ Toko dan Alamat Lokasi

Nomor	Kode	Nama Agen	Alamat Agen
1	G	Gudang	Jl. Ringroad
2	A1	Lavima 2	Ganjuran
3	A2	Rahmi	Jl. Parangtritis km. 25
4	A3	DH	Kretek
5	A4	Mustika	Kretek
6	A5	Agung apotik	Jl. Ringroad
7	A6	Fatimah	Jl. Jendral sudirman, pundong
8	A7	Semi	Pasar bantul
9	A8	Oskar	Sono sewu, Ngestiharjo
10	A9	Utomo	Jl. Raya Bantul
11	A10	Sarjinem	Pasar Bantul
12	A11	Hans	Kasongan
13	A12	Bu midah	Kasongan

14	A13	Bu purwo	Jl. Jogonalan, Tirtonirmolo
15	A14	Tk lestari	Jl. Ringroad Selatan
16	A15	Bu sri	Jl. Bibis, Tamantirto
17	A16	Kios bagus	Jl. Karanggayam, Sitimulyo Piyungan
18	A17	Tk prasojo	Jl. Ringroad Taman Tirto
19	A18	Rangga	Jl. Srandakan
20	A19	Ani	Jl. Raya Srandakan
21	A20	Dipo	Jl. Dr Wahidin, Tirenggo
22	A21	Utomo	Jl. Hasyim Asyari, Mandingan
23	A22	Arsita	Monggalan
24	A23	Paryono	Pasar Celep, Sanden
25	A24	Unik	Pundong
26	A25	Tk bimo	Turi, Bambanglipuro
27	A26	Pojok	Pasar Gamping
28	A27	Mulyo	Pundong
29	A28	Yeye	Kretek
30	A29	Dev mart	Jl. Parangtritis Km. 5
31	A30	Tk toni	Jl. Parangtritis Km 15
32	A31	Hadi	Jl. Ambar Binangun
33	A32	Narti	Kasihani Bantul
34	A33	HR	Jl. A Dahlan
35	A34	Dahlan	Jl. Ibu Ruswo
36	A35	Tk sumber santoso	Jl. Brigjend Katamso
37	A36	Tk noto putro	Jl. Re Martadinata
38	A37	Sumber insani mandiri	Yogyakarta
39	A38	Maju jaya	Jl. Piere Tendean
40	A39	Ridha jaya	Jl. Ali Maksum
41	A40	Ida	Krapyak Lor
42	A41	Apotek diro	Bantul
43	A42	Notojoyo	Jl. Bantul Km 2.5
44	A43	Apotek christella	Jl. Re Martadinata
45	A44	Apotek K24	Jl. Wates

46	A45	Mami	Pasar Gamping
47	A46	HM	Pasar Gamping
48	A47	Tk setia	Jl. Dr Wahidin, Trirenggo
49	A48	Tk c2	Jl. Perum Pelemsewu Sewon
50	A49	Perdana	Jl. Rotowijayan 14
51	A50	Blimbing	Brosot
52	A51	Berkah	Jl. Srandakan
53	A52	Ana	Jl. Wates c
54	A53	Yanto b	Jl. Raya Srandakan
55	A54	Tri wahyuni	Jl. Cokrodipuran
56	A55	Semangat baru	Jl. Gandekan Kidul
57	A56	Waserda	Jl. Ks Tubun
58	A57	Kuncen	Jl. Hos Cokroaminoto
59	A58	69	Jl. Hos Cokroaminoto
60	A59	Linawati	Jl. Suryatmajan
61	A60	Remaja	Jl. A Dahlan

4.2.2 Data Jarak dari Gudang ke Agen dan Jarak Antar Agen

Data matrik jarak antara gudang ke para agen, dan satu agen ke agen lainnya dapat dilihat pada lampiran.

Tabel 4.2 Rute Pendistribusian Produk

Nomor	Rute Pendistribusian
1	G-A1-A2- A3-A4-A5-G
2	G-A6-A7-A8-A9-A10-G
3	G-A11-A12-A13-A14-A15-G
4	G-A16-A17-A18-A19-A20-G
5	G-A21-A22-A23-A24-A25-G
6	G-A26-A27-A28-A29-A30-G
7	G-A31-A32-A33-A34-A35-G
8	G-A36-A37-A38-A39-A40-G
9	G-A41-A42-A43-A44-A45-G
10	G-A46-A47-A48-A49-A50-G
11	G-A51-A52-A53-A54-A55-G
12	G-A56-A57-A58-A59-A60-G

4.2.4 Jumlah dan Kapasitas Kendaraan

Sistem pendistribusian pada PT. Tigaraksa Satria Tbk dalam penelitian ini menggunakan 1 unit kendaraan (mobil box) dengan kapasitas setiap kendaraan yaitu 150 kg. Dalam mendistribusikan produk perusahaan hanya memberikan batasan sudah berada kembali di gudang pada pukul 16.30. Adapun rentang waktu yang dibutuhkan dalam mendistribusikan produk rata-rata 6,5 jam. Jadi produk harus sudah berada di agen yang paling akhir dalam sebuah tur 6,5 jam setelah pemberangkatan dari gudang.

4.3 Pengolahan Data

4.3.1 Perhitungan Total Jarak Tempuh Rute Pendistribusian Awal

1. Menentukan jarak agen -i ke agen -i+1

Dari data rute pendistribusian produk pada tabel 4.2 yang ada diperoleh jarak agen -i ke agen -i+1 seperti pada tabel 4.3 :

Tabel 4.3 Jarak Tempuh Agen i ke Agen i+1

Agen i	Agen i+1	Jarak (Km)
G	1	17,5
1	2	4
2	3	3,3
3	4	0,2
4	5	18,8
5	G	7
Total		50,8

Menghitung total jarak agen -i ke agen-i+1 adalah dengan cara menambahkan jarak agen -i ke agen-i+1

$$17,5 + 4 + 3,3 + 0,2 + 18,8 + 7 = 50,8 \text{ Km}$$

Dan seterusnya sampai diperoleh total jarak tempuh seluruh rute yang ada seperti pada tabel 4.4.

Tabel 4.4 Total Jarak Tempuh Agen -i ke Agen -i+1

Nomor	Rute Pendistribusian	Jarak Tempuh (Km)
1	G-A1-A2- A3-A4-A5-G	50,8
2	G-A6-A7-A8-A9-A10-G	57,3
3	G-A11-A12-A13-A14-A15-G	23,7
4	G-A16-A17-A18-A19-A20-G	60,1
5	G-A21-A22-A23-A24-A25-G	67,8
6	G-A26-A27-A28-A29-A30-G	75,1
7	G-A31-A32-A33-A34-A35-G	22,4
8	G-A36-A37-A38-A39-A40-G	24,4
9	G-A41-A42-A43-A44-A45-G	30,2
10	G-A46-A47-A48-A49-A50-G	78,1
11	G-A51-A52-A53-A54-A55-G	83,9
12	G-A56-A57-A58-A59-A60-G	14,4

2. Menghitung total jarak tempuh dalam rute pendistribusian produk yang dilakukan oleh perusahaan, dengan cara menambahkan jarak tempuh semua rute yang ada pada tabel 4.4 :

$$50,8 + 57,3 + 23,7 + 60,1 + 67,8 + 75,1 + 22,4 + 24,4 + 30,2 + 78,1 + 83,9 + 14,4 = 588,2 \text{ Km.}$$

4.3.2 Membangkitkan Populasi Awal dengan Inisialisasi Populasi

Tahap ini dilakukan dengan membangkitkan kromosom sejumlah ukuran populasi (*pop size*) yang telah ditentukan dengan representasi kromosom model integer. Dalam kasus ini, nama-nama agen disimbolkan dengan angka 1 sampai 60.

Contoh :

Dibangkitkan 3 populasi dengan angka random dan panjang kromosom 5

V1 = [1 4 5 8 9]

V2 = [4 1 8 5 9]

V3 = [8 9 5 4 1]

4.3.3 Menentukan Evaluasi Fungsi

Tujuan dari penyelesaian kasus yang dihadapi adalah meminimasi total panjang lintasan. Konsep yang dipakai untuk menyelesaikan permasalahan yang dihadapi adalah konsep minimasi. Hal ini berlawanan dengan konsep yang dipakai dalam AG. Karena didalam AG, konsep utamanya adalah mencari kromosom yang mempunyai nilai kesesuaian (*fitness value*) yang besar. Dalam penelitian ini fungsi tujuan yang dipakai untuk mencari nilai kesesuaian setiap kromosom dalam AG adalah sebagai berikut :

Sebagai contoh akan dihitung nilai *fitness* kromosom 1, 2, 3 dari tahap inisialisasi diatas sebagai berikut :

$$\begin{aligned}
 \text{Fitness 1} &= G - 1 + 1 - 4 + 4 - 5 + 5 - 8 + 8 - 9 + 9 - G \\
 &= 17,5 + 7,4 + 18,8 + 4 + 4,9 + 8,9 \\
 &= 61,7
 \end{aligned}$$

$$\begin{aligned}
 \text{Fitness 2} &= G - 4 + 4 - 1 + 1 - 8 + 8 - 5 + 5 - 9 + 9 - G \\
 &= 24,2 + 7,4 + 13,7 + 4 + 1,9 + 8,9 \\
 &= 60,1
 \end{aligned}$$

$$\begin{aligned}
 \text{Fitness 3} &= G - 8 + 8 - 9 + 9 - 5 + 5 - 4 + 4 - 1 + 1 - G \\
 &= 9,1 + 4,9 + 1,9 + 18,8 + 24,2 + 17,5 \\
 &= 76,4
 \end{aligned}$$

$$\begin{aligned}
 \text{Total Fitness} &= 61,7 + 60,1 + 76,4 \\
 &= 198,2
 \end{aligned}$$

4.3.4 Tahap Seleksi Kromosom Induk (*Parent Selection*)

Dalam tahap ini akan dilakukan seleksi pemilihan kromosom-kromosom induk yang akan mengalami proses genetika dalam tahap –tahap selanjutnya. Metode pemilihan kromosom induk yang digunakan dalam penelitian ini adalah metode pemilihan induk dengan seleksi roda roulette (*roulette wheel parent selection*).

Langkah pertama dalam pemilihan kromosom yang akan menjadi induk pada generasi selanjutnya adalah dengan melakukan perhitungan *fitness* relatif setiap kromosom, yang dapat dihitung dengan formula :

Menghitung *fitness* relatif

$$P_k = \frac{\text{fitness}}{\text{TotFitness}}$$

$$\begin{aligned}
 P_1 &= \frac{61,7}{198,2} \\
 &= 0,31
 \end{aligned}$$

$$\begin{aligned}
 P_2 &= \frac{60,1}{198,2} \\
 &= 0,30
 \end{aligned}$$

$$P1 = \frac{76,4}{198,2}$$

$$= 0,38$$

Kemudian dilakukan perhitungan *fitness* kumulatif setiap kromosom, yang dapat dihitung dengan formula :

$$q_k = q_{k-1} + P_k$$

$$k = 1, 2, 3, \dots, \text{popsize}$$

jika $k = 1$ maka $q_{k-1} = 0$

Contoh :

$$q_1 = 0 + 0,31$$

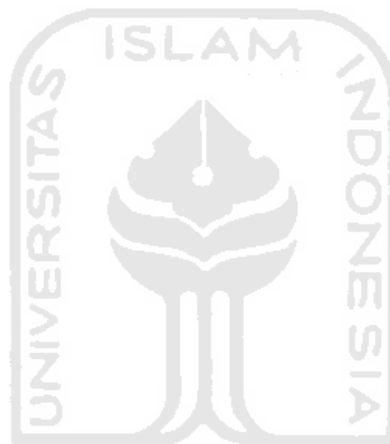
$$= 0,31$$

$$q_2 = 0,31 + 0,30$$

$$= 0,61$$

$$q_3 = 0,61 + 0,38$$

$$= 0,99$$



Dengan cara yang sama, kemudian dihitung *fitness* relatif dan *fitness* kumulatif untuk semua kromosom. Langkah setelah perhitungan *fitness* relatif dan *fitness* kumulatif setiap kromosom adalah melakukan seleksi dengan *roulette wheel* dengan cara membangkitkan bilangan random dari 0 sampai 1 sejumlah *popsize* untuk memilih kromosom. Jika bilangan random lebih kecil dari *fitness* kumulatif maka kromosom dipilih sebagai induk.

Contoh :

Dibangkitkan 3 bilangan acak :

0,52 0,25 0,96

$$q_1' = 0,52 < 0,61$$

$$q2' = 0,25 < 0,31$$

$$q3' = 0,96 < 0,99$$

Sehingga populasi baru terbentuk :

$$V1' = V2$$

$$= [4 \quad 1 \quad 8 \quad 5 \quad 9]$$

$$V2' = V1$$

$$= [1 \quad 4 \quad 5 \quad 9 \quad 9]$$

$$V3' = V3$$

$$= [8 \quad 9 \quad 5 \quad 4 \quad 1]$$

4.3.5 Tahap Rekombinasi

Tahap setelah dilakukan pemilihan kromosom-kromosom induk adalah melakukan operasi genetika untuk rekombinasi yang dikenakan pada kromosom-kromosom induk yaitu dengan operasi persilangan (*crossover*).

Dalam operasi persilangan ini, langkah pertama yang diambil adalah menentukan jumlah kromosom induk yang diharapkan melakukan persilangan. Penentuan jumlah kromosom induk yang diharapkan melakukan persilangan dilakukan dengan cara menentukan probabilitas persilangan dari *popsiz*e kromosom induk. Dalam penelitian ini probabilitas persilangannya dilakukan sebesar 0.5 (Psilang = 0,5), yang artinya maksimal sejumlah 0,5 x *popsiz*e akan mengalami persilangan.

Metode persilangan yang digunakan dalam penelitian ini adalah *order crossover* karena permasalahan dalam kasus TSP adalah bagaimana melakukan pemilihan rute urutan kota pendistribusian produk. Konsep dari metode ini adalah menentukan interval titik potong kromosom induk pertama secara random, kemudian meletakkan posisi gen yang ada pada interval tersebut pada kromosom anak seperti

posisi induk pertama dan mengisi ruangan kosong dalam kromosom anak dengan gen-gen dari induk kedua yang belum ada pada ruangan gen anak yang sudah terisi dari induk pertama. Pindah silang dengan *order crossover* dapat diimplementasikan pada fungsi TSPPindahSilang.m.

Contoh :

Psilang = 0,5

Bangkitkan bilangan acak sebanyak jumlah populasi yaitu 3 kali

0,13 0,03 0,67

Induk yang terseleksi untuk dilakukan *crossover* :

V1' dan V2'

Menentukan posisi *crossover* dengan membangkitkan bilangan acak, misalnya didapat bilangan acak adalah 2 maka maka posisi *crossover* ada di bit 2:

V1' = [4 1 8 5 9]

V2' = [1 4 5 8 9]

Hasil *crossover* :

V1'' = [4 1 5 8 9]

V2'' = [1 4 8 5 9]

4.3.6 Mutasi

Langkah pertama dalam operasi mutasi adalah menentukan jumlah kromosom induk yang akan terkena mutasi. Penentuan jumlah kromosom induk yang diharapkan akan mengalami mutasi pada gen-nya dilakukan dengan cara menentukan probabilitas mutasi dari *popsiz*e kromosom induk. Dalam penelitian ini, probabilitas mutasi ditentukan sebesar 0.01 ($P_{mutasi} = 0.01$), yang artinya maksimal sejumlah $0.01 \times \text{popsize akan mengalami mutasi.$

Mutasi membutuhkan satu kromosom induk untuk menghasilkan kromosom anak. Mekanisme dari operator mutasi *random invers* adalah cara memilih dua posisi gen secara acak dari kromosom dan membalik urutan diantara posisi yang telah dipilih sebelumnya. Mutasi dapat diimplementasikan dengan fungsi TSPMutasi.m. Hasil dari fungsi ini adalah Mutkrom, yaitu kromosom yang sudah termutasi.

Contoh :

Bangkitkan bilangan random sejumlah ukuran populasi dikali panjang kromosom misalnya $3 \times 5 = 15$, bilangan yang lebih besar dari Pmutasi tidak terkena mutasi.

Pmutasi = 0,01

0,008 0,003 0,031 0,711 0,563

0,298 0,002 0,498 0,007 0,287

0,754 0,132 0,906 0,465 0,397

Bit-bit pada kromosom yang mengalami mutasi :

Kromosom ke 1, bit ke 1 dan ke 2

Kromosom ke 2, bit ke 2 dan ke 4

Didapat kromosom baru :

V1" = [4 1 5 8 9]

= [1 4 5 8 9]

V2" = [1 4 8 5 9]

= [1 5 8 4 9]

Setelah dilakukan mutasi, akan terbentuk populasi baru yang juga berjumlah 3 kromosom :

V1" = [1 4 5 8 9]

V2" = [1 5 8 4 9]

V3' = [8 9 5 4 1]

Dengan nilai *fitness* :

$$\begin{aligned}\text{Fitness 1} &= G - 1 + 1 - 4 + 4 - 5 + 5 - 8 + 8 - 9 + 9 - G \\ &= 17,5 + 7,4 + 18,8 + 4 + 4,9 + 8,9 \\ &= 61,7\end{aligned}$$

$$\begin{aligned}\text{Fitness 2} &= G - 1 + 1 - 5 + 5 - 8 + 8 - 4 + 4 - 9 + 9 - G \\ &= 17,5 + 11,6 + 4 + 21,2 + 16,9 + 8,9 \\ &= 80,1\end{aligned}$$

$$\begin{aligned}\text{Fitness 3} &= G - 8 + 8 - 9 + 9 - 5 + 5 - 4 + 4 - 1 + 1 - G \\ &= 9,1 + 4,9 + 1,9 + 18,8 + 24,2 + 17,5 \\ &= 76,4\end{aligned}$$

Kromosom – kromosom pada populasi ini nantinya akan menjadi populasi awal pada generasi kedua. Proses seleksi, rekombinasi (*crossover*), dan mutasi ini dilakukan lagi sampai pada generasi terakhir (MaxGen).

4.3.7 Elitisme

Karena seleksi dilakukan secara random, maka tidak ada jaminan bahwa suatu individu bernilai *fitness* terbaik akan selalu terpilih. Kalaupun individu bernilai *fitness* terbaik terpilih, mungkin saja individu tersebut akan rusak selama proses evolusi. Untuk menjaga agar individu bernilai *fitness* terbaik tersebut tidak hilang selama evolusi, maka perlu dibuat satu atau beberapa kopinya. Prosedur ini dikenal dengan elitisme.

4.3.8 Implementasi TSP dalam Matlab

Implementasi TSP dilakukan dengan tahap-tahap berdasarkan komponen-komponen AG adalah sebagai berikut :

1. Penentuan Parameter

Parameter-parameter yang digunakan adalah

Popsize/ UkPop = 100

Psilang = 0,5

Pmutasi = 0,01

Maksimum generasi/ MaxG = 2000

2. Skema Pengkodean

Skema ini diimplementasikan secara sederhana menggunakan baris-baris perintah pada fungsi TSPInisialisasiPopulasi.m. Input untuk fungsi ini adalah ukuran populasi atau jumlah kromosom dalam populasi dan jumlah gen dalam kromosom yang juga menyatakan jumlah agen.

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

UkPop = Ukuran populasi atau jumlah kromosom dalam populasi

JumGen = Jumlah gen dalam kromosom yang juga menyatakan jumlah kota

Perintah rand (1,JumGen) menyatakan pembangkitan matriks berukuran 1 x JumGen yang berisi bilangan random dalam interval (0,1).

Xval,Ind]=sort(rand(1,JumGen)) menyatakan bilangan-bilangan dalam interval (0,1) hasil pengurutan dari kecil ke besar (*ascending*). Ind menyatakan indeks dari bilangan-bilangan yang dibangkitkan secara random tetapi belum berurutan. Jadi Ind merupakan nomor urut kota yang akan dibangkitkan secara random melalui

perintah sort (rand(1, JumGen)). Populasi(ii,:) menyatakan kromosom ke-ii pada populasi yang jumlah kolomnya sama dengan jumlah gen.

3. Nilai *Fitness*

Penghitungan nilai *fitness* diimplementasikan dalam fungsi TSPEvaluasiIndividu.m. Input untuk fungsi ini adalah koordinat dari semua agen, kromosom dan jumlah gen dalam kromosom yang menyatakan jumlah agen.

```
%TSPEvaluasiIndividu%
function
fitness=TSPEvaluasiIndividu(Kromosom, JumGen, XYkota)
TB=0;
Gudang=[43.4880 913.8180];
for ii=1:JumGen-1
    TB=TB+norm(XYkota(Kromosom(ii),:)-
XYkota(Kromosom(ii+1),:));
end
TB=TB+norm(Gudang-XYkota(Kromosom(1),:));
TB=TB+norm(XYkota(Kromosom(JumGen),:)-Gudang);
fitness=1/TB
```

Variabel XYkota berisi koordinat-koordinat dari semua agen. Variabel TB menyatakan total jarak yang ditempuh, dan nilai *fitness* suatu kromosom dinyatakan oleh $fitness=1/TB$.

4. *Linear Fitness Ranking*

Untuk menghindari kecenderungan konvergen pada optimum lokal, maka dilakukan proses penskalaan nilai *fitness*. Sehingga diperoleh nilai *fitness* baru yang lebih baik, yaitu yang memiliki variansi tinggi. *Linear fitness ranking* dapat diimplementasikan dalam fungsi LinearFitnessRanking.m.

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

Fungsi sort sudah tersedia dalam matlab, digunakan untuk mengurutkan nilai *fitness* dari kecil ke besar (*ascending*). Variabel SF berisi nilai-nilai *fitness* hasil pengurutan, sedangkan IndF berisi indeks dari nilai-nilai *fitness* tersebut.

5. *Roulette Wheel*

Fungsi *roulette wheel* diimplementasikan diimplementasikan dalam fungsi RouletteWheel.m. Hasil dari fungsi ini adalah pindex, yaitu indeks dari individu yang terpilih sebagai orang tua.

```
%RouletteWheel%
function Pindex=RouletteWheel(UkPop,LinearFitness);
JumFitness=sum(LinearFitness);
KumulatifFitness=0;
RN=rand;
ii=1;
while ii<=UkPop,
    KumulatifFitness=KumulatifFitness+LinearFitness(ii);
    if(KumulatifFitness/JumFitness)>RN,
        Pindex=ii;
        break;
    end
    ii=ii+1;
end
```

6. *Pindah Silang (Crossover)*

Pindah silang digambarkan dengan skema *order crossover* dapat diimplementasikan pada fungsi TSPPindahSilang.m.

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

```

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

```

7. Mutasi

Proses mutasi menggunakan skema mutasi *invers mutation*. Mutasi dapat diimplementasikan dengan fungsi TSPMutasi.m. Hasil dari fungsi ini adalah Mutkrom, yaitu kromosom yang sudah termutasi.

```

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

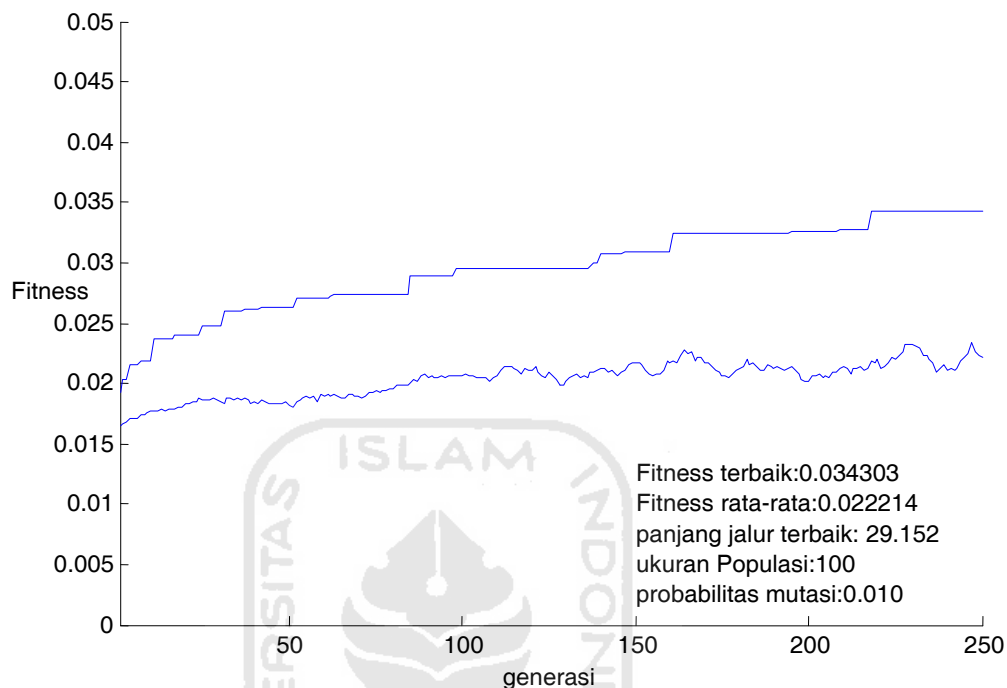
```

8. Program Utama

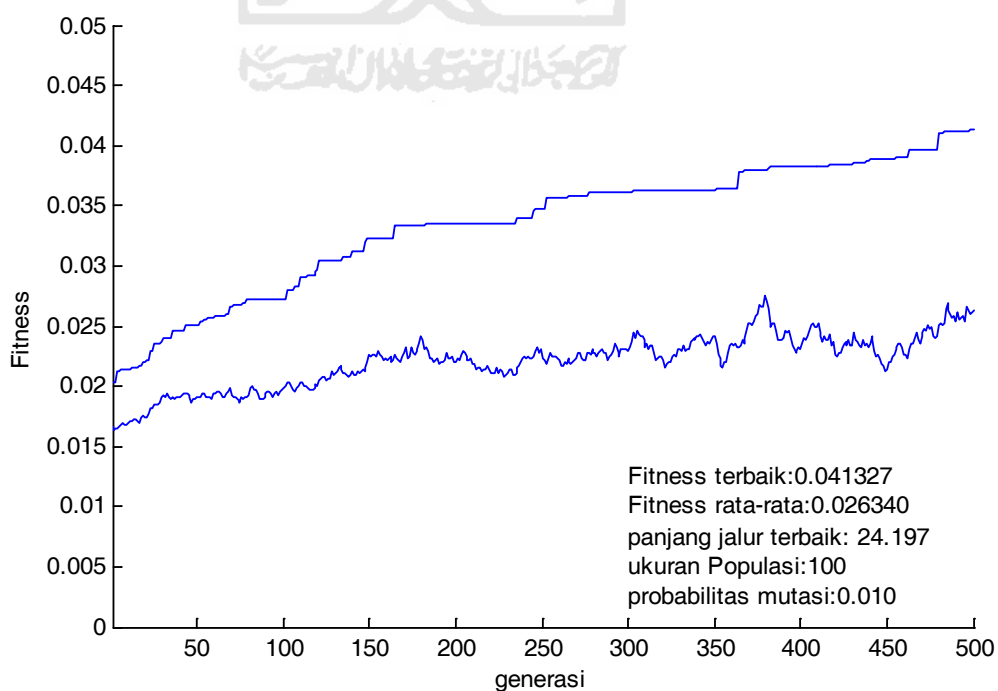
Sebagai program utama, *file* ProgramUtamaYuni.m memanggil semua fungsi diatas. Pada program utama ditambahkan perintah-perintah untuk menampilkan grafik evolusi generasi yang juga menampilkan *fitness* terbaik dan grafik rute baru.

4.3.9 Menjalankan Evolusi per 250 Generasi

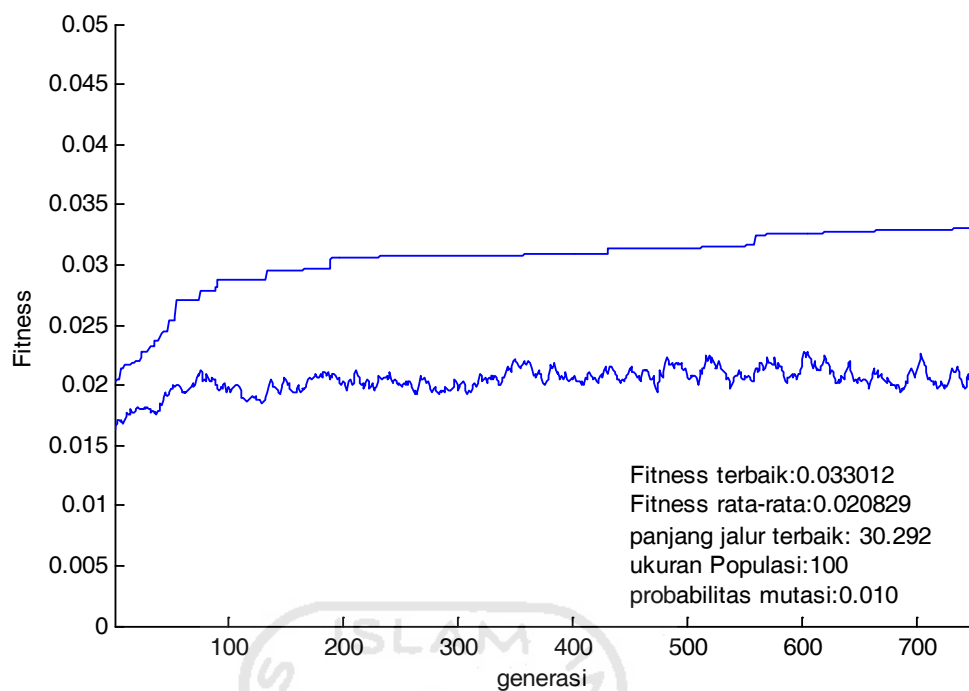
Untuk mendapatkan fitness dan panjang jalur terbaik maka evolusi dijalankan per 250 generasi sampai generasi 2000 dan hasilnya sebagai berikut :



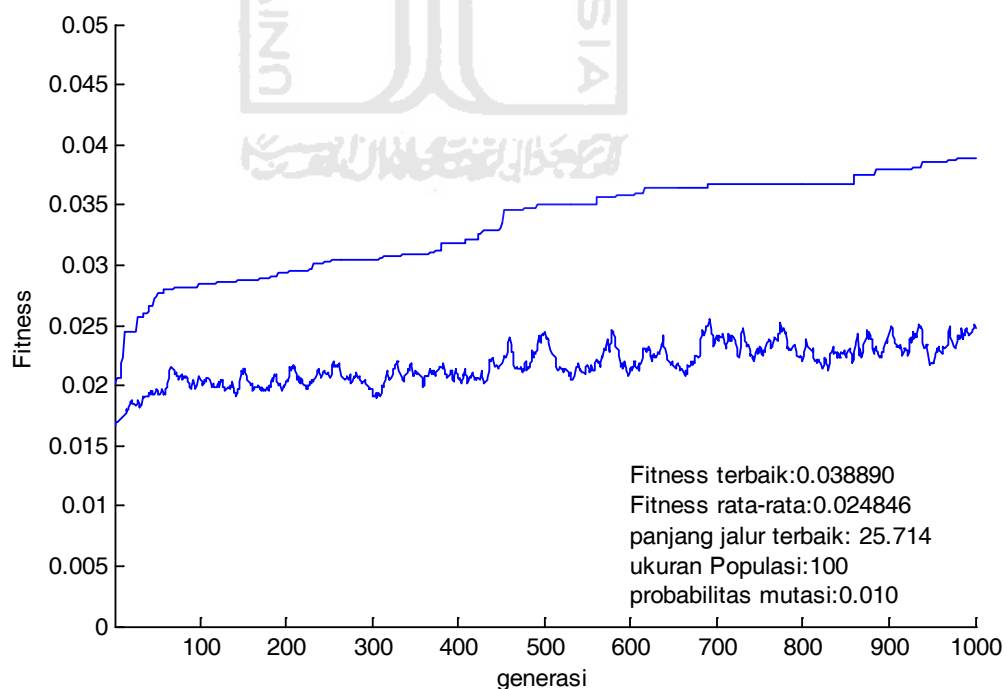
Gambar 4.2 Grafik Evolusi dari Generasi 1 sampai 250



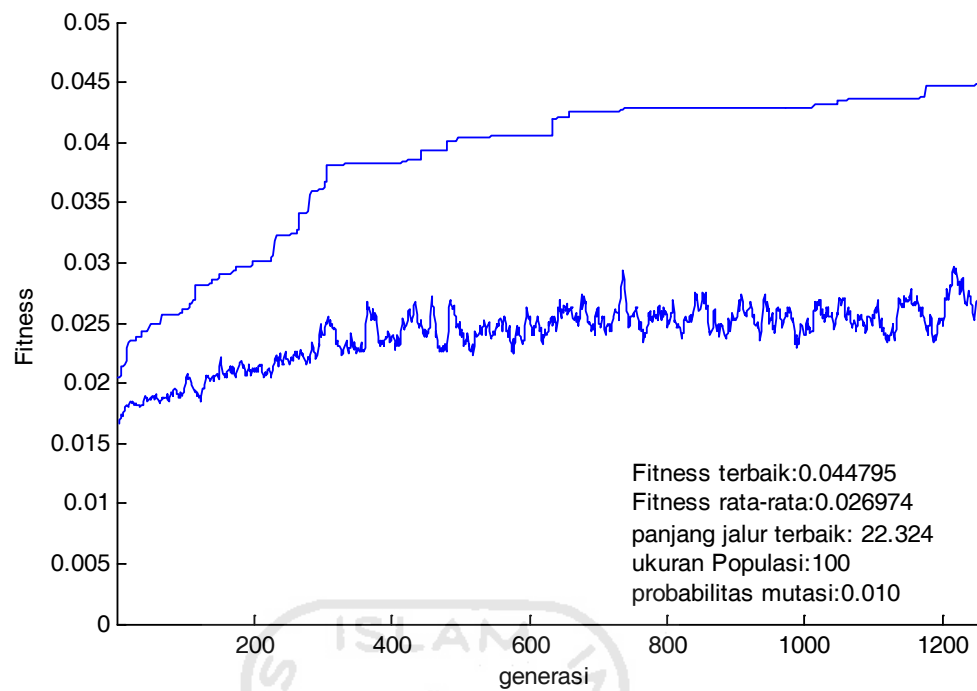
Gambar 4.3 Grafik Evolusi dari Generasi 1 sampai 500



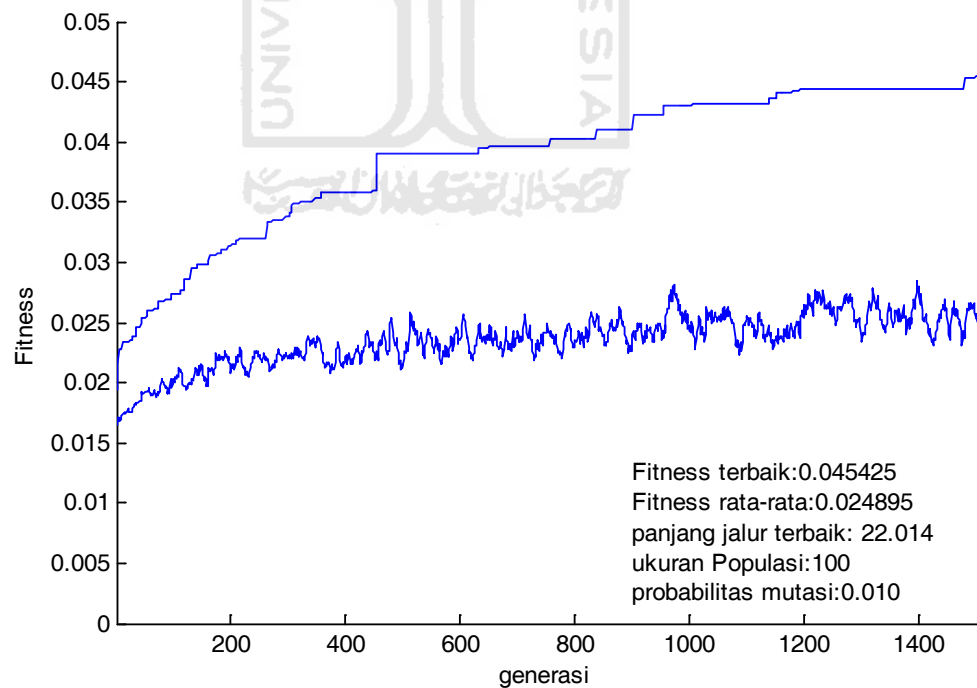
Gambar 4.4 Grafik Evolusi dari Generasi 1 sampai 750



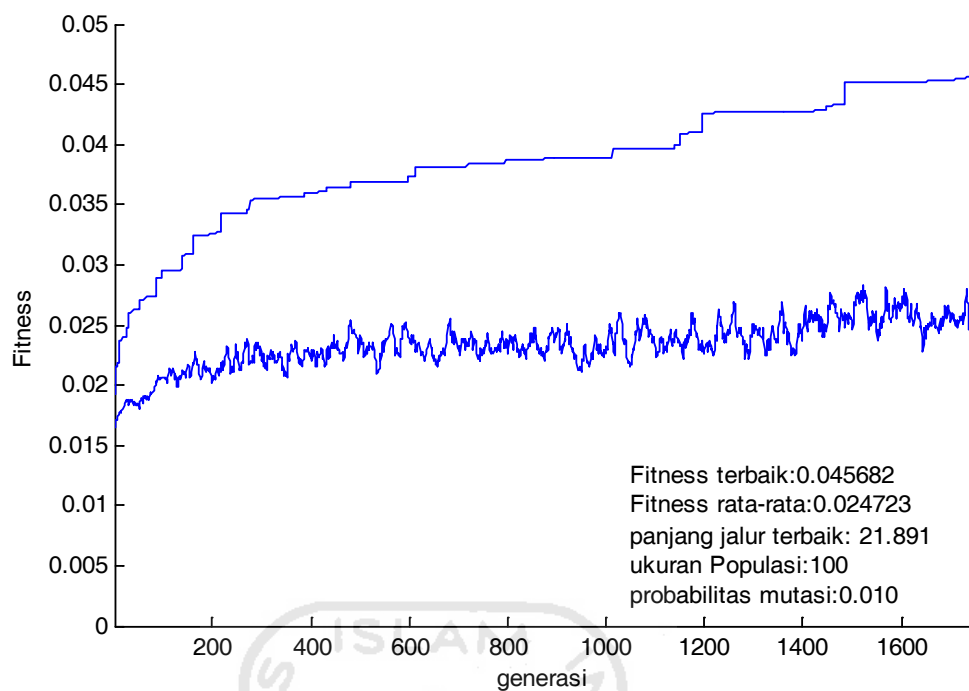
Gambar 4.5 Grafik Evolusi dari Generasi 1 sampai 1000



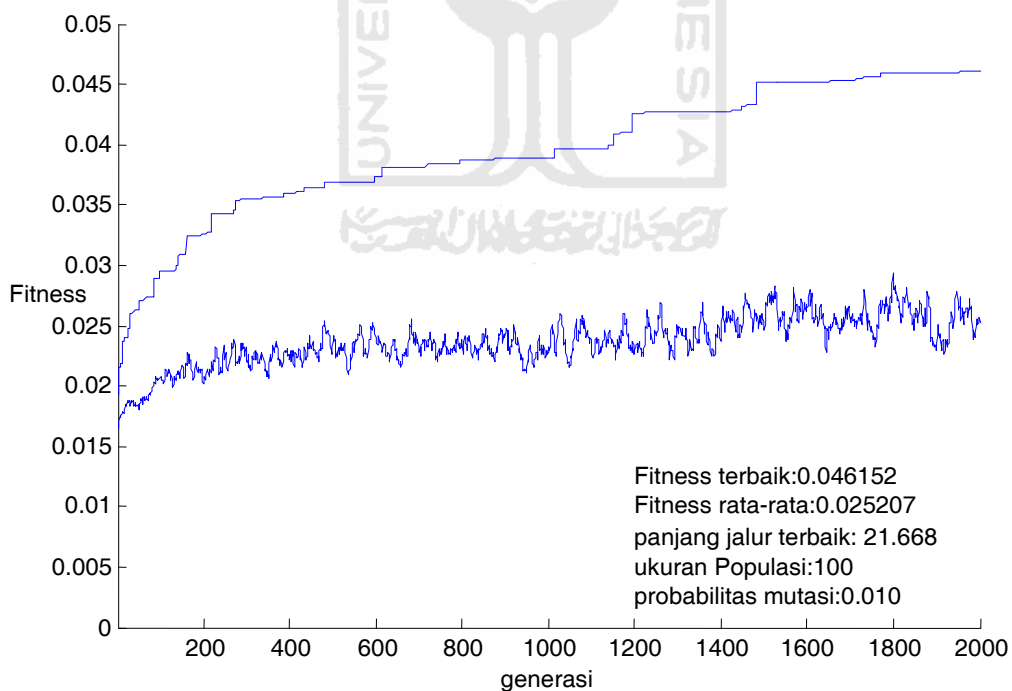
Gambar 4.6 Grafik Evolusi dari Generasi 1 sampai 1250



Gambar 4.7 Grafik Evolusi dari Generasi 1 sampai 1500



Gambar 4.8 Grafik Evolusi dari Generasi 1 sampai 1750



Gambar 4.9 Grafik Evolusi dari Generasi 1 sampai 2000

Dari 8 grafik evolusi diatas maka nilai *fitness* terbaik dan panjang jalur terbaik dapat dilihat dalam tabel berikut :

Tabel 4.5 *Fitness* dan Panjang Jalur Terbaik per 250 Generasi

Generasi	Fitness Terbaik	Panjang Jalur Terbaik
1 – 250	0,034303	29,152
1 – 500	0,041327	24,197
1 – 750	0,033012	30,292
1 – 1000	0,038890	25,714
1 – 1250	0,044795	22,324
1 – 1500	0,045425	22,014
1 – 1750	0,045682	21,891
1 – 2000	0,046152	21,668

Dari tabel diatas dapat dilihat bahwa panjang jalur terbaik adalah 21,668 kartesian. Maka solusi urutan kota dari kasus TSP dengan AG adalah hasil dari evolusi 1 sampai 2000 generasi yang ditunjukkan pada variabel editor Jalur Terbaik.

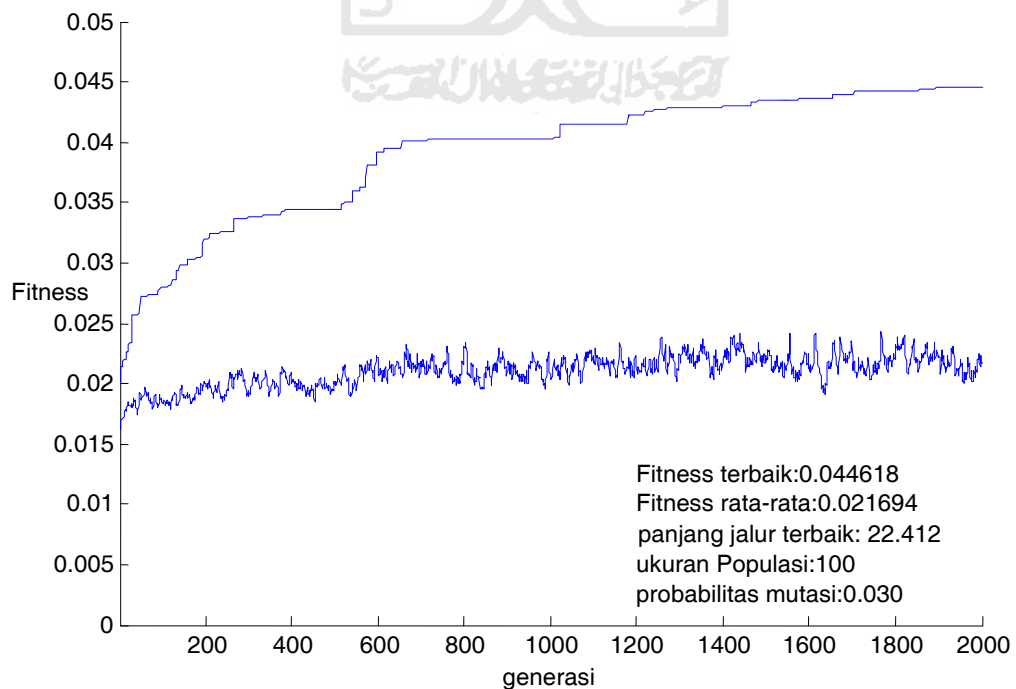
4.3.10 Menjalankan Evolusi dengan Beberapa Parameter

Untuk mendapatkan fitness dan panjang jalur terbaik maka evolusi dijalankan dengan menggunakan beberapa parameter sebagai berikut :

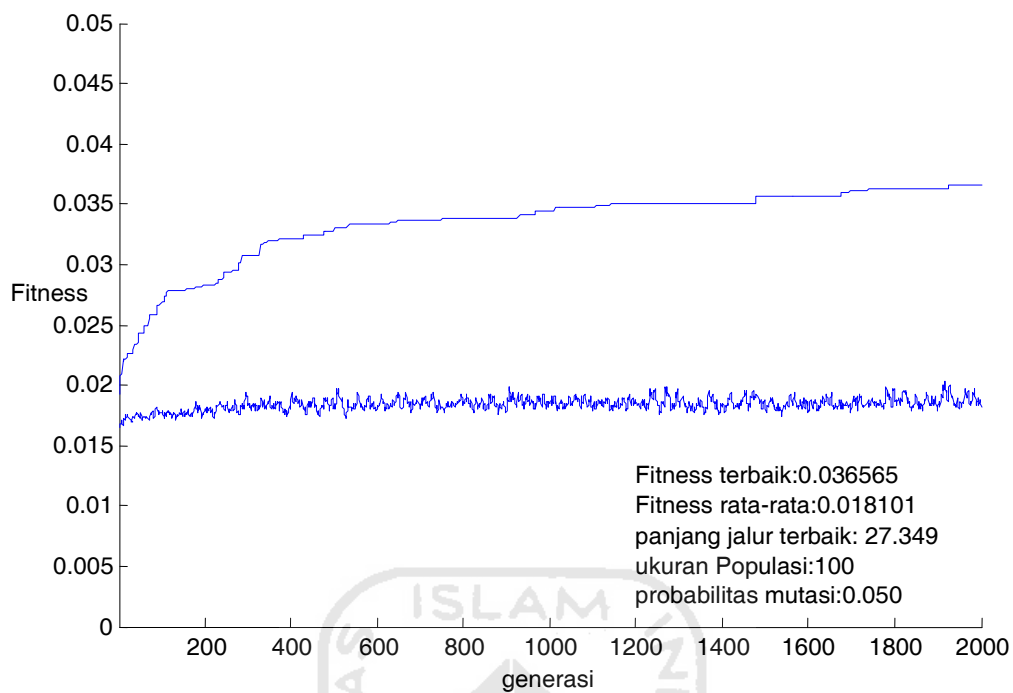
Tabel 4.6 Parameter Evolusi

Parameter	Psilang	Pmutasi
1	0,3	0,03
2	0,5	0,05
3	0,7	0,07
4	0,9	0,09
5	1,0	0,1

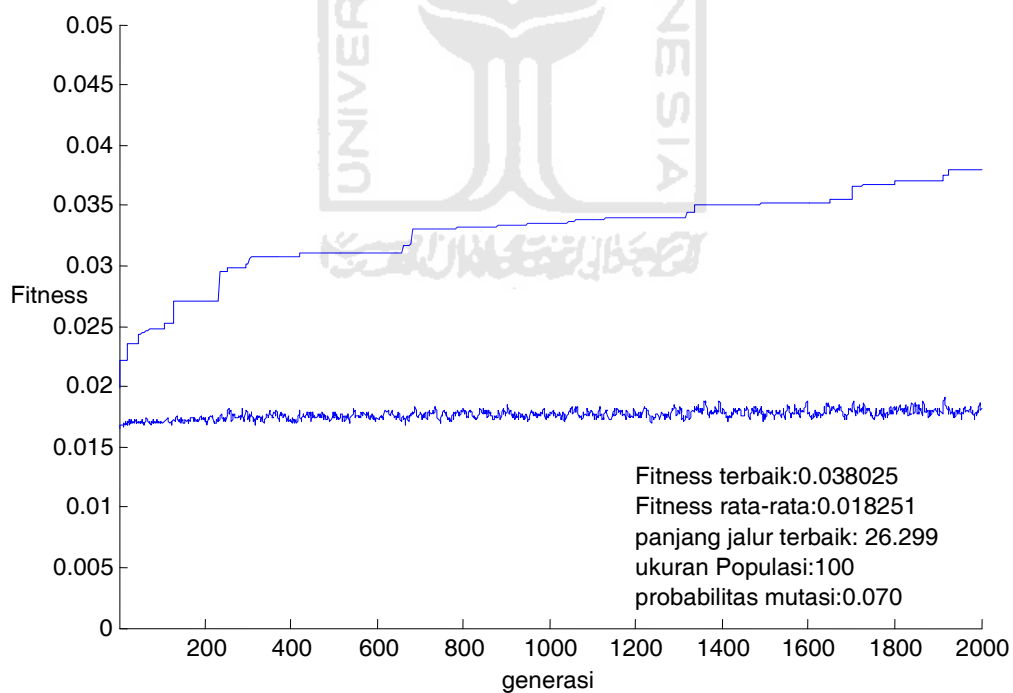
Dengan menggunakan parameter dalam tabel diatas dan evolusi generasi 2000 maka di dapat hasil sebagai berikut :



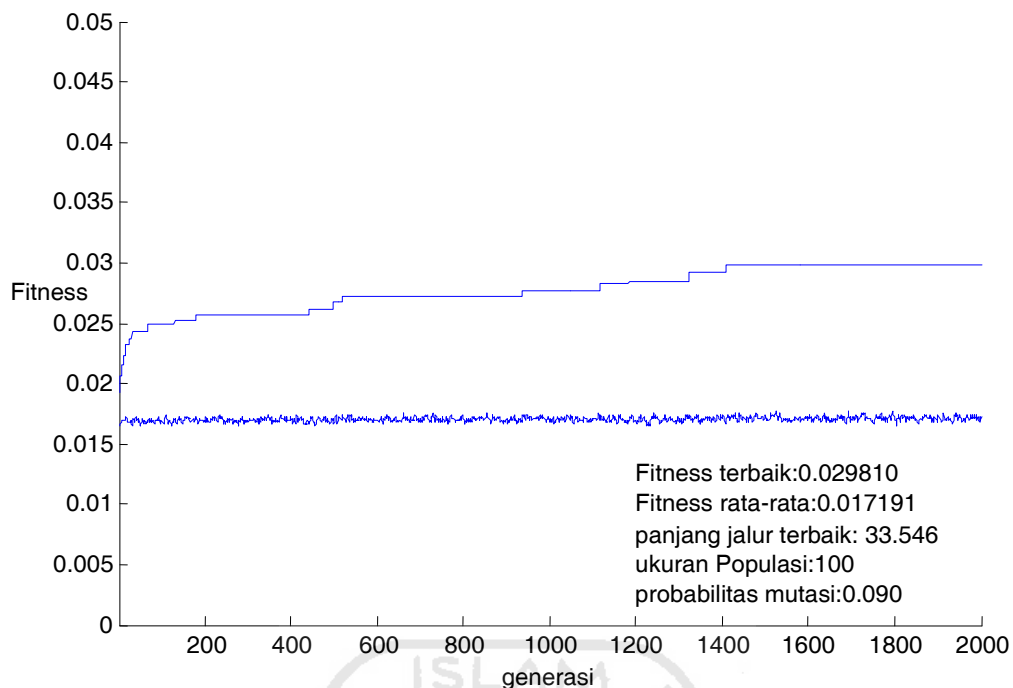
Gambar 4.10 Grafik Evolusi 2000 Generasi dengan Parameter 1



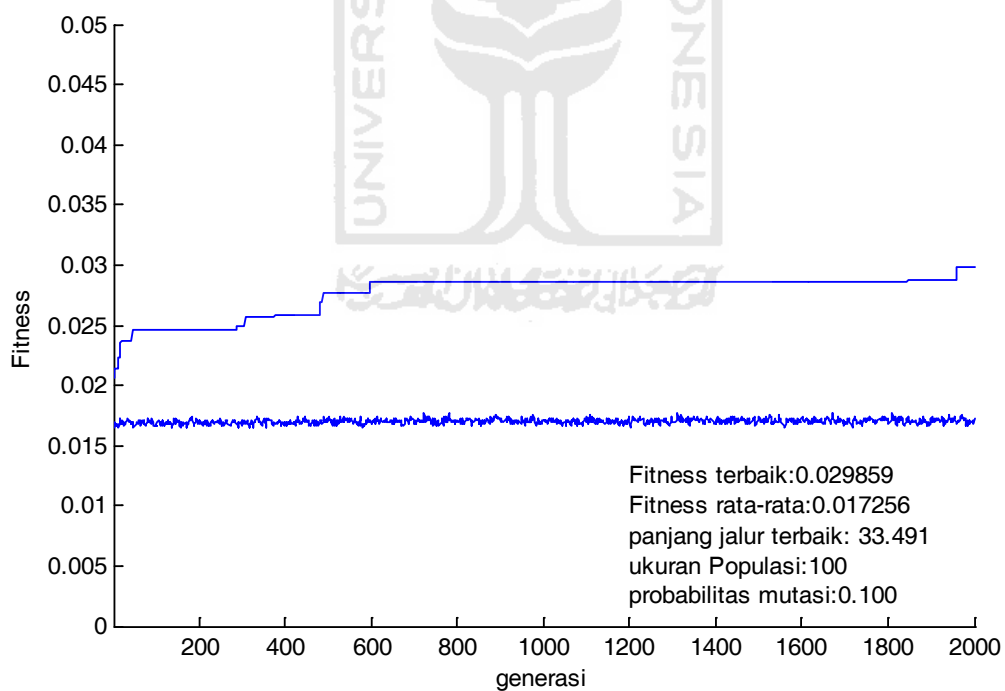
Gambar 4.11 Grafik Evolusi 2000 Generasi dengan Parameter 2



Gambar 4.12 Grafik Evolusi 2000 Generasi dengan Parameter 3



Gambar 4.13 Grafik Evolusi 2000 Generasi dengan Parameter 4



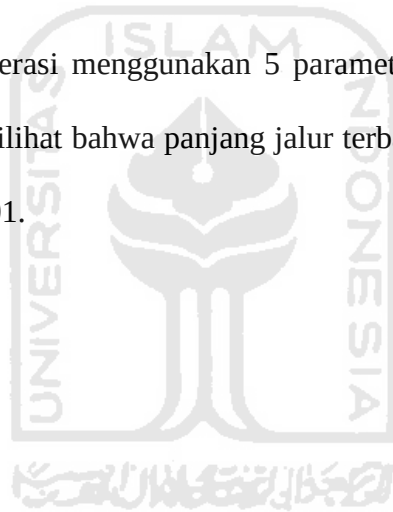
Gambar 4.14 Grafik Evolusi 2000 Generasi dengan Parameter 5

Dari grafik evolusi menggunakan 5 parameter diatas didapat hasil sebagai berikut :

Tabel 4.7 *Fitness* dan Panjang Jalur Terbaik Menggunakan 5 Parameter Evolusi

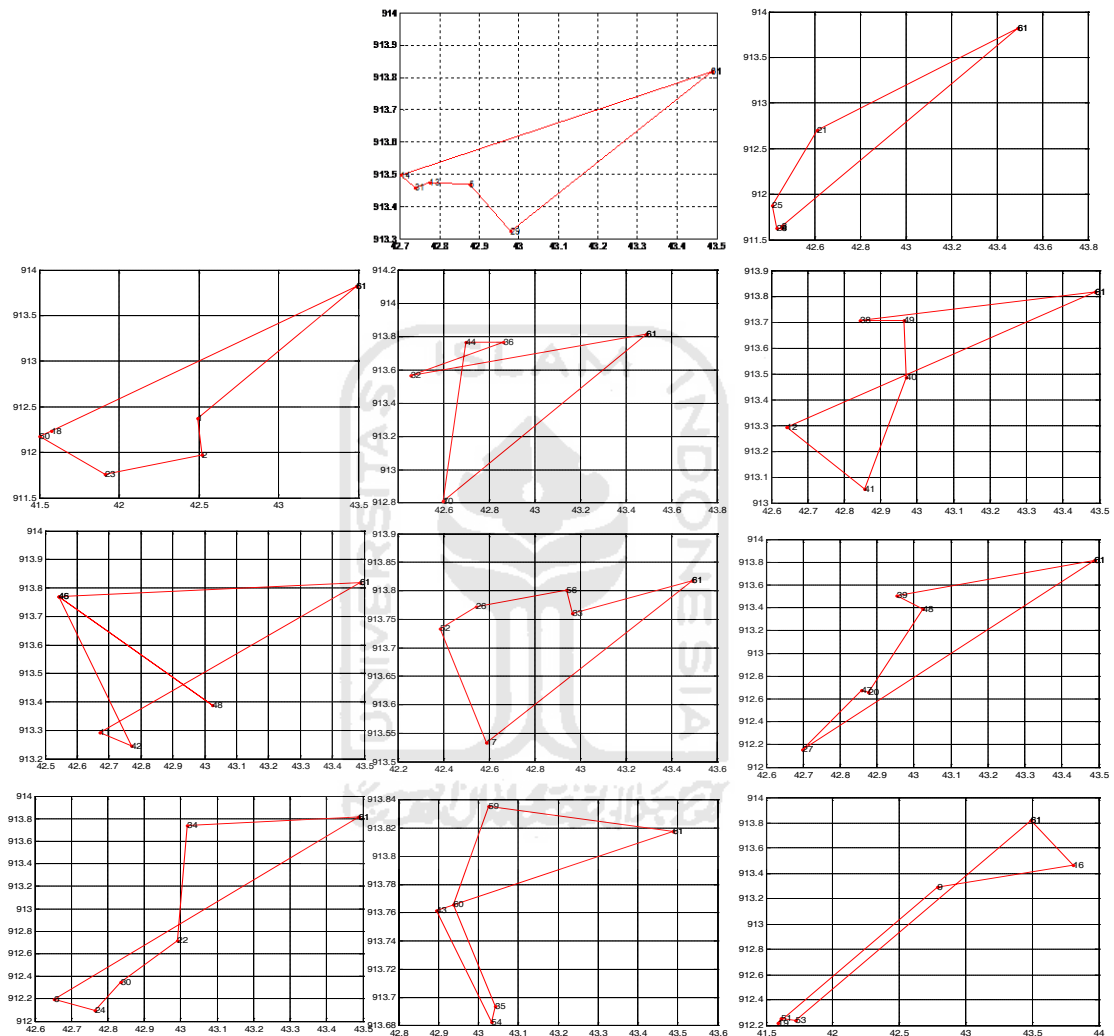
Parameter	Psilang	Pmutasi	<i>Fitness</i> terbaik	Panjang Jalur Terbaik
1	0,3	0,03	0,044618	22,412
2	0,5	0,05	0,036565	27,349
3	0,7	0,07	0,038025	26,299
4	0,9	0,09	0,029810	33,546
5	1,0	0,1	0,029856	33,491

Dari hasil evolusi generasi menggunakan 5 parameter diatas bila dibandingkan dengan gambar 4.9 dapat dilihat bahwa panjang jalur terbaik diperoleh dari parameter Psilang 0,5 dan Pmutasi 0,01.



4.3.11 Rute Baru Pendistribusian Produk

Setelah dilakukan evolusi 1 sampai 2000 dengan menggunakan parameter yang terbaik maka di dapat rute pendistribusian baru yang memiliki total panjang jarak tempuh yang lebih kecil, dengan grafik rute sebagai berikut :



Gambar 4.15 Grafik Rute Pendistribusian Setelah Menggunakan AG

Sehingga perusahaan dapat melakukan perubahan rute pendistribusian produk yang akan dilalui seorang salesman yaitu 5 agen selama 12 hari sebagai berikut :

Tabel 4.8 Rute Baru Pendistribusian Produk

Nomor	Rute Pendistribusian	Panjang Jarak Tempuh (KM)
1	G – A37 – A55 – A58 – A57 – A15 – G	24,4
2	G – A14 – A31 – A13 – A5 – A29 – G	19,2
3	G – A21 – A25 – A28 – A4 – A3 – G	49,4
4	G – A18 – A50 – A23 – A2 – A1 – G	59,5
5	G – A10 – A7 – A44 – A36 – A32 – G	48,05
6	G – A12 – A41 – A40 – A49 – A38 – G	27,3
7	G – A45 – A8 – A46 – A42 – A11 – G	34,9
8	G – A17 – A52 – A26 – A56 – A33 – G	23,5
9	G – A39 – A48 – A20 – A47 – A27 – G	38,9
10	G – A6 – A24 – A30 – A22 – A34 – G	41,2
11	G – A59 – A60 – A35 – A54 – A43 – G	14,7
12	G – A53 – A51 – A19 – A9 – A16 – G	56,2

Total jarak tempuh pendistribusian setelah menggunakan AG adalah :

$$24,4 + 19,2 + 49,4 + 59,5 + 48,05 + 27,3 + 34,9 + 23,5 + 38,9 + 41,2 + 14,7 + 56,2 = 437,25 \text{ Km}$$

BAB V

PEMBAHASAN

5.1 Perbandingan Total Panjang Lintasan Urutan Rute Pendistribusian Produk Awal dan Setelah Diminimasi Menggunakan AG

Berdasarkan pengolahan data yang sudah dilakukan, ternyata AG dapat mengurangi total jarak tempuh urutan rute pendistribusian produk. Hal ini dipengaruhi oleh pemilihan urutan rute pendistribusian produk yang bertujuan meminimasi total panjang lintasan. Selain itu AG juga mampu menghasilkan alternatif urutan rute pendistribusian produk yang lebih banyak, sehingga lebih memberikan keluasaan di dalam pemilihan urutan rute yang optimal dibanding kondisi awal perusahaan, dan dapat menghasilkan satu keputusan yang terbaik dari alternatif yang ada.

Perbandingan total panjang lintasan urutan rute pendistribusian produk dengan menggunakan AG dan tanpa AG terlihat bahwa total panjang lintasan AG lebih kecil dibandingkan dengan total panjang lintasan pendistribusian produk tanpa AG (awal).

Total jarak tempuh urutan rute pendistribusian awal yang diperoleh dari hasil perhitungan sebesar 588,2 km. Sedangkan total panjang lintasan urutan rute pendistribusian yang diperoleh dari pengolahan menggunakan AG sebesar 437,25 km. Sehingga total jarak tempuh urutan rute pendistribusian produk dengan menggunakan AG mengalami pengurangan sebesar 150,95 km dan perbaikan jalur distribusi sebesar 25,66%.

Algoritma genetika adalah suatu metode untuk mendapatkan alternatif solusi dari sekian banyaknya solusi-solusi yang dibangkitkan saat inisialisasi berlangsung. AG dianjurkan dan cocok untuk masalah yang memiliki karakteristik dengan alternatif

solusi yang sangat besar serta evolusi yang rumit. Tujuan dari penyelesaian kasus yang dihadapi adalah mengurutkan kota dalam rute pendistribusian produk untuk meminimasi total panjang lintasan, sehingga konsep yang dipakai untuk menyelesaikan permasalahan yang dihadapi adalah konsep minimasi. Hal ini berlawanan dengan konsep yang dipakai dalam AG. Karena didalam AG, konsep utamanya adalah mencari kromosom yang mempunyai nilai kesesuaian (*fitness value*) yang besar.

Dalam AG perlu adanya memodelkan kasus yang ada dalam sebuah pemrograman komputer alami yang sedikit rumit dalam hal ini peneliti menggunakan *software Matlab 7.6*. Dalam menentukan parameter (Probabilitas *crossover* dan probabilitas mutasi) menggunakan beberapa parameter sebagai pertimbangan dan hasil yang didapat adalah dari 5 parameter bila dibandingkan dengan gambar 4.9 dapat dilihat bahwa panjang jalur terbaik diperoleh dari parameter P_{silang} 0,5 dan P_{mutasi} 0,01. Seleksi induk dengan *roulette wheel* memberikan pendekatan yang lebih baik, karena kromosom yang memiliki *fitness* yang baik akan cenderung lebih besar peluangnya untuk terpilih menjadi induk untuk proses modifikasi kromosom selanjutnya.

5.2 Pemilihan Urutan Rute Pendistribusian Produk

Hal yang menjadi bahan pertimbangan dalam pemilihan urutan rute pendistribusian produk adalah total panjang lintasan terpendek. Kriteria tersebut diidentifikasi dari kenyataan yang terjadi di lapangan, bahwa dengan total panjang lintasan terpendek dapat memberikan keuntungan diantaranya meminimalkan total biaya pendistribusian produk, mengurangi waktu pendistribusian produk, dan akhirnya hal-hal tersebut dapat meningkatkan keuntungan perusahaan.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan pengolahan data dan pembahasan yang telah dilakukan, maka dapat diambil kesimpulan, sebagai berikut:

1. Dengan menggunakan algoritma genetika dapat menyelesaikan permasalahan TSP yaitu memperoleh rute distribusi produk dengan total jarak tempuh yang minimum sebesar 437,25 km.
2. Algoritma genetika dapat memberikan solusi yang optimal dari kasus TSP yang dihadapi yakni untuk meminimasi total jarak tempuh dalam pemilihan urutan rute pendistribusian produk. Terjadi pengurangan panjang lintasan distribusi sebesar 150,95 km. Sehingga diperoleh perbaikan jalur distribusi dalam persentase yaitu sebesar 25,66%.

6.2 Saran

Berdasarkan kesimpulan yang diperoleh dari analisa data perusahaan dan sebagai referensi untuk penelitian lanjutan dari penelitian ini, untuk pengembangan aplikasi Algoritma Genetik dalam kasus TSP, maka diberikan saran-saran yang sekiranya dapat menjadi masukan bagi perusahaan dan peneliti lain. Adapun saran-saran itu adalah sebagai berikut:

1. Perlu diperhatikan waktu pendistribusian produk dari satu kota ke kota berikutnya, karena panjang lintasan terpendek belum tentu memiliki waktu pendistribusian terpendek pula. Misalnya rute G – A37 – A55 – A58 – A57 – A15 – G pada jam-

jam tertentu cenderung macet sehingga menyebabkan waktu pendistribusian menjadi lebih lama.

2. Perlu adanya komunikasi antara agen dan pihak distributor minimal 1 hari sebelum pengiriman produk agar jumlah produk yang di bawa tidak kurang dan tidak berlebihan. Komunikasi tersebut juga bertujuan agar waktu pelayanan menjadi lebih cepat karena seorang salesman harus menunggu agen menghitung jumlah produk yang dibutuhkan.
3. Bagi para peneliti disarankan untuk melakukan penelitian lanjutan berdasarkan hasil penelitian ini. Adapun penelitian lanjutan yang dapat dilakukan adalah pemilihan urutan rute pendistribusian produk pada kasus asimetris.



DAFTAR PUSTAKA

- Androvov, D., (2005). *Minimasi Total Panjang Lintasan Pendistribusian Produk menggunakan Aplikasi Algoritma Tabu Search*. Fakultas Teknologi Industri, UII, Yogyakarta
- Amin, A.R., dan Ikhsan, M., dan Wibisono, L., (2008). *Travelling Salesman Problem*. Departemen Teknik Informatika, ITB, Bandung
- Cheng Runwei, dan Gen Mitsuo (1997). *Genetic Algorithms and Engineering Design*, John Wiley & Sons, Inc., New York
- Kurnianto, W.A., dan Fahrudin, T., dan Hardono, A., (2006). *Pemakaian Algoritma Brote Force pada Permasalahan TSP*, Teknik Informatika, STTK, Bandung
- Kusumadewi, Sri., (2003) *Artifitial Intelegence: Teknik dan Aplikasinya*, Graha ilmu, Yogyakarta
- Lalena, M., (1998). *Travelling Salesman Problem Using Genetic Algorithms*. www.LaLena.com
- Lukas, S., Anwar, T., Yuliani, W., (2005). Penerapan Algoritma Genetika Untuk Travelling Salesman Problem Dengan Menggunakan Metode *Order Crossover* Dan *Insertion Mutation*. Seminar Nasional Aplikasi Teknologi Informasi (SNATI) 18 Juni. Yogyakarta
- Ningtyas, D.K., dan Vina, E., dan Ernastuti., (2008). *Evaluasi Kinerja Algoritma Travelling Salesman Problem dengan Teknik Pemrograman Dinamik*. *Proceeding Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT)* 20-21 Agustus. Universitas Gunadarma, Depok
- Novan., (2004), *Aplikasi Jaringan Syaraf Tiruan Kohonen untuk menyelesaikan masalah Travelling Salesman Problem*, Fakultas Teknologi Industri, UII, Yogyakarta
- Nudu, J.H., (2007). *Kombinasi Strategi Distribusi Untuk Menurunkan Biaya Logistik*. *Jurnal Teknologi Industri*, Vol.XI, No. 2, April, 163-172, Universitas Atma Jaya, Yogyakarta
- Nugraha, I., (2008). *Aplikasi Algoritma Genetik Untuk Optimasi Penjadwalan Kegiatan Belajar Mengajar*. Makalah Strategi Algoritmik, Teknik Informatika, ITB, Bandung
- Paputungan, I.V., (2004). *Perbandingan Metode-Metode dalam Algoritma Genetika untuk Travelling Salesman Problem*. Seminar Nasional Teknologi Informasi 19 Juni, Yogyakarta
- Purnomo, I., dan Oskar., (2000). *Algoritma Genetik Pada Pencarian Rute Terpendek Dalam Masalah Travelling Salesman Problem (TSP)*. *Proceedings, Komputer dan Sistem Intelijen KOMMIT* 23-24 Agustus, Universitas Gunadarma, Jakarta

- Puspitorini, Sukma., (2008). *Penyelesaian Masalah Travelling Salesman Problem dengan Jaringan Saraf Self Organizing*. Media Informatika, Vol.6, Juni, 39-55, STMIK Nurdin Hamzah, Jambi
- Rewoldt., dan Stewart., (1987). *Pemasaran dan Strategi Pendistribusian*, Aneka Ilmu, Semarang
- Sanjoyo., (2006) *Aplikasi Algoritma Genetika*, Sanjoyo55.files.wordpress.com
- Saptono, F., dan Hidayat, T., (2007). *Perancangan Algoritma Genetika Untuk Menentukan Jalur Terpendek*. Seminar Nasional Aplikasi Teknologi Informasi (SNATI), 16 Juni, Yogyakarta
- Suyanto., (2005). *Algoritma Genetika dalam Matlab*, Penerbit Andi, Yogyakarta



LAMPIRAN

Nama Agen/ Toko dan Alamat Lokasi

Nomor	Nama Agen/ Toko	Alamat Lokasi
1	Lavima 2	Ganjuran
2	Rahmi	Jl. Parangtritis km. 25
3	DH	Kretek
4	Mustika	Kretek
5	Agung apotik	Jl. Ringroad
6	Fatimah	Jl. Jendral sudirman, pundong
7	Semi	Pasar bantul
8	Oskar	Sono sewu, Ngestiharjo
9	Utomo	Jl. Raya Bantul
10	Sarjinem	Pasar Bantul
11	Hans	Kasongan
12	Bu midah	Kasongan
13	Bu purwo	Jl. Jogonalan, Tirtonirmolo
14	Tk lestari	Jl. Ringroad Selatan
15	Bu sri	Jl. Bibis, Tamantirto
16	Kios bagus	Jl. Karanggayam, Sitimulyo Piyungan
17	Tk prasojo	Jl. Ringroad Taman Tirto
18	Rangga	Jl. Srandakan
19	Ani	Jl. Raya Srandakan
20	Dipo	Jl. Dr Wahidin, Tirenggo
21	Utomo	Jl. Hasyim Asyari, Mandingan
22	Arsita	Monggalan
23	Paryono	Pasar Celep, Sanden
24	Unik	Pundong
25	Tk bimo	Turi, Bambanglipuro
26	Pojok	Pasar Gamping
27	Mulyo	Pundong
28	Yeye	Kretek
29	Dev mart	Jl. Parangtritis Km. 5
30	Tk toni	Jl. Parangtritis Km 15
31	Hadi	Jl. Ambar Binangun
32	Narti	Kasihani Bantul
33	HR	Jl. A Dahlan
34	Dahlan	Jl. Ibu Ruswo
35	Tk sumber santoso	Jl. Brigjend Katamso
36	Tk noto putro	Jl. Re Martadinata
37	Sumber insani mandiri	Yogyakarta
38	Maju jaya	Jl. Piere Tendean
39	Ridha jaya	Jl. Ali Maksum
40	Ida	Krapyak Lor
41	Apotek diro	Bantul

42	Notojoyo	Jl. Bantul Km 2.5
43	Apotek christella	Jl. Re Martadinata
44	Apotek K24	Jl. Wates
45	Mami	Pasar Gamping
46	HM	Pasar Gamping
47	Tk setia	Jl. Dr Wahidin, Tirenggo
48	Tk c2	Jl. Perum Pelemsewu Sewon
49	Perdana	Jl. Rotowijayan 14
50	Blimbing	Brosot
51	Berkah	Jl. Srandakan
52	Ana	Jl. Wates c
53	Yanto b	Jl. Raya Srandakan
54	Tri wahyuni	Jl. Cokrodipuran
55	Semangat baru	Jl. Gandekan Kidul
56	Waserda	Jl. Ks Tubun
57	Kuncen	Jl. Hos Cokroaminoto
58	69	Jl. Hos Cokroaminoto
59	Linawati	Jl. Suryatmajan
60	Remaja	Jl. A Dahlan

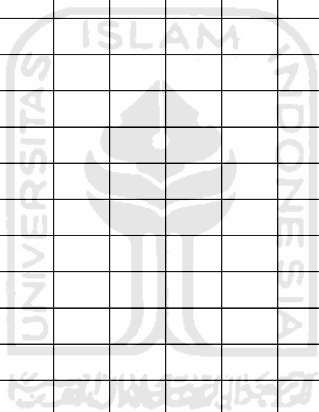


Matrik Jarak dari Gudang ke Agen (distributor) dan Jarak antar Agen

	G	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22	A23	A24	A25	A26	A27	A28	A29	A30	A31
G																																
A1	17.5																															
A2	20.8	4.0																														
A3	24	7.3	3.3																													
A4	24.2	7.4	3.5	0.2																												
A5	7	11.6	15.4	18.7	18.8																											
A6	18.3	2.4	2.6	5.8	5.9	12.9																										
A7	13.5	4.5	8.4	11.7	11.9	7.2	6.2																									
A8	9.1	13.7	17.7	21.1	21.2	4	15.5	9.3																								
A9	8.9	9.6	13.4	16.8	16.9	1.9	11	5.1	4.9																							
A10	13.5	4.5	8.4	11.7	11.8	7.1	6.1	8.25	9.3	5.1																						
A11	9.7	9.3	13.3	17.5	16.4	2.7	11	4.8	4.6	1.1	4.9																					
A12	9.9	9.5	13.2	16.6	16.7	2.9	10.9	4.9	4.5	1.4	4.8	0.2																				
A13	7.9	11.3	15.2	18.5	18.7	1	12.8	6.8	3.3	1.8	6.8	2.1	2.2																			
A14	8.5	11.4	15.3	18.6	18.8	1.7	13	6.9	2.7	2.2	6.9	2	2.1	0.7																		
A15	9.5	11.9	15.9	19.2	19.4	3.1	13.7	7.6	1.7	3.4	7.6	2.9	2.8	2.2	1.4																	
A16	4.7	17.1	19.7	22.6	22.4	9.3	17.1	13.7	12.5	10.3	13.7	11.4	11.7	10.3	11	12.3																
A17	9.5	11.7	15.6	18.9	19	2.9	13.4	7.2	2.1	3.1	7.2	2.5	2.4	1.9	1.2	0.3	12.2															
A18	25.3	9.3	7.8	10.5	10.6	18	10.7	11.6	18.2	16.1	11.7	15.3	15.1	17.3	17	16.8	25.5	16.5														
A19	25	9.2	9.6	10.4	10.4	17.9	10.6	11.7	18.1	16	11.6	15.2	15	17.2	16.9	16.7	25.4	16.4	0.1													
A20	13.1	4.8	7.7	10.9	11.1	8	5.2	3.2	11.2	6.3	3.1	6.6	6.7	8.2	8.5	9.5	12.3	9.2	13.6	13.6												
A21	14.2	3.4	7.3	10.6	10.7	8.1	5.1	1.1	10.4	6.2	1	5.9	0.6	7.9	8	8.7	14.2	8.3	11.2	11.2	2.7											
A22	12.1	6	8.8	11.9	12.1	7.6	6.2	4.1	11	6.1	4	6.5	6.7	7.8	8.3	9.4	11	9.1	14.9	14.8	1.2	3.8										
A23	25.9	8.5	6.5	5.5	5.5	19.6	8.6	12.5	20.9	17.6	12.5	17.1	17	19.2	19.1	19.3	25.5	18.9	5.8	5.6	13.2	11.7	14.4									
A24	18.6	3.9	2.7	5.4	5.6	13.7	1.4	7.3	16.5	11.9	7.2	11.9	12	13.7	13.9	14.8	17.2	14.4	11.9	11.8	5.7	6.2	6.5	9.1								
A25	22.2	5	1.4	2.4	2.5	16.5	3.9	9.4	18.7	14.6	9.4	14.3	14.3	16.3	16.4	16.9	21.1	16.6	9	8.9	9	8.4	10.1	5.1	4.1							
A26	9.5	13	19	21.3	21.4	4.5	15.8	9.6	0.4	5.3	9.6	4.9	4.8	3.7	3.1	3	12.9	2.4	18.1	18.2	11.6	10.7	11.4	21.1	16.8	18.9						
A27	18.4	3	2.5	5.6	5.7	13.3	0.6	6.6	15.9	11.4	6.6	11.3	11.4	13.2	13.4	14.2	17.1	13.8	11.2	11.1	5.3	5.5	6.3	8.8	0.8	3.9	16.2					

A28	24.3	7.5	3.6	0.3	0.2	18.9	6	11.0	21.1	16.9	11.9	16.7	16.7	18.7	18.8	19.4	22.9	19	10.4	10.2	11.2	10.8	12.2	5.3	5.7	2.4	21.4	5.8				
A29	7.1	10.7	14.3	17.6	17.7	1.8	11.7	6.4	5.7	1.9	6.4	3.1	3.3	2.5	3.2	4.7	8.3	4.4	17.7	17.7	6.7	7.2	6	18.9	12.4	15.5	6.2	12	17.7			
A30	16.1	3.4	4.9	7.9	8.1	11.2	2.4	5.2	14.2	9.4	5.2	9.6	9.6	11.3	11.5	12.5	14.8	12.1	12.6	12.5	3.1	4.2	4	10.9	2.5	6.2	14.5	2.3	8.2	9.8		
A31	8.3	11.1	15	18.3	18.5	1.4	12.7	6.6	3.2	1.7	6.6	1.8	1.9	0.3	0.5	1.9	10.6	1.6	16.8	16.9	8.1	7.7	7.8	18.9	13.6	16.1	3.6	13	18.5	2.7	11.1	
A32	10.6	12	15.9	19.2	19.3	4.3	13.8	7.6	2.1	4.3	7.7	3.4	3.3	3.3	2.5	1.2	13.5	1.3	15.9	16	9.9	8.7	10	18.9	14.9	16.8	2.2	14.4	19.3	5.7	12.7	3
A33	5.3	14.6	18.4	21.7	21.9	3	15.9	10.1	3.8	5.1	10.3	5.5	5.6	3.4	3.7	4.3	8.9	4.4	20.6	20.6	11	11.2	10.4	22.6	16.7	19.6	4.2	16.2	21.9	4.3	14.1	3.7
A34	4.8	14.5	18.3	21.6	21.8	3	15.8	10.2	4.3	5	10.2	5.6	5.8	3.5	4	4.7	8.3	4.7	20.8	20.8	10.8	11.1	10.2	22.7	16.5	19.5	4.7	16.1	21.8	4.1	14	3.9
A35	4.6	14.3	17.9	21.3	21.4	2.8	15.5	9.9	4.6	4.7	10	5.4	5.6	3.4	3.9	4.8	7.9	4.8	20.6	20.7	10.4	10.8	9.7	22.4	16.1	19.2	5	15.7	21.5	3.7	13.6	3.8
A36	6.3	14.4	18.2	21.5	21.7	2.9	15	10	2.8	4.8	9.9	5.1	5.2	3	3.1	3.4	9.9	3.6	19.9	20	11	10.9	10.5	22.2	16.7	19.3	3.1	16.2	21.7	4.5	14.2	3.3
A37	6.6	18.9	22.8	26.1	26.3	7.4	20.4	14.4	6	9.3	14.4	9.6	9.7	7.5	7.5	7.4	11.2	7.7	24.1	24.1	15.5	15.5	14.9	26.6	21.2	23.9	6	20.7	26.3	8.8	18.6	7.8
A38	6.5	13.8	17.6	20.9	21.1	2.4	15.2	9.3	2.6	4.1	9.3	4.4	4.5	2.4	2.6	3	9.9	3.1	19.4	19.4	10.4	10.3	10	21.6	16	18.7	3	15.6	21.1	4	13.6	2.6
A39	6.2	12.2	16	19.3	19.4	0.8	13.5	7.8	4.4	2.7	7.8	3.6	3.7	1.8	2.5	3.8	8.5	3.6	18.8	18.8	8.5	8.7	7.9	20.3	14.2	17.1	4.8	13.7	19.4	1.8	11.6	2.2
A40	6.1	12.1	15.8	19.1	19.2	0.9	13.3	7.7	4.6	2.6	7.7	3.5	3.8	1.9	2.7	4	8.3	3.8	18.7	18.7	8.3	8.6	7.6	20.2	14	17	5.1	13.6	19.3	1.6	11.4	2.3
A41	9.9	7.7	11.3	14.6	14.7	4.2	8.8	3.5	7.4	2.4	3.5	3	3.1	4.2	4.6	5.8	10.3	5.4	15.1	15.1	3.9	4.3	3.6	16	9.5	12.5	7.8	9.1	14.8	2.9	7	4.2
A42	9.2	9.1	13	16.3	16.4	2.5	10.5	4.6	5.3	0.4	4.7	1	1.3	2.2	2.6	3.8	10.5	3.4	15.6	15.6	5.9	5.6	5.7	17.1	11.4	14.1	5.7	10.9	16.4	2.2	9	2.1
A43	6.1	14.4	18.3	21.6	21.7	2.9	15.8	9.9	3.1	4.8	1	5.2	5.3	3.1	3.2	3.7	9.5	3.8	20.1	20.2	11	11	10.4	22.3	16.6	19.4	3.5	16.2	21.8	4.4	14.1	3.4
A44	7.9	14.1	18	21.3	21.5	3.5	15.7	9.6	1.1	4.7	9.6	4.7	4.7	3	2.7	2.3	11.4	2.5	18.9	19	11.2	10.8	10.9	21.5	16.7	19	1.5	16.1	21.5	5.2	14.2	3.1
A45	9.5	14	17.9	21.2	21.4	4.5	15.6	9.7	4.6	5.3	9.6	5	4.8	3.7	3.1	2	12.9	2.4	18.1	18.2	11.6	10.7	11.4	21	16.8	18.9	0.2	16.2	21.4	6.2	14.5	3.6
A46	9.5	14	17.9	21.2	21.4	4.5	15.6	9.7	4.6	5.3	9.6	5	4.8	3.7	3.1	2	12.9	2.4	18.1	18.2	11.6	10.7	11.4	21	16.8	18.9	0.2	16.2	21.4	6.2	14.5	3.6
A47	13	4.7	7.8	11	11.2	7.9	5.3	2.9	11	6.1	2.9	6.4	6.5	8	8.3	9.3	12.3	8.9	13.5	13.4	0.2	2.5	1.3	13.2	5.8	9.1	11.3	5.4	11.3	6.5	3.3	7.9
A48	6.3	11.4	15	18.3	18.4	1.7	12.5	7.2	5.6	2.5	7.2	3.6	3.9	2.6	3.4	4.8	7.8	4.6	18.5	18.5	7.4	8	6.7	19.7	13.1	16.2	6.1	12.7	18.5	0.7	10.6	2.9
A49	5.4	14.1	18	21.2	21.4	2.5	15.4	9.6	3.8	4.5	9.7	5	5.2	2.9	3.3	4.1	8.7	4.1	20.2	20.2	10.5	10.6	9.8	22.1	16.1	19	4.2	15.7	21.4	3.8	13.6	3.3
A50	25.7	10.1	10.3	10.9	10.9	18.8	11.5	12.6	18.9	16.9	12.6	16.1	15.9	18.1	17.7	17.5	26.3	17.3	0.4	0.9	14.5	12.1	15.8	5.8	12.6	9.5	18.9	11.9	10.7	18.6	13.4	17.7
A51	24.4	8.9	9.5	10.4	10.4	17.5	10.4	11.2	17.7	15.6	11.2	14.8	14.6	16.8	16.5	16.2	25	16	0.4	0.4	13.2	10.8	14.5	5.7	11.6	8.8	17.7	10.8	10.2	17.3	12.2	16.4
A52	11.1	13.6	17.6	20.9	21	5.6	15.6	9.4	1.9	5.9	9.5	5.2	5.1	4.6	3.9	2.5	14.4	2.8	17	17	11.8	10.5	11.8	20.3	16.7	18.5	1.6	16.1	21	7.2	14.6	4.4
A53	23.7	7.8	8.4	9.4	9.5	16.8	9.3	10.4	17.2	14.9	10.4	14.1	13.9	16.2	15.9	15.7	24.1	15.5	1.4	1.3	12.2	9.9	13.5	5.1	10.5	7.7	17.3	9.7	9.3	16.5	11.1	15.8
A54	4.7	14.1	17.8	21.1	21.3	2.6	15.3	9.7	4.5	4.6	9.7	5.3	5.5	3.3	3.8	4.7	8	4.7	20.5	20.5	10.3	10.7	9.6	22.2	16	19	5	15.6	21.3	3.6	13.5	3.7
A55	5.3	15.6	19.4	22.7	22.8	3.9	16.9	11.1	4	5.9	11.1	6.3	6.4	4.2	4.4	4.8	9.3	4.9	21.3	21.3	12	12.1	11.4	23.5	17.7	20.5	4.3	17.2	22.9	5.3	15.1	4.5
A56	5.5	15	19	22	22.2	3.4	16.3	10.5	3.6	5.4	10.5	5.7	5.8	3.6	3.8	4.2	9.2	4.4	20.7	20.7	11.4	11.5	10.8	22.8	17.1	19.9	3.9	16.6	22.2	4.7	14.5	3.9
A57	6.2	14.9	18.7	22.1	22.3	3.5	16.2	10.4	2.9	5.3	10.4	5.6	5.6	3.5	3.6	3.8	10.1	3.9	20.4	20.4	11.5	11.4	11	22.7	17.2	20	3.2	16.7	22.3	5	14.7	3.8
A58	6.2	15.2	19	22.4	22.6	3.8	16.7	10.7	3.1	5.6	10.7	5.9	6	3.8	3.9	4	10	4.2	20.6	20.7	11.9	11.7	11.3	23	17.5	20.2	3.3	17	22.6	5.3	15.1	4.1
A59	4.6	15.5	19.2	22.6	22.8	3.9	16.8	11.1	4.5	5.9	11.1	6.4	6.6	4.4	4.6	5.2	8.6	5.3	21.5	21.6	11.8	12.1	11.1	23.5	17.6	20.4	4.8	17.1	22.8	5.1	15	4.7
A60	5.5	14.6	18.3	21.7	21.8	3	16	10.1	3.5	4.9	10.1	5.4	5.5	3.3	3.5	4.1	9.1	4.2	20.4	20.5	11	11.1	10.5	22.5	16.7	19.5	3.9	16.3	21.9	4.4	14.2	3.6

A32	A33	A34	A35	A36	A37	A38	A39	A40	A41	A42	A43	A44	A45	A46	A47	A48	A49	A50	A51	A52	A53	A54	A55	A56	A57	A58	A59	A60
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----



[illegible]

Lampiran

Program Utama

```
%ProgramUtamaYuni%
clear;
clc;
datajarak=xlsread('matrik.xls');
%inisialisasi parameter
XYkota=[42.4943 912.3748; 42.5209 911.9746; 42.4543 911.6475; 42.4547
911.6315; 42.8754 913.4692;
42.6531 912.1934; 42.5960 912.8096; 42.5794 913.7434; 42.7851 913.2913;
42.5958 912.8088;
42.6707 913.2915; 42.6434 913.2932; 42.7741 913.4748; 42.7011 913.4968;
42.5753 913.5698;
43.8052 913.4680; 42.5854 913.5329; 41.5783 912.2337; 41.5887 912.2202;
42.8765 912.6601;
42.6072 912.6996; 42.9917 912.7170; 41.9117 911.7571; 42.7652 912.0991;
42.4127 911.8796;
42.5415 913.7715; 42.6986 912.1526; 42.4307 911.6308; 42.9798 913.3239;
42.8363 912.3457;
42.7378 913.4595; 42.4541 913.5658; 42.9624 913.7598; 43.0169 913.7378;
43.0424 913.6939;
42.8611 913.7663; 42.9594 914.2108; 42.8428 913.7067; 42.9540 913.5081;
42.9709 913.4859;
42.8552 913.0537; 42.7704 913.2454; 42.8942 913.7618; 42.6956 913.7681;
42.5417 913.7707;
42.5421 913.7700; 42.8583 912.6775; 43.0237 913.3892; 42.9642 913.7065;
41.5044 912.1798;
41.6148 912.2540; 42.3829 913.7337; 41.7236 912.2403; 43.0340 913.6825;
42.9632 913.8590;
label=1:length(XYkota(:,1));
JumGen=length(XYkota(:,1));
hasil=XYkota;
Gudang=[43.4880 913.8180];
UkPop=100;
Psilang=0.5;
Pmutasi=0.01;
MaxG=2000;
PanjJalHarp=40;
%inisialisasi grafis;
Fthreshold=0.05;
Bgraf=Fthreshold;
hfig=figure;
hold on
set(hfig, 'position', [50,50,600,400]);
set(hfig, 'DoubleBuffer', 'on');
axis([1 MaxG 0 Bgraf]);
hbestplot1=plot(1:MaxG,zeros(1,MaxG));
hbestplot2=plot(1:MaxG,zeros(1,MaxG));
htext1=text(0.6*MaxG,0.25*Bgraf,sprintf('Fitness terbaik:%7.6f',0.0));
htext2=text(0.6*MaxG,0.20*Bgraf,sprintf('Fitness rata-rata:%7.6f',0.0));
htext3=text(0.6*MaxG,0.15*Bgraf,sprintf('panjang jalur
terbaik:%7.3f',0.0));
htext4=text(0.6*MaxG,0.10*Bgraf,sprintf('ukuran Populasi:%3.0f',0.0));
xlabel('generasi');
ylabel('Fitness');
hold off
drawnow
%inisialisasi Populasi
```

```

Populasi=TSPInisialisasiPopulasi(UkPop, JumGen);
figure;
for generasi=1:MaxG
    MaxF=TSPEvaluasiIndividu(Populasi(1,:), JumGen, datajarak);
    MinF=MaxF;
    IndeksIndividuTerbaik=1;
    for ii=2:UkPop
        Fitness(ii)=TSPEvaluasiIndividu(Populasi(ii,:), JumGen, datajarak);
    end
    end
    FitnessRataRata=mean(Fitness);
    plotvector1=get(hbestplot1, 'YData');
    plotvector1(generasi)=MaxF;
    set(hbestplot1, 'YData', plotvector1);
    plotvector2=get(hbestplot2, 'YData');
    plotvector2(generasi)=FitnessRataRata;
    set(hbestplot2, 'YData', plotvector2);
    set(htext1, 'string', sprintf('Fitness terbaik:%7.6f', MaxF));
    set(htext2, 'string', sprintf('panjang jalur terbaik:%7.3f', 1/MaxF));
    set(htext3, 'string', sprintf('ukuran Populasi:%3.0f', UkPop));
    set(htext4, 'string', sprintf('probabilitas mutasi:%4.3f', Pmutasi));
    drawnow
    if MaxF>Fthreshold
        break;
    end
    TemPopulasi=Populasi;
    %elitisme
    if mod(UkPop, 2)==0
        IterasiMulai=3;
        TemPopulasi(1,:)=Populasi(IndeksIndividuTerbaik,:);
        TemPopulasi(2,:)=Populasi(IndeksIndividuTerbaik,:);
    else
        IterasiMulai=2;
        TemPopulasi(1,:)=Populasi(IndeksIndividuTerbaik,:);
    end
    LinearFitness=linearFitnessRanking(UkPop, Fitness, MaxF, MinF)
    %Proses pindah silang dengan metode roulette wheel
    for jj=IterasiMulai:2:UkPop
        IP1=RouletteWheel(UkPop, LinearFitness);
        IP2=RouletteWheel(UkPop, LinearFitness);
        if (rand<Psilang)
            Anak=TSPPindahSilang(Populasi(IP1,:), Populasi(IP2,:), JumGen);
            TemPopulasi(jj,:)=Anak(1,:);
            TemPopulasi(jj+1,:)=Anak(2,:);
        end
    end
    %proses terjadinya mutasi
    for kk=IterasiMulai:UkPop
        TemPopulasi(kk,:)=TSPMutasi(TemPopulasi(kk,:), JumGen, Pmutasi);
    end
    %replacemenet generasi baru
    Populasi=TemPopulasi;
    for i=1:length(XYkota(:,1))
        hasil(i,:)=XYkota(Populasi(1,i),:);
    end
    plot(hasil(:,1), hasil(:,2), 'r.-');grid;
    axis([41.25 44 911.5 914.25])

end

end

```

