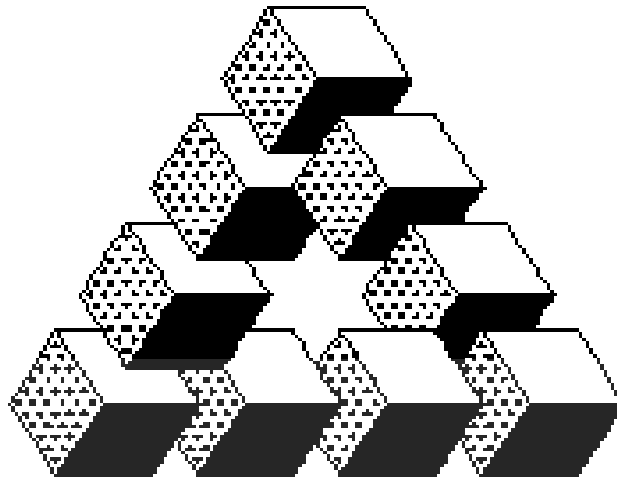


**OPTIMASI PENJADWALAN DAN RUTE PENGIRIMAN
BARANG PADA PELAYARAN PT. SURYA DENGAN
MENGUNAKAN PENDEKATAN METODE ALGORITMA
GENETIKA**



STIKOM
Dinamika

Oleh :

Nama : Wina Kusumawaty

NIM : 00.41010.0051

Program : S1 (Strata Satu)

Jurusan : Sistem Informasi

**SEKOLAH TINGGI
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER
SURABAYA**

2006

**OPTIMASI PENJADWALAN DAN RUTE PENGIRIMAN BARANG PADA
PELAYARAN PT. SURYA DENGAN MENGGUNAKAN PENDEKATAN
METODE ALGORITMA GENETIKA**

SKRIPSI

Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Komputer



Oleh :

Nama : Wina Kusumawaty

NIM : 00.41010.0051

Program : S1 (Strata Satu)

Jurusan : Sistem Informasi

SEKOLAH TINGGI

MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER

SURABAYA

2006

**OPTIMASI PENJADWALAN DAN RUTE PENGIRIMAN BARANG PADA
PELAYARAN PT. SURYA DENGAN MENGGUNAKAN PENDEKATAN
METODE ALGORITMA GENETIKA**

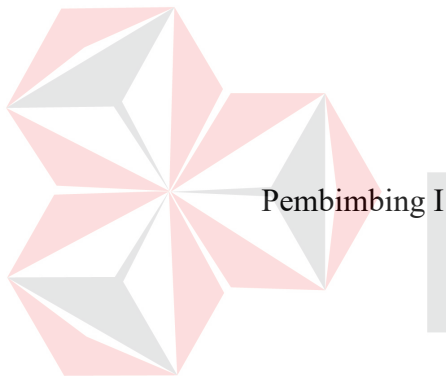
Disusun Oleh :

Nama : Wina Kusumawaty

NIM : 00.41010.0051

Surabaya, September 2006

Telah diperiksa, diuji dan disetujui :



Pembimbing I

Pembimbing II

UNIVERSITAS
Dinamika

Tutut Wuriyanto, M.Kom
NIDN 0703056702

Teguh Sutanto, S.Kom
NIDN

Mengetahui :

Wakil Ketua Bidang Akademik

Drs. Antok Supriyanto, M.MT
NIDN. 0703056702

ABSTRAKSI

Pelayaran PT Surya bergerak dalam bidang jasa pendistribusian, dengan menggunakan alat transportasi berupa kapal. Dalam suatu transaksi perusahaan memiliki tanggung jawab untuk membuat jadwal pengiriman barang bagi setiap kapal. Dari jadwal yang dibuat diharapkan pengiriman barang datang tepat waktu, sehingga biaya operasional kapal dapat diminimalkan. Yaitu dengan melakukan optimalisasi rute pengiriman.

Algoritma genetika merupakan suatu metode yang menggunakan seleksi alam yang merupakan bagian utama dari prinsip evolusi yaitu setiap individu yang mampu beradaptasi dengan lingkungannya akan mempunyai kesempatan hidup yang lebih besar. Prinsip tersebut digunakan sebagai dasar pemikiran untuk menyelesaikan suatu permasalahan.

Permasalahan dari tugas Akhir ini adalah bagaimana mengoptimalkan penjadwalan dan rute pengiriman barang menggunakan algoritma genetika pada pelayaran PT. Surya, sehingga jumlah kapal yang dibutuhkan dan waktu tempuh yang dilalui adalah minimum.

Berdasarkan hasil uji coba yang telah dilakukan pada Tugas Akhir ini dengan menggunakan generasi 10, populasi sebesar 40, probabilitas crossover 65%, probabilitas mutasi 5% diperoleh solusi terbaik dari kasus optimasi penjadwalan. Hasil nilai fitness terbaik yang diperoleh sebesar 507,8 jam.

KATA PENGANTAR

Dengan rasa syukur ke hadirat Tuhan Yang Maha Esa, berkat rahmat dan karunia-Nya penulis dapat menyelesaikan Tugas Akhir ini. Penyusunan Tugas Akhir ini adalah syarat dalam menyelesaikan Program Studi Strata Satu di Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya. Terima kasih kepada Bapak Tutut Wuriyanto, M.Kom selaku dosen pembimbing I yang telah banyak membimbing dan membantu dalam menyelesaikan Tugas Akhir ini, Bapak Teguh Sutanto, S.Kom, selaku dosen pembimbing II yang telah membimbing penulis dalam menyelesaikan Tugas Akhir.

Banyak pihak yang telah memberikan bantuan dalam menyusun Tugas Akhir ini, untuk itu penulis juga menyampaikan rasa terima kasih kepada :

1. Kepada Papa, Mama, Oscar, Monica, Yunita, Aaron dan Kakak ipar Suryanto yang telah memberikan do'a, dukungan dan perhatian kepada penulis.
2. Sahabatku Herny, Vivi *thanks a lot* buat dukungan tenaga, waktu, dan pengorbanan kalian.
3. Teman-temenku Rizki, Yiyik, Andri, Pandugo "Crew", Polenk, Erwin, Crist Jumbo, Riza, Kirman yang sudah mendukung penulis dalam menyelesaikan Tugas Akhir.
4. Semampir "Genk" yang sudah menyediakan waktu dan mau direpotkan penulis untuk menyelesaikan Tugas Akhir ini.
5. Tak lupa untuk semua orang yang telah membantu penulis yang tidak bisa disebutkan namanya, penulis ucapkan terima kasih.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna, namun penulis berharap semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak dan dapat ikut menunjang perkembangan teknologi dan ilmu pengetahuan, khususnya ilmu komputer.

Surabaya, September 2006

Penulis



UNIVERSITAS
Dinamika

DAFTAR ISI

	Halaman
ABSTRAKSI	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xiii
BAB I : PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Sistematika Penulisan	3
BAB II : LANDASAN TEORI	5
2.1 Algoritma Genetika	5
2.2 Rute Kendaraan	17
BAB III : METODE PENELITIAN.....	26
3.1 Identifikasi Masalah.....	27
3.2 Perancangan Algoritma Genetika	30
3.3 Desain Sistem.....	45

BAB IV	:	IMPLEMENTASI DAN EVALUASI	55
4.1		Implementasi Sistem.....	56
4.2		Evaluasi Sistem	62
BAB V	:	PENUTUP	74
5.1		Kesimpulan	74
5.2		Saran	74
DAFTAR PUSTAKA		76
LAMPIRAN		78



UNIVERSITAS
Dinamika

DAFTAR TABEL

	Halaman
Tabel 3.1 Tabel Data Kapal.....	52
Tabel 3.2 Tabel Data Tujuan.....	53
Tabel 3.3 Tabel Data Jarak.....	53
Tabel 3.4 Tabel Pemesanan	53
Tabel 3.5 Tabel Input Kapal.....	53
Tabel 3.6 Tabel Input Tujuan.....	53
Tabel 3.7 Tabel Laporan	54
Tabel 4.1 Uji Coba I Generasi.....	62
Tabel 4.2 Uji Coba II Populasi	65
Tabel 4.3 Uji Coba III Crossover.....	68
Tabel 4.4 Uji Coba IV Mutasi.....	71

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Struktur Umum Algoritma Genetika.....	6
Gambar 2.2 Kromosom dalam Algoritma Genetika	7
Gambar 2.3 Permutation Encoding.....	8
Gambar 2.4 Binary Encoding.....	8
Gambar 2.5 Proses Crossover dengan Satu Crossover Point.....	14
Gambar 2.6 Ilustrasi dari Operator Mutasi Inversi	16
Gambar 2.7 Ilustrasi dari Operator Mutasi Insersi.....	17
Gambar 3.1 Diagram Alir Penyelesaian Penjadwalan	29
Gambar 3.2 Kromosom Penjadwalan	30
Gambar 3.3 Kromosom Setelah Disesuaikan Dengan Batasan	35
Gambar 3.4 Proses Crossover	42
Gambar 3.5 Proses Mutasi	44
Gambar 3.6 Proses Optimasi Penjadwalan Pengiriman Barang	46
Gambar 3.7 Context Diagram Penjadwalan Pengiriman Barang.....	47
Gambar 3.8 DFD Level 0.....	48
Gambar 3.9 DFD Level 1 Proses 1	49
Gambar 3.10 DFD Level 2 Proses 1.2	50
Gambar 3.11 ERD <i>Conceptual Data Model</i>	51
Gambar 3.12 ERD <i>Physical Data Model</i>	52
Gambar 4.1 Menu Login.....	56

Gambar 4.2	Menu Utama.....	56
Gambar 4.3	Master Kapal.....	57
Gambar 4.4	Master Tujuan.....	57
Gambar 4.5	Master Jarak.....	58
Gambar 4.6	Master Costumer.....	58
Gambar 4.7	Form Proses Optimasi Jadwal danRute.....	59
Gambar 4.8	Input Kapal.....	60
Gambar 4.9	Input Tujuan.....	60
Gambar 4.10	Detail Proses.....	60
Gambar 4.11	Laporan Jadwal dan Rute Pengiriman Barang.....	61
Gambar 4.12	Grafik Uji Coba I Generasi.....	63
Gambar 4.13	Grafik Uji Coba 10 Generasi.....	63
Gambar 4.14	Jadwal Hasil Uji Coba 10 Generasi.....	64
Gambar 4.15	Grafik Uji Coba II Populasi.....	65
Gambar 4.16	Grafik Uji Coba 40 Populasi.....	66
Gambar 4.17	Jadwal Hasil Uji Coba 40 Populasi.....	67
Gambar 4.18	Grafik Uji Coba III Probabilitas Crossover.....	68
Gambar 4.19	Grafik Uji Coba Probabilitas Crossover 65%.....	69
Gambar 4.20	Jadwal Uji Coba Probabilitas Crossover 65%.....	70
Gambar 4.21	Grafik Uji Coba IV Probabilitas Mutasi.....	71
Gambar 4.22	Grafik Uji Coba Probabilitas Mutasi 5%.....	72
Gambar 4.23	Jadwal dan Rute Pengirim Barang.....	73

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Data Jarak.....	78
Lampiran 2 Biodata Penulis	79
Lampiran 3 Listing Program	80



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Distribusi suatu produk memegang peranan penting dalam suatu mata rantai produksi. Permasalahan distribusi tersebut mencakup kemudahan untuk mendapatkan suatu produk kapan dan dimana saja. Kemudahan mendapatkan produk ini mengandung suatu biaya kesempatan (opportunity cost) yaitu peluang memenangkan pasar ketika produksi lain tidak tersedia. Untuk itu distribusi merupakan hal yang sangat penting bagi suatu produk untuk memenangkan pasar.

Dalam pendistribusian suatu produk tentu saja hal yang paling relevan di dalamnya adalah mengenai transportasi produk. Di satu pihak, transportasi hanyalah merupakan suatu proses pemindahan suatu material yang tidak memberikan penambahan nilai pada produk dan sifatnya merupakan biaya (cost). Atau dengan kata lain, penambahan jarak transportasi akan memberikan biaya transportasi yang lebih besar pula. Oleh karena itu, salah satu upaya pengurangan biaya transportasi dapat dilakukan dengan mengoptimalkan waktu transportasi dimana waktu transportasi didapat dari jarak transportasi.

Pelayaran PT Surya bergerak dalam bidang jasa pendistribusian, yaitu mengirimkan barang suatu perusahaan dari suatu tujuan ke tujuan lain melalui laut dengan menggunakan kapal. Setiap kapal pengiriman barang memiliki rute tujuan pengiriman yang berbeda – beda. Dengan banyaknya kapal pengiriman yang memiliki rute tujuan yang berbeda ini maka pelayaran PT Surya memiliki tanggung jawab untuk membuat jadwal bagi setiap kapal agar mendapatkan hasil

pengiriman barang datang tepat waktu serta biaya operasional kapal juga dapat diminimalisasi. Petugas yang menyusun jadwal pengiriman barang tersebut menyusun jadwal rute tujuan kapal secara manual sehingga minimalisasi rute kapal tersebut masih belum dapat di katakan optimal.

Berdasarkan kondisi tersebut untuk menghindari pembengkakan biaya operasional, maka salah satu cara yang dapat ditempuh adalah dengan melakukan optimalisasi rute pengiriman. Karena jumlah rute tujuan yang harus dioptimasi adalah lebih dari satu tujuan pengiriman dengan jarak tempuh yang berbeda-beda, maka hal tersebut akan sulit untuk melaksanakannya dengan menggunakan proses manual. Sehingga dalam penelitian ini digunakan metode optimasi dengan algoritma genetik.

Dengan algoritma genetik diharapkan dapat membantu dalam penentuan jumlah kapal yang digunakan serta rute pengiriman barang yang harus di tempuh, sehingga menghasilkan jadwal rute pengiriman yang optimal serta biaya operasional kapal dapat di minimalkan.

1.2 Perumusan Masalah

Mengacu pada latar belakang masalah di atas, maka perumusan masalah adalah “Bagaimana mengoptimalkan penjadwalan dan rute pengiriman barang menggunakan algoritma genetika pada pelayaran PT. Surya, sehingga dihasilkan waktu tempuh yang dilalui adalah minimum”.

1.3 Batasan Masalah

Batasan masalah dalam perancangan dan penjadwalan pada tugas akhir ini adalah sebagai berikut :

1. Layanan pengiriman barang yang dimaksud adalah pengiriman barang dari PT Surya yaitu yang bertempat di Surabaya ke berbagai tujuan pengiriman barang di Indonesia.
2. Tidak membahas penataan barang dalam kapal.
3. Jadwal pengiriman barang yang dibuat berdasarkan dari jumlah kapal yang kesemuanya dalam kondisi siap pakai, jika terjadi kerusakan maka kapal yang rusak atau tidak siap pakai tidak akan diinputkan.
4. Dapat mengangkut barang dari suatu tujuan pengiriman ke tujuan lainnya yang sesuai dengan rute dari jadwal pengiriman barang tiap kapal.
5. Cuaca laut diasumsikan dalam kondisi baik. Sehingga tidak mengulur waktu pengiriman.

1.4 Tujuan

Tujuan dari penelitian ini adalah untuk melakukan rancang bangun sistem optimasi rute dan penjadwalan pengiriman barang pelayaran PT Surya dengan menggunakan algoritma genetika sebagai algoritma untuk mencari jalur optimal dengan waktu yang minimum.

1.5 Sistematika Penulisan

Di dalam penulisan Tugas Akhir ini secara sistematika diatur dan disusun dalam lima bab, yaitu:

BAB I : Pendahuluan

Bab ini akan menjelaskan tentang seluruh latar belakang, perumusan masalah, pembatasan masalah, tujuan, dan

sistematika penulisan yang berhubungan dengan penelitian tugas akhir ini.

BAB II : Landasan Teori

Berisikan teori – teori yang digunakan sebagai landasan dalam desain dan implementasi sistem.

BAB III : Metode Penelitian

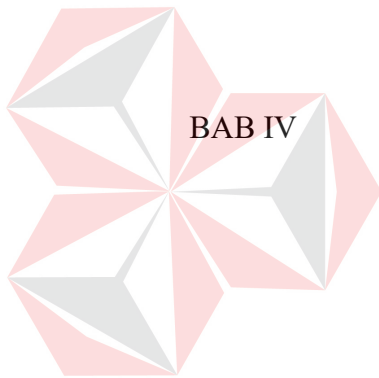
Berisikan tentang langkah-langkah mendesain sistem mulai dari identifikasi, perumusan masalah sampai dengan mendesain Sistem Flow, ERD, Context Diagram, DFD, Struktur File, Sistem Informasi, desain input-output dan kebutuhan perangkat lunak serta perangkat kerasnya.

BAB IV : Evaluasi dan Implementasi

Bab ini adalah merupakan proses evaluasi perangkat lunak sehingga menghasilkan perangkat lunak sesuai dengan kebutuhan dan di lanjutkan ke proses implementasi.

BAB V : Penutup

Bab ini berisi kesimpulan dan saran yang diambil dari penelitian dan pembahasan.



UNIVERSITAS
Dinamika

BAB II

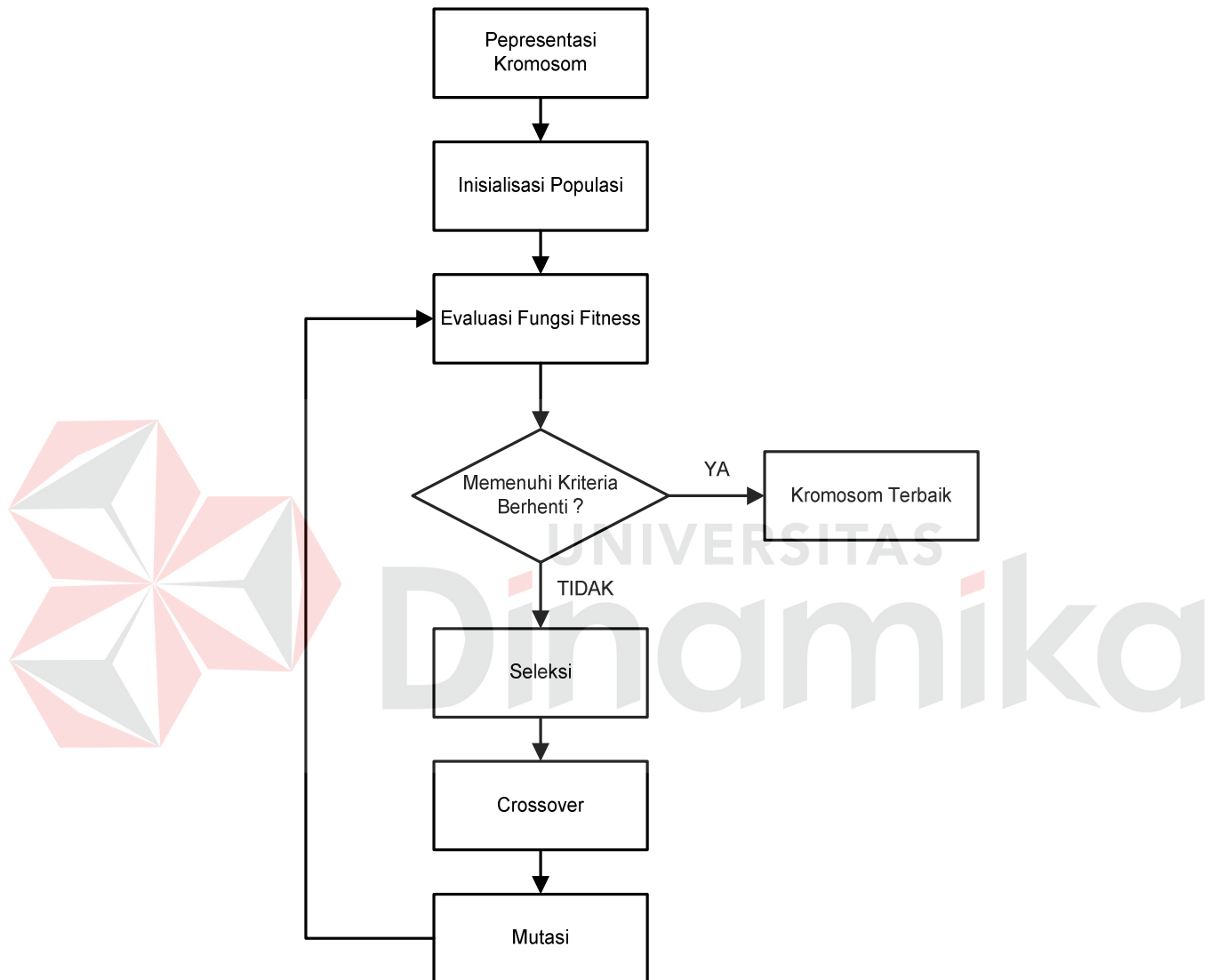
LANDASAN TEORI

2.1 Algoritma Genetika

Algoritma genetika merupakan suatu metode yang menggunakan seleksi alam yang merupakan bagian utama dari prinsip evolusi sebagai dasar pemikiran untuk menyelesaikan suatu permasalahan. Prinsip ini dikemukakan oleh Charles Darwin, bahwa individu yang mampu beradaptasi dengan lingkungannya akan mempunyai kesempatan hidup yang lebih besar. Penggunaan prinsip genetika pada komputer dimulai pada tahun 1950 ketika beberapa ahli Biologi menggunakan komputer untuk simulasi sistem biologi. Akhir tahun 1975 John Holland dari Universitas Michigan melalui paper yang berjudul "*Adaption in Natural and Artificial System*" menggunakan konsep dasar algoritma genetika.

Algoritma genetika bekerja dengan suatu populasi string dan melakukan proses pencarian nilai optimal secara paralel, dengan menggunakan operator genetika. Algoritma genetika akan melakukan rekombinasi antar individu. Algoritma genetika memiliki elemen dasar berupa string yang tersusun dari rangkaian substring (gen), yang masing-masing merupakan kode dari parameter dalam ruang solusi dimana suatu string (kromosom) menyatakan kandidat solusi. Kumpulan string dalam populasi berkembang dari generasi ke generasi melalui operator genetika. Pada setiap iterasi, individu-individu (kromosom) dalam populasi itu akan dievolusi dan diseleksi untuk menentukan populasi pada generasi berikutnya. Populasi ini akan terus berulang sampai menemukan suatu

parameter dengan nilai yang paling optimal sesuai dengan yang diinginkan. Adapun struktur umum algoritma genetika dapat diilustrasikan pada Gambar 2.1. Kusumadewi (2005:250).



Gambar 2.1 Struktur Umum Algoritma Genetika

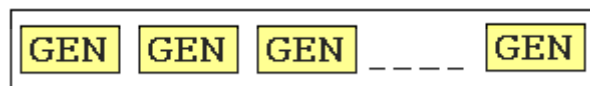
Istilah yang digunakan dalam algoritma genetika ini hampir sama dengan istilah yang dipakai dalam bidang biologi genetika, antara lain gen, kromosom, populasi, fungsi *fitness*, dan operator genetika yang meliputi mutasi dan *crossover*.

1. Gen

Gen adalah suatu sel dari suatu kromosom atau nilai yang terdapat dalam algoritma genetika ini dapat dibentuk oleh sebuah *byte* bahkan tidak menutup kemungkinan suatu string. Gen ini mewakili sebagian kecil dari solusi permasalahan.

2. Kromosom

Individu dalam populasi disebut kromosom. Kromosom terdiri dari unit yang dinamakan gen atau karakter. Kromosom ini dapat mewakili suatu solusi, dimana dapat diilustrasikan dalam Gambar 2.2.



Gambar 2.2 Kromosom dalam Algoritma Genetika

3. Populasi

Populasi adalah sekumpulan kromosom yang berisi solusi- solusi.

4. Fungsi Evaluasi (Fungsi Fitness)

Fungsi evaluasi adalah suatu formula yang digunakan untuk menilai kualitas suatu kromosom dan digunakan pada setiap generasi.

2.1.1 Menentukan Representasi Kromosom

Untuk mempresentasikan kromosom dilakukan dengan proses *encoding*, dibawah ini akan dijelaskan beberapa proses *encoding* yang biasa digunakan dalam beberapa kasus tertentu.

A. *Permutation Encoding*

Untuk jenis *Permutation Encoding* ini digunakan untuk permasalahan proses pengurutan, misalnya terdapat kasus optimasi jadwal atau pada kasus

traveling salesman. Pada *Permutation Encoding*, setiap Gen pada kromosom berupa angka dimana dapat ditampilkan seperti Gambar 2.3.

Kromosom A	1 5 3 1 6 4 8 9 8
Kromosom B	8 5 6 7 2 3 1 4 9

Gambar 2.3 Permutation Encoding

Permutation Encoding berlaku untuk permasalahan pengurutan. Contoh penggunaan *Permutation Encoding* ini adalah pada kasus *travelling sales problem*, dimana terdapat beberapa kota dengan jarak masing-masing. Pada kasus *traveling sales problem* ini seorang *salesman* harus mengunjungi semua kota yang ada, dan menentukan urutan kota yang akan dikunjungi untuk meminimalisasi jarak yang harus ditempuh.

B. Binary Encoding

Binary encoding adalah jenis encoding yang paling sering digunakan karena kasus pertama yang ada pada Algoritma Genetika menggunakan algoritma jenis ini. Setiap kromosom pada *Binary encoding* merupakan bit 0 dan 1 di mana dapat ditampilkan pada Gambar 2.4.

Kromosom A	1 0 0 1 1 0 1 0 1 1 0 1 0 0 1 0
Kromosom B	1 0 0 1 0 1 1 0 0 1 0 1 1 0 0 1

Gambar 2.4 Binary Encoding

Binary encoding dapat memberikan banyak kemungkinan pada kromosom meskipun pada jumlah gen yang sedikit. Di lain pihak jenis *encoding* ini tidak cukup natural untuk beberapa kasus tertentu dan kadang-kadang harus dilakukan koreksi setelah melakukan *crossover* atau mutasi, contoh penggunaan *Binary encoding* adalah pada permasalahan *knapsack* atau pengepakan, di mana

ada beberapa barang dengan jumlah dan ukuran masing-masing dan *knapsack* harus memberikan kapasitasnya untuk barang-barang tersebut, permasalahannya adalah bagaimana memilih barang untuk memaksimalkan jumlah barang sehingga dapat ditampung oleh *knapsack* tanpa harus menambah kapasitasnya.

C. *Value Encoding*

Penempatan nilai yang ada secara langsung dapat diterapkan pada *Value Encoding* jenis ini dapat menampung nilai yang rumit dan berbeda seperti tipe real. Penggunaan *Binary encoding* untuk jenis permasalahan ini sangatlah sulit dilakukan, setiap kromosom pada *Value Encoding* adalah suatu nilai tertentu, dimana nilai tersebut dapat berupa bilangan desimal (5,32,3,9 atau lainnya), bilangan real (1.4545,0.3543, 3.6,5.468 atau lainnya), *character* sampai dengan tipe obyek yang rumit

Value Encoding dapat digunakan untuk permasalahan yang spesial. Contoh penggunaan *Value Encoding* adalah menentukan bobot untuk jaringan neural, dimana jaringan neural tersebut telah diberikan arsitektur tertentu.

Permasalahannya adalah bagaimana mencari bobot untuk masukan neural yang digunakan untuk memproses jaringan neural untuk menghasilkan output yang diharapkan.

2.1.2 Inisialisasi Populasi

Proses inisialisasi adalah pembentukan populasi awal yang diperoleh dengan cara rekombinasi kromosom sebanyak ukuran populasi yang ditentukan. Pada inisialisasi populasi ditentukan nilai awal dari sejumlah untaian kromosom yang juga merupakan kemungkinan solusi permasalahan yang dicari. Populasi

awal ini dapat dibangun dengan cara acak. Dalam inisialisasi populasi ini tetap memperhatikan batasan-batasan yang akan digunakan dalam suatu permasalahan yang akan diselesaikan.

Adapun dalam inisialisasi populasi menggunakan batasan-batasan sebagai berikut :

$$\sum_{j=1}^n C_j \leq C \dots\dots\dots (1)$$

$$T_{tot} = \sum_{i,j=1}^n (t_{ij} + St_j) \dots\dots\dots (2)$$

dimana $t_{ij} = \frac{d_{ij}}{v} \dots\dots\dots (3)$

Batasan (1) menunjukkan bahwa permintaan (order) tujuan tidak boleh melebihi dari kapasitas dari angkut kapal. Sedangkan batasan (2) menunjukkan total waktu satu kapal mengunjungi seluruh tujuan yaitu dengan menjumlahkan waktu tempuh tujuan awal ke tujuan akhir dengan waktu yang diperlukan untuk melayani (bongkar muat) per tujuan. Dimana rumusan (3) adalah waktu tempuh tujuan awal ke tujuan akhir didapat dari jarak tempuh tujuan awal ke tujuan akhir dibagi dengan kecepatan kapal. Parameter yang digunakan adalah : n sebagai banyaknya tujuan, C_j sebagai permintaan (order) tujuan ke- j , C sebagai kapasitas angkut kapal, $T_{tot}(r)$ sebagai total waktu kapal (r) mengunjungi seluruh tujuan, St_j sebagai waktu yang diperlukan untuk melayani per tujuan (bongkar muat), t_{ij} sebagai waktu tempuh dari tujuan ke- i ke tujuan ke- j , d_{ij} sebagai jarak tempuh dari tujuan ke- i ke tujuan ke- j , v sebagai kecepatan kapal.

2.1.3 Menentukan Rumusan Fungsi Fitness

Perhitungan pada permasalahan penjadwalan yang timbul dapat di formulasikan sebagai berikut :

$$Eval(K_i) = \sum_{r=1}^n Ttot(r) \dots\dots\dots (4)$$

dimana, $r = 1,2,3,\dots,n$

$i = 1,2,3,\dots,max\ pop_size$

Untuk rumusan fungsi fitness (4) menunjukkan jumlah total waktu tempuh dari suatu kapal. Parameter yang digunakan adalah : r sebagai kapal, K_i sebagai kromosom ke $-i$.

2.1.4 Parameter Genetika

A. Ukuran Populasi (uk_pop)

Ukuran populasi mempengaruhi kinerja dan efektifitas dari algoritma genetika. Jika ukuran populasi kecil maka populasi tidak menyediakan cukup materi untuk mencakup ruang permasalahan, sehingga pada umumnya kinerja algoritma genetika menjadi buruk dan apabila penggunaan populasi yang besar dapat mencegah terjadinya konvergensi pada wilayah lokal.

B. Jumlah Generasi

Merupakan jumlah perulangan (iterasi) dilakukannya rekombinasi dan seleksi. Jumlah generasi ini mempengaruhi kestabilan output dan lama iterasi (waktu proses algoritma genetika). Jumlah generasi yang besar dapat mengarahkan ke arah solusi yang optimal, namun akan membutuhkan waktu

komputasi yang lama. Sedangkan jika jumlah generasinya terlalu sedikit maka solusi akan terjebak pada lokal optimal.

C. Probabilitas Crossover (P_c)

Probabilitas crossover digunakan untuk mengendalikan frekuensi operator crossover. Probabilitas *crossover* menyatakan seberapa banyak *crossover* yang dilakukan sehingga kromosom yang melakukan *crossover* akan menghasilkan suatu individu baru yang sama dengan induknya. Jika melakukan *crossover* maka akan dihasilkan individu yang memiliki sifat dari kromosom induknya. Jika probabilitas *crossover*nya 100% maka semua kromosom yang terbentuk adalah dari proses crossover dan jika probabilitas *crossover*nya 0% maka kromosom yang terbentuk berasal dari kromosom sebelumnya tetapi belum tentu generasi yang baru sama dengan generasi sebelumnya. Pada umumnya sebaiknya angka probabilitas crossover adalah tinggi, sekitar 80% sampai 95% (tetapi pada kasus-kasus tertentu angka probabilitas crossover adalah 60% akan menghasilkan solusi yang lebih baik). Dengan proses *crossover* diharapkan generasi yang baru akan membawa sifat yang baik dari kromosom induknya sehingga kromosom baru yang terbentuk akan memberikan solusi yang optimal. Dalam hal ini, pada populasi terdapat $P_c * uk_pop$ struktur (individu) yang melakukan crossover.

D. Probabilitas Mutasi (P_m)

Mutasi digunakan untuk meningkatkan variasi populasi. Probabilitas mutasi ini menentukan berapa banyak gen dari kromosom melakukan proses mutasi karena frekuensi terjadinya mutasi tersebut menjadi $P_m * uk_pop * N$, dimana N adalah panjang struktur (gen) dalam suatu individu.. Apabila tidak

dilaksanakan proses mutasi maka kromosom yang dihasilkan tepat sama dengan induknya. Probabilitas mutasi 100% berarti semua kromosom akan berubah dan jika probabilitas 0% berarti tidak ada bagian kromosom yang berubah. Mutasi dilakukan untuk mencegah algoritma genetika terjebak dalam *local extreme*, tetapi proses ini tidak boleh dilakukan terlalu sering karena algoritma genetika akan berubah menjadi pencarian acak atau *random search*.

2.1.5 Proses Seleksi

Proses ini digunakan untuk menentukan kromosom-kromosom yang akan bertahan dan masuk pada generasi selanjutnya. Penentuan suatu kromosom akan dimasukkan pada generasi selanjutnya didasarkan pada nilai fitness untuk tiap-tiap kromosom. Semakin baik nilai fitness maka probabilitas kromosom tersebut akan terpilih semakin besar. Metode yang sering digunakan dalam proses seleksi ini adalah roulette wheel.

2.1.6 Operasi Genetik

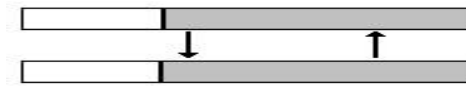
A. *Crossover*

Crossover adalah operator algoritma genetika yang membutuhkan parameter dua kromosom. Dua buah kromosom tersebut disebut kromosom induk. Operator ini akan menghasilkan dua buah kromosom baru. Ada beberapa jenis *crossover* yang sering digunakan dalam algoritma genetika antara lain:

A.1 *One-Point Crossover*

Contoh kerja operator ini adalah dengan menentukan *crossover point* (gen tertentu). Kromosom baru pertama berisi gen pertama sampai gen *crossover*

point dari kromosom induk pertama ditambah dengan gen dari *crossover point* sampai gen terakhir dari kromosom induk kedua. Kromosom baru kedua berisi gen pertama sampai gen *crossover point* dari induk kedua ditambahkan dengan gen dari *crossover point* sampai gen dari kromosom induk pertama.



Gambar 2.5. Proses Crossover dengan Satu Crossover Point

Dari ilustrasi di atas maka contoh penerapan metode *One-Point Crossover* adalah sebagai berikut:

Parent 1: 1 2 3 4 5 | 6 7 8 9

Parent 2: 4 5 3 6 8 | 9 7 2 1

Setelah proses *crossover* turunan yang dapat dihasilkan adalah dari kedua *parent* di atas adalah:

Parent 1: 1 2 3 4 5 | 9 7 2 1

Parent 2: 4 5 3 6 8 | 6 7 8 9

A.2 *Two-Point Crossover*

Proses *Two-Point Crossover* hampir sama dengan prosedur *One-Point Crossover*, kecuali pada *Two-Point Crossover* harus dipilih dua *crossover point* dan hanya gen yang ada di antara kedua *crossover point* itu yang akan ditukarkan.

Metode ini dapat menjadi bagian awal dan akhir dari kromosom dan hanya menukar bagian tengahnya saja.

A.3 *N-Point Crossover*

Prosedur *N-Point Crossover* hampir sama baik dengan prosedur *one-point crossover* maupun *two-point crossover*, hanya saja dalam *n-point crossover* ini harus dipilih n *crossover point* dan hanya gen di antara *crossover point* ganjil

dan genap yang dapat ditukarkan sedangkan gen diantara genap dan ganjil operator *crossover* tidak berubah. Atau dengan kata lain harus dipilih posisi n dan hanya bit antara ganjil dan genap posisi *crossover* yang akan dihilangkan.

Contoh: $P1 = 9\ 7\ 6\ 3\ 2\ 8$
 $P2 = 2\ 1\ 9\ 7\ 4\ 5$

Jika didapatkan angka random untuk $n=3$ dan diacak 2, 5 dan 6 sebagai posisi dari gen yang akan di *crossover*, didapatkan kromosom turunan:

$T1 = 9\ 1\ 6\ 3\ 4\ 5$
 $T2 = 2\ 7\ 9\ 7\ 2\ 8$

B. Operator Mutasi

Mutasi adalah operator yang membutuhkan satu parameter. Kromosom operator ini merupakan nilai suatu gen dari sebuah kromosom sehingga kromosom yang baru ini berbeda dengan kromosom yang lama. Seluruh proses mutasi ini menjanjikan keuntungan melalui pengarahannya ke mana mutasi ini tersebut sangat dibutuhkan. Operator mutasi digunakan untuk melakukan modifikasi satu atau lebih dari nilai gen dalam individu yang sama.

Langkah-langkah algoritma genetika mutasi :

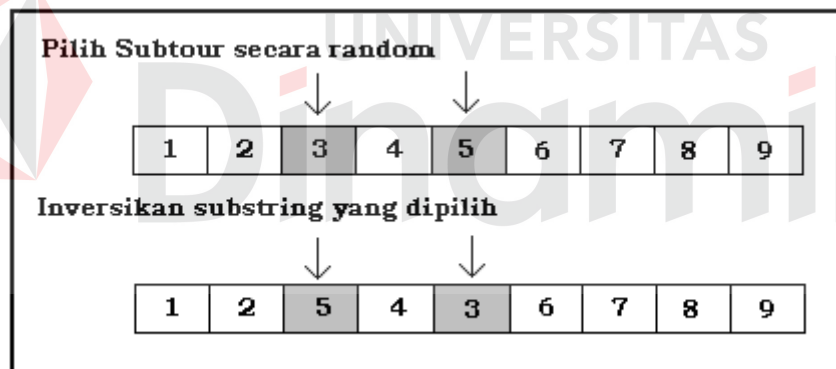
Step 1 = [Tentukan kromosom atau *parent* yang akan dimutasi]
 $i \leftarrow \text{random } N-1$
 $\text{Pos} \leftarrow \text{Pop}[i]$
 Step 2 = [Tentukan posisi *allele* yang akan dimutasi]
 $K \leftarrow \text{random } N-1$
 $L \leftarrow \text{random } N-1$
 Step 3 = [salin *allele* pada posisi K ke dalam variable temp]
 $\text{Temp} \leftarrow \text{Pop}[K]$
 Step 4 = [Lakukan pertukaran *allele*]
 $\text{Pop}[K] \leftarrow \text{Pop}[L]$
 $\text{Pop}[L] \leftarrow \text{temp}$
 Step 5 = [Selesai]
 Return

N adalah jumlah kromosom yang akan diproses. Pop adalah kromosom yang akan dimutasi. Variabel K dan L digunakan untuk menyimpan posisi dari *allele* yang akan di tukarkan dalam proses mutasi. Variabel temp digunakan untuk menyimpan nilai *allele* yang ditunjuk oleh posisi K.

Terdapat empat operator yang biasa digunakan untuk representasi permutasi, yaitu :

B.1 Mutasi Inversi (Inversion Mutation)

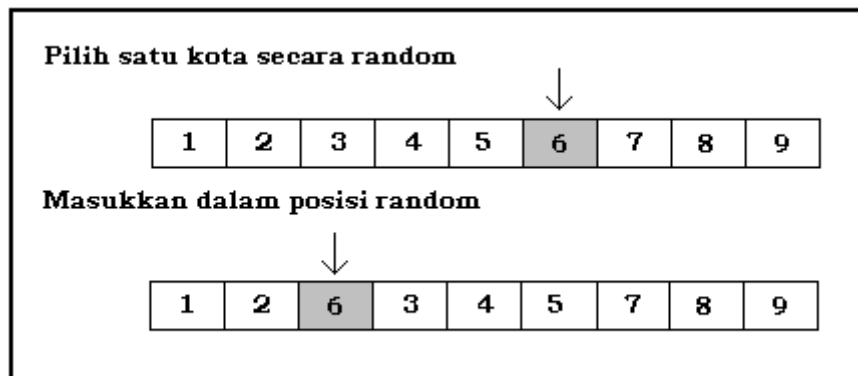
Mutasi ini bekerja pada satu kromosom dan membalik urutan diantara dua gen yang dipilih secara acak pada kromosom. Mutasi Inversi memilih dua posisi didalam kromosom secara random dan kemudian menginversikan gen antara dua posisi ini.



Gambar 2.6 Ilustrasi dari Operator Mutasi Inversi

B.2 Mutasi Inseri (Insertion Mutation)

Proses yang dilakukan untuk jenis mutasi inseri ini adalah dengan memilih secara random gen dalam sebuah kromosom dan menyisipkannya pada posisi lain secara random juga sehingga dihasilkan sebuah kromosom baru.



Gambar 2.7 Ilustrasi dari Operator Mutasi Insersi

2.1.7 Kriteria Berhenti

Setelah melakukan setiap proses diatas, maka kromosom–kromosom tersebut telah melakukan satu iterasi algoritma genetik yang disebut generasi. Dan proses algoritma genetika akan dilakukan berulang-ulang sampai memenuhi salah satu kriteria dibawah ini yaitu :

1. Setelah dalam beberapa generasi berturut-turut didapatkan nilai fitness yang tidak berubah.
2. Bila dalam generasi ke-n tidak didapatkan nilai fitness yang lebih baik.
3. Telah mencapai generasi ke – N.

2.2 Rute Kendaraan

2.2.1 Vehicle Routing Problem (VRP)

Dasar permasalahan pembentukan rute adalah adanya sekumpulan node atau busur yang harus dilayani oleh suatu armada kapal. Permasalahannya adalah untuk membentuk suatu biaya atau waktu yang rendah dari sekumpulan rute yang memungkinkan untuk masing – masing kapal angkut.

Kallehaug et al. (2001) mendefinisikan permasalahan *vehicle routing problem* (VRP) bahwa sebuah kota diasosiasikan sebagai sebuah *demand* atau konsumen, dan setiap kendaraan yang dipakai untuk perjalanan dianggap memiliki kapasitas tertentu. Total jumlah *demand* dalam suatu rute, tidak boleh melebihi kapasitas dari kendaraan yang ditugasi melewati rute tersebut. Hal ini membuat VRP kadang juga disebut sebagai *Capacitated Vehicle Routing Problem*. Sama seperti permasalahan *Travelling Salesman Problem* (TSP), dalam VRP juga terdapat suatu *depot*, dimana tiap kendaraan harus berangkat dan kembali ke *depot* itu. Dalam VRP, selain bertujuan untuk meminimalkan total jarak atau total biaya *travel*, dapat juga untuk meminimalkan jumlah kendaraan yang digunakan (m).

Prins (2001) menggambarkan permasalahan VRP sebagai suatu *undirected network* $G=(V, E)$, dengan sebuah *node set* $V=\{0, 1, \dots, n\}$, dan sebuah *edge set* E . *Node 0* adalah sebuah *depot*, dengan sejumlah kendaraan yang mempunyai kapasitas yang sama atau identik, Q . Tiap klien *node* $i>0$, memiliki suatu *demand non negatif* q_i , dan tiap *edge* $[i, j]$ memiliki biaya *non negatif* $c_{ij}=c_{ji}$. Permasalahan VRP bertujuan untuk menentukan suatu *set trips* kendaraan dengan total biaya minimal, dimana tiap *trip* berawal dan berakhir di *depot*, tiap klien dikunjungi tepat satu kali, total *demand* yang dibawa oleh tiap kendaraan tidak melebihi kapasitas kendaraan Q , dan biaya dari tiap *trip* tidak melebihi *upper limit* L , yang telah ditentukan. Pada penelitian yang dilakukan oleh Prins (2001), variabel keputusannya adalah jumlah kendaraan.

Thangiah (1995) merumuskan model *mixed-integer programming* untuk permasalahan VRP. Parameter yang digunakan antara lain: K sebagai nomer

kendaraan, N sebagai nomer konsumen (0 menunjukkan *depot*), C_i menunjukkan konsumen i , C_0 menyatakan *depot*. Selanjutnya V_k sebagai rute kendaraan k , c_{ijk} adalah biaya *travel* antara konsumen i dan j untuk kendaraan k , q_{ik} menyatakan total demand kendaraan k sampai konsumen i , dan v_k adalah kapasitas maksimum kendaraan k .

Sedangkan, variabelnya didefinisikan sebagai :

$$y_{ik} = \begin{cases} 1 & \text{1, jika konsumen } i \text{ dilayani oleh kendaraan } k; \\ 0 & \text{0, jika tidak demikian} \end{cases}$$

$$x_{ijk} = \begin{cases} 1 & \text{1, jika kendaraan } k \text{ dari konsumen } i \text{ langsung ke konsumen } j \\ 0 & \text{0, jika tidak demikian} \end{cases}$$

Formulasi mixed-integer programming untuk VRP ini adalah :

Fungsi tujuan :

$$\text{Min} \sum_{i=0}^N \sum_{j=0}^N \sum_k^K c_{ij} x_{ijk} \dots\dots\dots (5)$$

dengan kendala – kendala :

$$\sum_{i=0}^N q_{ik} y_{ik} \leq v_k, k = 1, 2, \dots, K \dots\dots\dots (6)$$

$$y_{ik} = 0 \text{ atau } 1 ; i = 1, 2, \dots, N ; k = 1, 2, \dots\dots\dots (7)$$

$$x_{ijk} = 0 \text{ atau } 1 ; i = 1, 2, \dots, N ; k = 1, 2, \dots, K \dots\dots\dots (8)$$

$$\sum_{k=1}^K y_{ik} = \begin{cases} K, & i=0 \\ 1, & i=1, \dots, N \end{cases} \dots\dots\dots (9)$$

$$\sum_{i=0}^N x_{ijk} = y_{jk}, j = 0, \dots, N; k = 1, \dots, K \dots\dots\dots (10)$$

$$\sum_{j=0}^N x_{ijk} = y_{jk}, i = 1, \dots, N; k = 1, \dots, K \dots\dots\dots (11)$$

Tujuan dari model ini adalah untuk meminimalkan total biaya *travel*. Kendala (6) membatasi bahwa total jumlah *demand* yang dibawa oleh kendaraan k tidak boleh melebihi kapasitas dari kendaraan tersebut. Kendala (9) untuk menunjukkan bahwa tiap konsumen hanya dapat dilayani oleh satu kendaraan saja. Kendala (10), dan (11) digunakan untuk memastikan bahwa tiap konsumen dikunjungi oleh kendaraan yang sama dengan yang sudah dijadwalkan untuk konsumen tersebut.

Sedangkan, Kallehauge dkk.(2001) memodelkan VRP sebagai:

Fungsi tujuan :

$$\text{Min } \sum_{k=1}^K \sum_{i=0}^{N+1} \sum_{j=0}^{N+1} c_{ij} x_{ijk} \dots\dots\dots (12)$$

dengan kendala-kendala :

$$\sum_{k=1}^K \sum_{j=0}^{N+1} x_{ijk} = 1; \quad i = 1, 2, \dots, N \dots\dots\dots (13)$$

$$\sum_{i=1}^N d_i \sum_{j=0}^{N+1} x_{ijk} \leq v_k; \quad k = 1, 2, \dots, K \dots\dots\dots (14)$$

$$\sum_{j=0}^{N+1} x_{ojk} = 1; \quad k = 1, 2, \dots, K \dots\dots\dots (15)$$

$$\sum_{i=0}^{N+1} x_{ihk} - \sum_{j=0}^{N+1} x_{hjk} = 0; \quad h = 1, 2, \dots, N; k = 1, 2, \dots, K \dots\dots\dots (16)$$

$$\sum_{i=0}^{N+1} x_{i, N+1, k} = 1; \quad k = 1, 2, \dots, K \dots\dots\dots (17)$$

$$x_{ijk} \in \{0, 1\}; \quad i = 0, 2, \dots, N+1; k = 1, 2, \dots, K \dots\dots\dots (18)$$

Fungsi tujuan dari model VRP Kallehauge dkk. (2001), sama seperti fungsi tujuan dari model yang dibuat Thangiah (1995), yaitu untuk meminimalkan

total biaya *travel*. Tetapi Thangiah hanya memperhitungkan biaya *travel* untuk perjalanan awal dari *depot*, kemudian mengunjungi semua konsumen yang ada saja, tanpa memperhitungkan perjalanan kembali ke *depot*, pada akhir perjalanan tersebut. Sedangkan Kallehauge dkk. (2001), memperhitungkan biaya *travel* untuk perjalanan awal dari *depot*, kemudian mengunjungi semua konsumen yang ada saja, dan juga perjalanan kembali ke *depot*, pada akhir perjalanan tersebut.

Kendala (13) mempunyai kegunaan yang sama seperti kendala (9), yaitu untuk menunjukkan bahwa tiap konsumen hanya dapat dilayani oleh satu kendaraan saja. Kendala (14) mempunyai kegunaan yang sama seperti kendala (6), yaitu untuk membatasi bahwa total jumlah *demand* yang dibawa oleh kendaraan k , tidak boleh melebihi kapasitas dari kendaraan tersebut. Kendala (15)–(17) digunakan untuk memastikan bahwa tiap kendaraan berangkat dari *depot* 0, dan setelah selesai melayani seorang konsumen, kendaraan tersebut akan pergi, serta pada akhirnya, kendaraan tersebut akan kembali ke *depot* $N+1$. Pada model yang dibuat oleh Tangiah (1995), tidak terdapat kendala yang mempunyai kegunaan seperti kendala (15)–(17).

2.2.2 Vehicle Routing Problem Dengan Time Windows

Menurut Kallehauge dkk. (2001), *vehicle routing problem* dengan *time windows* (VRPTW) adalah perluasan dari VRP. Jika pada VRP ditambahkan *time window* pada masing–masing konsumen, maka permasalahan tersebut menjadi VRPTW. Untuk VRPTW, selain adanya kendala kapasitas kendaraan, terdapat tambahan kendala yang mengharuskan kendaraan untuk melayani tiap konsumen pada *time frame* tertentu. Kendaraan boleh datang sebelum *time window* “*opens*”,

tetapi konsumen tersebut tidak dapat dilayani sampai *time window* “opens”. Kendaraan tidak diperbolehkan untuk datang setelah *time window* “closed”.

Tangiah (1995) mendefinisikan VRPTW sebagai permasalahan untuk menjadwalkan sekumpulan kendaraan, dengan kapasitas dan *travel time* terbatas, dari *central depot* ke sekumpulan konsumen yang tersebar secara geografis, dengan *demand* diketahui, dalam *time windows* tertentu. *Time windows* adalah *two sided*, yang berarti bahwa tiap konsumen harus dilayani saat atau setelah *earliest time*, dan sebelum *latest time* dari konsumen tersebut. Jika kendaraan datang ke konsumen sebelum *earliest time* dari konsumen tersebut, maka akan menghasilkan *idle* atau waktu tunggu. Kendaraan yang datang ke konsumen setelah *latest time* adalah *tardy*. Terdapat pula waktu *service* yang diperlukan untuk melayani tiap konsumen. Biaya rute dari suatu kendaraan adalah total dari waktu *travel* (proporsional dengan jarak), waktu tunggu, dan waktu *service*, yang diperlukan untuk mengunjungi sekumpulan konsumen.

Homberger dkk. (1999) mendefinisikan permasalahan VRPTW sebagai berikut: n konsumen akan dilayani dari sebuah *depot*, dengan sejumlah kendaraan yang memiliki kapasitas, Q , yang sama. Untuk tiap konsumen i , $i=1, 2, \dots, n$, terdapat *demand* q_i , waktu *service* s_i , dan *service time window* $z_i=[e_i, f_i]$. *Lower bound* e_i merupakan waktu paling awal untuk melakukan *service*, dan *upper bound* f_i , waktu paling lambat untuk melakukan *service*. *Demand* q_i dari konsumen i harus dipenuhi dengan sekali *service* saja, dalam batas *time window* z_i . Sebagai tambahan, e_0 merupakan waktu paling awal untuk kendaraan berangkat dari *depot* i , $i=0$, dan f_0 merupakan waktu paling lambat untuk kendaraan kembali ke *depot*. Data mengenai lokasi dari *depot*, dan konsumen,

jarak terpendek d_{ij} , serta waktu $travel$ d'_{ij} antara dua lokasi, diketahui. Tujuannya untuk menentukan jadwal rute yang *feasible*, yaitu pertama untuk meminimalkan jumlah kendaraan, dan kedua untuk meminimalkan total jarak $travel$. Konsumen tidak dapat dilayani diluar *time window* mereka masing-masing. Tetapi kendaraan diperbolehkan untuk datang sebelum *lower bound* dari *time window*. Apabila hal ini terjadi, maka kendaraan harus menunggu sampai batas waktu paling awal *service* tersebut dapat dilakukan.

Ketiga definisi VRPTW diatas, membahas VRPTW dengan *hard time windows*, yaitu tiap kendaraan diperbolehkan untuk sampai ke konsumen i sebelum waktu pelayanan paling awal konsumen tersebut (e_i), tetapi tidak diperbolehkan datang melewati batas waktu pelayanan paling akhir konsumen itu (l_i). Bila kendaraan datang ke konsumen i sebelum batas waktu e_i , maka akan dikenakan waktu tunggu sampai batas waktu e_i tersebut.

Untuk memodelkan VRPTW, Tangiah memperkenalkan beberapa parameter yang perlu ditambahkan, yaitu: R_k sebagai total waktu rute untuk kendaraan k , t_{ij} untuk waktu $travel$ antara konsumen i dan j (proporsional dengan jarak *Euclidean*), t_i menyatakan waktu kedatangan di konsumen i , f_i sebagai waktu *service* di konsumen i , w_i adalah waktu tunggu sebelum melayani konsumen i , $w_i = \max\{0, (e_i - t_i)\}$, e_i sebagai waktu paling awal untuk pelayanan di konsumen i , dan l_i adalah waktu paling akhir untuk pelayanan di konsumen i .

Seperti halnya dengan VRP, fungsi tujuan VRPTW yaitu meminimalkan total biaya $travel$ semua kendaraan (8). Sedangkan semua kendalanya juga sama dengan kendala VRP [(9)-(14)], tetapi perlu ditambahkan beberapa kendala lagi

yang berhubungan dengan kendala *time windows*. Kendala-kendala yang perlu ditambahkan tersebut adalah:

$$\sum_{i=0}^N \sum_{i=0}^N y_{ik} (t_{ij} + f_i + w_i) \leq R_k; k = 1, \dots, K \dots\dots\dots (19)$$

$$t_j \geq t_i + w_i + f_i + t_{ij} - M(1 - x_{ijk}); i, j = 1, \dots, N; k = 1, \dots, K \dots\dots\dots (20)$$

$$e_i \leq t_i \leq l_i; i = 1, \dots, N \dots\dots\dots (21)$$

$$t_i \geq 0; i = 1, \dots, N \dots\dots\dots (22)$$

Kendala (19) digunakan untuk memastikan bahwa tiap kendaraan melayani semua konsumen yang dijadwalkan untuk kendaraan tersebut, tanpa melebihi waktu *travel* dari kendaraan tersebut. Kendala (20) untuk memastikan waktu kedatangan dari kedua konsumen adalah *compatible*. M merupakan bilangan riil yang sangat besar. Kendala (21) mengharuskan kendaraan untuk sampai di tiap-tiap konsumen selama batas *time window* dari konsumen tersebut. Kendala (22) memastikan bahwa waktu kedatangan kendaraan ke tiap konsumen selalu positif.

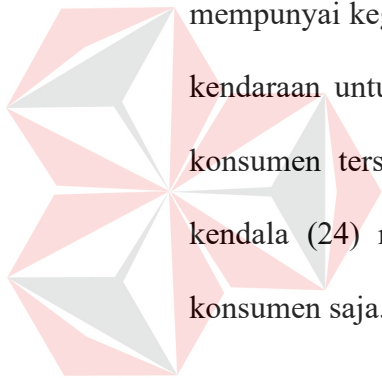
Untuk memodelkan VRPTW, Kallehauge dkk. (2001) juga menambahkan beberapa kendala ke model permasalahan VRP sebelumnya, yaitu yang terdapat pada persamaan [(12) – (18)]. Kendala yang ditambahkan tersebut adalah :

$$t_i + t_{ij} - M(1 - x_{ijk}) \leq t_j; i, j = 0, 1, \dots, N+1; k = 1, 2, \dots, K \dots\dots\dots (23)$$

$$e_i \leq t_i \leq l_i; i = 0, 1, \dots, N+1 \dots\dots\dots (24)$$

Parameter t_{ij} yang digunakan pada model ini memiliki arti yang berbeda dari parameter t_{ij} yang digunakan pada model yang dibuat oleh Tangiah. Pada

model yang dibuat oleh Kallehauge dkk. (2001) ini, t_{ij} berarti suatu waktu yang dimiliki oleh setiap $arc(i, j)$, dimana $i \neq j$. Jadi, t_{ij} pada model ini memuat waktu perjalanan antara konsumen i dan j , waktu tunggu di konsumen i , serta waktu *service* di konsumen i . Kendala (23) mirip dengan kendala (20), dan juga mempunyai kegunaan yang sama dengan kendala (20), yaitu untuk memastikan bahwa waktu kedatangan dari kedua konsumen adalah *compatible*. Perbedaan antara kendala (20), dan (23) adalah, bahwa kendala (23) mengikut sertakan *depot*, sedangkan kendala (20) hanya untuk konsumen saja, tanpa mengikut sertakan *depot*. Sedangkan kendala (24) juga mirip dengan kendala (21), dan juga mempunyai kegunaan yang sama dengan kendala (21), yaitu untuk mengharuskan kendaraan untuk sampai di tiap-tiap konsumen selama batas *time window* dari konsumen tersebut. Perbedaan antara kendala (21), dan (24) adalah, bahwa kendala (24) mengikut sertakan *depot*, sedangkan kendala (21) hanya untuk konsumen saja.



BAB III

METODE PENELITIAN

Dalam memecahkan masalah dalam tugas akhir ini ditempuh langkah: indentifikasi masalah, perumusan masalah, batasan masalah, tujuan, studi lapangan dan studi pustaka yang di lanjutkan dengan perancangan model.

Algoritma pencarian dengan menggunakan model reproduksi dinamakan *genetic algorithm* (GA). Proses GA dalam mencari suatu nilai optimal adalah:

1. Menentukan model representasi genetika dari masalah yang dihadapi yang berupa model dari sistem buatan, representasi individu, pembatasan, fungsi evaluasi.
2. Pembangkitan generasi awal.
3. Menentukan nilai *fitness* (nilai tujuan) dari tiap individu berdasarkan struktur gennya. Nilai *fitness* ini dijadikan sebagai ukuran apakah individu tersebut sudah optimal atau belum.
4. Pemilihan individu dengan nilai *fitness* terbaik untuk dijadikan induk dalam menghasilkan individu baru.
5. Proses reproduksi yang terdiri dari *crossover* dan mutasi, dari proses reproduksi ini dihasilkan individu baru.

Dengan melakukan proses di atas secara berulang-ulang, diharapkan induk yang baik akan menghasilkan suatu generasi dengan individu yang lebih baik.

3.1 Identifikasi Masalah

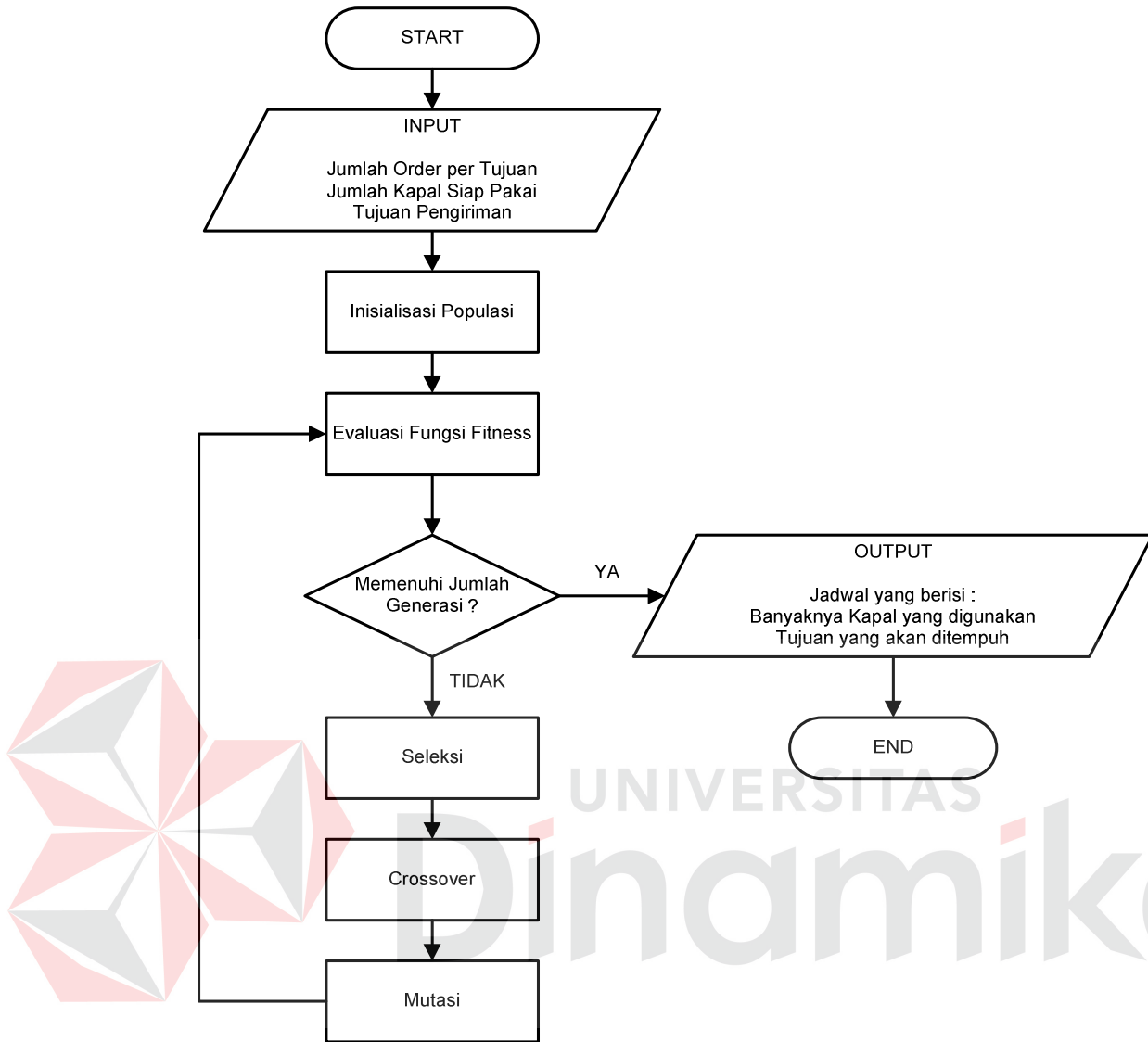
PT. Surya sendiri merupakan pelayaran yang dipercaya oleh beberapa perusahaan sebagai pelayaran yang mengirimkan barang produksi perusahaan tersebut dari Surabaya menuju tempat-tempat tujuan yang telah ditentukan sebelumnya. Setiap kapal pengiriman barang memiliki lebih dari satu rute tujuan pengiriman serta rute tujuan yang berbeda-beda pula. Dengan banyaknya kapal pengiriman yang memiliki rute tujuan yang berbeda ini maka pelayaran PT Surya memiliki tanggung jawab untuk membuat sebuah jadwal. Jadwal yang dimaksudkan adalah jadwal pengiriman barang, dimana pada jadwal tersebut berisi kapal yang tersedia yang akan digunakan beserta rute-rute yang akan dilewati oleh masing-masing kapal.

Permasalahan yang timbul adalah dalam pembuatan penjadwalan dengan menentukan rute pengiriman barang dari Surabaya (PT. Surya) ke tiap tujuan pengiriman dapat dikatakan masih manual, sebab meskipun menggunakan komputer tetapi program yang digunakan bukanlah program yang ditujukan untuk menunjang kegiatan penjadwalan pengiriman barang. Penggunaan komputer hanya sebatas untuk menginputkan data perusahaan serta dokumentasi perusahaan. Hal ini menyebabkan terjadinya penjadwalan rute pengiriman barang yang kurang optimal.

Dari analisis diatas, untuk dapat menyelesaikan permasalahan penjadwalan rute kapal, akan diterapkan konsep penjadwalan dengan pendekatan metode algoritma genetika dengan mengabaikan sistem sebelumnya (manual) yang digunakan oleh PT.Surya.

Dalam suatu pengiriman, perusahaan (costumer) yang akan mengirimkan barangnya ke beberapa tujuan pengiriman akan menghubungi PT.Surya. Dari pesanan pengiriman tersebut PT.Surya akan menentukan kapal serta jumlah kapal yang akan digunakan. Penentuan kapal-kapal yang akan digunakan berdasarkan dari kapasitas pengiriman yang disesuaikan dengan kapasitas masing-masing kapal yang dimiliki PT.Surya. Setelah PT.Surya menentukan kapal pengiriman maka akan diteruskan dengan penentuan rute tujuan yang akan dilalui tiap kapal. Penentuan rute tujuan pengiriman tiap kapal didasarkan dari waktu tempuh kapal yang terkecil (minimum), sehingga dari tiap pemilihan kapal serta rute tujuan akan dihasilkan suatu jadwal pengiriman barang yang optimal.

Data yang dipakai adalah data kapal yang meliputi nama kapal, kecepatan kapal, kapasitas kapal. Data tujuan pengiriman yang meliputi jarak dari tujuan pengiriman. Dari data tersebut dapat diselesaikan masalah yaitu masalah pemilihan kapal (dengan berdasar pada data kapal) sampai dengan masalah pemilihan rute yang akan dilewati kapal (berdasar pada data tujuan).



Gambar 3.1. Diagram Alir Penyelesaian Penjadwalan

Beberapa batasan yang digunakan dalam masalah penjadwalan pengiriman barang :

1. Dalam satu kali pengiriman kapal, hanya berjalan sekali dalam melakukan tugas pengirimannya.
2. Jumlah order tiap tujuan pengiriman dalam rute pengiriman tidak boleh melebihi kapasitas kapal.
3. Jumlah waktu total pengiriman ke seluruh tujuan dalam satu rute digunakan sebagai nilai fitness.

3.2 Perancangan Algoritma Genetika

Pada tahap perancangan ini akan dilakukan penerapan proses algoritma genetika dari permasalahan jadwal pengiriman barang dengan mengimplementasikan data yang ada dalam masing–masing proses yang ada pada algoritma genetika.

3.2.1 Pengkodean (Representasi Kromosom)

Pengkodean adalah proses menterjemahkan suatu obyek permasalahan ke dalam metode algoritma genetika. Dalam permasalahan penjadwalan dan rute ini kromosom dipresentasikan dalam bentuk list, yang mempresentasikan tujuan–tujuan yang akan dikunjungi. Setiap kromosom harus memenuhi batasan–batasan yang sudah dijelaskan diatas. Representasi kromosom dapat digambarkan seperti gambar dibawah ini.

Dengan contoh terdapat 8 tujuan dengan 3 kapal yang siap pakai, maka dapat dipresentasikan sebagai berikut :

Kapal A			Kapal B		Kapal C		
3	7	8	1	5	4	2	6

Gambar 3.2 Kromosom Penjadwalan

Pada gambar 3.2 menunjukkan bahwa kapal A akan mengirimkan barang pada tujuan 3 yang dilanjutkan ke tujuan 7 dilanjutkan ke tujuan 8 dan berakhir pada tujuan 1. Sedangkan pada kapal B akan mengirimkan barang pada tujuan ke 5 dan berakhir pada tujuan 4. Demikian pula dengan kapal C yang mengirimkan barang pada tujuan 2 dan tujuan 6.

3.2.2 Inisialisasi Populasi

Proses inisialisasi adalah pembentukan populasi awal yang diperoleh dengan cara rekombinasi kromosom sebanyak ukuran populasi yang ditentukan. Dalam permasalahan ini proses rekombinasi kromosom ini harus tetap memperhatikan batasan kapasitas yang telah ditetapkan, yaitu tiap tujuan hanya dilewati atau dilayani oleh satu kapal saja dan setiap order tujuan tidak boleh melebihi kapasitas kapal.

Dimana populasi menyatakan banyaknya kromosom yang terbentuk dalam populasi dalam satu generasi. Apabila ukuran populasinya kecil maka algoritma genetika memiliki kemungkinan yang sedikit untuk melakukan proses kromosom dan hanya sebagian kecil dari ruang pencarian yang dapat digunakan.

Sedangkan jika terlalu banyak kromosom maka proses dalam algoritma genetika menjadi lambat.

Contoh proses inisialisasi populasi berdasar dari batasan kapasitas kapal adalah sebagai berikut :

Dimisalkan diinputkan jumlah (ukuran) sebagai berikut :

Tujuan ke	1	2	3	4	5	6	7	8	
Order	40	70	30	20	60	30	20	30	(ton)

Kapal ke	A	B	C	
Kapasitas Kapal	120	100	140	(ton)
Rata-rata Kecepatan Kapal	50	55	65	(mill/jam)

Waktu melayani tiap tujuan = 2 hari = 48 jam.

Setiap tujuan akan dikelompok-kelompokkan dalam setiap kapal. Proses pengelompokkan tujuan ke kapal dimulai dengan mengambil tujuan secara acak.

- Tujuan 4

Cek kapasitas A :

$$\text{TotalOrder} = \text{TotalOrder} + \text{order}(4)$$

$$20 = 0 + 20 \leq 120 \text{ (Memenuhi)}$$

Karena memenuhi batasan yang ada, maka tujuan 4 masuk dalam kapal A.

Cek Waktu :

$$t(i)(j) = d(i)(j) / v$$

$$t(0)(4) = d(0)(4) / 50$$

$$= 1193 / 50 = 23,86 \text{ jam}$$

$$St(j) = 48 \text{ jam}$$

$$T_{tot} = t(i)(j) + St(j)$$

$$T_{tot} = t(0)(4) + St(4)$$

$$= 23,86 + 48 = 71,86 \text{ jam}$$

- Tujuan 6

Cek Kapasitas kapal A :

$$\text{TotalOrder} = \text{TotalOrder} + \text{order}(6)$$

$$50 = 20 + 30 \leq 120 \text{ (Memenuhi)}$$

Karena memenuhi batasan yang ada, maka tujuan 6 masuk dalam kapal A.

Cek Waktu :

$$t(i)(j) = d(i)(j) / v$$

$$t(4)(6) = d(4)(6) / 50$$

$$= 701 / 50 = 14,02 \text{ jam}$$

$$St(j) = 48 \text{ jam}$$

$$T_{tot} = t(i)(j) + St(j)$$

$$T_{tot} = t(4)(6) + St(6)$$

$$= 14,02 + 48 = 62,02 \text{ jam}$$

- Tujuan 8

Cek Kapasitas kapal A :

$$\text{TotalOrder} = \text{TotalOrder} + \text{order}()$$

$$80 = 50 + 30 \leq 120 \text{ (Memenuhi)}$$

Karena memenuhi batasan yang ada, maka tujuan 8 masuk dalam kapal A.

Cek Waktu :

$$t(i)(j) = d(i)(j) / v$$

$$t(6)(8) = d(6)(8) / 50$$

$$= 1482 / 50 = 29,64 \text{ jam}$$

$$St(j) = 48 \text{ jam}$$

$$T_{tot} = t(i)(j) + St(j)$$

$$T_{tot} = t(6)(8) + St(8)$$

$$= 29,64 + 48 = 33,64 \text{ jam}$$

- Tujuan 5

Cek Kapasitas kapal A :

$$\text{TotalOrder} = \text{TotalOrder} + \text{order}(5)$$

$$140 = 80 + 60 \leq 120 \text{ (Tidak memenuhi)}$$

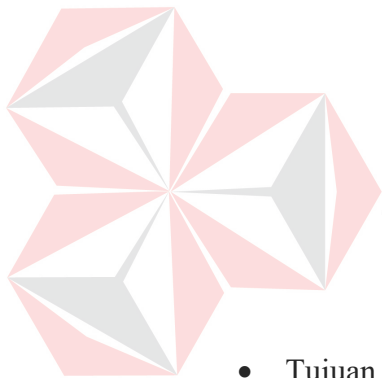
Cek Kapasitas kapal B :

$$\text{TotalOrder} = \text{TotalOrder} + \text{order}(5)$$

$$60 = 0 + 60 \leq 100 \text{ (Memenuhi)}$$

Karena tidak memenuhi batasan maka tujuan 5 masuk dalam kapal B.

Cek Waktu :



$$t(i)(j) = d(i)(j) / v$$

$$t(0)(5) = d(0)(5) / 55$$

$$= 1047 / 55 = 19,03 \text{ jam}$$

$$St(j) = 48 \text{ jam}$$

$$T_{tot} = t(i)(j) + St(j)$$

$$T_{tot} = t(0)(5) + St(5)$$

$$= 19,03 + 48 = 67,03 \text{ jam}$$

- Tujuan 2

Cek Kapasitas kapal A :

$$\text{TotalOrder} = \text{TotalOrder} + \text{order} (2)$$

$$150 = 80 + 70 \leq 120 \text{ (Tidak memenuhi)}$$

Cek Kapasitas kapal B :

$$\text{TotalOrder} = \text{TotalOrder} + \text{order} (2)$$

$$130 = 60 + 70 \leq 100 \text{ (Tidak memenuhi)}$$

Cek Kapasitas kapal C :

$$\text{TotalOrder} = \text{TotalOrder} + \text{order} (2)$$

$$70 = 0 + 70 \leq 140 \text{ (Memenuhi)}$$

Karena tidak memenuhi batasan baik dalam kapal A maupun kapal B maka tujuan 2 masuk dalam kapal C.

Cek Waktu :

$$t(i)(j) = d(i)(j) / v$$

$$t(0)(2) = d(0)(2) / 50$$

$$= 331 / 50 = 7,35 \text{ jam}$$

$$St(j) = 48 \text{ jam}$$

$$T_{tot} = t(i)(j) + St(j)$$

$$\begin{aligned} T_{tot} &= t(0)(2) + St(2) \\ &= 5,09 + 48 = 55,35 \text{ jam} \end{aligned}$$

Proses ini berlangsung terus sampai seluruh tujuan masuk dalam kapal. Dari contoh diatas, akan didapatkan sebuah kromosom sebagai berikut :

Kapal A			Kapal B		Kapal C		
4	6	8	7	5	3	2	1

Gambar 3.3 Kromosom Setelah Disesuaikan Dengan Batasan

Dalam proses ini, akan dilakukan rekombinasi kromosom sampai diperoleh satu populasi sesuai jumlah inputan.

3.2.3 Fungsi Fitness

Setelah mendapatkan satu populasi awal dari algoritma inisialisasi populasi di atas, maka selanjutnya tiap-tiap kromosom akan di evaluasi. Populasi baru dari perhitungan batasan – batasan fungsi fitness akan dievaluasi yang gunanya adalah menghitung total waktu yang ditempuh oleh setiap kapal, dan pendefinisianya adalah sebagai berikut :

$$Eval(K_i) = \sum_{r=1}^n T_{tot}(r) \dots\dots\dots (4)$$

dimana, $r = 1,2,3,\dots,n$

$i = 1,2,3,\dots,\text{max pop_size}$

Untuk rumusan fungsi fitness (4) menunjukkan jumlah total waktu tempuh dari suatu kapal. Parameter yang digunakan adalah : r sebagai kapal, K_i sebagai kromosom ke $-i$.

Dari contoh diatas maka dari hasil random sesuai dengan jumlah populasi maka didapat 6 kromosom, dari tiap kromosom tersebut maka dihitung jarak total dari tiap kapal.

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

Eval (K_1) = 466,40 jam

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

Eval (K_2) = 525,78 jam

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

Eval (K_3) = 526,59 jam

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

Eval (K_4) = 504,91 jam

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

Eval (K_5) = 509,22 jam

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

Eval (K_6) = 510,69 jam

3.2.4 Seleksi

Proses seleksi bertanggung jawab untuk melakukan pemilihan terhadap individu (parent) yang hendak diikuti dalam proses reproduksi. Pada permasalahan ini akan digunakan metode seleksi *roulette whell*. Dalam penerapan *roulette whell*, terdapat dua bagian user fungsi yaitu fungsi inisialisasi *roulette*

whell dan fungsi penentuan roulette whell. Penentuan inisialisasi roulette whell bertujuan untuk mendapatkan probabilitas seleksi, dan probabilitas ini didapat berdasarkan nilai fitness tiap individu.

Adapun langkah-langkah yang diperlukan dalam inisialisasi roulette whell adalah sebagai berikut :

1. Menjumlahkan nilai fitness dari semua individu yang terdapat dalam populasi.

$$TotalFitness = \sum_{i=1}^{popsize} Eval(K_i)$$

2. Menghitung probabilitas terpilihnya setiap individu ($P[i]$) dengan membagi nilai fitness masing-masing individu dengan total nilai fitness populasi.

$$P[i] = Eval(K_i) / TotalFitness$$

3. Menghitung probabilitas kumulatif untuk setiap individu dalam populasi ($Q[i]$).

$$Q[i] = \sum_{j=1}^i P[j]$$

Tidak semua individu yang terdapat pada populasi melakukan operasi genetik. Hal itu disebabkan tidak semua individu lulus dalam seleksi. Untuk masalah seleksi, Roulette Whell dipergunakan dalam program untuk menentukan individu-individu mana yang akan dipakai dalam operasi genetik. Roulette whell dilakukan setelah didapatkan probabilitas seleksi.

Langkah-langkah dari Roulette whell adalah :

1. Membangkitkan bilangan acak (R) bertipe real dengan range $[0..1]$ sejumlah individu dalam populasi.
2. Apabila $Q[i] \notin R$, dan $Q[i+1] > R$ maka pilih kromosom ke $(i+1)$ sebagai kandidat induk.

Dari masalah di atas maka dalam langkah inisialisasi roulette wheel didapatkan hasil seperti dibawah ini :

1. Menjumlahkan nilai fitness

$$\begin{aligned} & \text{Eval}(K_1) + \text{Eval}(K_2) + \text{Eval}(K_3) + \text{Eval}(K_4) + \text{Eval}(K_5) + \text{Eval}(K_6) = \\ & 466,40 + 525,78 + 526,59 + 504,91 + 509,22 + 510,69 = 3043,59 \text{ jam} \end{aligned}$$

2. Menghitung probabilitas setiap individu

$$P[1] = 466,40 / 3043,59 = 0,153 \text{ (dibulatkan } 0,15)$$

$$P[2] = 525,78 / 3043,59 = 0,172 \text{ (dibulatkan } 0,17)$$

$$P[3] = 526,59 / 3043,59 = 0,173 \text{ (dibulatkan } 0,17)$$

$$P[4] = 504,91 / 3043,59 = 0,165 \text{ (dibulatkan } 0,17)$$

$$P[5] = 509,22 / 3043,59 = 0,167 \text{ (dibulatkan } 0,17)$$

$$P[6] = 510,69 / 3043,59 = 0,167 \text{ (dibulatkan } 0,17)$$

3. Menghitung probabilitas kumulatif

$$Q[1] = 0,15$$

$$Q[2] = 0,15 + 0,17 = 0,32$$

$$Q[3] = 0,15 + 0,17 + 0,17 = 0,49$$

$$Q[4] = 0,15 + 0,17 + 0,17 + 0,17 = 0,66$$

$$Q[5] = 0,15 + 0,17 + 0,17 + 0,17 + 0,17 = 0,83$$

$$Q[6] = 0,15 + 0,17 + 0,17 + 0,17 + 0,17 + 0,17 = 1,0$$

Dan Langkah dari roulette wheel-nya adalah :

1. Membangkitkan bilangan acak R dengan range [0..1] sebanyak 6 atau sejumlah dengan individu dalam populasi.

Misalnya didapatkan dari hasil random sebagai berikut :

$$\begin{array}{lll} R[1] = 0,28 & R[3] = 0,47 & R[5] = 0,35 \\ R[2] = 0,88 & R[4] = 0,78 & R[6] = 0,06 \end{array}$$

2. Apabila $Q[i] \leq R$, dan $Q[i+1] > R$ maka pilih kromosom ke (i+1) sebagai kandidat induk.

$$\begin{array}{lll} R[1] & \leq & Q[1] \\ 0,28 & \leq & 0,15 \\ & \text{dan} & \\ Q[2] & > & R[1] \\ 0,32 & > & 0,28 \end{array}$$

Jadi Q[2] menjadi kromosom baru yang pertama.

$$\begin{array}{lll} R[2] & \leq & Q[5] \\ 0,88 & \leq & 0,83 \\ & \text{dan} & \\ Q[6] & > & R[2] \\ 1 & > & 0,88 \end{array}$$

Jadi Q[6] menjadi kromosom baru yang kedua.

$$\begin{array}{lll} R[3] & \leq & Q[2] \\ 0,47 & \leq & 0,32 \\ & \text{dan} & \\ Q[3] & > & R[3] \\ 0,49 & > & 0,47 \end{array}$$

Jadi Q[3] menjadi kromosom baru yang ketiga.

$$\begin{array}{lll} R[4] & \leq & Q[4] \\ 0,78 & \leq & 0,66 \\ & \text{dan} & \\ Q[5] & > & R[4] \\ 0,83 & > & 0,78 \end{array}$$

Jadi Q[5] menjadi kromosom baru yang keempat.

$$\begin{array}{lll} R[5] & \leq & Q[2] \\ 0,36 & \leq & 0,32 \\ & \text{dan} & \\ Q[3] & > & R[5] \\ 0,49 & > & 0,35 \end{array}$$

Jadi Q[3] menjadi kromosom baru yang kelima.

$$R[6] \leq Q[1]$$

0,06 £ 0,15
 dan
 Q[1] > R[6]
 0,15 > 0,06

Jadi Q[1] menjadi kromosom baru yang keenam.

Kromosom yang telah mengalami seleksi :

K[1] =	K[2] =	6	3	7	1	2	4	8	5
K[2] =	K[6] =	4	2	6	8	1	7	3	5
K[3] =	K[3] =	4	2	3	7	8	1	6	5
K[4] =	K[5] =	1	6	8	4	2	3	5	7
K[5] =	K[3] =	4	2	3	7	8	1	6	5
K[6] =	K[1] =	4	6	8	7	5	3	2	1

Pseudocode Seleksi Roulette Wheel :

Procedure SelectionRouletteWheel(indexGenerasi as integer)

dim EvalKromosom(max_pop_size) { Nilai fitness tiap kromosom }

dim P(max_pop_size) { Nilai relatif tiap kromosom }

dim Q(max_pop_size) { Nilai kumulatif tiap kromosom }

dim R(max_pop_size) { Nilai random tiap kromosom }

dim K(max_pop_size) { Kromosom dalam populasi }

{ hitung total nilai fitness populasi }

for i=1 to pop_size { Ukuran populasi }

 TotalFitness = TotalFitness + EvalKromosom(i)

next i

{ mencari nilai relatif tiap kromosom }

for i=1 to pop_size

 P(i) = EvalKromosom(i) / TotalFitness

next i

{ mencari nilai kumulatif tiap kromosom }

for i=1 to pop_size

 for j=1 to i

 Q(i) = Q(i) + P(i)

 next j

next i

```

{ mencari nilai acak tiap kromosom }
for i=1 to pop_size
    R(i) = Random(1) { Random nilai antara 0 .. 1 }
next i

{ jika nilai acak lebih kecil dari nilai kumulatif dan terdapat nilai
  kumulatif maka tukar posisi kromosom dalam populasi }
for i=1 to pop_size
    if Q(i) < R(i) and Q(i+1) > R(i) then
        tmp = K(i)
        K(i+1) = K(i)
        K(i) = tmp
    end if
end if
next i
end procedure

```

3.2.5 Operator Genetika

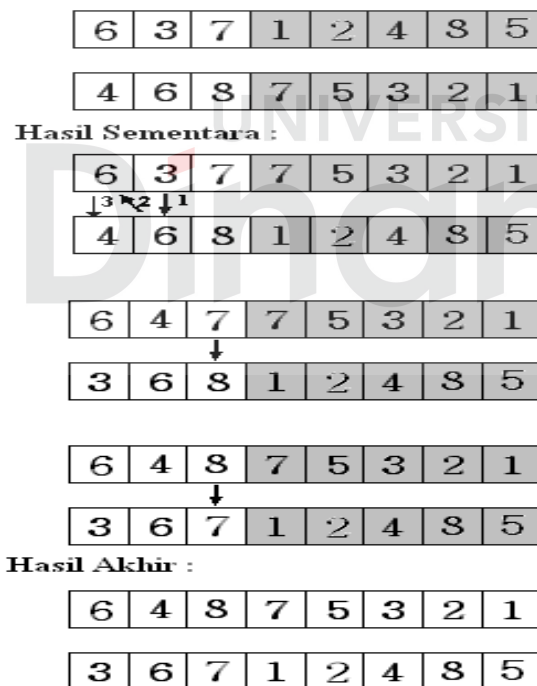
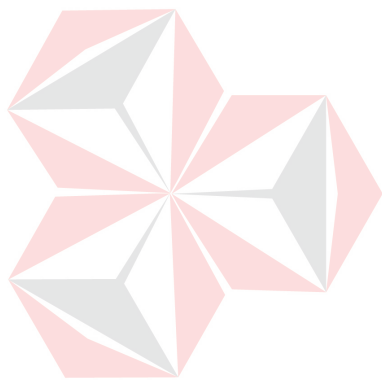
Setelah dilakukannya seleksi maka dari parent – parent yang terpilih akan dilakukan proses kerja yang ada pada algoritma genetika, yaitu diperlukan operator–operator genetika. Pada permasalahan optimasi jadwal pengiriman ini memerlukan dua jenis operator genetika yaitu crossover dan mutasi.

A. Crossover

Pada permasalahan optimasi jadwal pengiriman akan menggunakan jenis *one point* crossover sebab menggunakan permutation encoding yang dalam pembentukan kromosom dilakukan dengan mengambil kode nama tujuan yang sesuai dengan permasalahan penjadwalan. One point crossover dilakukan dengan cara memilih secara acak posisi dalam kromosom induk (parent) pertama kemudian saling menukar gen dengan induk kedua dengan posisi yang sama,

sebab dalam pembentukan kromosom terbentuk dari permutation encoding dimana permutation encoding digunakan untuk permasalahan proses pengurutan. Dan jumlah kromosom yang mengalami crossover dipengaruhi oleh parameter crossover (probabilitas crossover) yang telah ditentukan yaitu dengan cara menghitung $P_c * uk_pop$.

Berdasar dari contoh di atas maka diinputkan probabilitas crossover (P_c) adalah 50%(0,5). Maka diharapkan jumlah kromosom yang mengalami crossover adalah $0,5 * 6 = 3$ kromosom. Dalam permasalahan ini dipakai one point crossover, dan setelah didapat kromosom induk (parent) secara acak serta posisi pemotongan didapat secara acak pula maka proses crossoversnya adalah sebagai berikut :



Gambar 3.4 Proses Crossover

Setelah kedua individu selesai dicrossover, dimungkinkan terjadi kekacauan dalam sebuah kromosom, yaitu terjadi perulangan dari tujuan pengiriman. Karena itu setelah terjadi crossover, maka dilakukan perbaikan pada

masing-masing kromosom. Setiap tujuan yang kembar pada kromosom 1 (dari bagian yang tidak di tukar) dibandingkan dengan tujuan pada posisi yang sama pada kromosom 2, bila tujuan pada kromosom 2 ini juga kembar, maka kedua tujuan akan ditukar, bila tidak maka dicari tujuan pada kromosom 2 yang posisinya sama dengan posisi tujuan pada kromosom 2 tadi, tetapi pada kromosom 1.

Berikut adalah kromosom-kromosom setelah di crossover :

K[1]" =	6	4	8	7	5	3	2	1
K[2]' =	4	2	6	8	1	7	3	5
K[3]' =	4	2	3	7	8	1	6	5
K[4]' =	1	6	8	4	2	3	5	7
K[5]' =	4	2	3	7	8	1	6	5
K[6]" =	3	6	7	1	2	4	8	5

Pseudocode Proses Crossover :

```

Procedure CrossOver(indexGenerasi as integer)
dim R(max_pop_size) { Nilai random tiap kromosom }
dim KTerpilih(max_pop_size) { Kromosom yang terpilih untuk di cross over }
  { mencari nilai acak tiap kromosom }
  for i=1 to pop_size
    R(i) = Random(1) { Random nilai antara 0 .. 1 }
  next i

  { jika nilai acak lebih kecil dari nilai probabilitas crossover
  maka terpilih kromosom yang di cross over }
  j = 1
  for i=1 to pop_size
    if R(i) < Pc then
      { pemilihan kromosom yang akan di crossover }
      KTerpilih(j) = K(i)
      j = j + 1
    end if
  next i

  { kromosom yang terpilih untuk cross over harus berjumlah genap }
  if ubound(KTerpilih) mod 2 = 0 then
    { pemilihan posisi awal pada kromosom yang akan di cross over }
    posisiAwal = Random(panjang_kromosom)
    for i=1 to pop_size step 2
      j = i
    
```

```

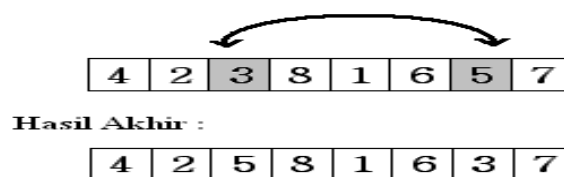
{ pemilihan induk kromosom harus 2 kromosom }
IndukPertama = KTerpilih(i)
IndukKedua = KTerpilih(j+1)
  { proses cross over kromosom }
  for pos=posisiAwal to panjang_kromosom
    tukar gen kromosom IndukPertama dengan
    gen kromosom IndukKedua sesuai
    dengan posisi yang telah diacak
    {posisiAwal};
    perbaikan gen kromosom IndukPertama
    dengan gen kromosom IndukKedua
    sehingga tidak terdapat duplikasi gen pada
    masing-masing kromosom;
  next pos
next i
end if
end procedure

```

B. Mutasi

Dalam permasalahan ini digunakan mutasi inversi, sebab menggunakan permutation encoding dimana mutasi yang sesuai dengan penjadwalan. Proses mutasi dilakukan dengan cara mengganti satu sub kromosom (gen) yang dipilih secara acak dengan suatu nilai baru. Untuk memilih posisi gen yang mengalami mutasi prosesnya adalah kita hitung dengan $P_m * uk_pop * N$.

Dalam masalah penjadwalan misalnya ditentukan probabilitas mutasinya (P_m) adalah 5% (0,05) , Maka diharapkan bahwa jumlah gen yang akan dimutasi adalah $0,05 * 6 * 8 = 2,4$ (dibulatkan menjadi 2) gen. Kromosom yang mengalami mutasi dipilih secara acak, serta pemilihan gen yang dimutasi juga ditentukan secara acak. Dan proses mutasi pada penjadwalan ini adalah sebagai berikut :



Gambar 3.5 Proses Mutasi

Setelah proses mutasi selesai, maka dalam permasalahan penjadwalan ini telah dilakukan satu iterasi dalam algoritma genetika atau disebut dengan satu generasi.

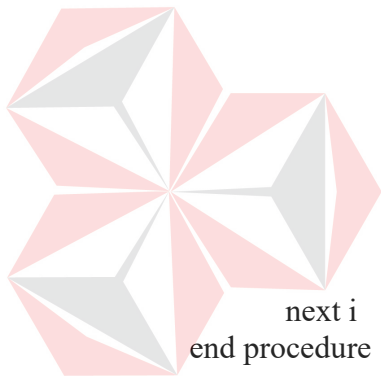
Pseudocode proses mutasi :

```

Procedure Mutation(indexGenerasi as integer)
dim R(max_pop_size) { Nilai random tiap kromosom }
  { mencari nilai acak tiap kromosom }
  for i=1 to pop_size
    R(i) = Random { Random nilai }
  next i

  { jika nilai acak lebih kecil dari nilai probabilitas mutasi
    maka terpilih kromosom yang dimutasi }
  for i=1 to pop_size
    if R(i) < Pm then
      { pemilihan posisi gen dalam kromosom yang akan
        dimutasi }
      posisiAwal = Random( pnj_kromosom )
      posisiAkhir = Random(pnj_kromosom)
      if posisiAwal <> posisiAkhir then
        tukar kedua gen dalam kromosom sesuai dengan
        posisi yang telah diacak;
      end if
    end if
  next i
end procedure

```

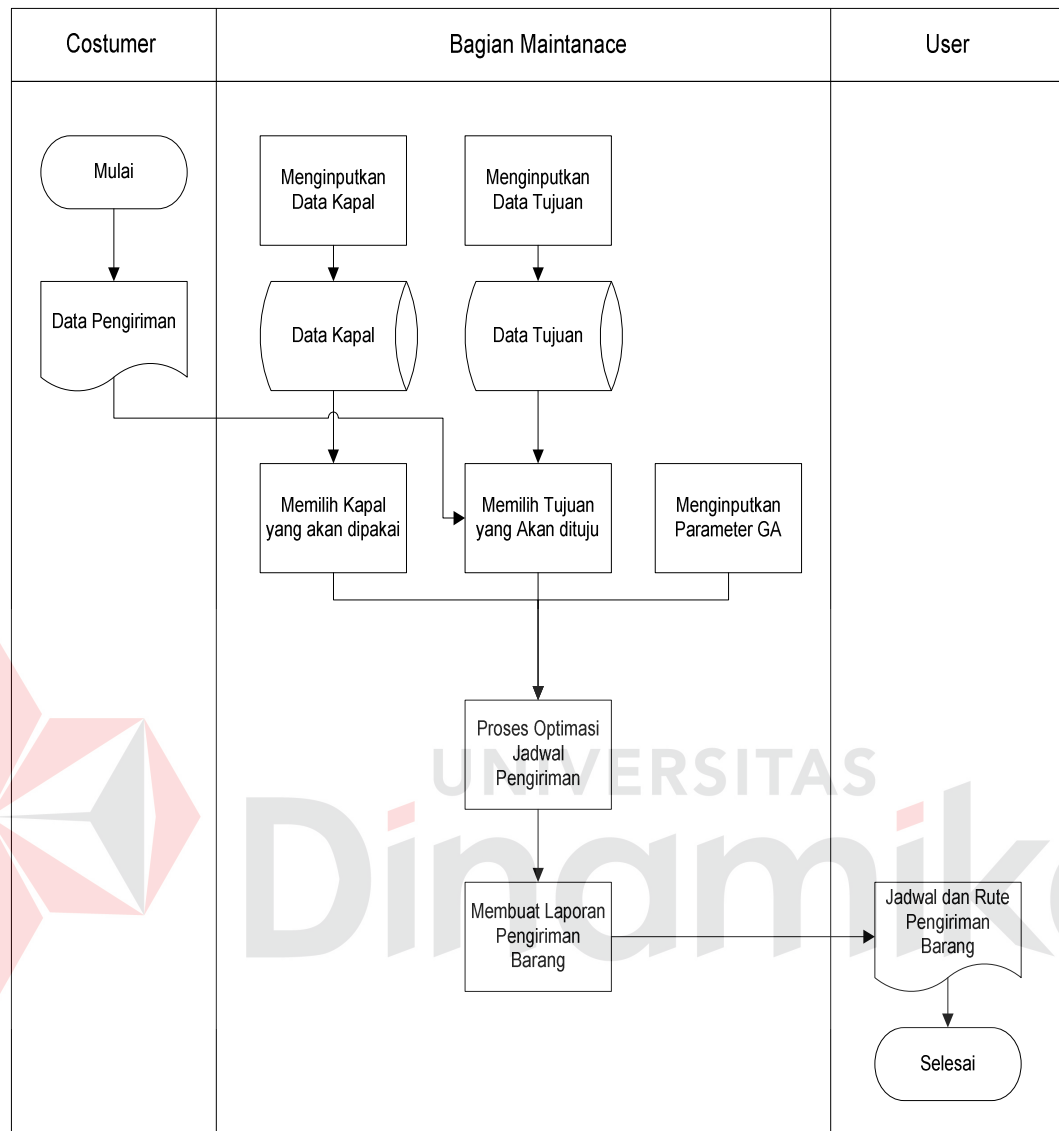


Dinamika

3.2.6 Kriteria Berhenti

Setelah melakukan perancangan sistem maka hasil rancangan tersebut akan diuji untuk mengetahui apakah rancangan yang telah dibuat sudah memenuhi kebutuhan sistem. Pada penjadwalan kapal, proses akan berhenti setelah mencapai generasi ke – N (sesuai dengan inputan pada generasi).

3.3 Desain Sistem

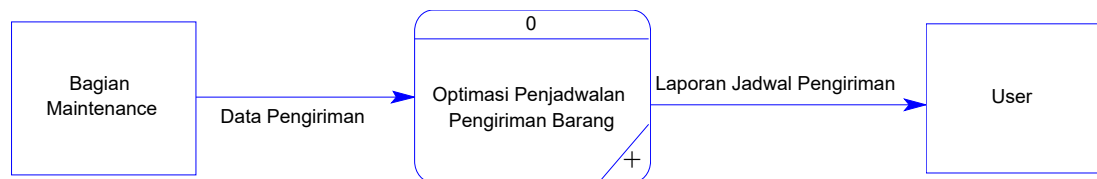


Gambar 3.6 Proses Optimasi Penjadwalan Pengiriman Barang

Berdasarkan diagram alur diatas, proses algoritma genetika dimulai dengan mengambil data kapal dan data tujuan, selanjutnya dilakukan proses optimasi seperti pada gambar 3.6. Output dari proses ini akan menghasilkan suatu jadwal pengiriman barang yang optimal.

3.3.1 Context Diagram

Context diagram menggambarkan sistem pertama kali secara garis besar dari semua hubungan antara sistem pengendalian bahan baku ini dengan lingkungan sekitarnya. Pembentukan Context diagram dapat dilihat pada Gambar 3.7.

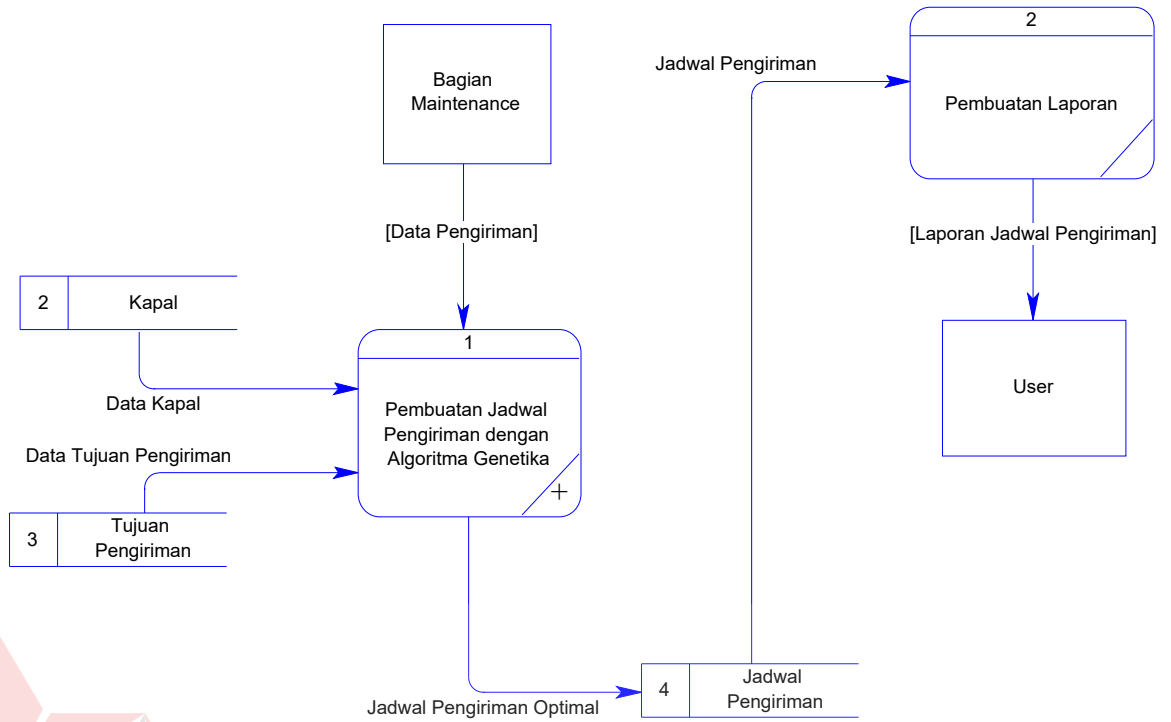


Gambar 3.7 Context Diagram Penjadwalan Pengiriman Barang

3.3.2 Data Flow Diagram (DFD)

Representasi dari sebuah sistem secara grafis yang digambarkan dengan sejumlah simbol tertentu untuk menunjukkan perpindahan data dalam proses suatu sistem.

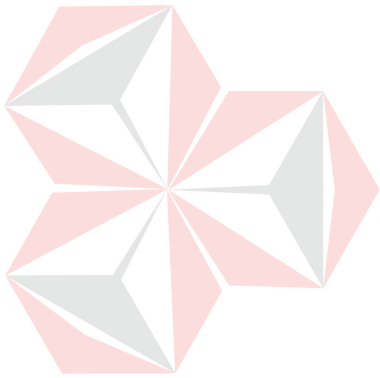
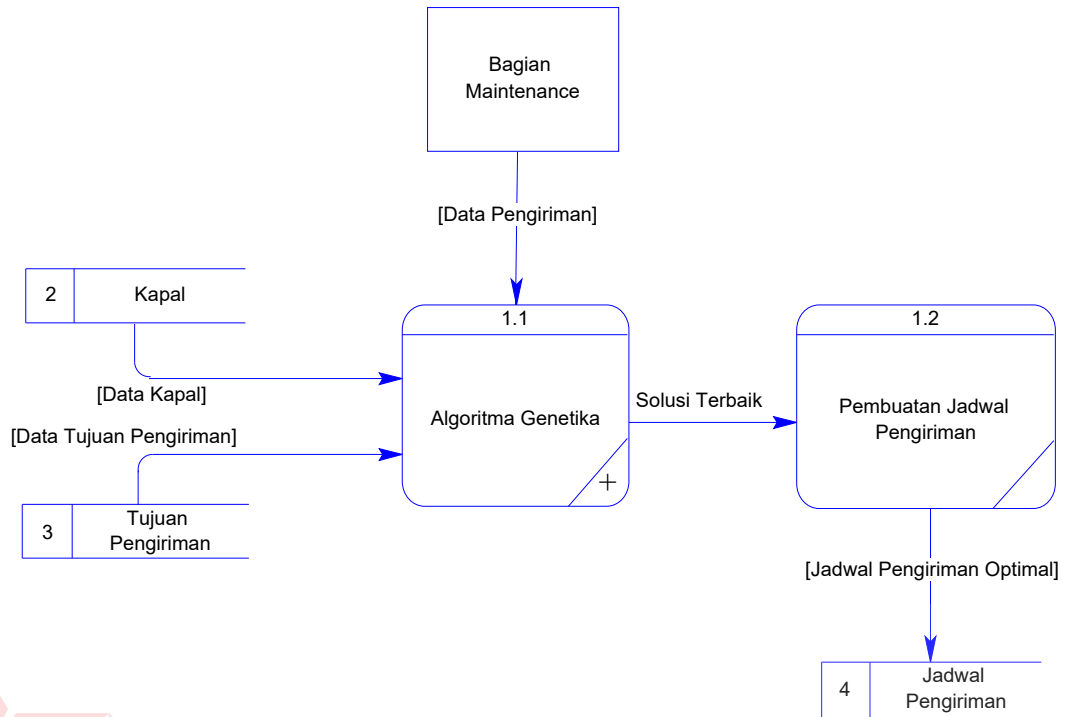
A. DFD Level 0



Gambar 3.8 DFD Level 0

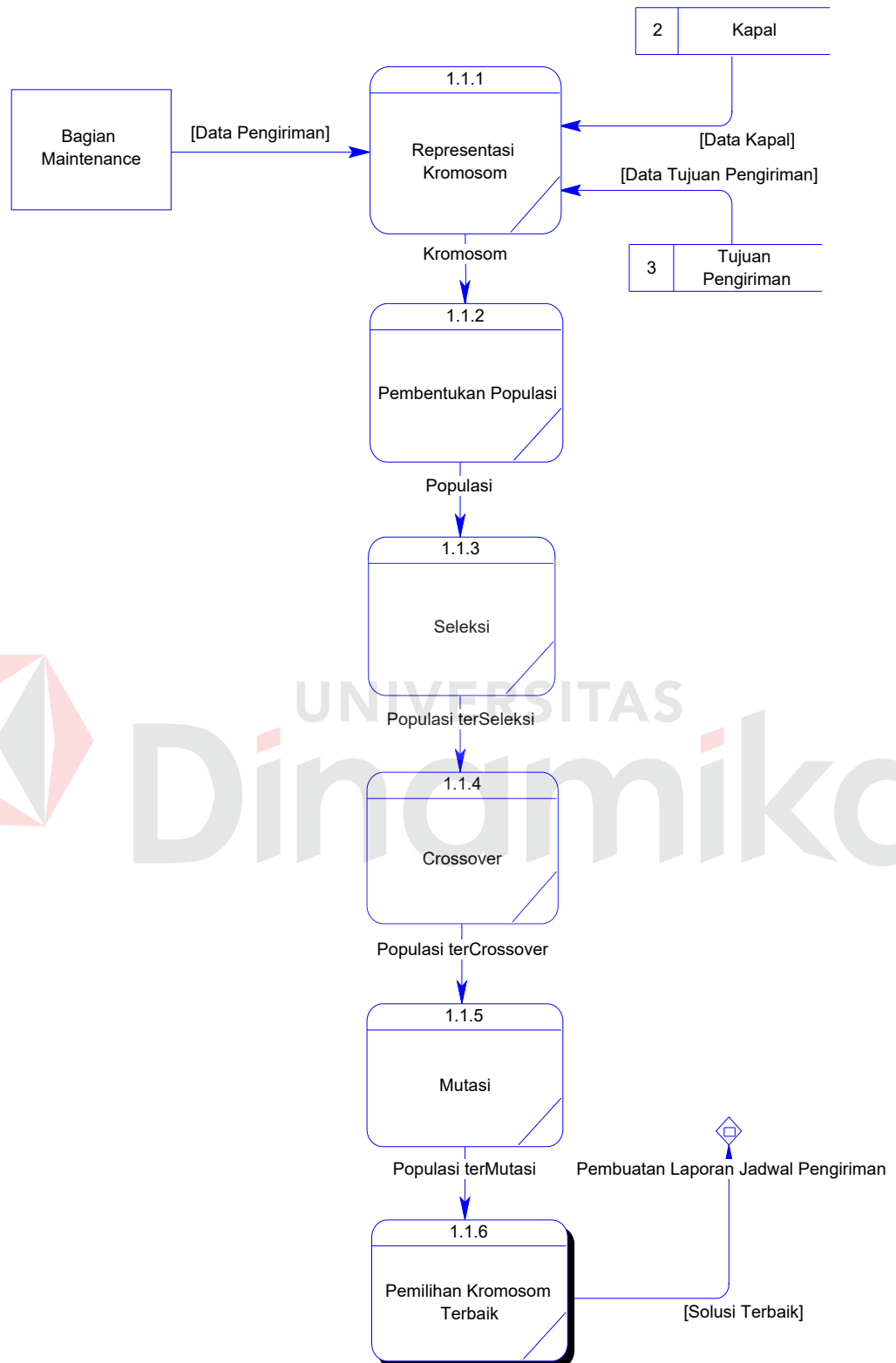
Sedangkan DFD pada level dibawahnya dari masing-masing proses di

DFD level 0 penjadwalan pengiriman barang dengan menggunakan Algoritma Genetika dapat dilihat pada gambar 3.9 sampai dengan gambar 3.10 berikut ini.

B. DFD Level 1 Proses 1

Gambar 3.9 DFD Level 1 Proses 1

C. DFD Level 2 Proses 1.2

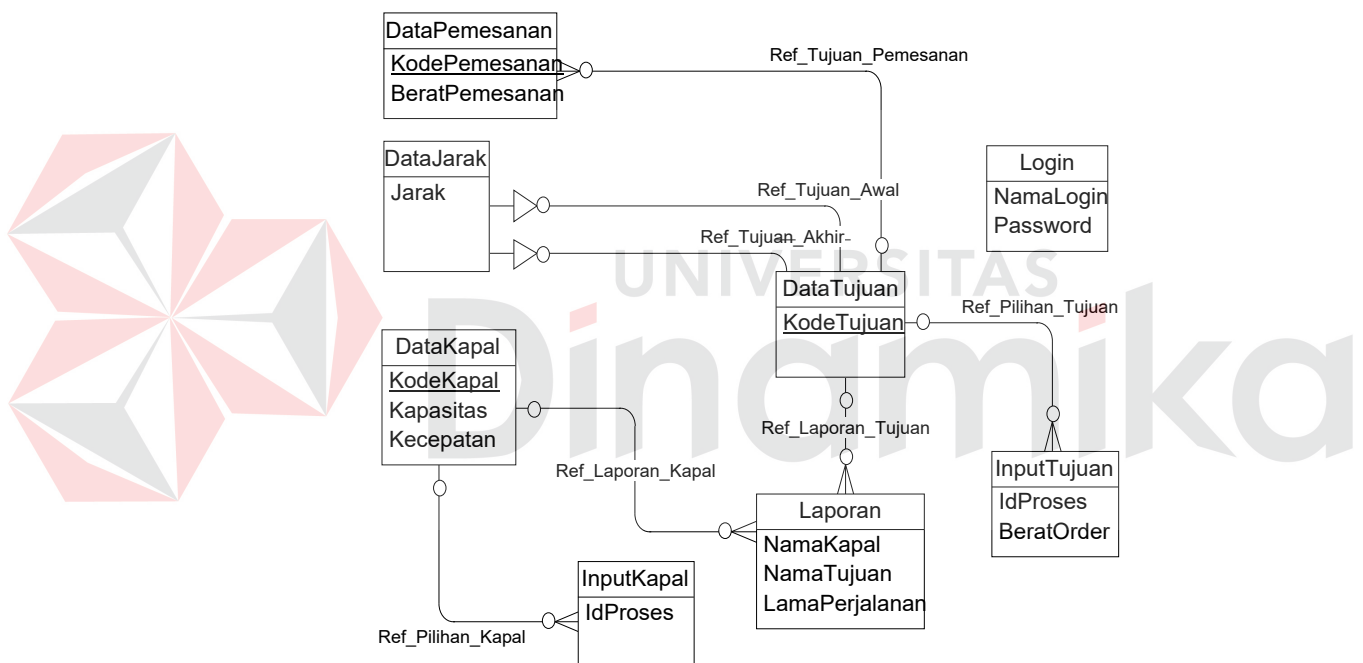


Gambar 3.10 DFD Level 2 Proses 1.2

3.3.3 Entity Relationship Diagram

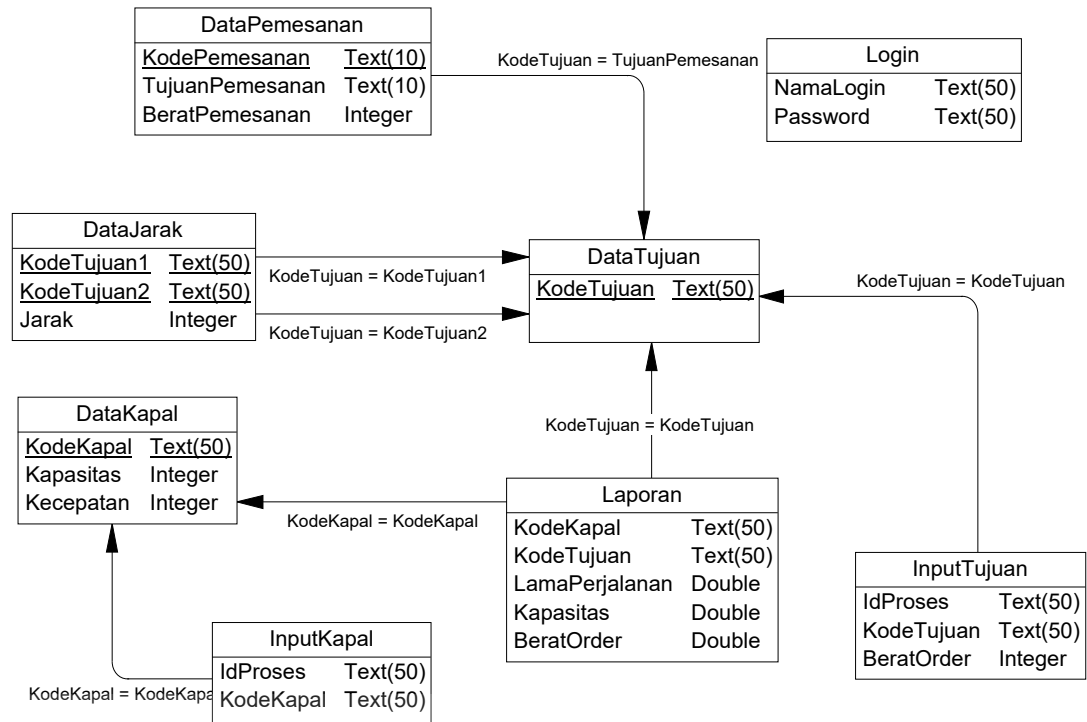
Entity Relationship Diagram (ERD) digunakan untuk menggambarkan hubungan data yang digunakan dalam sistem. ERD juga menunjukkan struktur keseluruhan kebutuhan data yang diperlukan, dalam ERD data tersebut digambarkan dengan menggunakan simbol entity.

Pada ERD *Conceptual Data Model* (CDM) dapat dijelaskan hubungan kardinalisasi yang terjadi antar tabel.



Gambar 3.11 Gambar ERD Conceptual Data Model

Sedang pada ERD *Physical Data Model* (PDM) dapat dijelaskan struktur database secara lengkap beserta nama *field* serta *primary key* dan *foreign key*.



Gambar 3.12 Gambar ERD Physical Data Model

3.3.4 Struktur Database

Adapun struktur database yang digunakan dengan menggunakan sistem database Microsoft® Access® dalam aplikasi ini adalah:

1. Tabel Data Kapal

Fungsi : Menyimpan data kapal.

Tabel 3.1 Tabel Data Kapal

No	Nama Field	Key	Tipe Data	Lebar Field	Keterangan
1	KodeKapal	Pk	Varchar	50	Kode Kapal
2	NamaKapal		Varchar	50	Nama Kapal
3	Kapasitas		Integer	-	Kapasitas Kapal
4	Kecepatan		Integer	-	Kecepatan Kapal

2. Tabel Data Tujuan

Fungsi : Menyimpan data tujuan.

Tabel 3.2 Tabel Data Tujuan

No	Nama Field	Key	Tipe Data	Lebar Field	Keterangan
1	KodeTujuan	Pk	Varchar	50	Kode Tujuan
2	NamaTujuan		Varchar	50	Nama Tujuan

3. Tabel Data Jarak

Fungsi : Menyimpan data jarak.

Tabel 3.3 Tabel Data Jarak

No	Nama Field	Key	Tipe Data	Lebar Field	Keterangan
1	KodeTujuan1	Pk	Varchar	50	Kode Tujuan Awal
2	KodeTujuan2	Pk	Varchar	50	Kode Tujuan Akhir
3	Jarak		Integer	-	Jarak

4. Tabel Data Pemesan

Fungsi : Menyimpan data costumer

Tabel 3.4 Tabel Data Pemesan

No	Nama Field	Key	Tipe Data	Lebar Field	Keterangan
1	KodePemesanan	Pk	Text	10	Kode Kapal
2	NamaPemesanan		Text	100	Nama Kapal
3	TujuanPemesanan		Text	10	Kapasitas Kapal
4	BeratPemesanan		Integer	-	Kecepatan Kapal

5. Tabel Input Kapal

Fungsi : Menyimpan data kapal yang digunakan dalam proses GA.

Tabel 3.5 Tabel Input Kapal

No	Nama Field	Key	Tipe Data	Lebar Field	Keterangan
1	IdProses	Pk	Varchar	50	Id Proses
2	KodeKapal		Varchar	50	Kode Kapal

6. Tabel Input Tujuan

Fungsi : Menyimpan data tujuan, digunakan dalam proses GA.

Tabel 3.6 Tabel Input Tujuan

No	Nama Field	Key	Tipe Data	Lebar Field	Keterangan
1	IdProses	Pk	Varchar	50	Id Proses
2	KodeTujuan		Varchar	50	Kode Tujuan

3	NamaTujuan		Varchar	50	
4	BeratOrder		Integer	-	Berat Order tiap tujuan

7. Tabel Laporan

Fungsi : Menyimpan laporan proses genetika algoritma.

Tabel 3.7 Tabel Laporan

No	Nama Field	Key	Tipe Data	Lebar Field	Keterangan
1	KodeKapal	Pk	Varchar	50	Id Proses
2	NamaKapal		Varchar	50	Generasi
3	KodeTujuan		Varchar	50	Populasi
4	NamaTujuan		Varchar	50	Kromosom Tujuan
5	LamaPerjalanan		Integer	-	Kromosom Kapal
6	Kapasitas		Integer	-	Fitness
7	BeratOrder		Integer	-	Probabilitas Crossover



BAB IV

EVALUASI DAN IMPLEMENTASI

Pembahasan dan pengujian sistem adalah tahap penjelasan secara nyata penggunaan dari rancang bangun program yang telah dibuat, mulai dari evaluasi sampai dengan implementasi dengan memproses database GA

Aplikasi jadwal pengiriman dan rute barang adalah suatu aplikasi untuk menjadwalkan rute pengiriman barang sesuai dengan permintaan yang diterbitkan dari bagian pengiriman dengan menggunakan metode GA. Untuk menjalankan aplikasi ini dibutuhkan perangkat lunak sebagai berikut:

1. Tersedia perangkat lunak Microsoft Access xp.
2. Membuat koneksi ODBC.
3. Tersedia perangkat lunak Microsoft Visual Basic 6.0.
4. Tersedia perangkat lunak Crystal Reports 8
5. Menjalankan setup program.
6. Menjalankan program.

Sedangkan kebutuhan perangkat keras minimum adalah sebagai berikut:

- a. Intel Pentium IV 2,26Ghz
- b. Memori DDR sebesar 224 MB.
- c. Kapasitas harddisk sebesar 20 GB.
- d. Virtual memori sistem operasi initial size 200 MB dan maximum size 400 MB.
- e. Mouse, printer, keyboard.

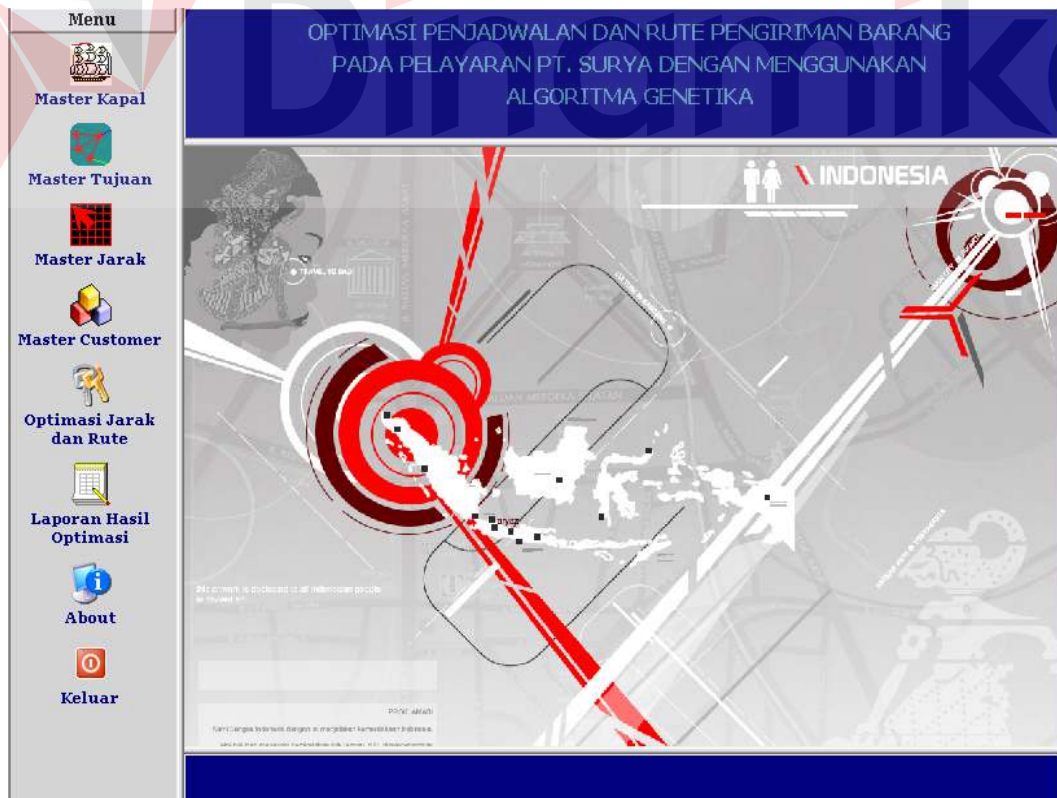
4.1 Implementasi Sistem

Untuk dapat menjalankan aplikasi setiap pengguna harus melakukan login terlebih dahulu. Pengguna memasukan nama *login* dan *password* untuk dapat membuka aplikasi ke menu selanjutnya. Tampilan menu login seperti pada Gambar 4.1.



Gambar 4.1 Menu Login

Aplikasi ini mempunyai lima menu utama yang terletak di bagian kiri dari tampilan aplikasi. Lima menu tersebut yaitu menu Master Kapal, Master Tujuan, Master Jarak, Optimasi Jarak dan Rute dan Laporan Hasil Optimasi.



Gambr 4.2 Menu Utama

Pada tampilan utama aplikasi terdapat tombol master kapal, tujuan serta jarak yang dapat ditambah, diubah, dan dihapus yang kemudian dapat disimpan dalam database yang bersangkutan.


Kode Kapal

Nama Kapal

Kapasitas TON

Kecepatan Mill/Jam

Kode Kapal	Nama Kapal	Kapasitas	Kecepatan
KPL001	M.V. Surya Tulus	6470	10
KPL002	M.V. Surya Kartika	6700	10
KPL003	M.V. Caraka Jaya Niaga III-12	4371	10
KPL004	M.V. Caraka Jaya Niaga III-6	4367	10
KPL005	M.V. Maluku	4600	11
KPL006	M.V. Sungai Bone	5340	9
KPL007	M.V. Kalsel	4860	9
KPL008	M.V. Surya Karya	6455	8
KPL009	M.V. Surya Citra I	5530	8
KPL010	M.V. Surya Citra II	5430	8

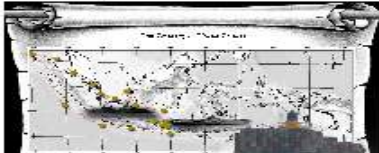


Gambar 4.3 Master Kapal

Kode Tujuan

Nama Tujuan

Kode Tujuan	Nama Tujuan
TJN0001	Surabaya
TJN0002	Banyuwangi
TJN0003	Batulicin
TJN0004	Biringkasi
TJN0005	Bitung
TJN0006	Gorontalo
TJN0007	Jayapura
TJN0008	Labuhan UKI
TJN0009	Makassar
TJN0010	Manukwari
TJN0011	Pare-pare
TJN0012	Tobelo
TJN0013	Toli-toli



Gambar 4.4 Master Tujuan

Dari:

Ke:

Jarak: Mil

	Surabaya	Banyuwangi	Batulicin	Biringkasi	B
Surabaya	0	143	331	433	
Banyuwangi	143	0	373	290	
Batulicin	331	373	0	265	
Biringkasi	433	290	265	0	
Bitung	1193	1050	713	758	
Gorontalo	1047	904	885	627	
Jayapura	1940	1797	1724	1458	
Labuhan UKI	1053	910	580	664	
Makassar	457	314	242	23	
Manukwari	1648	1505	1320	1055	
Pare-pare	514	371	397	151	



Gambar 4.5 Master Jarak

Form customer digunakan sebagai form yang menunjukkan data dari customer atau perusahaan pemesan dimana menggunakan jasa dari PT. Surya sebagai pelayaran yang mengirimkan barang.

Kode Pemesanan:

Nama:

Tujuan:

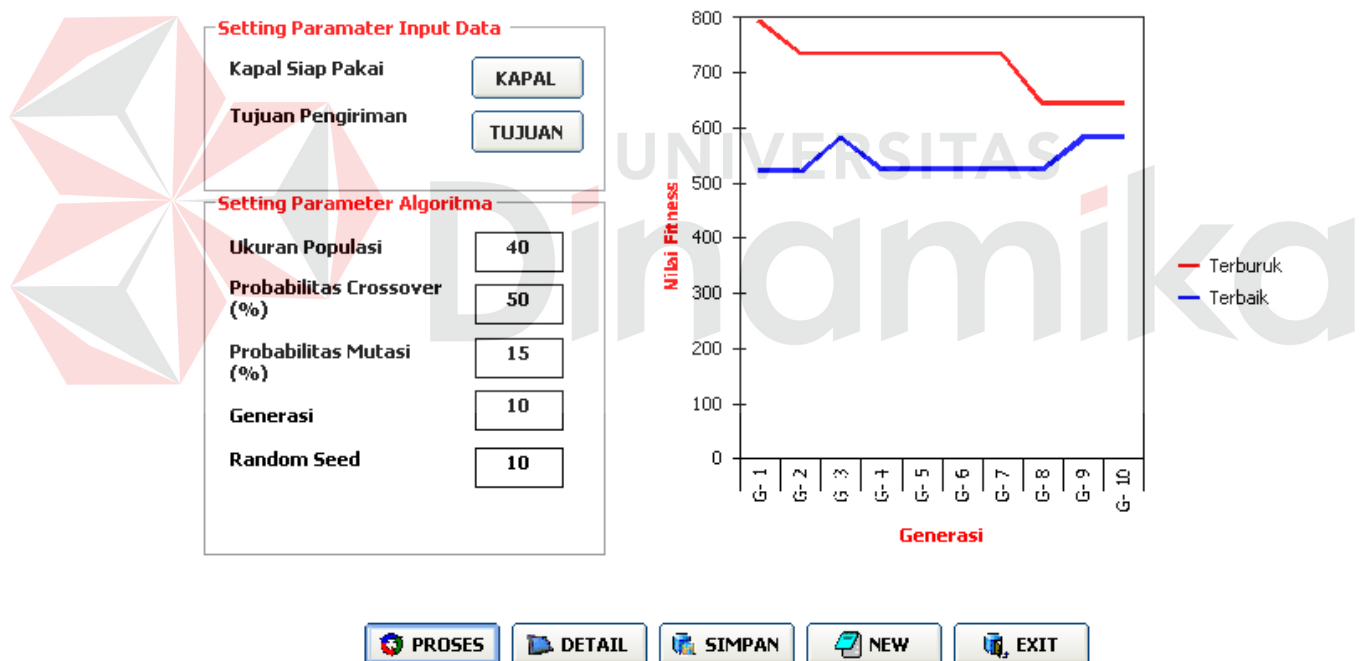
Berat Order:

Kode Pemesa...	Nama	Kode Tujuan	Nama Tujuan	Berat
CST001	PT KARISMA J...	TJN0007	Jayapura	250
CST002	PT DUTA KENC...	TJN0008	Labuhan UKI	2500
CST003	PT GERHANA	TJN0009	Makassar	1500
CST004	CV SANJAYA	TJN0007	Jayapura	900
CST005	PT HANS	TJN0005	Bitung	600
CST006	PT WISATA	TJN0013	Toli-toli	1200
CST007	CV SILOAM	TJN0009	Makassar	500

Gambar 4.6 Master Customer

Pada gambar 4.7 menunjukkan form proses dari optimasi jadwal dan rute. Pada form tersebut terdapat setting parameter input data berupa tombol, yang berisi data kapal dan data tujuan yang dituju yang akan digunakan. Inputan setting

parameter algoritma yaitu ukuran populasi, probabilitas crossover, probabilitas mutasi, generasi, serta random seed. Hasil proses dalam bentuk grafik, dimana garis terbaik menunjukkan fitness terbaik yang diperoleh, dan garis terburuk menunjukkan fitness terburuk yang diperoleh dari proses. Pada form proses terdapat juga tombol proses, detail, simpan, new, dan exit. Untuk tombol proses digunakan untuk memproses data kapal, tujuan, serta parameter-parameter genetika. Tombol detail proses digunakan untuk menampilkan detail dari proses genetik algoritma berupa fitness terbaik, terburuk, serta rata-rata dari proses genetik, tombol baru untuk memulai proses baru.



Gambar 4.7 Form Proses Optimasi Jadwal dan Rute

Gambar dari tampilan setting parameter input data kapal serta tujuan dapat dilihat pada gambar 4.8 dan gambar 4.9. sedangkan gambar 4.10 menunjukkan form detail proses.

Kode Kapal	Nama Kapal	Kapasitas	Kecepatan
<input checked="" type="checkbox"/> KPL001	M.V. Surya Tulus	6470	10
<input type="checkbox"/> KPL002	M.V. Surya Kartika	6700	10
<input checked="" type="checkbox"/> KPL003	M.V. Caraka Jaya Niaga III-12	4371	10
<input type="checkbox"/> KPL004	M.V. Caraka Jaya Niaga III-6	4367	10
<input type="checkbox"/> KPL005	M.V. Maluku	4600	11
<input type="checkbox"/> KPL006	M.V. Sungai Bone	5340	9
<input type="checkbox"/> KPL007	M.V. Kalsel	4860	9
<input type="checkbox"/> KPL008	M.V. Surya Karya	6455	8
<input type="checkbox"/> KPL009	M.V. Surya Citra I	5530	8
<input type="checkbox"/> KPL010	M.V. Surya Citra II	5430	8

 SAVE  EXIT


Gambar 4.8 Input Kapal

Kode Tujuan	Tujuan	Order (ton)
TJN0003	Batulicin	0
TJN0004	Biringkasi	0
TJN0005	Bitung	600
TJN0006	Gorontalo	0
TJN0007	Jayapura	1150
TJN0008	Labuhan UKI	2500
TJN0009	Makassar	2000
TJN0010	Manukwari	0

 KALKULASI PEMESANAN  EXIT

Gambar 4.9 Input Tujuan

Generasi	Fitness Terbaik	Fitness Terburuk
1	507.8	631.5
2	507.8	577.9
3	507.8	507.8
4	507.8	747.5
5	507.8	631.5
6	507.8	507.8
7	507.8	747.5
8	507.8	507.8
9	507.8	507.8
10	507.8	507.8

 CLOSE

Gambar 4.10 Detail Proses

Untuk melihat hasil proses optimasi penjadwalan yaitu tampilan laporan untuk jadwal pengiriman barang, dapat dilihat seperti pada gambar 4.11.

28/9/2006



Jadwal dan Rute Pengiriman Barang Pelayaran PT. Surya , Surabaya

Tanggal Keberangkatan Kapal (3 hari dari tanggal cetak) : 1/10/2006

<u>KodeKapal</u>	<u>KPL001</u>	<u>Kapasitas Kapal</u>	6,470
<u>NamaKapal</u>	M.V. Surya Tulus	<u>Sisa Kapasitas Kapal</u>	220

<u>KodeTujuan</u>	<u>NamaTujuan</u>	<u>LamaPerjalanan</u>	<u>BeratOrder</u>
TJN0009	Makassar	93.70	2,000
	<u>Kode</u>	<u>Nama</u>	<u>Tujuan</u> <u>Berat</u>
	CST003	PT GERHANA	TJN0009 1,500
	CST007	CV SILOAM	TJN0009 500
TJN0008	Labuhan UKI	112.10	2,500
	<u>Kode</u>	<u>Nama</u>	<u>Tujuan</u> <u>Berat</u>
	CST002	PT DUTA KENCANA	TJN0008 2,500
TJN0005	Bitung	62.00	600
	<u>Kode</u>	<u>Nama</u>	<u>Tujuan</u> <u>Berat</u>
	CST005	PT HANS	TJN0005 600
TJN0007	Jayapura	118.10	1,150
	<u>Kode</u>	<u>Nama</u>	<u>Tujuan</u> <u>Berat</u>
	CST001	PT KARISMA JAYA	TJN0007 250
	CST004	CV SANJAYA	TJN0007 900
		385.90 Jam	6,250 Ton

<u>KodeKapal</u>	<u>KPL003</u>	<u>Kapasitas Kapal</u>	4,371
<u>NamaKapal</u>	M.V. Caraka Jaya Niaga III-12	<u>Sisa Kapasitas Kapal</u>	3.171

<u>KodeTujuan</u>	<u>NamaTujuan</u>	<u>LamaPerjalanan</u>	<u>BeratOrder</u>
TJN0013	Toli-toli	121.90	1,200
	<u>Kode</u>	<u>Nama</u>	<u>Tujuan</u> <u>Berat</u>
	CST006	PT WISATA	TJN0013 1,200
		121.90 Jam	1,200 Ton

Catatan :

Lama Perjalanan = Waktu dari Surabaya ke Tujuan + Waktu Bongkar Muat 48 jam

Hasil Perolehan Waktu yang Optimal (jam) : **507.80**

Gambar 4.11 Laporan Jadwal dan Rute Pengiriman Barang

4.2 Evaluasi Sistem

Pada bagian evaluasi ini akan dilakukan uji coba aplikasi yang telah dibuat. Kapal yang digunakan sejumlah dua kapal dengan kode kapal KPL001 dan KPL003. Terdapat 5 data tujuan pengiriman adalah Bitung dengan berat total order 600 ton, Jayapura dengan berat total order 1150 ton, Labuhan UKI dengan berat total order 2500 ton, Makassar dengan berat total order 2000 ton, Toli-toli dengan berat total order 1200 ton. Dalam bagian ini dilakukan proses membandingkan hasil proses genetika. Hasil optimal dapat dilihat dari nilai fitness terkecil (minimal) yang dihasilkan oleh setiap kromosom yang terbentuk dari proses genetika. Berikut adalah hasil yang diperoleh dari proses genetika yang telah terbentuk berdasar beberapa kriteria yang digunakan :

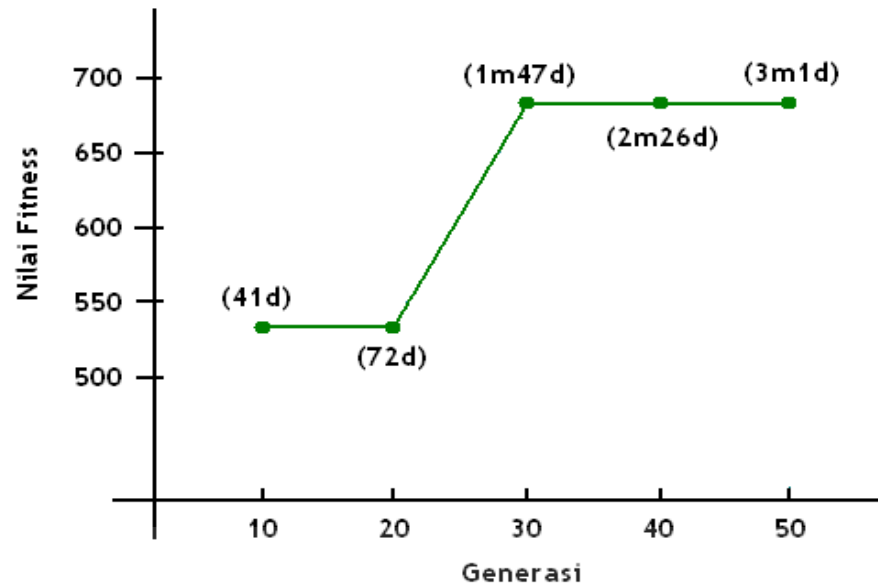
4.2.1 Uji Coba I Terhadap Generasi

Pada uji coba terhadap generasi ini, akan dilakukan pengujian proses genetika algoritma dengan melakukan uji coba yang memberikan nilai generasi yang berbeda-beda.

Tabel 4.1 Uji Coba I Generasi

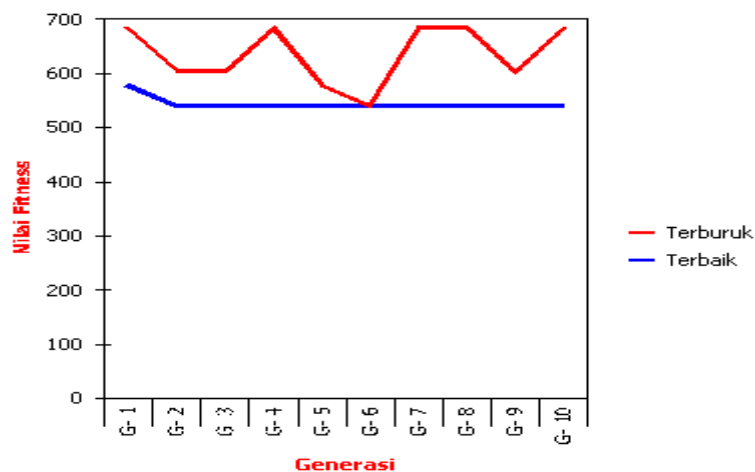
	Generasi				
	10	20	30	40	50
Populasi	30	30	30	30	30
Prob Crossover	65	65	65	65	65
Prob Mutasi	5	5	5	5	5
Fitness Min	539,7	539,7	683,4	683,4	683,4
Waktu Proses	41d	72d	1m47d	2m26d	3m1d
Random Seed	100	100	100	100	100

Grafik dari perubahan nilai fitness dari uji coba I dapat di lihat dalam gambar berikut, dimana nilai dalam tanda ”()” menunjukkan waktu proses.



Gambar 4.12 Grafik Uji Coba I Generasi

Dari hasil uji coba I, diperoleh nilai optimal untuk jumlah generasi yang digunakan adalah 10 generasi. Dibawah ini adalah gambar grafik dari hasil uji coba tersebut.



Gambar 4.13 Grafik Uji Coba 10 Generasi

Dan jadwal yang dihasilkan dari uji coba tersebut dapat dilihat pada gambar 4.14.

28/9/2006



Jadwal dan Rute Pengiriman Barang Pelayaran PT. Surya , Surabaya

Tanggal Keberangkatan Kapal (3 hari dari tanggal cetak): 1/10/2006

<u>KodeKapal</u>	<u>KPL001</u>	<u>Kapasitas Kapal</u>	6,470
<u>NamaKapal</u>	M.V. Surya Tulus	<u>Sisa Kapasitas Kapal</u>	820

<u>KodeTujuan</u>	<u>NamaTujuan</u>	<u>LamaPerjalanan</u>	<u>BeratOrder</u>
TJN0009	Makassar	93.70	2,000
	<u>Kode</u>	<u>Nama</u>	<u>Tujuan</u>
	CST003	PT GERHANA	TJN0009
	CST007	CV SILOAM	TJN0009
TJN0008	Labuhan UKI	112.10	2,500
	<u>Kode</u>	<u>Nama</u>	<u>Tujuan</u>
	CST002	PT DUTA KENCANA	TJN0008
TJN0007	Jayapura	132.10	1,150
	<u>Kode</u>	<u>Nama</u>	<u>Tujuan</u>
	CST001	PT KARISMA JAYA	TJN0007
	CST004	CV SANJAYA	TJN0007
		337.90 Jam	5,650 Ton

<u>KodeKapal</u>	<u>KPL003</u>	<u>Kapasitas Kapal</u>	4,371
<u>NamaKapal</u>	M.V. Caraka Jaya Niaga III-12	<u>Sisa Kapasitas Kapal</u>	2,571

<u>KodeTujuan</u>	<u>NamaTujuan</u>	<u>LamaPerjalanan</u>	<u>BeratOrder</u>
TJN0013	Toli-toli	121.90	1,200
	<u>Kode</u>	<u>Nama</u>	<u>Tujuan</u>
	CST006	PT WISATA	TJN0013
TJN0005	Bitung	79.90	600
	<u>Kode</u>	<u>Nama</u>	<u>Tujuan</u>
	CST005	PT HANS	TJN0005
		201.80 Jam	1,800 Ton

Catatan :

Lama Perjalanan = Waktu dari Surabaya ke Tujuan + Waktu Bongkar Muat 48 jam

Hasil Perolehan Waktu yang Optimal (jam) : 539.70

Gambar 4.14 Jadwal Hasil Uji Coba 10 Generasi

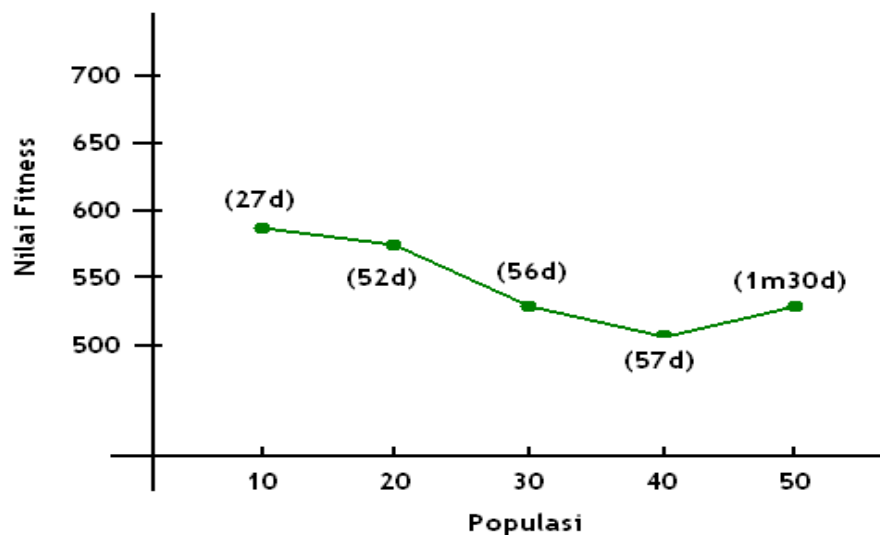
4.2.2 Uji Coba II Terhadap Populasi

Pada uji coba terhadap jumlah populasi ini, akan dilakukan pengujian proses genetika algoritma dengan melakukan uji coba yang memberikan jumlah populasi yang berbeda-beda.

Tabel 4.2 Uji Coba II Populasi

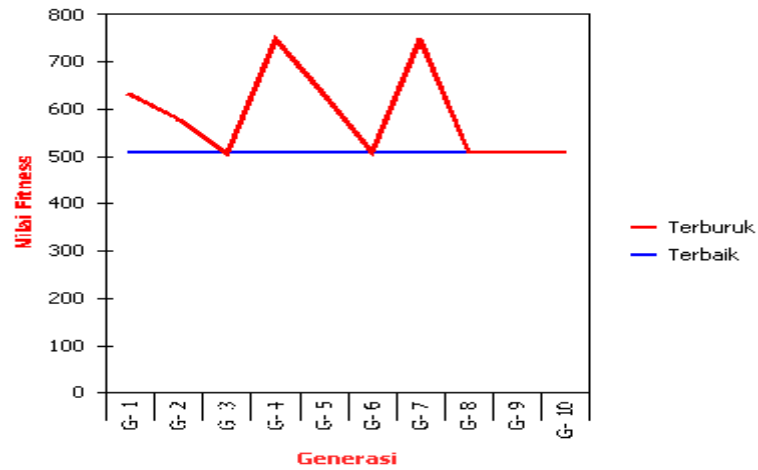
	Populasi				
	10	20	30	40	50
Generasi	10	10	10	10	10
Prob Crossover	65	65	65	65	65
Prob Mutasi	5	5	5	5	5
Fitness Min	589,7	577,1	539,7	507,8	539,7
Waktu Proses	27d	52d	56d	57d	1m30d
Random Seed	100	100	100	100	100

Grafik dari perubahan nilai fitness dari uji coba II dapat di lihat dalam gambar berikut, dimana nilai dalam tanda "()" menunjukkan waktu proses.



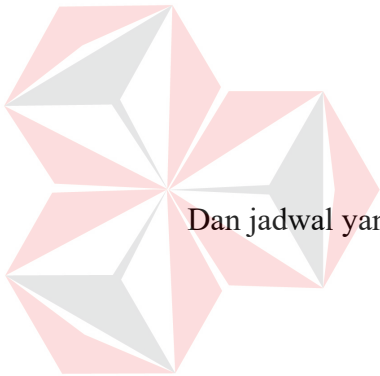
Gambar 4.15 Grafik Uji Coba II Populasi

Dari hasil uji coba II, diperoleh nilai optimal untuk jumlah populasi yang digunakan adalah 40. Dibawah ini adalah gambar dari grafik nilai fitness dari hasil uji coba II diatas.



Gambar 4.16 Grafik Uji Coba 40 Populasi

Dan jadwal yang dihasilkan dari uji coba tersebut dapat dilihat pada gambar 4.17.



28/9/2006



Jadwal dan Rute Pengiriman Barang Pelayaran PT. Surya , Surabaya

Tanggal Keberangkatan Kapal (3 hari dari tanggal cetak) : 1/10/2006

KodeKapal	KPL001	Kapasitas Kapal	6,470	
NamaKapal	M.V. Surya Tulus	Sisa Kapasitas Kapal	220	
KodeTujuan	NamaTujuan	LamaPerjalanan	BeratOrder	
TJN0009	Makassar	93.70	2,000	
	Kode	Nama	Tujuan	Berat
	CST003	PT GERHANA	TJN0009	1,500
	CST007	CV SILOAM	TJN0009	500
TJN0008	Labuhan UKI	112.10	2,500	
	Kode	Nama	Tujuan	Berat
	CST002	PT DUTA KENCANA	TJN0008	2,500
TJN0005	Bitung	62.00	600	
	Kode	Nama	Tujuan	Berat
	CST005	PT HANS	TJN0005	600
TJN0007	Jayapura	118.10	1,150	
	Kode	Nama	Tujuan	Berat
	CST001	PT KARISMA JAYA	TJN0007	250
	CST004	CV SANJAYA	TJN0007	900
		385.90 Jam	6,250 Ton	
KodeKapal	KPL003	Kapasitas Kapal	4,371	
NamaKapal	M.V. Caraka Jaya Niaga III-12	Sisa Kapasitas Kapal	3,171	
KodeTujuan	NamaTujuan	LamaPerjalanan	BeratOrder	
TJN0013	Toli-toli	121.90	1,200	
	Kode	Nama	Tujuan	Berat
	CST006	PT WISATA	TJN0013	1,200
		121.90 Jam	1,200 Ton	

Catatan :

Lama Perjalanan = Waktu dari Surabaya ke Tujuan + Waktu Bongkar Muat 48 jam

Hasil Perolehan Waktu yang Optimal (jam) : 507.80

Gambar 4.17 Jadwal Hasil Uji Coba 40 Populasi

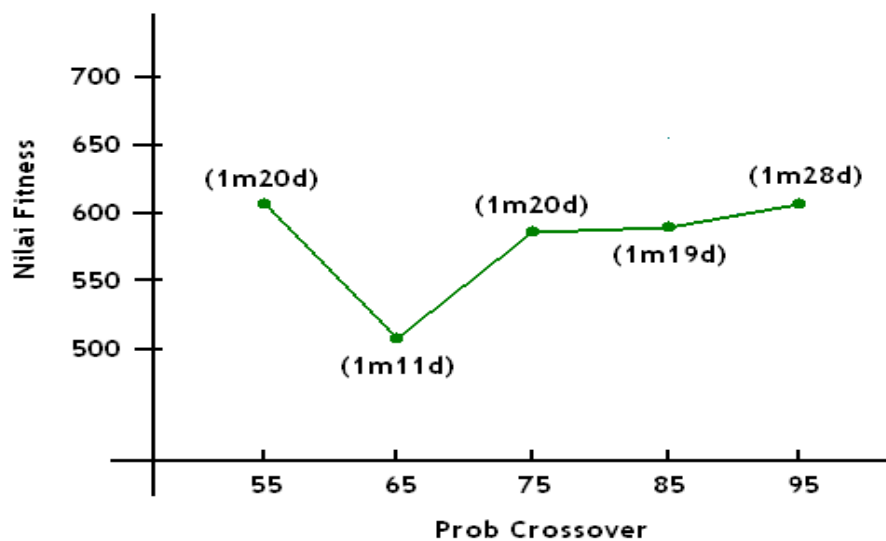
4.2.3 Uji Coba III Terhadap Probabilitas Crossover

Pada uji coba terhadap probabilitas crossover ini, akan dilakukan pengujian proses genetika algoritma dengan melakukan uji coba yang memberikan nilai probabilitas crossover yang berbeda-beda.

Tabel 4.3 Uji Coba III Probabilitas Crossover

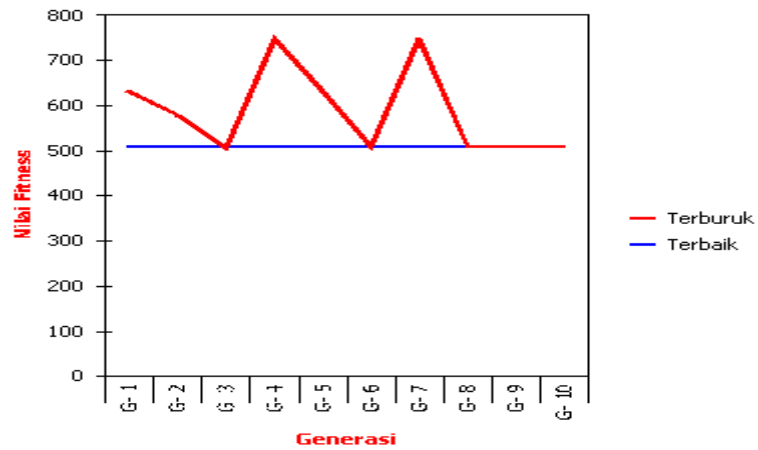
	Probabilitas Crossover (%)				
	55	65	75	85	95
Generasi	10	10	10	10	10
Populasi	40	40	40	40	40
Prob Mutasi	5	5	5	5	5
Fitness Min	603	507,8	589	593,4	603
Waktu Proses	1m20d	1m11d	1m20d	1m19d	1m28d
Random Seed	100	100	100	100	100

Grafik dari perubahan nilai fitness dari uji coba III dapat di lihat dalam gambar berikut, dimana nilai dalam tanda "()" menunjukkan waktu proses.



Gambar 4.18 Grafik Uji Coba III Probabilitas Crossover.

Dari hasil uji coba III, diperoleh nilai optimal untuk probabilitas crossover yang digunakan adalah 65%. Dibawah ini adalah gambar dari grafik nilai fitness dari hasil uji coba III diatas.



Gambar 4.19 Grafik Uji Coba Prob Cross 65%

Dan jadwal yang dihasilkan dari uji coba tersebut dapat dilihat pada gambar 4.20.



28/9/2006



Jadwal dan Rute Pengiriman Barang Pelayaran PT. Surya , Surabaya

Tanggal Keberangkatan Kapal (3 hari dari tanggal cetak) : 1/10/2006

<u>KodeKapal</u>	<u>KPL001</u>	<u>Kapasitas Kapal</u>	6,470
<u>NamaKapal</u>	M.V. Surya Tulus	<u>Sisa Kapasitas Kapal</u>	220
<u>KodeTujuan</u>	<u>NamaTujuan</u>	<u>LamaPerjalanan</u>	<u>BeratOrder</u>
TJN0009	Makassar	93.70	2,000
	<u>Kode</u>	<u>Nama</u>	<u>Tujuan</u> <u>Berat</u>
	CST003	PT GERHANA	TJN0009 1,500
	CST007	CV SILOAM	TJN0009 500
TJN0008	Labuhan UKI	112.10	2,500
	<u>Kode</u>	<u>Nama</u>	<u>Tujuan</u> <u>Berat</u>
	CST002	PT DUTA KENCANA	TJN0008 2,500
TJN0005	Bitung	62.00	600
	<u>Kode</u>	<u>Nama</u>	<u>Tujuan</u> <u>Berat</u>
	CST005	PT HANS	TJN0005 600
TJN0007	Jayapura	118.10	1,150
	<u>Kode</u>	<u>Nama</u>	<u>Tujuan</u> <u>Berat</u>
	CST001	PT KARISMA JAYA	TJN0007 250
	CST004	CV SANJAYA	TJN0007 900
		385.90 Jam	6,250 Ton
<u>KodeKapal</u>	<u>KPL003</u>	<u>Kapasitas Kapal</u>	4,371
<u>NamaKapal</u>	M.V. Caraka Jaya Niaga III-12	<u>Sisa Kapasitas Kapal</u>	3.171
<u>KodeTujuan</u>	<u>NamaTujuan</u>	<u>LamaPerjalanan</u>	<u>BeratOrder</u>
TJN0013	Toli-toli	121.90	1,200
	<u>Kode</u>	<u>Nama</u>	<u>Tujuan</u> <u>Berat</u>
	CST006	PT WISATA	TJN0013 1,200
		121.90 Jam	1,200 Ton

Catatan :

Lama Perjalanan = Waktu dari Surabaya ke Tujuan + Waktu Bongkar Muat 48 jam

Hasil Perolehan Waktu yang Optimal (jam) : 507.80

Gambar 4.20 Jadwal Hasil Uji Coba Probabilitas Crossover 65%

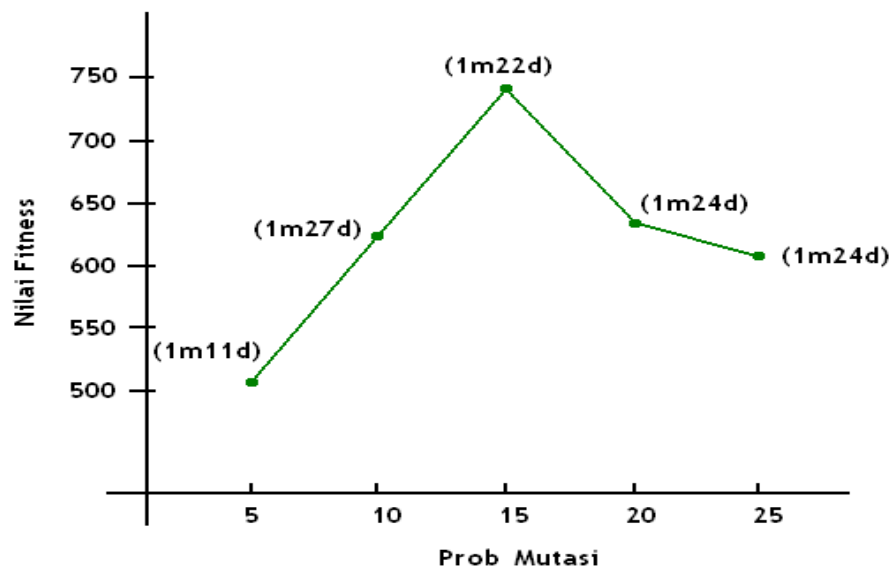
4.2.4 Uji Coba IV Terhadap Probabilitas Mutasi

Pada uji coba terhadap probabilitas mutasi ini, akan dilakukan pengujian proses genetika algoritma dengan melakukan uji coba yang memberikan nilai probabilitas mutasi yang berbeda-beda.

Tabel 4.4 Uji Coba IV Probabilitas Mutasi

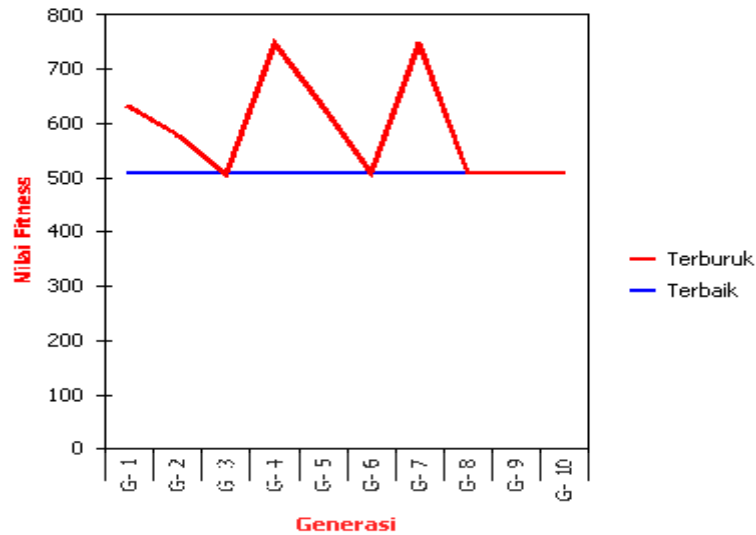
	Probabilitas Mutasi (%)				
	5	10	15	20	25
Generasi	10	10	10	10	10
Populasi	40	40	40	40	40
Prob Crossover	65	65	65	65	65
Fitness Min	507,8	623,8	743,6	631,8	609
Waktu Proses (dtk)	1m11d	1m27d	1m22d	1m24d	1m24d
Random Seed	100	100	100	100	100

Grafik dari perubahan nilai fitness dari uji coba IV dapat di lihat dalam gambar berikut, dimana nilai dalam tanda "()" menunjukkan waktu proses.



Gambar 4.21 Grafik Uji Coba IV Probabilitas Mutasi

Dari hasil uji coba IV, diperoleh nilai optimal untuk probabilitas mutasi yang digunakan adalah 5%. Dibawah ini adalah gambar dari grafik nilai fitness dari hasil uji coba IV diatas.



Gambar 4.22 Grafik Uji Coba Prob Mutasi 5%

Dari beberapa uji coba yang telah dilakukan dalam proses pengujian program, sebagai pengukur hasil percobaan digunakan kriteria-kriteria nilai terbaik atau rata-rata. Dan hasil uji coba diatas dapat diperoleh solusi optimal berdasar nilai fitness terkecil (minimal), dengan generasi 10, populasi 40, probabilitas crossover 65%, probabilitas mutasi 5% dan nilai fitness sebesar 507,8 jam dengan penggunaan random seed 100. Jadwal pengiriman yang telah dihasilkan dapat dilihat pada gambar 4.23 dibawah ini.

28/9/2006



Jadwal dan Rute Pengiriman Barang Pelayaran PT. Surya , Surabaya

Tanggal Keberangkatan Kapal (3 hari dari tanggal cetak) : 1/10/2006

KodeKapal	KPL001	Kapasitas Kapal	6,470	
NamaKapal	M.V. Surya Tulus	Sisa Kapasitas Kapal	220	
KodeTujuan	NamaTujuan	LamaPerjalanan	BeratOrder	
TJN0009	Makassar	93.70	2,000	
	Kode	Nama	Tujuan	Berat
	CST003	PT GERHANA	TJN0009	1,500
	CST007	CV SILOAM	TJN0009	500
TJN0008	Labuhan UKI	112.10	2,500	
	Kode	Nama	Tujuan	Berat
	CST002	PT DUTA KENCANA	TJN0008	2,500
TJN0005	Bitung	62.00	600	
	Kode	Nama	Tujuan	Berat
	CST005	PT HANS	TJN0005	600
TJN0007	Jayapura	118.10	1,150	
	Kode	Nama	Tujuan	Berat
	CST001	PT KARISMA JAYA	TJN0007	250
	CST004	CV SANJAYA	TJN0007	900
		385.90 Jam	6,250 Ton	
KodeKapal	KPL003	Kapasitas Kapal	4,371	
NamaKapal	M.V. Caraka Jaya Niaga III-12	Sisa Kapasitas Kapal	3,171	
KodeTujuan	NamaTujuan	LamaPerjalanan	BeratOrder	
TJN0013	Toli-toli	121.90	1,200	
	Kode	Nama	Tujuan	Berat
	CST006	PT WISATA	TJN0013	1,200
		121.90 Jam	1,200 Ton	

Catatan :

Lama Perjalanan = Waktu dari Surabaya ke Tujuan + Waktu Bongkar Muat 48 jam

Hasil Perolehan Waktu yang Optimal (jam) : 507.80

Gambar 4.23 Jadwal dan Rute Pengiriman Barang

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil uji coba yang diperoleh dari proses yang telah dilakukan dengan menggunakan aplikasi yang telah dibuat maka dapat diambil kesimpulan bahwa penerapan algoritma genetika pada pengaturan jadwal pengiriman barang pada pelayaran PT.Surya memberikan kemudahan kepada pihak pengguna untuk memperoleh jadwal pengiriman barang yang optimal.

Dengan penggunaan kombinasi rata-rata parameter genetika yaitu 10 generasi, jumlah populasi sebesar 40, probabilitas crossover 65%, dan probabilitas mutasi 5% dalam penggunaan random seed 100, dapat menghasilkan solusi yang lebih optimal dalam menghasilkan jadwal dan rute pengiriman barang dengan menggunakan kapal sebagai alat transportasinya.

5.2 Saran

Permasalahan masalah optimasi penjadwalan dan rute pengiriman barang pada tugas akhir ini dapat dikembangkan lebih lanjut dengan beberapa saran sebagai berikut :

1. Pada form input jarak dapat dibuat tampilan yang menampilkan peta dari tujuan pengiriman.
2. Dapat mengoptimasi jadwal pengiriman barang pada saat kapal mengangkut barang dari tujuan lain yang tidak ada pada jadwal awal yang

telah dibuat, tanpa harus melalui rute yang sama sesuai jadwal awal pengiriman.

3. Dapat mengoptimalkan penataan barang dalam kapal, apabila komoditi yang diangkut oleh kapal lebih dari satu jenis barang.



UNIVERSITAS
Dinamika

DAFTAR PUSTAKA

Gen, Mutsuo and Cheng, Runwei, 1997. *Generic Algorithm And Engineering Design*, A Shikaga Institute of Techonology A Shikaga, Japan.

Homberger, J., dan H. Gehring, 1999, *Two Evolutionary Metaheuristics for The Vehicle Routing Problem with Time Windows*, *INFOR*, vol. 37, 297-318.

Houpt R.L. And Houpt S.E., 1998, *Practical Genetic Algorithms*, John Wiley, London.

Kallehauge, B., J. Larsen, dan O.B.G. Marsen, 2001, *Lagrangean Duality Applied on Vehicle Routing with Time Windows*, Technical Report, IMM, Technical University of Denmark.

Kendall, Kenneth E. and Kendall, Julie E., 2002, *System Analisis And Design*, Rutgers University School of Businnes-Camden, Camden, New Jersey.

Kusumadewi, Sri, 2005, *Penyelesaian Masalah Optimasi Dengan Teknik-Teknik Heuristik*, Graha Ilmu, Yogyakarta.

Michalewicz, Zbigniew, 1996, *Genetic Algorithm + Data Structure = Evolution Programs*, Third Revised and Extended Edition, Springer-Verlag Berlin Heidelberg, New York.

Prins, C., 2001, *A Simple and Effective Evolutionary Algorithms for The Vehicle Routing Problem*, 4th Metaheuristics International Conference, 143-147.

Smith, Alies E., 1997, *Local Search Genetic Algorithms for Optimal Design of Reliable Networks*, (online), (<http://www.cs.nott.ac.uk/~gxx/papers/ieecc.pdf>) diakses 10 maret 2006.

Setiawan, Kuswara, 2003, *Paradigma Sistem Cerdas*, Banyumedia Publising, Malang.

Sutapa, I Nyoman, dan Christine, 2003, *Studi Tentang Travelling Salesman Dan Vehicle Routing Problem Dengan Time Window*, (online), (<http://puslit.petra.ac.id/journals/industrial>) diakses 15 april 2006.

Thangiah, S.R., 1995, *Vehicle Routing with Time Windows Using Genetic Algorithms*, Application Handbook of Genetic Algorithms: New Frontiers, Vol. II, Lance Chambers, 253-277.



UNIVERSITAS
Dinamika