

**PENERAPAN ALGORITMA GENETIKA  
UNTUK PERMASALAHAN OPTIMASI DISTRIBUSI BARANG  
DUA TAHAP**

**SKRIPSI**

**LABORATORIUM KOMPUTASI CERDAS VISUALISASI**

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun oleh :

**RISKA SULISTIYORINI**

**NIM. 115060801111030**

**KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI  
PROGRAM STUDI INFORMATIKA/ ILMU KOMPUTER  
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA**

**MALANG**

**2015**

**LEMBAR PERSETUJUAN**

**PENERAPAN ALGORITMA GENETIKA  
UNTUK PERMASALAHAN OPTIMASI DISTRIBUSI BARANG  
DUA TAHAP**

**SKRIPSI**

**LABORATORIUM KOMPUTASI CERDAS VISUALISASI**

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun oleh :

**RISKA SULISTIYORINI**

**NIM. 115060801111030**

Skripsi ini telah disetujui oleh dosen pembimbing pada tanggal 24 April 2015

**Dosen Pembimbing**

**Wayan Firdaus Mahmudy, S.Si., M.T.,Ph.D**

**NIP. 19720919 199702 1 001**

**LEMBAR PENGESAHAN**

**PENERAPAN ALGORITMA GENETIKA  
UNTUK PERMASALAHAN OPTIMASI DISTRIBUSI BARANG  
DUA TAHAP**

**SKRIPSI**

**LABORATORIUM KOMPUTASI CERDAS DAN VISUALISASI**  
Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun Oleh :

**RISKA SULISTIYORINI**  
**NIM. 115060801111030**

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 24 April 2015

**Penguji I**

**Drs. Achmad Ridok, M.Kom.**  
**NIP. 19680825 199403 1 002**

**Penguji II**

**Candra Dewi, S.Kom., M.Sc**  
**NIP. 19771114 200312 2 001**

**Penguji III**

**Budi Darma Setiawan S.Kom., M.Sc**  
**NIP. 198410152014041002**

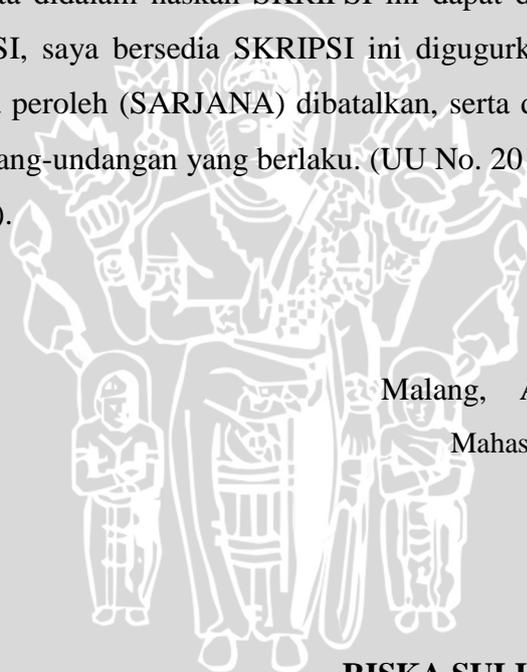
**Mengetahui,**  
**Ketua Program Studi Informatika/Illmu Komputer**

**Drs. Marji, M.T.**  
**NIP. 19670801 199203 1 001**

## PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).



Malang, April 2015

Mahasiswa,

**RISKA SULISTIYORINI**

**NIM. 115060801111030**

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena atas segala rahmat dan karunia-Nya, penulis dapat menyelesaikan skripsi yang berjudul **“Penerapan Algoritma Genetika untuk Permasalahan Optimasi Distribusi Barang Dua Tahap”**.

Skripsi ini merupakan tugas akhir yang diajukan untuk memenuhi syarat dalam memperoleh gelar Sarjana Komputer pada Program Teknologi informasi dan Ilmu Komputer Universitas Brawijaya Malang . Penyusunan skripsi ini juga tidak lepas dari bantuan berupa bimbingan, kritik, saran, dukungan, motivasi maupun doa banyak pihak. Oleh karena itu pada kesempatan ini penulis menyampaikan banyak terima kasih kepada :

1. Wayan Firdaus Mahmudy, S.Si.,M.T.,P.hd selaku dosen pembimbing atas segala bimbingan, waktu yang diluangkan serta kritik dan saran yang telah diberikan kepada penulis.
2. Ir. Sutrisno, M.T, Ir. Heru Nurwasito, M.Kom., Himawat Aryadita, S.T, M.Sc., dan Edy Santoso, S.Si., M.Kom., selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
3. Drs. Marji, M.T., dan Issa Arwani, S.Kom, M.Sc., selaku Ketua dan Sekretaris Program Studi Teknik Informatika Universitas Brawijaya.
4. Seluruh bapak dan ibu dosen Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang atas segala bimbingan serta ilmu yang telah diajarkan kepada penulis.
5. Kedua orang tua penulis Ibunda Supiyatun, Ayahanda Sunyoto yang tersayang dan tercinta atas segala dukungan, motivasi, doa dan segala pengorbanan baik materil maupun moril. Serta seluruh keluarga atas segenap dukungan dan kasih sayang yang telah diberikan.
6. Shindu Nata Gama atas bantuan, dukungan dan doa pada masa penyelesaian skripsi ini.

7. Teman seperjuangan dalam mengerjakan skripsi, Candra Bella, Fauzia, Yasmin atas bantuan dan motivasi dalam menyelesaikan skripsi.
8. Keluarga One Day One Juz yang telah memberikan semangat dan doa pada penyelesaian skripsi.
9. Keluarga Display PTIHK UB atas semangat dan doanya.
10. Teman-teman Informatika 2011 yang telah memberikan bantuan baik moril maupun spiritual.
11. Teman-teman kos Terusan Cikampek no.34 atas semangat dan doa yang telah diberikan.
12. Seluruh Civitas Akademika Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini. Dan semua pihak yang tidak bisa disebutkan satu per satu. Terima kasih atas segala bantuannya.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna serta banyak kekurangan disebabkan oleh keterbatasan kemampuan dan pengalaman, dengan kerendahan hati penulis mengharapkan kritik dan saran yang bersifat membangun dari pembaca.

Malang, 4 Mei 2015

Penulis

## ABSTRAK

**Riska Sulistiyorini, 2015 : Penerapan Algoritma Genetika untuk Permasalahan Optimasi Distribusi Barang Dua Tahap. Skripsi Program Studi Informatika / Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang.**

**Dosen Pembimbing : Wayan Firdaus Mahmudy, S.Si., MT., Ph.D**

Saluran distribusi merupakan aspek penting bagi perusahaan untuk memasarkan produk. Jalur distribusi yang optimal mampu memberikan keuntungan untuk menekan biaya distribusi. Permasalahan inti pada distribusi adalah apabila wilayah pemasaran produk semakin luas, produsen akan mengeluarkan biaya transportasi yang besar untuk mendistribusikan produk. Distribusi barang dua tahap digunakan untuk mempermudah pendistribusian barang dari produsen kepada konsumen dengan membangun sebuah distributor yang berdekatan dengan wilayah pemasaran konsumen. Algoritma genetika diterapkan pada kasus distribusi barang dua tahap dengan mengoptimasi rute distribusi barang untuk mendapatkan biaya distribusi minimum. Optimasi rute distribusi menggunakan teknik *one cut point crossover*, mutasi dengan *exchange mutation* dan seleksi menggunakan *elitism selection*. Pada penelitian ini digunakan 2 pabrik, 5 distributor dan 10 agen yang direpresentasikan menggunakan representasi permutasi. Solusi optimal diperoleh pada ukuran populasi sebanyak 80, kombinasi *crossover rate* (*cr*) 0,4 dan *mutation rate* (*mr*) 0,6, jumlah generasi sebanyak 80 dan mutasi segmen campuran yang memperoleh rata-rata nilai *fitness* tertinggi yaitu 0,166514.

**Kata Kunci :** Algoritma genetika, optimasi rute distribusi, distribusi dua tahap.

## ABSTRACT

**Riska Sulistiyorini, 2015** : *Genetic Algorithm for Optimization Two Stage Distribution Problems. Undergraduate Thesis of Informatics / Computer Science Study Program, Program on Informatic Technology and Computer Science, University of Brawijaya, Malang.*

**Advisor : Wayan Firdaus Mahmudy, S.Si., MT., Ph.D**

*Distribution channels is an important aspect for companies to market their products. Optimal distribution channels give benefits to reduce distribution costs. The main problem in distribution is reducing transportation costs to distribute the product when the product marketing area becomes more wide. Two-stage distribution is used to ease the distribution of goods from producers to consumers by building a distributor that is near to the area of consumer marketing. Genetic algorithm is applied to the case of two-stage distribution to optimise the distribution route of the goods to get the minimum distribution costs. Distribution route optimization uses one cut point crossover techniques, mutation by exchange mutation and selection using elitism selection. The case study for testing uses 2 factories, 5 distributors and 10 agents which are represented by using permutation representation. The optimal solution is obtained from a population size of 80, combination crossover rate (cr) of 0.4 and mutation rate (mr) of 0.6, the generations size of 80 and mix mutation segments which obtain the highest fitness value of 0.166514.*

**Keywords** : *Genetic algorithms, distribution route optimization, two stages distribution*

## DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
LEMBAR PENGESAHAN .....	ii
PERNYATAAN.....	iii
KATA PENGANTAR .....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR .....	xii
DAFTAR TABEL.....	xiv
DAFTAR PERSAMAAN.....	xvi
DAFTAR KODE PROGRAM.....	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Sistematika Penulisan.....	5
BAB II KAJIAN PUSTAKA DAN DASAR TEORI.....	6
2.1 Kajian Pustaka.....	6
2.2 Distribusi .....	7
2.3 Distribusi Dua Tahap.....	7
2.4 Penyelesaian dan Pemodelan Matematis Transportasi pada Distribusi Dua Tahap .....	8

2.4.1 Variabel.....	9
2.4.2 Fungsi Obyektif .....	9
2.4.3 <i>Constraint</i> (Batasan).....	9
2.5 Algoritma Genetika .....	10
2.5.1 Siklus Algoritma Genetika.....	11
2.6 Penerapan Algoritma Genetika .....	11
2.6.1 Representasi Kromosom.....	12
2.6.2 Nilai <i>Fitness</i> .....	13
2.6.3 Membangkitkan Populasi Awal.....	13
2.6.4 Fase Reproduksi.....	14
2.6.5 Seleksi.....	16
2.7 Parameter Genetika .....	18
<b>BAB III METODOLOGI PENELITIAN.....</b>	<b>20</b>
3.1 Tahapan Penelitian .....	20
3.1.1 Studi Literatur .....	21
3.1.2 Analisis Kebutuhan dan Perancangan.....	21
3.1.3 Implementasi Perangkat Lunak.....	22
3.1.4 Uji Coba Perangkat Lunak.....	22
3.1.5 Evaluasi Hasil Uji Coba dan Pengambilan Kesimpulan.....	23
3.2 Kebutuhan Sistem.....	23
3.2.1 Kebutuhan Perangkat Keras.....	23
3.2.2 Kebutuhan Perangkat Lunak.....	23
3.3 Formulasi Masalah .....	24
3.3.1 Deskripsi Umum .....	24
3.3.2 Deskripsi Data.....	26

3.4	Siklus Penyelesaian Masalah Optimasi Distribusi Barang Dua Tahap dengan GA's.....	27
3.4.1	Representasi Kromosom dan Perhitungan Nilai <i>Fitness</i> .....	29
3.4.2	Inisialisasi Populasi Awal.....	37
3.4.3	Proses <i>Crossover</i> .....	38
3.4.4	Proses Mutasi.....	41
3.4.5	Seleksi.....	42
BAB IV PERANCANGAN.....		47
4.1	Perancangan Database.....	47
4.2	Perancangan Antarmuka.....	48
4.2.1	Halaman Beranda.....	49
4.2.2	Menu Pabrik.....	49
4.2.3	Manu Distributor.....	50
4.2.4	Menu Agen.....	51
4.2.5	Menu Biaya Transportasi.....	51
4.2.6	Menu Perhitungan.....	52
4.3	Perancangan Pengujian.....	54
4.3.1	Pengujian Ukuran Populasi.....	54
4.3.2	Perancangan Pengujian <i>Crossover Rate</i> (cr) dan <i>Mutation Rate</i> (mr) 54	
4.3.3	Perancangan Pengujian Jumlah Generasi.....	55
4.3.4	Perancangan Pengujian Jenis Segmen Mutasi.....	56
BAB V IMPLEMENTASI.....		57
5.1	Implementasi Tabel Database.....	57
5.3	Implementasi Kode Program.....	57
5.3.1	Implementasi Inisialisasi Populasi Awal.....	57



5.3.2	Implementasi Perhitungan Nilai <i>Fitness</i> .....	59
5.3.3	Implementasi <i>Crossover</i> .....	61
5.3.3	Implementasi Mutasi.....	63
5.3.5	Implementasi Proses Seleksi.....	66
5.4	Implementasi Antar Muka.....	67
<b>BAB VI PENGUJIAN DAN ANALISIS .....</b>		<b>76</b>
6.1	Sistematika Pengujian .....	76
6.2	Hasil dan Pembahasan.....	76
6.2.1	Hasil Pengujian Terhadap Ukuran Populasi .....	76
6.2.2	Hasil Pengujian Terhadap <i>Crossover Rate</i> dan <i>Mutation Rate</i> .....	78
6.2.3	Hasil Pengujian Terhadap Jumlah Generasi .....	80
6.2.4	Hasil Pengujian Terhadap Perbandingan Segmen Mutasi .....	82
6.3	Solusi Terbaik yang Pernah Didapatkan .....	83
<b>BAB VII PENUTUP .....</b>		<b>86</b>
7.1	Kesimpulan.....	86
7.2	Saran.....	87
<b>DAFTAR PUSTAKA .....</b>		<b>88</b>

## DAFTAR GAMBAR

Gambar 2.1 Model Distribusi Dua Tahap .....	8
Gambar 2.2 Siklus algoritma genetika Zbigniew Michalewicz .....	11
Gambar 2.3 Bentuk kromosom pada representasi permutasi .....	12
Gambar 2.4 <i>Crossover</i> pada representasi permutasi .....	15
Gambar 2.5 Proses mutasi dengan <i>reciprocal exchange mutation</i> .....	16
Gambar 2.6 Proses mutasi dengan <i>insertion mutation</i> .....	16
Gambar 2.7 Pseudocode seleksi dengan elitism .....	18
Gambar 3.1 Diagram alir metode penelitian .....	20
Gambar 3.2 Diagram Alir Proses Optimasi Distribusi Dua Tahap dengan GA's .....	28
Gambar 3.3 Contoh Rancangan Kromosom .....	30
Gambar 3.4 Diagram Alir Proses Perhitungan <i>Fitness</i> .....	34
Gambar 3.5 Diagram Alir Pembentukan Populasi Awal .....	37
Gambar 3.6 Diagram Alir Proses <i>Crossover</i> .....	39
Gambar 3.7 <i>Child</i> Hasil <i>Crossover</i> P2 dan P4 .....	40
Gambar 3.8 <i>Child</i> Hasil <i>Crossover</i> P3 dan P5 .....	41
Gambar 3.9 Diagram Alir Proses Mutasi .....	41
Gambar 3.10 Hasil Mutasi .....	42
Gambar 3.11 Diagram Alir Proses Seleksi .....	43
Gambar 4.1 Tabel Pabrik .....	47
Gambar 4.2 Tabel Distributor .....	47
Gambar 4.3 Tabel Agen .....	48
Gambar 4.4 Tabel Biaya Transportasi .....	48
Gambar 4.5 Perancangan Antarmuka Halaman Beranda .....	49
Gambar 4.6 Perancangan Antarmuka Halaman Pabrik .....	50
Gambar 4.7 Perancangan Antarmuka Halaman Distributor .....	50
Gambar 4.8 Perancangan Antarmuka Halaman Agen .....	51
Gambar 4.9 Biaya Transportasi .....	51
Gambar 4.10 Perancangan Antarmuka Halaman Perhitungan .....	52
Gambar 4.11 Perancangan Antarmuka Halaman Hasil Perhitungan .....	53
Gambar 5.1 Implementasi Tabel dalam Sistem. ....	57

Gambar 5.2 Halaman <i>Home</i> .....	68
Gambar 5.3 Implementasi Halaman Pabrik .....	68
Gambar 5.4 Implementasi Halaman Distributor .....	69
Gambar 5.5 Implementasi Halaman Agen .....	70
Gambar 5.6 Implementasi Halaman Biaya Distribusi .....	70
Gambar 5.7 Halaman <i>Input</i> Parameter .....	71
Gambar 5.8 Halaman Hasil <i>Input</i> -an Parameter .....	72
Gambar 5.9 Tampilan Antarmuka Representasi Kromosom .....	72
Gambar 5.10 Tampilan Halaman <i>Crossover</i> .....	73
Gambar 5.11 Tampilan Halaman Mutasi .....	73
Gambar 5.12 Tampilan Halaman Evaluasi .....	74
Gambar 5.13 Tampilan Halaman Seleksi .....	74
Gambar 5.14 Jalur Distribusi Optimal .....	75
Gambar 5.15 Individu yang Lolos Seleksi .....	75
Gambar 6.1 Grafik Hasil Pengaruh Pengujian Ukuran Pupulasi .....	77
Gambar 6.2 Grafik Hasil Pengaruh Pengujian Kombinasi <i>cr</i> dan <i>mr</i> .....	79
Gambar 6.3 Grafik Hasil Pengaruh Pengujian Jumlah Generasi .....	81
Gambar 6.4 Grafik Hasil Pengaruh Pengujian Segmen Mutasi .....	83

## DAFTAR TABEL

Tabel 3.1 Data Pabrik dan Kapasitas Persediaan Pabrik.....	26
Tabel 3.2 Data Distributor dan Kapasitas Permintaan Distributor .....	26
Tabel 3.3 Data Agen dan Kapasitas Permintaan Agen .....	26
Tabel 3.4 Biaya Distribusi Pabrik ke Distributor.....	27
Tabel 3.5 Biaya Distribusi Distributor ke Agen .....	27
Tabel 3.6 Tabel Kapasitas Barang .....	30
Tabel 3.7 Biaya Distribusi Tahap 1 pabrik ke distributor.....	31
Tabel 3.8 Biaya Distribusi Tahap 2 distributor ke Agen .....	31
Tabel 3.9 Alokasi Distribusi dari Gen 3 .....	32
Tabel 3.10 Alokasi Distribusi dari Gen 4 .....	32
Tabel 3.11 Alokasi Distribusi dari Gen ke 6 dan 1 .....	32
Tabel 3.12 Alokasi Distribusi dari Gen ke 2.....	33
Tabel 3.13 Alokasi Distribusi dari Gen ke 5.....	33
Tabel 3.14 Alokasi Distribusi tahap 2.....	33
Tabel 3.15 Tabel biaya rute dari Pabrik ke Distributor.....	35
Tabel 3.16 Tabel biaya rute dari Distributor ke Agen .....	36
Tabel 3.17 Tabel Populasi Awal dan Nilai <i>Fitness</i> .....	38
Tabel 3.18 Pemilihan P2 dan P4 Sebagai <i>Parent</i> .....	40
Tabel 3.19 Pemilihan P3 dan P5 sebagai <i>Parent</i> .....	40
Tabel 3.20 P1 sebagai <i>Parent</i> Proses Mutasi.....	42
Tabel 3.21 Kumpulan Individu Populasi Awal dan Individu Hasil Reproduksi... 44	44
Tabel 3.22 Hasil Sorting Kumpulan Populasi.....	44
Tabel 3.23 Populasi Baru dan Individu Terbaik .....	45
Tabel 3.24 Tabel Matriks Distribusi Tahap 1 .....	45
Tabel 3.25 Tabel Matriks Distribusi Tahap 2 .....	45
Tabel 3.26 Tabel biaya rute dari Pabrik ke Distributor.....	46
Tabel 3.27 Tabel biaya rute dari Distributor ke Agen .....	46
Tabel 4.1 Pengujian Ukuran Populasi.....	54

Tabel 4.2 Perancangan Pengujian kombinasi *Crossover rate* (*cr*) dan *mutation rate* (*mr*) ..... 55

Tabel 4.3 Perancangan Pengujian Jumlah Generasi ..... 55

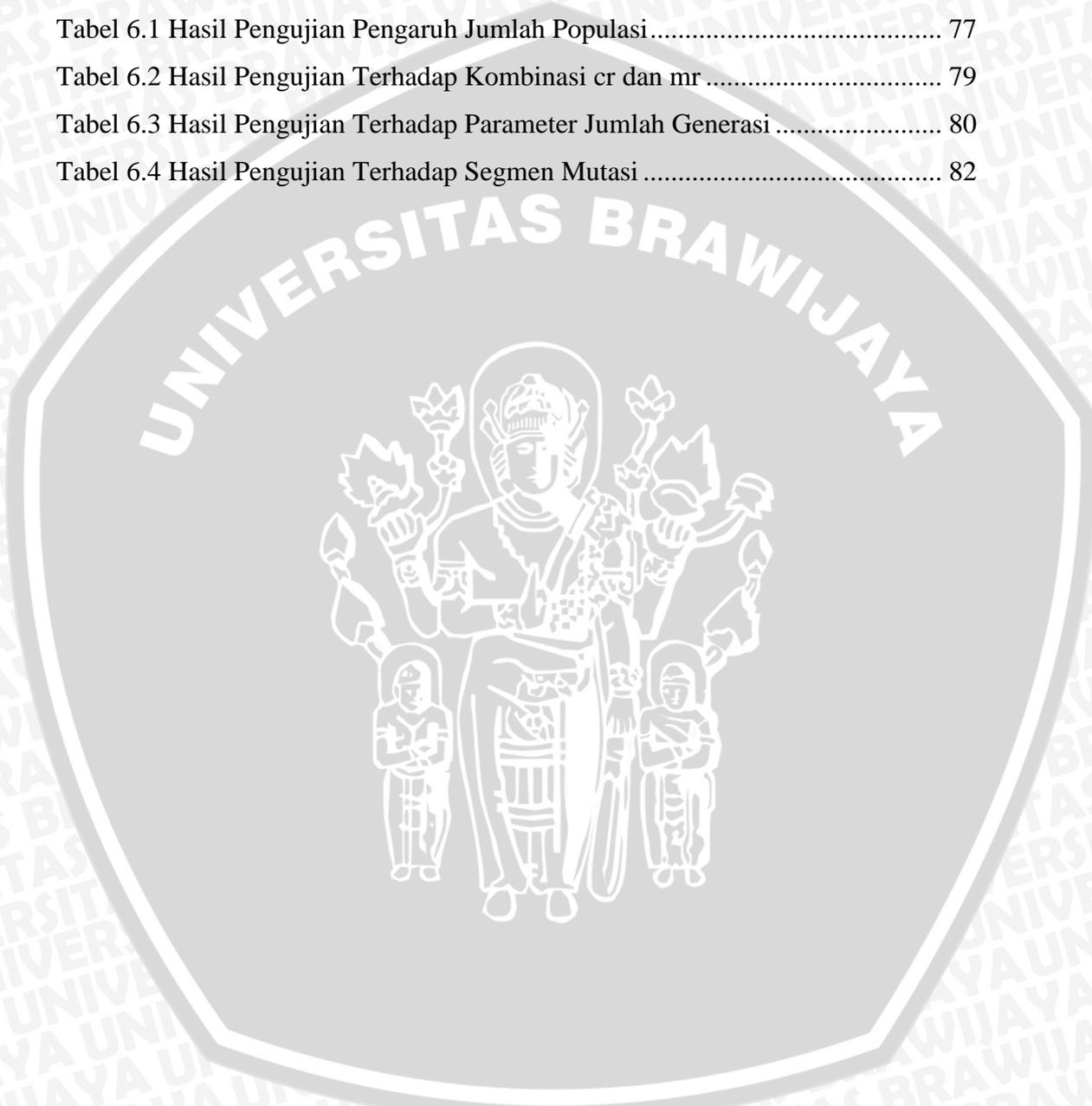
Tabel 4.4 Perancangan Pengujian Jenis Segmen Mutasi ..... 56

Tabel 6.1 Hasil Pengujian Pengaruh Jumlah Populasi..... 77

Tabel 6.2 Hasil Pengujian Terhadap Kombinasi *cr* dan *mr* ..... 79

Tabel 6.3 Hasil Pengujian Terhadap Parameter Jumlah Generasi ..... 80

Tabel 6.4 Hasil Pengujian Terhadap Segmen Mutasi ..... 82



**DAFTAR PERSAMAAN**

Persamaan 2-1 Fungsi Obyektif ..... 9

Persamaan 2-2 *Constrain* Kapasitas Barang Pabrik ..... 10

Persamaan 2-3 *Constrain* Kapasitas Barang Distributor ..... 5810

Persamaan 2-4 *Constrain* Permintaan Agen Terpenuhi..... 10

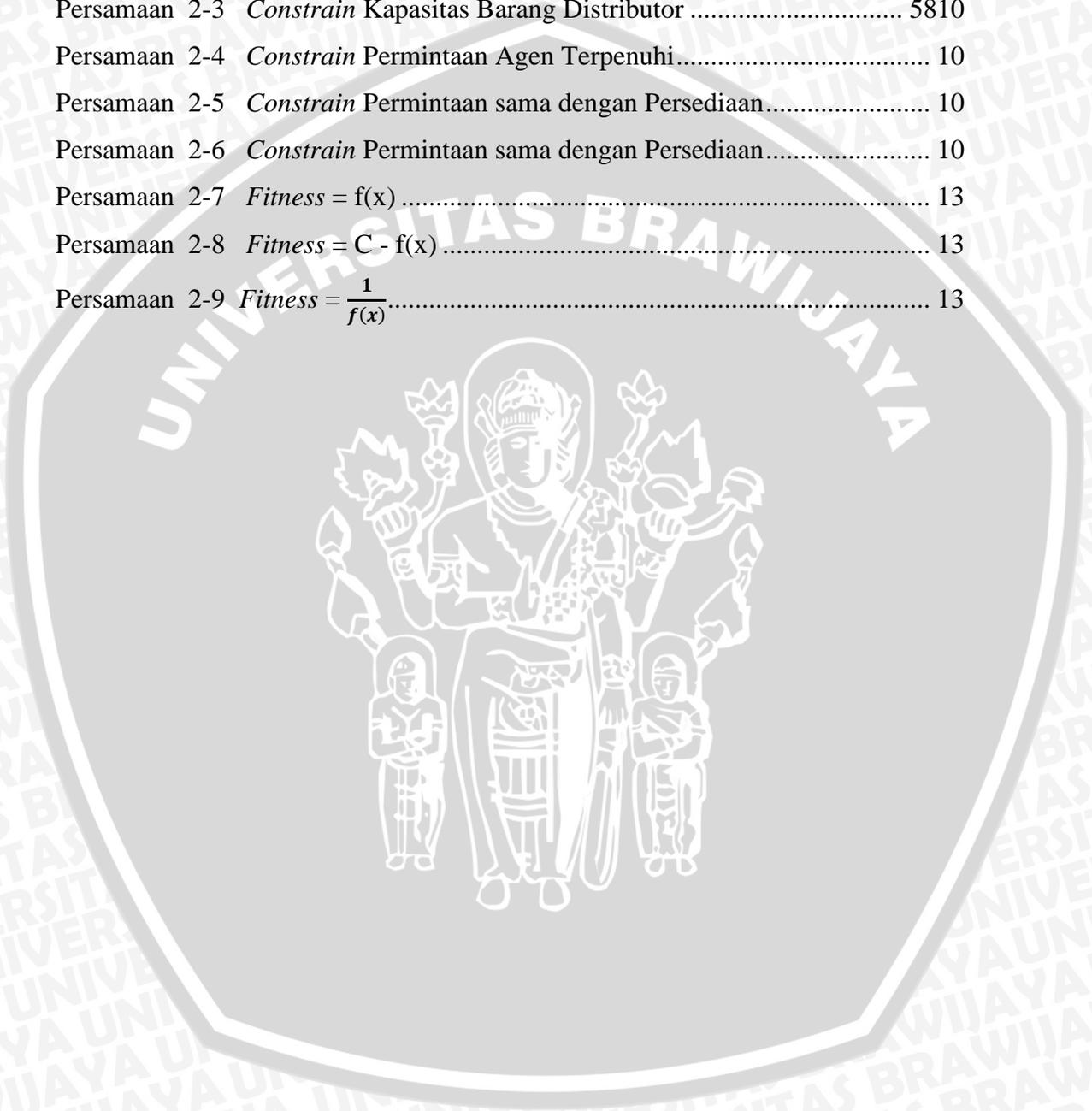
Persamaan 2-5 *Constrain* Permintaan sama dengan Persediaan..... 10

Persamaan 2-6 *Constrain* Permintaan sama dengan Persediaan..... 10

Persamaan 2-7 *Fitness* =  $f(x)$  ..... 13

Persamaan 2-8 *Fitness* =  $C - f(x)$  ..... 13

Persamaan 2-9 *Fitness* =  $\frac{1}{f(x)}$  ..... 13



**DAFTAR KODE PROGRAM**

Kode Program 5.1 Membentuk Individu ..... 58

Kode Program 5.2 Pembangkitan Populasi Awal..... 58

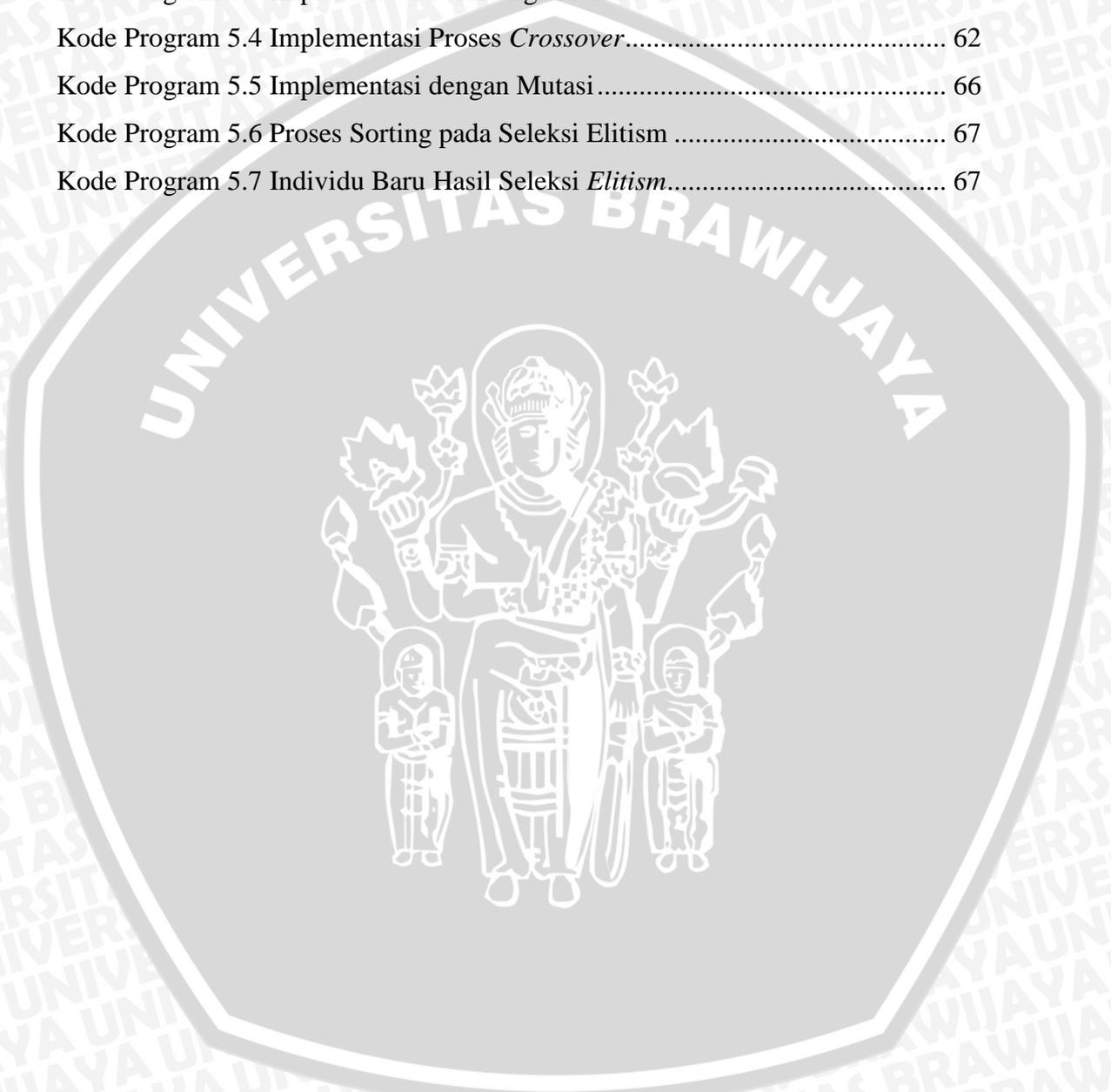
Kode Program 5.3 Implementasi Perhitungan Nilai *Fitness*..... 60

Kode Program 5.4 Implementasi Proses *Crossover*..... 62

Kode Program 5.5 Implementasi dengan Mutasi..... 66

Kode Program 5.6 Proses Sorting pada Seleksi Elitism ..... 67

Kode Program 5.7 Individu Baru Hasil Seleksi *Elitism*..... 67



## BAB I

### PENDAHULUAN

Pada bab ini dijelaskan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan dalam penelitian.

#### 1.1 Latar Belakang

Distribusi merupakan proses untuk memindahkan dan menyimpan barang dari produsen sampai pada tingkat konsumen pada suatu manajemen rantai pasok (Pujawan & Mahendrawati, 2010). Permasalahan utama pada distribusi adalah ketika wilayah pemasaran produk semakin luas, jalur distribusi yang tidak optimal menyebabkan produsen mengeluarkan biaya distribusi yang besar. Hal ini dikarenakan produsen harus mengirimkan barang secara langsung kepada konsumen akhir, dimana semakin jauh jangkauan tempat tujuan dari pusat distribusi maka biaya transportasi yang dikeluarkan juga akan semakin besar. Biaya distribusi menghabiskan sebanyak 30% dari total penjualan barang, oleh karena itu dibutuhkan saluran distribusi yang optimal. Saluran distribusi yang optimal memberikan kontribusi yang cukup besar dalam memangkas biaya transportasi dan lama waktu pendistribusian (Ballou dalam Sari, 2008). Saluran distribusi yang optimal merupakan suatu penunjang keberhasilan perusahaan dalam memasarkan produk dan dapat meningkatkan penjualan perusahaan.

Pada konsep dunia bisnis saat ini dibentuk rantai pasok atau *supply chain*. *Supply chain management* merupakan perusahaan-perusahaan yang bekerja bersama untuk menciptakan kemudahan pendistribusian barang sampai pada tingkat konsumen akhir (Pujawan & Mahendrawati, 2010). Pertama kali barang disediakan oleh produsen, kemudian dilanjutkan pendistribusian barang oleh perantara (distributor dan agen) kepada konsumen akhir. Hal yang penting pada jaringan rantai pasok adalah masalah pendistribusian produk, distribusi berfokus untuk membuat jalur penyebaran produk dari beberapa sumber menuju beberapa tujuan dengan jumlah produk yang telah ditentukan. Distribusi pada rantai pasok dapat dibagi menjadi distribusi satu tahap, distribusi dua tahap, distribusi tiga tahap

dan lain sebagainya tergantung pada wilayah cakupan distribusi (Kotler, 2003). Distribusi barang dua tahap dibangun untuk memudahkan pendistribusian barang dari produsen menuju konsumen. Pada distribusi dua tahap dibangun sebuah distributor dan agen yang berdekatan dengan wilayah konsumen. Tujuan utama dari distribusi barang dua tahap adalah untuk membuat jalur pendistribusian barang secara tepat sehingga dapat meminimumkan biaya distribusi.

Permasalahan untuk menentukan jalur distribusi yang optimal pada distribusi dua tahap dapat diselesaikan menggunakan metode *heuristic* yang didasarkan pada mekanisme evolusi biologis. Salah satu dari metode *heuristic* yang dapat digunakan untuk mendapatkan solusi yang optimal adalah algoritma genetika (Mahmudy, 2013). Algoritma genetika merupakan suatu metode algoritma optimasi pada prinsip genetika dan seleksi natural yang merupakan bagian dari algoritma evolusi. Secara umum algoritma genetika memiliki representasi genetik, cara membentuk populasi, pencarian *fitness*, memiliki nilai dan parameter genetik (Haupt, 2004). Algoritma genetika dinilai mempunyai hasil yang optimal untuk menyelesaikan masalah, hal ini telah dibuktikan bahwa algoritma genetika dapat menghasilkan himpunan solusi optimal yang sangat berguna dengan banyak obyek (Mahmudy, 2013). Kekuatan utama algoritma genetika adalah kemampuannya untuk menyelesaikan masalah kompleks dalam waktu yang relative cepat (Mahmudy, WF, Marian, RM, & Luong, 2014). Dengan kemampuannya ini, algoritma genetika sangat layak diterapkan untuk permasalahan distribusi barang dua tahap.

Pada penelitian sebelumnya telah dibahas mengenai algoritma genetika untuk permasalahan distribusi. Algoritma genetika telah diterapkan pada kasus penentuan rute optimum jalur distribusi menggunakan metode *weight mapping crossover* (WMX) dan *swapp mutation* untuk teknik mutasi mampu menentukan rute distribusi (Toni, 2009). Penerapan lainnya adalah algoritma genetika dengan dua populasi pada kasus transportasi dua tahap menggunakan metode *weight mapping crossover* (WMX) dan *swapp mutation* untuk teknik mutasi digunakan untuk menentukan jalur transportasi yang mendekati optimal pada pendistribusian barang (Prabowo, 2011).

Oleh karena itu, pada penelitian ini menggunakan algoritma genetika untuk menyelesaikan permasalahan optimasi distribusi barang dua tahap melalui penelitian yang berjudul “**Penerapan Algoritma Genetika untuk Permasalahan Optimasi Distribusi Barang Dua Tahap**”. Untuk membedakan penelitian ini dengan penelitian sebelumnya digunakan representasi permutasi dengan perkalian setiap subjek distribusi, metode *crossover* berupa *one cut point crossover*, metode mutasi *exchange mutation* dan seleksi menggunakan *elitism selection*. Dengan menerapkan algoritma tersebut, diharapkan dapat menghasilkan solusi yang lebih baik pada kasus optimasi distribusi barang.

## 1.2 Rumusan Masalah

Berdasarkan permasalahan yang diangkat pada bagian latar belakang, maka rumusan masalah dikhususkan pada :

1. Bagaimana menerapkan algoritma genetika untuk menyelesaikan masalah distribusi barang dua tahap.
2. Bagaimana menentukan representasi kromosom untuk kasus optimasi distribusi barang dua tahap.
3. Bagaimana pengaruh parameter algoritma genetika (ukuran populasi, *crossover rate* , *mutation rate* , dan jumlah generasi) terhadap nilai *fitness* dari hasil optimasi distribusi barang dua tahap.

## 1.3 Batasan Masalah

Dari permasalahan yang telah dipaparkan, berikut ini diuraikan batasan masalah untuk menghindari melebarnya masalah yang akan diselesaikan pada penelitian kali ini adalah :

1. Produk yang didistribusikan hanya satu macam produk.
2. Pengangkutan barang dari pabrik ke distributor dan pengangkutan barang dari distributor ke agen menggunakan jenis kendaraan yang sama.
3. Parameter penentuan optimasi adalah jalur distribusi barang yang optimal, dan total biaya distribusi tanpa memperhitungkan jadwal produksi, jumlah kendaraan, beban gaji supir dan pengaruh lainnya.

4. *Input* data ke sistem terdiri dari parameter algoritma genetika dan parameter kasus. Parameter algoritma genetika adalah ukuran populasi, *crossover rate* dan *mutation rate* jumlah generasi, dan perbandingan segmen mutasi. Sedangkan parameter kasus merupakan data *dummy* yang dibuat peneliti sendiri, yaitu jumlah pabrik, distributor, dan agen, kapasitas pabrik, kebutuhan distributor dan agen, serta biaya distribusi dari pabrik ke distributor, maupun distributor ke agen.
5. Menggunakan model distribusi seimbang, dimana setiap sumber diasumsikan selalu memiliki persediaan dan setiap tujuan diasumsikan memiliki permintaan sesuai dengan jumlah persediaan dan tidak nol.
6. *Crossover* menggunakan *one cut point crossover*, mutasi menggunakan *exchange mutation* dan seleksi menggunakan *elitism selection*.
7. Pengujian berfokus pada pengaruh parameter pada algoritma genetika dengan melakukan pengujian terhadap parameter ukuran populasi, kombinasi *crossover* dan *mutation rate*, ukuran populasi jumlah generasi dan segmen mutasi.
8. *Output* yang dihasilkan oleh sistem adalah jalur distribusi mana saja yang akan dilalui yang menghasilkan *fitness* terbesar dan biaya minimum.

#### 1.4 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah :

1. Untuk menerapkan Algoritma Genetika dalam menyelesaikan masalah distribusi barang dua tahap.
2. Untuk mengetahui representasi kromosom untuk kasus distribusi barang dua tahap.
3. Untuk mengetahui pengaruh parameter algoritma genetika (ukuran populasi, *crossover rate*, *mutation rate*, dan jumlah generasi) terhadap terhadap nilai *fitness* dari hasil optimasi distribusi barang dua tahap.

#### 1.5 Manfaat

Manfaat yang dapat diambil dari penulisan skripsi ini adalah menyediakan suatu sistem yang dapat membantu dalam pemilihan jalur distribusi yang optimal dan dapat menunjukkan jalur dengan biaya distribusi minimum.

## **1.6 Sistematika Penulisan**

Adapun sistematika penulisan yang digunakan dalam menyusun tugas akhir ini adalah :

### **BAB I PENDAHULUAN**

Bab ini menguraikan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

### **BAB II KAJIAN PUSTAKA DAN DASAR TEORI**

Bab ini menguraikan beberapa teori mengenai distribusi dua tahap, teori mengenai algoritma genetika dan teori-teori yang berhubungan dengan optimasi distribusi dengan pola transportasi dua tahap.

### **BAB III METODOLOGI PENELITIAN**

Pada bab ini akan dijelaskan mengenai langkah-langkah atau alur dalam pencarian jalur distribusi dua tahap yang optimal dengan menggunakan algoritma genetika.

### **BAB IV PERANCANGAN**

Bab ini dijelaskan mengenai perancangan sistem untuk implementasi, merancang database dan perancangan pengujian.

### **BAB V IMPLEMENTASI**

Bab ini dijelaskan mengenai implementasi sistem yang telah dijabarkan sebelumnya mengenai implementasi database dan implementasi perangkat lunak

### **BAB VI PEGUJIAN DAN ANALISIS**

Pada bab ini membahas mengenai hasil uji coba dari sistem yang telah diimplementasikan serta menganalisis hasil uji coba tersebut.

### **BAB VII PENUTUP**

Bab ini menguraikan kesimpulan dan saran dari penelitian untuk pengembangan penelitian lebih lanjut.

## BAB II

### KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab ini dijelaskan mengenai dasar teori yang berkaitan dengan penerapan algoritma genetika untuk permasalahan optimasi distribusi dua tahap. Bab tinjauan pustaka akan menjelaskan mengenai kajian pustaka, distribusi, distribusi dua tahap, pemodelan matematis transportasi pada distribusi dua tahap, algoritma genetika, penerapan algoritma genetika dan parameter genetika.

#### 2.1 Kajian Pustaka

Subbab kajian pustaka menjelaskan mengenai penelitian sebelumnya yang berkaitan dengan penelitian penulis. Penelitian sebelumnya yang dibahas pada subbab kajian pustaka digunakan sebagai pembanding antara penelitian yang dilakukan oleh penulis. Penelitian sebelumnya yang berkaitan dengan penelitian penulis berjudul “*Penentuan Rute Optimum Jalur Distribusi dalam Supply Chain Network Menggunakan Algoritme Genetika*”. Penelitian ini membahas tentang penggunaan algoritma genetika untuk mencari jalur optimal distribusi barang. Pada penelitian ini digunakan metode *crossover* berupa *weight mapping crossover* (WMX) dan metode mutasi berupa *swap mutation*, seleksi menggunakan metode *roulette wheel*. Algoritma genetika sudah dapat berjalan baik untuk menentukan rute optimum distribusi dengan menampilkan rute jalan yang dilalui (Toni, 2009).

Selain itu penulis juga melakukan kajian pustaka lainnya terhadap penelitian yang berjudul “*Penerapan Algoritma Genetik Dua Populasi pada Kasus Transportasi Dua Tahap*”. Penelitian ini membahas tentang penggunaan algoritma genetika untuk permasalahan distribusi dua tahap yang membagi populasinya menjadi populasi elit dan umum. Pada penelitian ini digunakan metode *crossover* berupa *weight mapping crossover* (WMX) dan metode mutasi berupa *swap mutation*. Menggunakan seleksi berupa metode *roulette wheel* (Prabowo, 2011). Algoritma genetika diterapkan dengan membagi populasi menjadi populasi elit dan umum, namun dengan membagi menjadi dua populasi mengakibatkan solusi yang dihasilkan kurang optimal karena pada populasi elit individu yang melakukan

reproduksi tidak selalu menghasilkan keturunan yang baik atau kecenderungan memiliki nilai *fitness* seperti induknya.

Dari 2 kajian pustaka yang dilakukan, penulis mendapatkan kesimpulan bahwa algoritma genetika mampu digunakan untuk menyelesaikan permasalahan optimasi distribusi barang. Oleh karena itu penulis menggunakan algoritma genetika untuk menyelesaikan permasalahan optimasi distribusi barang dua tahap. Sedangkan perbedaannya terletak pada representasi kromosom, teknik *crossover*, mutasi dan seleksi. Pada penelitian ini akan digunakan representasi permutasi dengan perkalian antar subjek distribusi, metode *crossover* berupa *one cut point crossover* yang merupakan metode *crossover* yang mampu menjaga konsistensi urutan nilai pada kromosom (Ningrum, 2010). Metode mutasi berupa *exchange mutation* dinilai lebih baik dari metode mutasi lainnya karena *exchange mutation* memberikan rata-rata nilai tertinggi dengan membentuk individu yang beraneka ragam tanpa melupakan struktur individu asal (Harun, 2014). Seleksi menggunakan *elitism* digunakan untuk memperoleh individu dengan nilai *fitness* tertinggi sejumlah ukuran populasi.

## 2.2 Distribusi

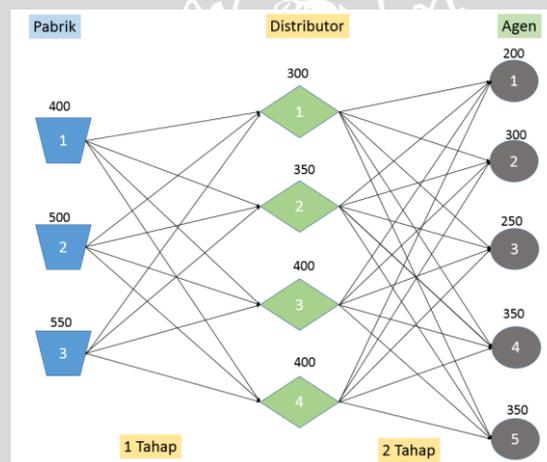
Secara umum masalah distribusi dianggap sebagai serangkaian proses untuk memindahkan barang dari produsen kepada konsumen. Proses distribusi banyak dilakukan oleh perusahaan-perusahaan yang memproduksi barang banyak, kegiatan distribusi merupakan suatu proses penting untuk kelancaran pemasaran produk. Distribusi adalah bagaimana memindahkan dan menyimpan barang dari sumber (*source*) untuk sampai ke tujuan (*destination*) dengan tujuan meminimalkan transportasi dan biaya pengiriman (Nirwansyah & Widowati, 2007).

## 2.3 Distribusi Dua Tahap

Distribusi dua tahap merupakan proses memindahkan barang secara dua tahap, barang dari sumber (*plants*) produksi barang yang dipindahkan kepada distributor (*distribution center*) yang kemudian dari distributor akan dipindahkan kepada agen (*customer*) (Natalisa, 2007). Pada distribusi dua tahap berkaitan erat dengan persoalan transportasi, dimana yang menjalankan proses distribusi adalah transportasi. Permasalahan distribusi dua tahap berfokus untuk membuat suatu jalur

transportasi barang dari beberapa sumber menuju beberapa tujuan dengan membawa jumlah dan jenis barang yang tepat pada waktu yang telah ditentukan dengan biaya yang minimum (Ballou dalam Sari, 2008). Ciri khusus pada model distribusi dua tahap adalah terdapat sumber dan sejumlah tujuan, kuantitas komoditas produk yang dipindahkan dari sumber sesuai dengan permintaan tujuan dan produk yang diminta harus sesuai dengan jumlah barang yang diproduksi oleh sumber.

Pada distribusi barang dua tahap digambarkan pada suatu model transportasi dengan sumber ( $I$ ), distributor ( $J$ ) dan Agen ( $K$ ) untuk memperjelas proses yang ada. Untuk lebih jelasnya pemodelan distribusi dua tahap dapat dilihat pada Gambar 2.1 .



**Gambar 2.1** Model Distribusi Dua Tahap

Sumber : (Gen, Altiparmak, & Lin dalam Prabowo, 2010)

## 2.4 Penyelesaian dan Pemodelan Matematis Transportasi pada Distribusi Dua Tahap

Inti dari distribusi dua tahap adalah proses memindahkan barang dari asal ke beberapa tujuan. Proses pemindahan tersebut dilakukan dengan transportasi. Persoalan transportasi merupakan suatu kasus yang dapat dimodelkan sebagai pemrograman linier (*linier programming*). Untuk masalah penyelesaian transportasi pada distribusi dua tahap harus memperhatikan masalah tiga elemen yang berupa variabel, fungsi obyektif dan *constraint* (batasan) (Gen, Altiparmak, & Lin dalam Prabowo, 2010).

#### 2.4.1 Variabel

Pada kasus transportasi untuk distribusi dua tahap, masalah variabel pada pemrograman linier menunjukkan jumlah produk atau kuantitas (Kumar, 2004). Variabel digunakan untuk menunjukkan jumlah produk yang dikirimkan dari sumber kepada tujuan dan biaya yang dibutuhkan dari sumber ke tujuan. Untuk penyelesaian masalah transportasi, variabel-variabel yang digunakan adalah :

- a.  $I$  : Jumlah sumber ( $i= 1,2,...i$ ).
- b.  $x_{ij}$  : Jumlah barang yang dikirimkan dari sumber (pabrik) menuju distributor.
- c.  $a_i$  : Unit persediaan (kapasitas produksi) sumber untuk dipindahkan ke distributor.
- d.  $J$  : Jumlah distributor ( $j= 1,2,...j$ ).
- e.  $t_{ij}$  : Biaya perjalanan dari sumber (pabrik) menuju distributor.
- f.  $y_{jk}$  : Jumlah barang yang dikirimkan dari distributor menuju agen.
- g.  $b_j$  : Jumlah kapasitas jual ditributor.
- h.  $c_{jk}$  : Biaya perjalanan dari distributor menuju agen.
- i.  $K$  : Jumlah agen ( $k= 1,2, ...,k$ ).
- j.  $d_k$  : Jumlah permintaan dari agen  $K$ .

#### 2.4.2 Fungsi Obyektif

Fungsi obyektif merupakan persamaan yang menggambarkan tujuan dari masalah yang akan dipecahkan (Kumar, 2004). Masalah yang akan dipecahkan dalam masalah distribusi dua tahap adalah bagaimana meminimalkan biaya transportasi. Untuk meminimalkan biaya transportasi diberikan fungsi obyektif hubungan antara biaya dan variabelnya. Fungsi obyektif untuk permasalahan transportasi pada distribusi dua tahap adalah dengan meminimalkan  $Z$  yang dinyatakan secara matematis pada Persamaan 2.1 (Gen, Altiparmak, & Lin, 2006):

$$Z = \sum_{i=1}^I \sum_{j=1}^J t_{ij} x_{ij} + \sum_{j=1}^J \sum_{k=1}^K c_{jk} y_{jk} \quad 2-1$$

#### 2.4.3 Constraint (Batasan)

Kondisi dimana banyaknya suplai dan permintaan terpenuhi merupakan *constrain* (Kumar, 2004). Masalah transportasi pada distribusi dua tahap, *constrain*

untuk memenuhi biaya minimum dinyatakan pada Persamaan 2.2 sampai 2.6 (Gen, Altiparmak, & Lin, 2006). Yang pertama distribusi barang tidak boleh melebihi kapasitas yang dinyatakan dalam Persamaan 2.2 dan 2.3.

$$\sum_{j=1}^j x_{ij} \leq a_i \quad \forall i \quad 2-2$$

$$\sum_{k=1}^k y_{jk} \leq b_j \quad \forall j \quad 2-3$$

Pada distribusi harus dipastikan bahwa semua permintaan agen terpenuhi, dengan maksud barang yang dikirimkan dari distributor menuju agen tidak boleh kurang dari permintaan agen, dinyatakan pada Persamaan 2.4

$$\sum_{j=1}^j y_{jk} \geq d_k \quad \forall k \quad 2-4$$

Batasan yang harus dipenuhi berupa sebuah penawaran total sama dengan permintaan total yang biasa disebut transportasi berimbang. Tidak boleh terdapat perhitungan negatif, dinyatakan dalam Persamaan 2.5 dan 2.6

$$\sum_{i=1}^i \sum_{j=1}^j x_{ij} = \sum_{j=1}^j \sum_{k=1}^k y_{jk} \quad 2-5$$

$$x_{ij}, y_{jk} \geq 0, \quad \forall i, j, k \quad 2-6$$

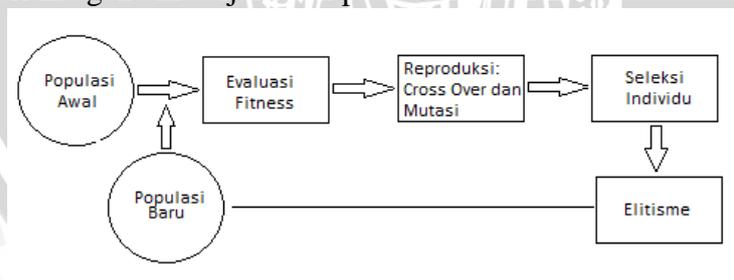
## 2.5 Algoritma Genetika

Algoritma genetika merupakan cabang dari algoritma evolusi dan algoritma yang paling populer. Dalam penerapannya algoritma genetika mampu menyelesaikan berbagai masalah kompleks dalam segala bidang permasalahan, algoritma genetika terinspirasi dari prinsip genetika dan seleksi alam yang diperkenalkan oleh John Holland dari Universitas Michigan pada tahun 1975. Konsep dasar dari algoritma genetika dirancang untuk menyimulasikan proses di alam yang digunakan untuk evolusi, dimana individu akan mengalami perubahan gen secara terus menerus untuk menyesuaikan dengan lingkungan hidupnya. Dasar dari algoritma genetika adalah individu dalam populasi bersaing untuk sumber daya dan pasangannya dengan melakukan reproduksi untuk memperoleh individu baru yang mempunyai gen lebih baik dari orang tuanya yang kemudian akan melalui proses genetik yaitu seleksi alam individu yang lebih kuat akan bertahan untuk mendapatkan generasi yang lebih baik (Sutojo, Mulyanto, & Suhartanto, 2011).

### 2.5.1 Siklus Algoritma Genetika

Siklus Algoritma genetika pertama kali diperkenalkan oleh David Goldberg, bermula dengan membuat populasi awal secara acak dengan ditentukan jumlah kromosomnya. Dari populasi yang ada ditentukan evaluasi dengan menghitung nilai *fitness*-nya yang kemudian dilakukan seleksi individu untuk memilih individu terbaik. Siklus selanjutnya adalah melakukan reproduksi dengan *crossover* dan mutasi untuk menciptakan individu baru yang kemudian akan dilakukan evaluasi terhadap *fitness*, proses ini akan terjadi secara berulang-ulang sampai generasi yang ditentukan (Sutojo, Mulyanto, & Suhartanto, 2011).

Sedikit berbeda dengan siklus yang diperkenalkan David Goldberg, Zbigniew Michalewicz memperbaikinya dengan menambahkan proses elitisme dan membalik proses reproduksi dahulu untuk kemudian dilakukan proses seleksi. Secara umum algoritma genetika dimulai dengan membangun populasi awal secara acak dan kemudian dilakukan evaluasi pada setiap kromosom dengan menggunakan fungsi *fitness* yang menunjukkan kualitas kromosom dalam populasi. Setelah melakukan evaluasi, muncul individu berikutnya yang didapatkan dari proses reproduksi dengan cara *crossover* dan mutasi. Dalam *crossover* dibutuhkan dua individu sebagai *parent*, pada mutasi hanya membutuhkan satu *parent* melalui modifikasi kromosom. Setelah reproduksi dilakukan seleksi untuk memperoleh populasi baru dan kemudian akan dilakukan evaluasi *fitness* (Michalewicz, 1996). Siklus algoritma genetika dijelaskan pada Gambar 2.2.



**Gambar 2.2** Siklus algoritma genetika Zbigniew Michalewicz  
**Sumber :** (Sutojo, 2011)

### 2.6 Penerapan Algoritma Genetika

Penerapan algoritma genetika dimulai dengan membangkitkan populasi awal dengan merepresentasikan permasalahan kedalam kromosom kemudian

mengevaluasi dengan nilai *fitness*. Dari ukuran populasi yang ditentukan, individu-individu melakukan reproduksi dengan *crossover* dan mutasi yang akan menghasilkan individu baru (*offspring*). Kemudian dilakukan seleksi untuk pemilihan terhadap individu dengan acuan nilai *fitness* dari masing-masing individu sampai dengan generasi yang ditentukan (Mahmudy, 2013).

### 2.6.1 Representasi Kromosom

Representasi kromosom merupakan suatu proses untuk menyelesaikan masalah, dimana suatu permasalahan dapat dikodekan kedalam kromosom (Gen & Cheng, 2000). Pada setiap permasalahan memiliki representasi kromosom yang berbeda-beda, karena tidak semua model representasi cocok untuk setiap permasalahan. Cara representasi permasalahan ke dalam suatu kromosom merupakan hal yang penting dalam algoritma genetika. Algoritma genetika memiliki beberapa jenis representasi kromosom untuk permasalahan yang berbeda, seperti representasi biner, integer, real dan permutasi.

Representasi permutasi telah sukses diterapkan pada berbagai masalah kombinatorial seperti Travelling Salesperson Problem, perencanaan dan penjadwalan produksi industri manufaktur (Mahmudy, 2013). Bentuk masalah semacam ini tidak dapat diselesaikan dengan menggunakan biner, integer maupun real karena dalam solusi permasalahan ini yang dicari adalah bagaimana menemukan kunjungan dan perencanaan yang total nilainya paling optimal. Representasi permutasi bisa digunakan untuk menyatakan sebuah solusi, dimana satu kromosom menyatakan satu solusi, nilai dan posisi. Setiap gen pada kromosom berupa angka integer yang menyatakan nomer dari setiap simpul untuk menyatakan kunjungan lokasi. Contoh penggunaan representasi permutasi dijelaskan pada Gambar 2.3.

<b>Node</b>	1	2	3	4	5	6	7
<b>Kromosom</b>	2	5	7	4	1	3	6

**Gambar 2.3** Bentuk kromosom pada representasi permutasi  
**Sumber :** (Mahmudy, 2013)

Permasalahan transportasi dua tahap dapat digunakan representasi permutasi dengan menyatakan prioritas alokasi setiap sel pada matriks. Intinya pada representasi kromosom dengan permutasi adalah semua operasi dikodekan dengan nilai atau *permutation encoding*, setiap individu berisi satu kromosom yang berisi gen – gen yang merepresentasikan urutan prioritas distribusi.

### 2.6.2 Nilai *Fitness*

Nilai *fitness* menyatakan nilai dan tujuan, algoritma genetika digunakan untuk memaksimalkan nilai *fitness*. Dalam transportasi nilai *fitness* berupa inversi dari biaya pelaksanaan kromosom. Nilai *fitness* membedakan kualitas kromosom satu dengan yang lain dan untuk membedakan kromosom yang dihasilkan. Untuk permasalahan maksimasi, pencarian nilai *fitness* menggunakan Persamaan 2.7 (Mahmudy, 2013):

$$Fitness = f(x) \quad 2-7$$

Fungsi *fitness* untuk persamaan 2.7 digunakan pada fungsi maksimasi dengan nilai  $f(x)$  berbanding lurus dengan *fitness*. Persamaan 2.7 tidak dapat digunakan untuk masalah minimasi karena pada masalah minimasi nilai  $f(x)$  dan *fitness* adalah berbanding terbalik. Persamaan yang dapat digunakan adalah 2.8 dan persamaan 2.9.

$$Fitness = C - f(x) \quad 2-8$$

$$Fitness = \frac{1}{f(x)} \quad 2-9$$

Dengan C merupakan nilai konstan yang harus ditetapkan sebelumnya. Akan tetapi dalam penggunaannya harus berhati-hati untuk menghindari pembagian dengan angka 0 (Mahmudy, 2013). Fungsi *fitness* pada distribusi barang dua tahap menggunakan persamaan 2.9,  $F(x)$  merupakan total biaya distribusi sesuai dengan fungsi obyektif persamaan 2.1.

### 2.6.3 Membangkitkan Populasi Awal

Pembangkitan populasi awal pada algoritma genetika dilakukan secara random sesuai jumlah individu dalam populasi yang ditentukan. Populasi awal

merupakan himpunan solusi yang terbentuk dari sejumlah string kromosom dan ditempatkan pada populasi (Mahmudy, 2013).

Tahap membangkitkan populasi awal adalah menentukan jumlah ukuran individu dalam suatu populasi atau yang biasa disebut *popsiz*e. Ukuran populasi menyatakan kemampuan eksplorasi dalam satu proses algoritma genetika. Ukuran populasi yang terlalu sedikit menyebabkan area eksplorasi kecil, sehingga algoritma genetika bekerja kurang maksimal dan konvergen. Ukuran populasi yang terlalu banyak mampu meningkatkan kemampuan eksplorasi, tetapi hal ini menyebabkan waktu komputasi yang lama (Pitaloka, Mahmudy, & Sutrisno, 2012)

#### 2.6.4 Fase Reproduksi

Fase reproduksi merupakan suatu fase dalam algoritma genetika untuk membentuk suatu individu baru. Individu baru didapat dari proses *crossover* (pindah silang) dan reproduksi menggunakan mutasi.

##### 2.6.4.1 *Crossover (Pindah Silang)*

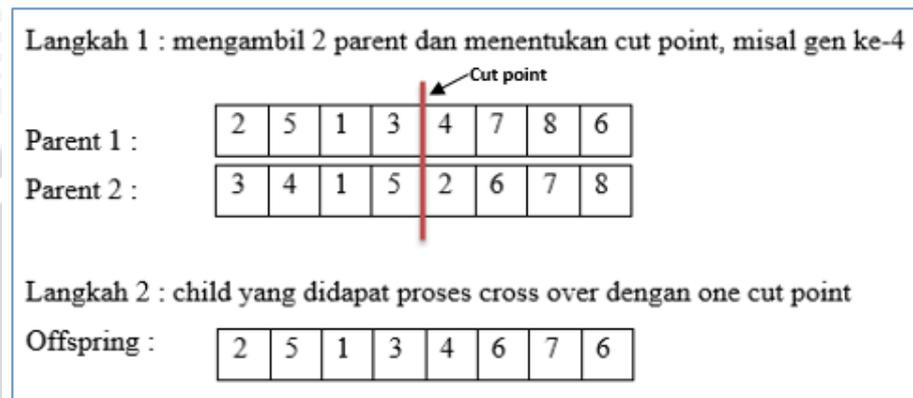
Sebuah individu baru yang mengarah pada solusi optimal bisa diperoleh melalui proses pindah silang. Proses reproduksi dengan *crossover* diperlukan dua *parent* terpilih untuk menghasilkan keturunan. Secara umum proses *crossover* adalah memilih dua buah kromosom sebagai *parent* yang dipilih secara random, sehingga masing-masing kromosom *parent* terpisah menjadi dua segmen. Setelah memilih secara acak, dilakukan penukaran segmen kromosom induk untuk menghasilkan *offspring* atau individu baru.

Pada algoritma genetika terdapat metode *crossover* seperti *one cut point crossover* yang digunakan pada representasi biner dan permutasi, dan *extended intermediete crossover* yang digunakan pada representasi *realcode*. Metode lain yang digunakan dalam *crossover* pada representasi permutasi adalah *position-based crossover*, *order-based crossover*, *partial mapped crossover (PMX)*, *order crossover (OX)*, *cycle crossover* dan *heuristic crossover* (Gen & Cheng dalam Mahmudy, 2013).

Pada representasi permutasi untuk masalah transportasi *crossover* dapat digunakan metode *one cut point crossover*. *One cut point crossover* merupakan

suatu metode *crossover* yang memilih dua individu terpilih untuk dijadikan sebagai *parent* kemudian menukar kromosom pada area yang terpilih.

Misalkan terdapat dua *parent* terpilih yang memiliki panjang kromosom delapan satuan. Dalam kromosom ditentukan titik potong atau *cut point* secara random dengan *cut point* pada posisi gen ke-4. *Offspring* yang dihasilkan dari *one cut point* dijelaskan pada Gambar 2.4.



**Gambar 2.4** *Crossover* pada representasi permutasi

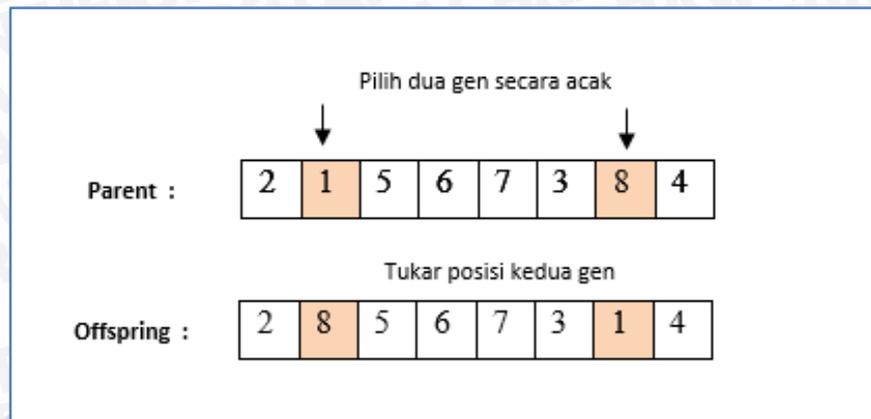
**Sumber :** (Mahmudy, 2013)

*Child* dihasilkan dengan *crossover* dari dua *parent*, segment kiri dari kromosom *child* didapatkan dari *parent* 1 dan segment kanan didapatkan dari urutan gen tersisa dari *parent* 2.

#### 2.6.4.2 Mutasi

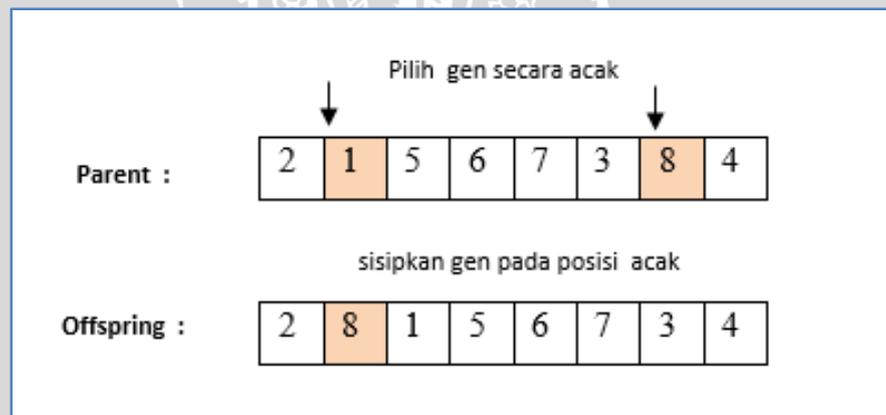
Mutasi merupakan fase reproduksi yang dilakukan untuk mendapatkan individu baru. Proses mutasi merupakan proses suatu operator genetika untuk menghasilkan perubahan acak pada satu kromosom. Metode mutasi yang biasa digunakan dalam representasi permutasi adalah *reciprocal exchange mutation* dan *insertion mutation*.

Metode mutasi dengan *reciprocal exchange mutation* bekerja dengan memilih dua posisi (*exchage point* /XP) secara random kemudian menukarkan nilai pada posisi tempatnya sehingga menghasilkan *offspring* baru. Metode *reciprocal exchange mutation* dijelaskan pada Gambar 2.5.



**Gambar 2.5** Proses mutasi dengan *reciprocal exchange mutation*  
**Sumber :** (Prabowo, 2011)

Metode *insertion mutation* merupakan metode mutasi lain dengan memilih secara acak sebuah gen yang kemudian disisipkan pada posisi yang dipilih secara random. Proses mutasi dengan *insertion point* dijelaskan pada Gambar 2.6.



**Gambar 2.6** Proses mutasi dengan *insertion mutation*  
**Sumber :** (Prabowo, 2011)

Penelitian sebelumnya yang membahas mengenai transportasi dua tahap digunakan metode *reciprocal exchange mutation* yang memberikan hasil yang lebih baik dibandingkan dengan menggunakan *insertion mutation* (Prabowo, 2010).

### 2.6.5 Seleksi

Seleksi merupakan fase untuk memilih individu yang akan dijadikan sebagai *parent* yang kemudian dilakukan proses reproduksi kembali. Seleksi dilakukan untuk memilih individu dari himpunan populasi dan *offspring* yang

dipilih untuk melakukan proses genetik pada generasi berikutnya (Mahmudy, 2013). Proses seleksi dilakukan agar generasi berikutnya lebih baik dari generasi sebelumnya, dengan memilih individu dengan nilai *fitness* yang tinggi. Ada berbagai macam metode seleksi pada algoritma genetika seperti *roulette wheel*, *binary tournament*, *elitism*, dan *replacement selection*.

Seleksi dengan menggunakan metode *roulette wheel* digunakan untuk memilih populasi yang dilakukan dengan menghitung nilai probabilitas seleksi tiap individu berdasarkan nilai *fitness*-nya. Metode seleksi dengan *binary tournament* dilakukan dengan mengambil secara acak sejumlah kecil individu dari penampungan populasi dan *offspring*, individu dengan nilai *fitness* lebih besar akan terpilih untuk masuk populasi berikutnya. Namun kelemahan dari metode ini adalah mudah terjadi konvergensi dini. *Elitism selection* bekerja dengan mengumpulkan semua individu dalam populasi, metode seleksi ini menjamin semua individu yang memiliki *fitness* terbaiklah yang akan selalu lolos. Kelemahan dari metode ini adalah tidak memberikan kesempatan kepada individu yang memiliki nilai *fitness* rendah. *Replacement selection* merupakan metode seleksi dengan aturan *offspring* yang diproduksi melalui proses *crossover* akan menggantikan induk yang lemah jika mempunyai nilai *fitness* yang lebih baik daripada induk yang terlemah (Mahmudy, Marian & Luong 2013a, 2013d).

#### **2.6.7.1 Seleksi Elitism**

Seleksi elitism merupakan suatu prosedur seleksi untuk memilih individu yang memiliki nilai *fitness* tertinggi untuk dijadikan generasi berikutnya. Metode seleksi ini bekerja dengan mengumpulkan semua *offspring* dan *parent* dalam satu penampungan, popsize dengan *fitness* individu terbaik akan lolos menjadi generasi selanjutnya. Seleksi elitism dijelaskan dalam pseudocode pada Gambar 2.7

```

PROCEDURE ElitismSelection
Input:
  POP: himpunan individu pada populasi
  pop_size: ukuran populasi
  OS: himpunan individu anak (offspring) hasil
      reproduksi menggunakan crossover dan mutasi
Output :
  POP: himpunan individu pada populasi setelah proses
  seleksi selesai
  /* gabungkan individu pada POP dan OS ke dalam TEMP */
  TEMP <- Merge (POP,OS)
  /* urutkan individu berdasarkan fitness secara ascending */
  OrderAscending (Temp)
  /* copy pop_size individu terbaik ke POP */
  POP <- CopyBest (Temp, pop_size)
END PROCEDURE

```

**Gambar 2.7** Pseudocode seleksi dengan elitism  
**Sumber :** (Mahmudy, 2013)

## 2.7 Parameter Genetika

Parameter genetika digunakan untuk menentukan kesuksesan suatu proses optimasi. Parameter yang digunakan pada algoritma genetika adalah sebagai berikut:

### 1. Ukuran Populasi (*pop\_size*)

Jumlah populasi dalam algoritma genetika menentukan keefektifan kinerja algoritma dalam mengatasi suatu permasalahan. Jumlah populasi yang terlalu sedikit kurang mampu menyediakan ruang solusi yang dapat merepresentasikan seluruh ruang persoalan. Sedangkan jumlah populasi yang terlalu banyak dapat memperlama waktu komputasi karena data yang diproses banyak dan lam dalam proses pencariannya (Prabowo, 2010). Jumlah populasi akan mempengaruhi proses persilangan dan mutasi, semakin besar jumlah populasi maka persilangan dan mutasi juga akan semakin banyak yang diproses.

### 2. *Crossover Rate* (*cr*)

*Crossover rate* menunjukkan ukuran terjadinya persilangan antara dua kromosom. *Crossover rate* merupakan banyaknya jumlah *offspring* yang dihasilkan dalam suatu populasi terhadap jumlah kromosom dalam satu populasi. *Crossover* bekerja dengan menyilangkan dua *parent* dan menghasilkan sebuah *offspring*, semakin besar *crossover rate* memungkinkan pencapaian alternatif solusi yang

bervariasi. *Crossover rate* yang terlalu besar juga berdampak pada waktu komputasi, karena semakin banyak *crossover rate* maka kemungkinan *parent* yang disilangkan juga semakin banyak dan memungkinkan kurang menjanjikan solusi yang kurang optimal (Ningrum, 2010).

### 3. *Mutation Rate* (mr)

Mutasi merupakan salah satu proses reproduksi dalam algoritma genetika. Mutasi digunakan untuk meningkatkan variasi dalam populasi, dibutuhkan satu kromosom untuk dapat melakukan mutasi. *Mutation rate* merupakan suatu parameter yang digunakan dalam memungkinkan berapa jumlah kromosom dalam populasi untuk dilakukan proses mutasi. *Mutation rate* yang kecil akan berpengaruh pada kromosom yang berpotensi untuk tidak dicoba (Ningrum, 2010).



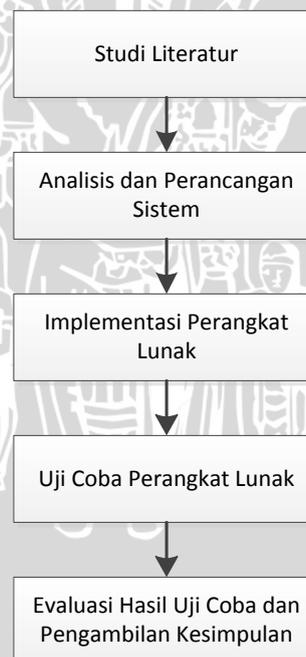
## BAB III

### METODOLOGI PENELITIAN

Pada bab metodologi penelitian dijelaskan mengenai tahapan penelitian, kebutuhan sistem, formulasi masalah dan siklus penyelesaian masalah optimasi distribusi barang dua tahap dengan algoritma genetika.

#### 3.1 Tahapan Penelitian

Tahapan penelitian pada kasus optimasi distribusi barang dua tahap menggunakan algoritma genetika menjelaskan langkah-langkah sistematis untuk menerapkan metode algoritma genetika ke dalam suatu perangkat lunak. Tahap-tahap penelitian untuk membangun sistem dengan menggunakan algoritma genetika untuk kasus optimasi distribusi barang dua tahap dijelaskan pada Gambar 3.1.



**Gambar 3.1** Diagram alir metode penelitian

Berdasarkan Gambar 3.1, penjelasan tahapan penelitian mengenai optimasi distribusi barang dua tahap menggunakan algoritma genetika adalah :

1. Mengumpulkan berbagai studi literatur mengenai metode algoritma genetika, jurnal dan literatur lain yang terkait dengan optimasi distribusi barang dua tahap.
2. Menganalisa dan merancang perangkat lunak untuk optimasi distribusi barang dua tahap dengan menggunakan algoritma genetika.
3. Mengimplementasikan hasil analisis dan perancangan yang telah dilakukan dalam bentuk perangkat lunak untuk optimasi distribusi barang dua tahap.
4. Melakukan uji coba terhadap perangkat lunak untuk distribusi barang dua tahap dengan menggunakan algoritma genetika.
5. Mengevaluasi hasil dari sistem yang dibangun untuk distribusi barang dua tahap berdasarkan kondisi nyata.

### 3.1.1 Studi Literatur

Studi literatur merupakan metode yang digunakan untuk mendapatkan dasar teori sebagai sumber untuk penulisan penelitian dan pengembangan aplikasinya. Studi literatur yang berkaitan dengan penulisan penelitian ini adalah :

1. Algoritma Evolusi
2. Algoritma Genetika
3. Distribusi barang dua tahap

Studi literatur merupakan dasar teori yang digunakan sebagai penunjang dan pendukung penulisan pada penelitian. Sumber atau referensi yang digunakan antara lain jurnal, laporan penelitian, buku dan bantuan mesin pencari (*search engine*) dengan bantuan koneksi internet.

### 3.1.2 Analisis Kebutuhan dan Perancangan

Analisa kebutuhan merupakan metode untuk menganalisis kebutuhan yang akan dibuat meliputi spesifikasi media yang akan digunakan. Analisa kebutuhan dari pengguna dapat dikelompokkan menjadi kebutuhan fungsional dan non fungsional yang akan digunakan dalam perancangan sistem. Analisa kebutuhan sistem berupa :

1. Sistem mampu meng-*input*-kan data parameter berupa jumlah pabrik, distributor dan agen, biaya distribusi, jumlah populasi, *cr*, *mr* dan jumlah generasi.
2. Sistem mampu menampilkan *input* parameter dari pengguna.
3. Sistem mampu menampilkan perhitungan hasil penerapan algoritma genetika untuk masalah distribusi barang dua tahap berupa, inisialisasi, reproduksi, seleksi dan jumlah generasi yang di *input*-kan user.
4. Sistem mampu menampilkan jalur distribusi optimal dari kromosom terpilih.

Perancangan sistem merupakan tahapan merancang suatu perangkat lunak sesuai dengan penerapan algoritma genetika untuk optimasi distribusi barang dua tahap. Teori dari pustaka akan digunakan untuk pengimplementasian pengembangan optimasi distribusi barang dua tahap. Perangkat lunak memiliki bagian sistem yang memproses data *input* pengguna untuk menghasilkan *output* berupa jalur distribusi yang optimal dengan biaya yang mendekati minimum.

### 3.1.3 Implementasi Perangkat Lunak

Implementasi merupakan tahapan yang dilakukan untuk mengaplikasikan perancangan sistem. Implementasi dilakukan dengan menggunakan bahasa pemrograman PHP dengan database MySQL. Pada implementasi perangkat lunak menampilkan *input* berupa kapasitas persediaan, permintaan dan matriks biaya. Setelah *input* diproses akan menghasilkan *output* berupa jalur distribusi yang optimal beserta biaya minimum dari distribusi barang dua tahap.

### 3.1.4 Uji Coba Perangkat Lunak

Uji coba dilakukan untuk mengetahui kinerja sistem sesuai dengan kebutuhan. Pengujian sistem yang digunakan pada optimasi distribusi barang dua tahap dengan menerapkan algoritma genetika dilakukan dengan menguji ukuran populasi, pengujian kombinasi *crossover rate* dan *mutation rate*, pengujian jumlah generasi dan pengujian segmen mutasi.

### 3.1.5 Evaluasi Hasil Uji Coba dan Pengambilan Kesimpulan

Evaluasi hasil uji coba dilakukan untuk mengetahui kinerja sistem berdasarkan hasil uji coba perangkat lunak. Evaluasi dilakukan untuk mengetahui pengaruh parameter-parameter yang digunakan pada optimasi distribusi barang dua tahap dengan menggunakan algoritma genetika.

Pengambilan kesimpulan dilakukan setelah semua tahapan analisa perancangan sistem, implementasi, pengujian dan evaluasi selesai dilakukan. Kesimpulan menjawab mengenai subbab perumusan masalah. Saran ditujukan untuk memperbaiki kesalahan yang terjadi pada penelitian ini.

## 3.2 Kebutuhan Sistem

Pada subbab kebutuhan sistem akan dibahas mengenai spesifikasi kebutuhan yang digunakan untuk membangun sistem dengan menerapkan algoritma genetika pada studi kasus optimasi distribusi barang dua tahap. Kebutuhan sistem yang akan dipaparkan meliputi kebutuhan perangkat keras dan kebutuhan perangkat lunak.

### 3.2.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang digunakan untuk mengembangkan sistem dengan menerapkan algoritma genetika untuk kasus optimasi distribusi barang dua tahap adalah sebagai berikut :

1. Prosesor AMD A8.
2. RAM 4 GB.
3. *Hardisk* dengan kapasitas 500 GB.
4. Monitor 14' Wide Screen LED Backlit Display.
5. Keyboard standart qwerty.

### 3.2.2 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan untuk membangun sistem dengan menerapkan algoritma genetika untuk kasus optimasi distribusi barang dua tahap adalah sebagai berikut :

1. Sistem operasi Microsoft Windows 8.1 Pro 64-bit.

2. Bahasa pemrograman PHP
3. *Integrated Development Environment* Sublime Text 2.0
4. *Database Management System* MySQL XAMPP 1.8.3

### 3.3 Formulasi Masalah

Pada subbab formulasi masalah berisi penjelasan mengenai deskripsi permasalahan yang akan diselesaikan dengan algoritma genetika dan data yang akan digunakan.

#### 3.3.1 Deskripsi Umum

Algoritma genetika diterapkan pada permasalahan distribusi barang dua tahap digunakan untuk mendapatkan biaya distribusi minimum. Permasalahan utama dalam penelitian ini adalah bagaimana menentukan jalur distribusi barang yang optimal untuk mendistribusikan barang dari pabrik ke agen dengan perantara distributor. Untuk meminimalkan biaya distribusi, pabrik harus mengetahui kemana dan berapa jumlah barang yang akan didistribusikan.

Pada distribusi barang dua tahap, biaya distribusi diperoleh dari biaya tahap 1 ditambah dengan biaya tahap 2 yang telah dijelaskan pada persamaan 2.1 pada bab 2. Biaya tahap 1 diperoleh dari biaya perjalanan dari pabrik ke distributor dikalikan jumlah barang yang dikirimkan. Biaya distribusi tahap 2, total biaya distribusi diperoleh dari biaya perjalanan dari distributor ke agen dikalikan dengan jumlah barang yang dikirimkan. Semakin kecil biaya distribusi, maka jalur distribusi yang digunakan semakin optimal

Sistem dengan menerapkan algoritma genetika untuk permasalahan optimasi distribusi barang dua tahap digunakan *input* data dari proses distribusi berupa data pabrik dengan kapasitas persediaan, distributor dan agen dengan kapasitas permintaan serta biaya distribusi untuk masing-masing tahap. Untuk optimasi rute dengan algoritma genetika digunakan *input* parameter berupa ukuran populasi, maksimum generasi, *crossover rate*, *mutation rate* dan persentase segmen mutasi. Representasi kromosom diperoleh dari jumlah subyek distribusi, untuk tahap 1 panjang kromosom diperoleh dari pabrik dikalikan jumlah distributor dan untuk tahap 2 panjang kromosom diperoleh dari jumlah distributor dikali dengan

jumlah agen. Tabel biaya akan terbentuk secara otomatis berdasarkan jumlah pabrik, distributor dan agen. Biaya distribusi akan dikalikan dengan kapasitas distribusi yang diperoleh dari prioritas distribusi pada representasi kromosom. Dari total biaya distribusi dapat dihitung nilai *fitness* berdasarkan persamaan 2.8 pada bab 2. Nilai *fitness* digunakan untuk membandingkan kebaikan solusi dari setiap kromosom. Secara lebih detail, proses yang terdapat dalam sistem adalah :

1. Sistem akan membentuk populasi awal yang terdiri dari sejumlah kromosom sesuai dengan *input* parameter ukuran populasi. Kromosom-kromosom tersebut akan dirandom untuk menentukan prioritas distribusi sesuai pada representasi kromosom. Pada inisialisasi populasi awal juga disertakan total biaya distribusi dan nilai *fitness*-nya.
2. Setelah proses inisialisasi populasi awal, sistem melakukan fase reproduksi dengan *crossover* dan mutasi. Reproduksi digunakan untuk menghasilkan solusi baru. *Crossover* dilakukan dengan mengambil dua *parent* dari populasi awal, jumlah *offspring* atau anak yang dihasilkan adalah sejumlah *crossover rate* dikali dengan ukuran populasi. Sedangkan mutasi merupakan proses reproduksi yang melibatkan 1 *parent* pada populasi awal, dengan jumlah *offspring* sebanyak *mutation rate* dikali dengan ukuran populasi.
3. Proses evaluasi dilakukan untuk mengumpulkan semua individu pada populasi awal dan hasil reproduksi untuk persiapan sebelum dilakukan seleksi.
4. Proses seleksi digunakan untuk mendapatkan individu baru pada generasi selanjutnya. Seleksi dilakukan dengan *elitism*, mengurutkan terlebih dahulu kromosom yang memiliki nilai terbesar hingga terkecil kemudian diambil individu baru sejumlah populasi awal.
5. Dari proses seleksi didapatkan individu-individu dengan nilai *fitness* terbesar. Hasil akhir berupa rute distribusi beserta total biaya biaya distribusi dari kromosom yang memiliki nilai *fitness* tertinggi.

### 3.3.2 Deskripsi Data

Optimasi distribusi barang dua tahap menggunakan data berupa jumlah pabrik, distributor dan agen, data kapasitas persediaan pabrik, data permintaan dari distributor dan agen. Pada sistem ini pengaruh utama pada distribusi berupa banyaknya kapasitas barang yang akan didistribusikan dan biaya yang akan dikeluarkan pada proses pendistribusian barang. Digunakan pabrik sebanyak 2 buah, distributor 5 buah dan agen sebanyak 10. Data distribusi barang dua tahap yang akan digunakan pada penelitian ini berupa tabel yang berisi jumlah produksi, jumlah permintaan, dan tabel biaya distribusi. Data distribusi dua tahap pada penelitian ini dijelaskan pada Tabel 3.1 sampai 3.3 mengenai data kapasitas produksi pabrik dan permintaan distributor dan agen.

**Tabel 3.1** Data Pabrik dan Kapasitas Persediaan Pabrik

<b>Id Pabrik</b>	<b>Nama Pabrik</b>	<b>Kapasitas Pabrik</b>
1	Pabrik 1	1000
2	Pabrik 2	1000

**Tabel 3.2** Data Distributor dan Kapasitas Permintaan Distributor

<b>Id Distributor</b>	<b>Nama Distributor</b>	<b>Kapasitas Distributor</b>
1	Distributor 1	300
2	Distributor 2	400
3	Distributor 3	250
4	Distributor 4	600
5	Distributor 5	450

**Tabel 3.3** Data Agen dan Kapasitas Permintaan Agen

<b>Id Pabrik</b>	<b>Nama Agen</b>	<b>Permintaan Agen</b>
1	Agen 1	150
2	Agen 2	175
3	Agen 3	200
4	Agen 4	225
5	Agen 5	200
6	Agen 6	150
7	Agen 7	150
8	Agen 8	300
9	Agen 9	200
10	Agen 10	250

Setelah dijelaskan mengenai data produksi dan permintaan, pada distribusi dua tahap dibutuhkan pula data biaya distribusi untuk pengiriman barang dari pabrik menuju distributor dan dari distributor menuju agen. Data biaya dijelaskan pada Tabel 3.4 mengenai data distribusi pabrik ke distributor dan Tabel 3.5 mengenai data distribusi dari distributor ke agen.

**Tabel 3.4** Biaya Distribusi Pabrik ke Distributor

Node	1	2	3	4	5
1	27	32	10	17	16
2	14	12	29	12	29

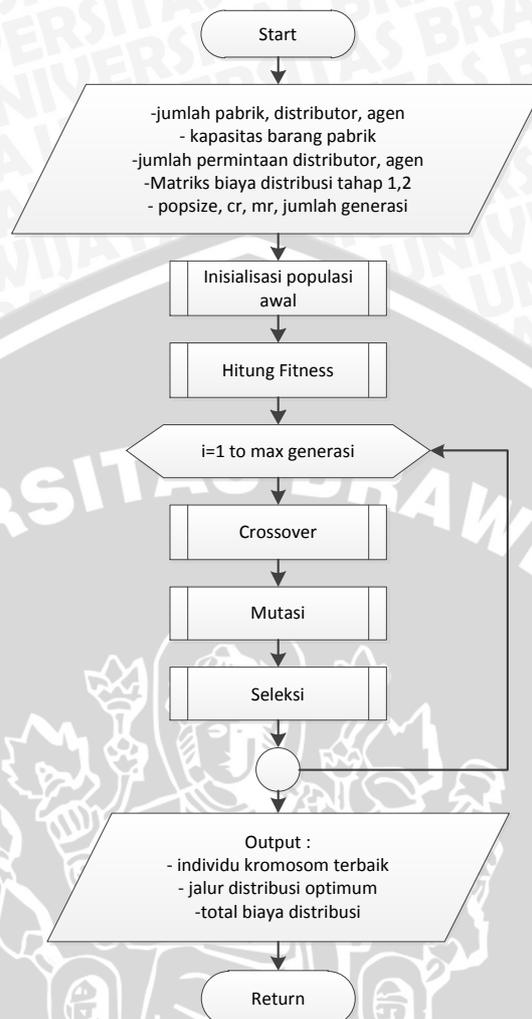
**Tabel 3.5** Biaya Distribusi Distributor ke Agen

Node	1	2	3	4	5	6	7	8	9	10
1	20	23	14	2	68	16	21	53	25	12
2	62	66	61	26	55	67	57	24	13	26
3	13	59	39	71	66	19	11	11	40	12
4	8	14	76	51	79	19	54	23	6	61
5	19	23	61	59	15	11	73	62	47	69

Data kapasitas dan biaya yang didefinisikan merupakan data yang akan digunakan pada kasus distribusi dua tahap. Pada penelitian ini dibahas mengenai distribusi barang dua tahap yang diharapkan optimasi dengan menerapkan algoritma genetika dapat diterapkan pada berbagai kasus distribusi dua tahap, sehingga data yang digunakan pada formulasi masalah merupakan data *dummy* mengenai distribusi dua tahap yang sesuai dengan berbagai literatur.

### 3.4 Siklus Penyelesaian Masalah Optimasi Distribusi Barang Dua Tahap dengan GA's

Siklus penyelesaian masalah optimasi distribusi barang dua tahap menggunakan algoritma genetika mengacu pada subbab 3.3 mengenai formulasi permasalahan. Data-data yang didefinisikan akan diolah menjadi sebuah solusi berupa jalur distribusi optimal dengan biaya minimum menggunakan algoritma genetika. Diagram alir proses optimasi distribusi barang dua tahap dengan algoritma genetika ditunjukkan pada Gambar 3.2.



**Gambar 3.2** Diagram Alir Proses Optimasi Distribusi Dua Tahap dengan GA's  
Proses optimasi distribusi dua tahap dengan algoritma genetika sesuai dengan diagram alir pada Gambar 3.2 adalah sebagai berikut :

1. *Input* data awal berupa jumlah pabrik beserta kapasitas persediaan, jumlah distributor dan agen beserta kapasitas permintaan, tabel biaya distribusi tahap 1 dan 2. Parameter genetika berupa ukuran populasi, maksimum generasi, *crossover rate*, *mutation rate* dan persentase segmen mutasi.
2. Inisialisasi populasi awal yang dibentuk dari nilai *inputan*, pembentukan populasi awal dilakukan secara random, untuk tahap satu diperoleh jumlah gen sebanyak pabrik dikalikan dengan agen sedangkan pada tahap dua diperoleh jumlah gen sebanyak agen dikalikan dengan sub agen. Gen pada kromosom mempresentasikan prioritas distribusi akan diletakkan pada node lokasi.

3. Melakukan perhitungan nilai *fitness* sebagai nilai pembandingan pada kebaikan solusi setiap kromosom. Perhitungan nilai *fitness* dilakukan sesuai dengan persamaan 2.8 pada bab 2.
4. Melakukan proses reproduksi dengan *crossover* menggunakan metode *one cut point crossover*, jumlah *offspring* yang dihasilkan adalah sebanyak *crossover rate (cr)* dikali dengan jumlah populasi awal.
5. Melakukan proses mutasi dengan metode *exchange mutation*, yang menghasilkan *offspring* sejumlah *mutation rate (mr)* dikali dengan populasi awal.
6. Melakukan evaluasi dengan mengumpulkan semua individu dari inisialisasi populasi awal dan *offspring* untuk proses seleksi.
7. Proses seleksi dilakukan dengan menggunakan metode *elitism selection*, seleksi dilakukan terhadap individu pada populasi awal dan individu baru hasil dari proses reproduksi. *Elitism selection* dilakukan dengan mengurutkan kromosom sesuai dengan nilai *fitness*nya dari besar ke kecil. Jika jumlah generasi sudah terpenuhi maka proses algoritma genetika untuk optimasi distribusi dua tahap berakhir dan menghasilkan solusi berupa jalur distribusi yang optimal beserta biaya distribusi minimum.

Secara lebih detail penerapan algoritma genetika pada kasus optimasi distribusi barang dua tahap dijelaskan pada langkah selanjutnya beserta perhitungan nilai *fitness*. Proses mengenai algoritma genetika secara lebih detail adalah sebagai berikut :

#### **3.4.1 Representasi Kromosom dan Perhitungan Nilai *Fitness***

Distribusi dua tahap merupakan masalah kombinatorial yang dapat diselesaikan menggunakan representasi permutasi. Representasi kromosom untuk distribusi dua tahap, setiap gen dalam kromosom menyatakan prioritas alokasi distribusi. Pada penelitian ini representasi kromosom menggunakan representasi permutasi. Panjang kromosom ditentukan oleh banyaknya pabrik, distributor dan agen yang terbentuk secara dinamis. Gen pada kromosom posisinya ditentukan secara random yang merepresentasikan prioritas distribusi. Untuk kromosom pada segmen 1 didapatkan dari jumlah pabrik dikalikan dengan jumlah distributor,

sedangkan untuk segmen kedua didapatkan dari jumlah distributor dikalikan jumlah agen. Untuk contoh kasus perhitungan manual digunakan 2 pabrik yang mendistribusikan barang ke 3 distributor dan 4 agen, maka terbentuk kromosom seperti ditunjukkan pada Gambar 3.3.

	Tahap 1						Tahap 2											
Posisi	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	11	12
P1	3	4	6	1	2	5	11	3	4	6	7	1	2	8	5	9	10	12

**Gambar 3.3** Contoh Rancangan Kromosom

Pada Gambar 3.3 posisi gen pada kolom tahap 1 menunjukkan representasi distribusi dari pabrik ke distributor, dan tahap 2 merepresentasikan jalur distribusi untuk distributor ke agen. Posisi merupakan node alokasi distribusi sedangkan pada P1 merupakan kromosom yang merepresentasikan nilai random dari posisi untuk menentukan alokasi distribusi pada kromosom pertama pada sebuah populasi.

Pada contoh kasus distribusi maka dapat diuraikan menjadi sebuah solusi dengan terlebih dahulu memasukkan data sebagai berikut :

1. Memasukkan jumlah pabrik sebanyak 2, distributor sebanyak 3 dan Agen sebanyak 4.
2. Jumlah populasi (*pop\_size*) sebanyak 5.
3. *Crossover Rate* = 0,4 dan *Mutation Rate* = 0,2
4. Jumlah maksimum generasi 1 kali.

Setiap pabrik memiliki kapasitas barang yang akan didistribusikan pada distributor dan agen. Kapasitas dari pabrik, distributor dan agen dijelaskan pada Tabel 3.6 sebagai berikut :

**Tabel 3.6** Tabel Kapasitas Barang

No	Pabrik	Distributor	Agen
1	1000	650	500
2	900	700	500
3		550	450
4			450

Pada distribusi barang dari masing-masing pabrik ke distributor maupun distributor ke agen memiliki biaya yang berbeda pada setiap pengirimannya. Biaya yang digunakan pada contoh kasus merupakan nilai kecil untuk mempermudah perhitungan, pada contoh kasus nyata biaya bisa jadi dalam bentuk ribuan, ratusan ataupun jutaan tergantung pada harga pengiriman barang. Tabel 3.7 merupakan tabel biaya distribusi menuju pabrik pada distributor dan Tabel 3.8 merupakan tabel biaya distribusi dari distributor ke agen.

**Tabel 3.7** Biaya Distribusi Tahap 1 pabrik ke distributor

Pabrik	Distributor		
	1	2	3
1	5	6	8
2	8	5	7

**Tabel 3.8** Biaya Distribusi Tahap 2 distributor ke Agen

Distributor	Agen			
	1	2	3	4
1	5	6	7	8
2	8	8	8	4
3	6	5	3	6

Setelah ditentukan kapasitas barang dan biaya dari masing-masing pendistribusian maka akan dibentuk populasi awal dari *popsi* yang telah ditentukan pada *input* parameter genetika. Contoh kasus distribusi barang maka dapat diuraikan menjadi sebuah solusi dengan langkah-langkah sebagai berikut :

1. Pada pencarian alokasi distribusi tahap 1, persediaan barang di pabrik sebesar 1000 pada pabrik 1 dan 900 pada pabrik 2. Jumlah permintaan dari masing-masing distributor sebanyak 650 pada distributor 1, 700 pada distributor 2 dan 550 pada distributor 3. Pada kasus penyelesaian distribusi dua tahap, kromosom akan disusun seperti matriks, jika pabrik terdiri dari 2 buah dan distributor 3 buah untuk tahap 1 dapat digambarkan menjadi matriks 2 x 3. Pada setiap kolom matriks diberikan node 1 sampai dengan 6. Sedangkan yang dicontohkan pada Gambar 3.4 pada P1 merupakan nilai random dari posisi node. Pada kromosom P1 untuk tahap 1 alokasi unit maksimum pada sel dengan nomor urut 3, dengan maksimal alokasi sebesar

550 unit barang.  $a_i$  dan  $b_j$  merupakan total baris dan kolom sementara untuk persediaan setiap pabrik dan permintaan dari setiap distributor. Pengisian gen ke 3 ditunjukkan pada Tabel 3.9.

**Tabel 3.9** Alokasi Distribusi dari Gen 3

Pabrik	Distributor			$a_i$	
	1	2	3		
1	1	2	3	550	1000
2	4	5	6		900
$b_j$	650	700	550		

2. Dari Tabel 3.9 permintaan pada agen 3 telah terpenuhi sehingga persediaan pada pabrik 1 menjadi 450. Selanjutnya isi sel 4 dengan unit maksimum distribusi sebanyak 650 unit. Pengisian barang pada gen ke 4 ditunjukkan pada Tabel 3.10.

**Tabel 3.10** Alokasi Distribusi dari Gen 4

Pabrik	Distributor			$a_i$	
	1	2	3		
1	1	2	3	550	450
2	4	5	6		900
$b_j$	650	700	0		

3. Isi sel 6 dan sel 1 dengan 0 karena permintaan pada agen 3 dan 1 sudah terpenuhi. Pengisian gen ke 6 dan 1 ditunjukkan pada Tabel 3.11.

**Tabel 3.11** Alokasi Distribusi dari Gen ke 6 dan 1

Pabrik	Distributor			$a_i$	
	1	2	3		
1	1	2	3	550	450
2	4	5	6	0	250
$b_j$	0	700	0		

4. Isi sel 2 dengan 450 dari sisa persediaan dari pabrik 1 sebanyak 450. Pengisian gen ke 2 ditunjukkan pada Tabel 3.12.

**Tabel 3.12** Alokasi Distribusi dari Gen ke 2

Pabrik	Distributor			a <sub>i</sub>
	1	2	3	
1	<sup>1</sup> 0	<sup>2</sup> 450	<sup>3</sup> 550	450
2	<sup>4</sup> 650	<sup>5</sup> 250	<sup>6</sup> 0	250
b <sub>j</sub>	0	250	0	

5. Isi sel 5 dengan 250 dari sisa persediaan dari pabrik dua sebanyak 250 dan total permintaan sebanyak 700 untuk distributor 2. Pengisian gen ke 5 ditunjukkan pada Tabel 3.13.

**Tabel 3.13** Alokasi Distribusi dari Gen ke 5

Pabrik	Distributor			a <sub>i</sub>
	1	2	3	
1	<sup>1</sup> 0	<sup>2</sup> 450	<sup>3</sup> 550	0
2	<sup>4</sup> 650	<sup>5</sup> 250	<sup>6</sup> 0	0
b <sub>j</sub>	0	0	0	

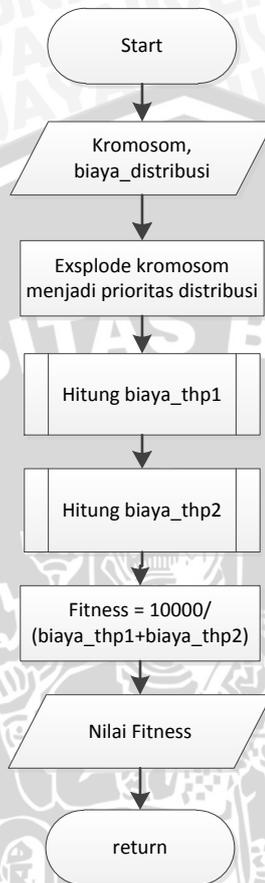
Untuk tahap dua juga dilakukan pengisian seperti tahap satu. Dengan mengalokasikan barang sesuai dengan prioritas gen yang telah direpresentasikan seperti pada Gambar 3.3. Berikut adalah alokasi untuk tahap 2, pengisian barang sesuai dengan gen pada kromosom tahap 2 ditunjukkan pada Tabel 3.14.

**Tabel 3.14** Alokasi Distribusi tahap 2

Distributor	Agen				a <sub>i</sub>
	1	2	3	4	
1	<sup>1</sup> 200	<sup>2</sup> 0	<sup>3</sup> 0	<sup>4</sup> 450	650
2	<sup>5</sup> 200	<sup>6</sup> 500	<sup>7</sup> 0	<sup>8</sup> 0	700
3	<sup>9</sup> 100	<sup>10</sup> 0	<sup>11</sup> 450	<sup>12</sup> 0	550
b <sub>j</sub>	500	500	450	450	

Dari hasil representasi kromosom didapatkan alokasi pada tahap 1 dan 2 adalah pada Tabel 3.11 dan Tabel 3.12. Setelah didapatkan alokasi barang pada setiap tahap maka dilakukan perhitungan nilai *fitness*. Nilai *fitness* digunakan untuk mencari jalur optimal distribusi barang dua tahap. Pencarian nilai *fitness* digunakan

persamaan 2.9 pada bab 2, diagram alir proses perhitungan nilai *fitness* ditunjukkan pada Gambar 3.4.



**Gambar 3.4** Diagram Alir Proses Perhitungan *Fitness*

Berdasarkan Gambar 3.4, langkah- langkah perhitungan nilai *fitness* adalah sebagai berikut :

1. *Inputan* yang digunakan sebagai perhitungan *fitness* setiap individu adalah kromosom dan total biaya.
2. Sistem akan mengubah kromosom menjadi prioritas distribusi.
3. Sistem akan mengecek lokasi distribusi sesuai dengan prioritas gen pada kromosom dengan jumlah unit distribusi dan tabel biaya distribusi sesuai dengan fungsi optimasi biaya distribusi dua tahap pada persamaan 2.1.

Proses perhitungan nilai *fitness* =  $10000/\text{total\_biaya}$

Pada contoh kasus perhitungan untuk kasus distribusi barang, berikut akan dijelaskan mengenai tahapan perhitungan nilai *fitness* dari kromosom yang ditunjukkan untuk Gambar 3.3.

1. Mengalokasikan unit maksimum barang pada tiap-tiap node sesuai random kromosom pada tahap 1 sesuai dengan persediaan setiap pabrik dan permintaan setiap distributor.
2. Mengulangi langkah alokasi barang untuk tahap 2 seperti langkah pada tahap 1 sesuai dengan prioritas setiap gen yang telah direpresentasikan. Pada tahap 2 merupakan pengalokasian unit maksimum barang pada tiap-tiap node sesuai random kromosom untuk pendistribusian barang dari distributor menuju pada agen.
3. Setelah persediaan yang ada sesuai dengan jumlah permintaan, didapatkan jalur distribusi, jumlah alokasi barang dan tabel biaya yang disesuaikan dengan tabel biaya distribusi pada tabel 3.7 dan 3.8. Biaya distribusi diperoleh dari jumlah barang yang dikirimkan dikalikan dengan biaya pengiriman barang per-unit. Tabel 3.15 merupakan biaya rute pengiriman dari pabrik ke distributor, dengan mengalikan sejumlah barang yang didistribusikan dari pabrik ke distributor dengan biaya pengiriman per-unit barang. Kolom  $i,j$  merupakan jalur pendistribusian dari pabrik ke distributor,  $x[i,j]$  merupakan jumlah barang yang didistribusikan,  $t[i,j]$  merupakan biaya pengiriman barang per-unit dan  $x[i,j]*t[i,j]$  merupakan total biaya distribusi per-jalur.

**Tabel 3.15** Tabel biaya rute dari Pabrik ke Distributor

$i,j$	$x[i,j]$	$t[i,j]$	$x[i,j]*t[i,j]$
[1,2]	450	6	2700
[1,3]	550	8	4400
[2,1]	650	8	5200
[2,2]	250	5	1250
Total			13.550

Tahap 2 merupakan biaya rute pengiriman dari distributor ke agen. Biaya diperoleh dari perkalian total barang yang didistribusikan dengan

biaya pengiriman barang per-unit. Kolom j,k merupakan jalur pendistribusian dari distributor ke agen,  $y[j,k]$  merupakan jumlah barang yang didistribusikan,  $c[j,k]$  merupakan biaya pengiriman barang per-unit dan  $y[j,k] * c[j,k]$  merupakan total biaya distribusi per-jalur. Tabel total biaya distribusi tahap ke-2 ditunjukkan pada Tabel 3.16.

**Tabel 3.16** Tabel biaya rute dari Distributor ke Agen

j,k	$y[j,k]$	$c[j,k]$	$y[j,k] * c[j,k]$
[1,1]	200	5	1000
[1,4]	450	8	3600
[2,1]	200	8	1600
[2,2]	500	8	4000
[3,1]	100	6	600
[3,3]	450	3	1350
Total			12.150

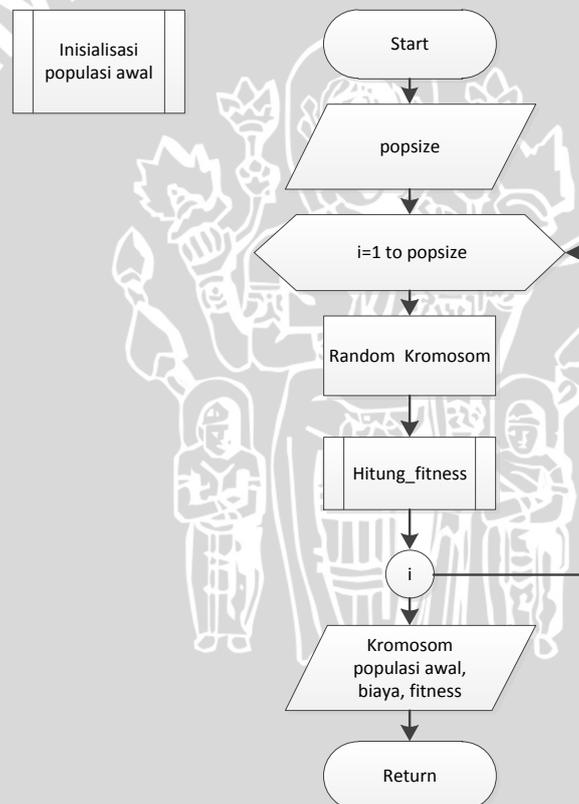
4. Dari tabel biaya rute didapatkan total biaya pengiriman untuk masing-masing tahap didapatkan total biaya distribusi pada tahap 1 sebesar 13.550 yang diperoleh dari total unit barang yang didistribusikan dikalikan dengan biaya pengiriman dari pabrik menuju distributor, sedangkan untuk tahap 2 sebesar 12.150 yang diperoleh dari unit barang yang didistribusikan dikali dengan biaya pengiriman barang dari distributor menuju agen.
5. Dari total biaya yang diperoleh pada tahap 1 dan 2 dapat dihitung *fitness* dari kromosom sesuai dengan persamaan 2.9 pada bab 2. Nilai *fitness* untuk kromosom adalah :

$$f = \frac{1}{((13.550) + (12.150))/10000} = 0,38911$$

Pada persamaan diatas nilai  $f(x)$  atau total biaya distribusi dibagi dengan angka 10000 yang bertujuan agar nilai koma tidak terlalu besar. Jika diterapkan pada kasus nyata dengan angka yang lebih besar maka dapat digunakan bilangan pembagi yang lebih besar agar nilai yang dihasilkan tidak memiliki nilai koma yang panjang. Sehingga untuk contoh kromosom pada gambar 3.3 didapatkan nilai *fitness* 0,38911.

### 3.4.2 Inisialisasi Populasi Awal

Inisialisasi populasi awal digunakan untuk membentuk kromosom pada satu populasi. Inisialisasi populasi awal dibentuk setelah representasi kromosom selesai dilakukan. Populasi awal berisi kumpulan kromosom yang dibentuk secara random pada tahap 1 dan tahap 2, panjangnya sesuai dengan jumlah pabrik, distributor dan jumlah agen. Pada inisialisasi populasi awal kromosom-kromosom yang berada pada satu populasi dihitung nilai *fitness* yang bertujuan untuk membedakan kebaikan solusi dari masing-masing kromosom pada permasalahan optimasi distribusi barang dua tahap. Diagram alir inisialisasi populasi awal ditunjukkan pada diagram alir pembentukan populasi awal Gambar 3.5.



**Gambar 3.5** Diagram Alir Pembentukan Populasi Awal

Pembentukan populasi awal yang ditunjukkan pada Gambar 3.5, langkah-langkahnya adalah sebagai berikut :

1. Menentukan jumlah populasi yang akan dibentuk.

2. Membentuk kromosom dari masukan berupa jumlah pabrik, jumlah distributor dan agen serta kapasitas persediaan dan permintaan.
3. Menghitung nilai *fitness* dan biaya distribusi hasil dari kromosom.
4. Melakukan perulangan hingga populasi awal terpenuhi sesuai dengan jumlah populasi yang telah ditentukan sebelumnya.
5. Setelah jumlah populasi awal sudah terbentuk akan ditampilkan populasi awal sesuai dengan *inputan popsize* nilai *fitness* dan total biaya distribusi yang dihasilkan. .

Pada tahap inisialisasi kromosom ditentukan jumlah populasi dan membangkitkan individu secara random untuk dilakukan proses perhitungan menggunakan algoritma genetika. Pada tahap insialisasi kromosom untuk distribusi dua tahap dibangkitkan sebanyak 5 populasi dengan panjang tahap 1 sebanyak 6 gen dari 2 pabrik dan 3 distributor, untuk tahap 2 sepanjang 12 gen dari 3 distributor dan 4 agen. Pembangkitan populasi awal dijelaskan pada Tabel 3.17.

**Tabel 3.17** Tabel Populasi Awal dan Nilai *Fitness*

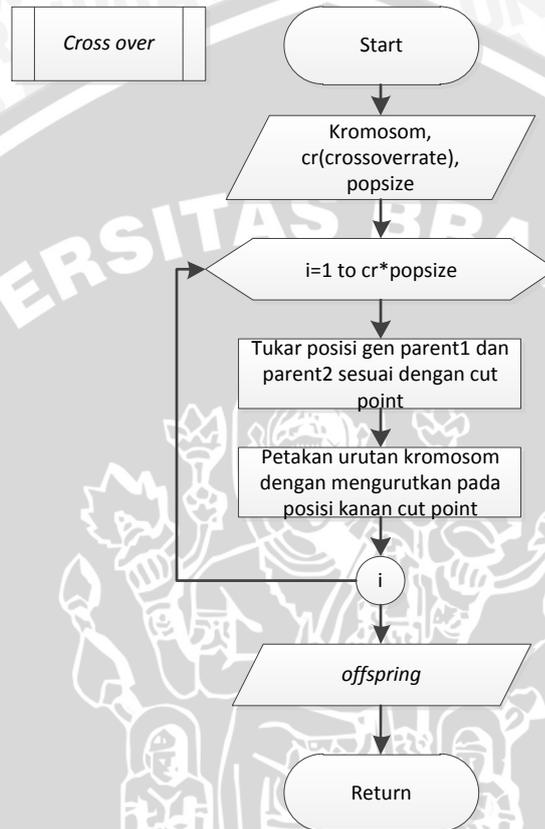
Posisi	Tahap 1						Tahap 2												FITNESS
	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	11	12	
P1	3	4	6	1	2	5	11	3	4	6	7	1	2	8	5	9	10	12	0,38911
P2	5	3	4	2	1	6	10	7	9	5	3	4	2	1	6	8	11	12	0,40404
P3	4	5	3	6	1	2	9	12	7	4	5	3	6	1	2	8	10	11	0,36832
P4	2	1	5	3	6	4	8	12	11	2	1	5	3	6	4	10	9	7	0,45767
P5	1	2	6	4	5	3	7	9	12	1	2	6	4	5	3	11	8	10	0,43384

Terdapat populasi awal sebanyak 5 kromosom dengan nilai *fitness* yang telah didefinisikan sesuai dengan perhitungan nilai *fitness* pada subbab 3.4.1. Populasi awal akan dilakukan perhitungan menggunakan algoritma genetika sesuai dengan contoh alokasi barang sesuai dengan prioritas gen sesuai pada penjelasan mengenai representasi kromosom. Populasi awal akan diambil secara random sesuai dengan *crossover rate* dan *mutation rate* dikalikan dengan jumlah populasi untuk proses reproduksi.

### 3.4.3 Proses *Crossover*

Proses *crossover* atau kawin silang merupakan proses reproduksi dengan melibatkan 2 *parent*, pada kasus ini menggunakan metode *one cut point*

pada setiap tahap distribusinya. Jumlah individu hasil *crossover* ditentukan sesuai dengan *inputan crossover rate* (*crossover rate*) dikali dengan *popsi*. Pada proses *Crossover* dipilih dua *parent* secara random dengan pemilihan titik potong juga dilakukan secara random. Gambar 3.6 menunjukkan diagram alir proses *crossover*.



**Gambar 3.6** Diagram Alir Proses *Crossover*

Berdasarkan Gambar 3.6, langkah-langkah proses *crossover* adalah sebagai berikut:

1. Pada proses *crossover* digunakan kromosom *parent* sebagai data *inputan* yang diambil secara random dan data *inputan crossover rate* (*cr*).
2. Sistem akan melakukan perulangan untuk menghasilkan *offspring* sejumlah *crossover rate* dikalikan dengan jumlah populasi awal (*popsi*).
3. Pada *parent* yang terpilih untuk proses *crossover* akan dipilih *cut point* secara random pada masing-masing tahapnya.

4. Tukar posisi gen untuk *parent* 1 dengan *parent* 2 sesuai dengan titik potong. Petakan urutan kromosom dengan mengurutkan pada posisi kanan *cut point*.
5. Setelah proses selesai maka akan tampil individu hasil dari persilangan dua kromosom yang dipilih secara random.

Setelah ditentukan populasi awal untuk contoh kasus distribusi dua tahap, akan diambil kromosom secara random yang akan menghasilkan individu baru sebanyak *crossover rate* dikalikan dengan jumlah populasi. Pada perhitungan manual digunakan *crossover rate* sebesar 0,4, maka jumlah individu baru yang dihasilkan dari proses *crossover* adalah sebanyak 2 individu baru. Untuk menghasilkan satu individu baru digunakan dua *parent*, diambil P2 dan P4 yang akan dijadikan sebagai *parent* yang ditunjukkan pada Tabel 3.18.

**Tabel 3.18** Pemilihan P2 dan P4 Sebagai *Parent*

P	Kromosom														Fitness				
P2	5	3	4	2	1	6	10	7	9	5	3	4	2	1	6	8	11	12	0,40404
P4	2	1	5	3	6	4	8	12	11	2	1	5	3	6	4	10	9	7	0,45767

*Parent* P2 disilangkan dengan P4 dengan menggunakan *one cut point crossover*, diambil titik potong pada gen ke 2 pada tahap 1 dan titik potong pada gen ke 4 pada tahap 2. Kromosom segmen kiri pada tahap 1 dari *child* di dapatkan dari *parent* P2 sedangkan segmen kanan pada tahap 1 didapatkan dari urutan gen sisa dari *parent* P4 begitu pula untuk *crossover* pada tahap dua. *Child* hasil *crossover* ditunjukkan pada Gambar 3.7.

C1	5	3	2	1	6	4	10	7	9	5	8	12	11	2	1	3	6	4	0,40404
----	---	---	---	---	---	---	----	---	---	---	---	----	----	---	---	---	---	---	---------

**Gambar 3.7** *Child* Hasil *Crossover* P2 dan P4

Berdasarkan *crossover rate* yang ditentukan sebesar 0,4 maka individu baru yang dihasilkan adalah sebanyak 2. Untuk *child* ke 2 digunakan P3 dan P5 yang ditunjukkan pada Tabel 3.19.

**Tabel 3.19** Pemilihan P3 dan P5 sebagai *Parent*

P	Kromosom														Fitness				
P3	4	5	3	6	1	2	9	12	7	4	5	3	6	1	2	8	10	11	0,36832
P5	1	2	6	4	5	3	7	9	12	1	2	6	4	5	3	11	8	10	0,43384

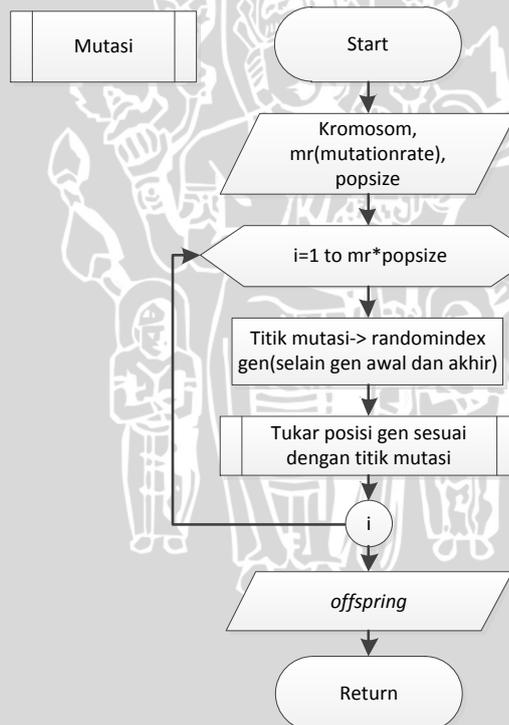
Hasil persilangan dari P3 dan P5 menghasilkan C2 yang dilakukan dengan metode *one cut point crossover* seperti yang telah dijelaskan sebelumnya. Hasil C2 ditunjukkan pada Gambar 3.8.

C2	4	5	1	2	6	3	9	12	7	4	1	2	6	5	3	11	8	10	0,36832
----	---	---	---	---	---	---	---	----	---	---	---	---	---	---	---	----	---	----	---------

**Gambar 3.8** Child Hasil Crossover P3 dan P5

### 3.4.4 Proses Mutasi

Proses mutasi yang digunakan pada penelitian ini adalah metode *exchange mutation* yang akan dilakukan pada setiap segmen. Jumlah individu baru hasil dari proses mutasi adalah sesuai dengan *input mutation rate* dikalikan dengan jumlah populasi awal atau *popsi*. Diagram alir proses mutasi ditunjukkan pada Gambar 3.9.



**Gambar 3.9** Diagram Alir Proses Mutasi

Berdasarkan Gambar 3.9, langkah-langkah proses reproduksi dengan mutasi adalah sebagai berikut :

1. Pada proses mutasi digunakan 1 kromosom *parent* sebagai *input*-an yang diambil secara random dan *inputan mutation rate*.

2. Sistem akan melakukan perulangan untuk menghasilkan individu baru hasil mutasi sejumlah *mutation rate* dikalikan dengan jumlah populasi awal (*popsize*).
3. Pada *parent* yang terpilih untuk proses mutasi akan dipilih secara random titik mutasi kecuali indeks awal dan indeks akhir.
4. Tukar posisi gen sesuai dengan titik mutasi.
5. Setelah proses selesai maka akan tampil individu dari hasil mutasi kromosom yang dipilih secara random.

Pada proses mutasi yang digunakan *mutation rate* sebesar 0,2, sehingga *child* yang dihasilkan dari proses mutasi adalah 1. Diambil P1 sebagai *parent* pada proses mutasi, Metode ini bekerja dengan memilih dua posisi (*exchange point / XP*) secara random kemudian menukarkan nilai pada posisi tersebut seperti pada Tabel 3.20.

**Tabel 3.20** P1 sebagai *Parent* Proses Mutasi

P	Kromosom															Fitness			
P1	3	4	6	1	2	5	11	3	4	6	7	1	2	8	5	9	10	12	0,38911

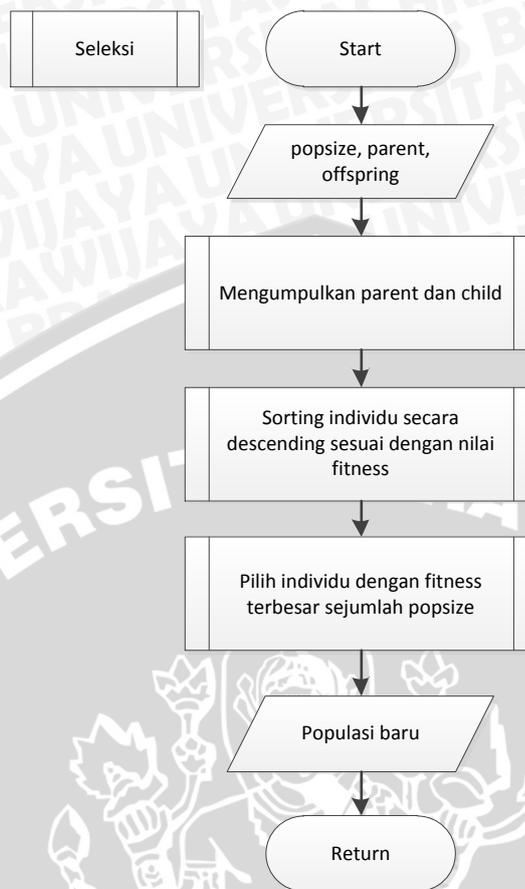
*Child* yang diperoleh dari proses mutasi adalah sebanyak 1, ditunjukkan pada Gambar 3.10.

C3	3	2	6	1	4	5	11	3	9	6	7	1	2	8	5	4	10	12	0,41152
----	---	---	---	---	---	---	----	---	---	---	---	---	---	---	---	---	----	----	---------

**Gambar 3.10** Hasil Mutasi

### 3.4.5 Seleksi

Seleksi dilakukan untuk memilih individu terbaik dari himpunan populasi awal dan individu hasil proses reproduksi yang akan digunakan pada generasi berikutnya. Pada penelitian ini seleksi dilakukan menggunakan seleksi *elitism*, dengan menggumpulkan induk beserta individu baru yang didapatkan pada proses *crossover* dan mutasi. Seleksi *elitism* merupakan metode seleksi yang hanya memilih individu yang memiliki nilai *fitness* terbesar saja yang akan masuk pada individu baru untuk proses regenerasi berikutnya. Diagram alir proses seleksi ditunjukkan pada Gambar 3.11.



**Gambar 3.11** Diagram Alir Proses Seleksi

Berdasarkan Gambar 3.11 mengenai diagram alir proses seleksi, langkah-langkah seleksi *elitism* adalah sebagai berikut :

1. Menggabungkan populai awal dan individu baru hasil proses reproduksi.
2. Menghitung nilai *fitness* untuk setiap individu.
3. Mengurutkan individu sesuai dengan nilai *fitness* terbesar hinggann individu yang memiliki nilai *fitness terkecil*.
4. Memilih individu yang memiliki nilai *fitness* terbesar sejumlah populasi awal atau *popsi* untuk dijadikan populasi baru pada generasi selanjutnya.

Pada contoh kasus perhitungan manual dilakukan proses reproduksi pada subbab 3.4.3. Langkah awal pada seleksi *elitism* adalah mengumpulkan semua individu dari populasi awal dan *child* hasil reproduksi menggunakan proses

*crossover* dan mutasi, Tabel 3.21 menunjukkan kumpulan individu populasi awal dan individu hasil reproduksi beserta nilai *fitness*.

**Tabel 3.21** Kumpulan Individu Populasi Awal dan Individu Hasil Reproduksi

Posisi	Tahap 1						Tahap 2												FITNESS
	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	11	12	
P1	3	4	6	1	2	5	11	3	4	6	7	1	2	8	5	9	10	12	0,38911
P2	5	3	4	2	1	6	10	7	9	5	3	4	2	1	6	8	11	12	0,40404
P3	4	5	3	6	1	2	9	12	7	4	5	3	6	1	2	8	10	11	0,36832
P4	2	1	5	3	6	4	8	12	11	2	1	5	3	6	4	10	9	7	0,45767
P5	1	2	6	4	5	3	7	9	12	1	2	6	4	5	3	11	8	10	0,43384
C1	5	3	2	1	6	4	10	7	9	5	8	12	11	2	1	3	6	4	0,40404
C2	4	5	1	2	6	3	9	12	7	4	1	2	6	5	3	11	8	10	0,36832
C3	3	2	6	1	4	5	11	3	9	6	7	1	2	8	5	4	10	12	0,41152

Setelah proses menggabungkan populasi awal maupun individu baru hasil reproduksi, proses selanjutnya adalah mengurutkan individu sesuai dengan individu yang memiliki nilai tertinggi hingga terendah. Berikut ini hasil pengurutan individu yang memiliki nilai tertinggi hingga terendah. Tabel 3.22 menunjukkan pengurutan populasi dari yang memiliki nilai *fitness* terbesar hingga terkecil.

**Tabel 3.22** Hasil Sorting Kumpulan Populasi

Posisi	Tahap 1						Tahap 2												FITNESS
	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	11	12	
P4	2	1	5	3	6	4	8	12	11	2	1	5	3	6	4	10	9	7	0,45767
P5	1	2	6	4	5	3	7	9	12	1	2	6	4	5	3	11	8	10	0,43384
C3	3	2	6	1	4	5	11	3	9	6	7	1	2	8	5	4	10	12	0,41152
P2	5	3	4	2	1	6	10	7	9	5	3	4	2	1	6	8	11	12	0,40404
C1	5	3	2	1	6	4	10	7	9	5	8	12	11	2	1	3	6	4	0,40404
P1	3	4	6	1	2	5	11	3	4	6	7	1	2	8	5	9	10	12	0,38911
P3	4	5	3	6	1	2	9	12	7	4	5	3	6	1	2	8	10	11	0,36832
C2	4	5	1	2	6	3	9	12	7	4	1	2	6	5	3	11	8	10	0,36832

Setelah proses pengurutan selesai hasil seleksi berupa individu baru yang memiliki nilai *fitness* tertinggi yang diambil sejumlah populasi awal dan akan digunakan pada generasi berikutnya. Tabel 3.23 menunjukkan individu baru hasil proses seleksi

**Tabel 3.23** Populasi Baru dan Individu Terbaik

Posisi	Tahap 1						Tahap 2												FITNESS
	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	11	12	
P4	2	1	5	3	6	4	8	12	11	2	1	5	3	6	4	10	9	7	0,45767
P5	1	2	6	4	5	3	7	9	12	1	2	6	4	5	3	11	8	10	0,43384
C3	3	2	6	1	4	5	11	3	9	6	7	1	2	8	5	4	10	12	0,41152
P2	5	3	4	2	1	6	10	7	9	5	3	4	2	1	6	8	11	12	0,40404
C1	5	3	2	1	6	4	10	7	9	5	8	12	11	2	1	3	6	4	0,40404

Dari hasil perhitungan menggunakan algoritma genetika untuk distribusi barang dua tahap maka didapatkan solusi optimal pada perhitungan manual yang ditunjukkan pada tabel 3.23 untuk satu generasi. Jalur distribusi optimal adalah pada kromosom P4 dengan nilai *fitness* 0,45767. Sehingga diperoleh matriks solusi untuk tahap 1 dan tahap 2 yang ditunjukkan pada tabel 3.24 dan 3.25.

**Tabel 3.24** Tabel Matriks Distribusi Tahap 1

Pabrik	Distributor			a <sub>i</sub>
	1	2	3	
1	<sup>1</sup> 300	<sup>2</sup> 700	<sup>3</sup> 0	1000
2	<sup>4</sup> 350	<sup>5</sup> 0	<sup>6</sup> 550	900
b <sub>j</sub>	650	700	550	

**Tabel 3.25** Tabel Matriks Distribusi Tahap 2

Distributor	Agen				a <sub>i</sub>
	1	2	3	4	
1	<sup>1</sup> 150	<sup>2</sup> 500	<sup>3</sup> 0	<sup>4</sup> 0	650
2	<sup>5</sup> 250	<sup>6</sup> 0	<sup>7</sup> 0	<sup>8</sup> 450	700
3	<sup>9</sup> 100	<sup>10</sup> 0	<sup>11</sup> 450	<sup>12</sup> 0	550
b <sub>j</sub>	500	500	450	450	

Setelah didapatkan matriks distribusi per tahap maka dapat ditentukan biaya distribusi per jalur yang berupa jumlah unit barang yang di distribusikan dikali dengan biaya pengiriman. Tabel 3.26 dan 3.27 merupakan biaya distribusi barang dari solusi kromosom P4.

**Tabel 3.26** Tabel biaya rute dari Pabrik ke Distributor

i,j	x[i,j]	t[i,j]	x[i,j]*t[i,j]
[1,1]	300	5	1500
[1,2]	700	6	4200
[2,1]	350	8	2800
[2,3]	550	7	3850
Total			12.350

**Tabel 3.27** Tabel biaya rute dari Distributor ke Agen

j,k	y[j,k]	c[j,k]	y[j,k]*c[j,k]
[1,1]	150	5	750
[1,2]	500	6	3000
[2,1]	250	8	2000
[2,4]	450	4	1800
[3,1]	100	6	600
[3,3]	450	3	1350
Total			9500

Jika diterapkan dalam proses nyata hasil solusi jalur distribusi paling optimal yang ditunjukkan oleh kromosom P4 berdasarkan Tabel 3.26 dan Tabel 3.27 adalah sebagai berikut :

Pabrik ke 1 menuju distributor ke 1, kapasitas 300, biaya distribusi 1500.

Pabrik ke 1 menuju distributor ke 2, kapasitas 700, biaya distribusi 4200

Pabrik ke 2 menuju distributor ke 1, kapasitas 350, biaya distribusi 2800

Pabrik ke 2 menuju distributor ke 3, kapasitas 550, biaya distribusi 3850

Distributor ke 1 menuju agen ke 1, kapasitas 150, biaya distribusi 750

Distributor ke 1 menuju agen ke 2, kapasitas 500, biaya distribusi 3000

Distributor ke 2 menuju agen ke 1, kapasitas 250, biaya distribusi 2000

Distributor ke 2 menuju agen ke 4, kapasitas 450, biaya distribusi 1800

Distributor ke 3 menuju agen ke 1, kapasitas 100, biaya distribusi 600

Distributor ke 3 menuju agen ke 3, kapasitas 450, biaya distribusi 1350

Total Biaya distribusi adalah 21850.

## BAB IV

### PERANCANGAN

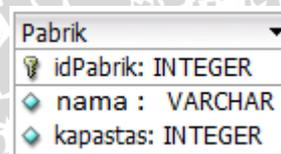
Pada bab perancangan ini dijelaskan mengenai perancangan database, perancangan user interface, dan perancangan uji coba dan evaluasi untuk penerapan algoritma genetika pada kasus distribusi dua tahap.

#### 4.1 Perancangan Database

Perancangan database bertujuan untuk memberikan gambaran data yang digunakan dan penyimpanan data. Pada perancangan database berisi struktur tabel yang berelasi untuk keperluan sistem. Perancangan database pada penerapan algoritma genetika untuk permasalahan distribusi barang dua tahap berisi tabel pabrik, distributor dan agen, struktur tabel database terdiri dari :

##### 1. Tabel Pabrik

Tabel pabrik digunakan untuk menyimpan data pabrik berupa idPabrik, namaPabrik dan kapasitas. Perancangan tabel pabrik dapat dilihat pada Gambar 4.1.

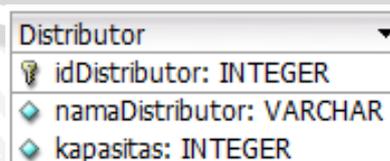


Pabrik	
idPabrik	INTEGER
nama	VARCHAR
kapasitas	INTEGER

Gambar 4.1 Tabel Pabrik

##### 2. Tabel Distributor

Tabel distributor berfungsi untuk menyimpan data distributor yang berupa idDistributor, namaDistributor dan kapasitas. Perancangan tabel distributor dapat dilihat pada Gambar 4.2.

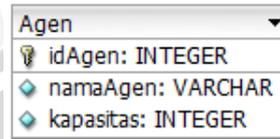


Distributor	
idDistributor	INTEGER
namaDistributor	VARCHAR
kapasitas	INTEGER

Gambar 4.2 Tabel Distributor

### 3. Tabel Agen

Tabel agen merupakan tabel yang berfungsi untuk menyimpan data agen yang didalamnya terdapat atribut idAgen, namaAgen dan kapasitas. Perancangan tabel agen dapat dilihat pada Gambar 4.3.

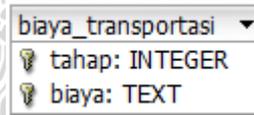


Agen	
idAgen	INTEGER
namaAgen	VARCHAR
kapasitas	INTEGER

**Gambar 4.3** Tabel Agen

### 4. Tabel Biaya Transportasi

Biaya transportasi digunakan untuk distribusi antara pabrik dan distributor. Pada perancangan tabel biaya transportasi terdapat atribut berupa tahap dan biaya. Perancangan tabel biaya distribusi tahap 1 dapat dilihat pada Gambar 4.4.



biaya_transportasi	
tahap	INTEGER
biaya	TEXT

**Gambar 4.4** Tabel Biaya Transportasi

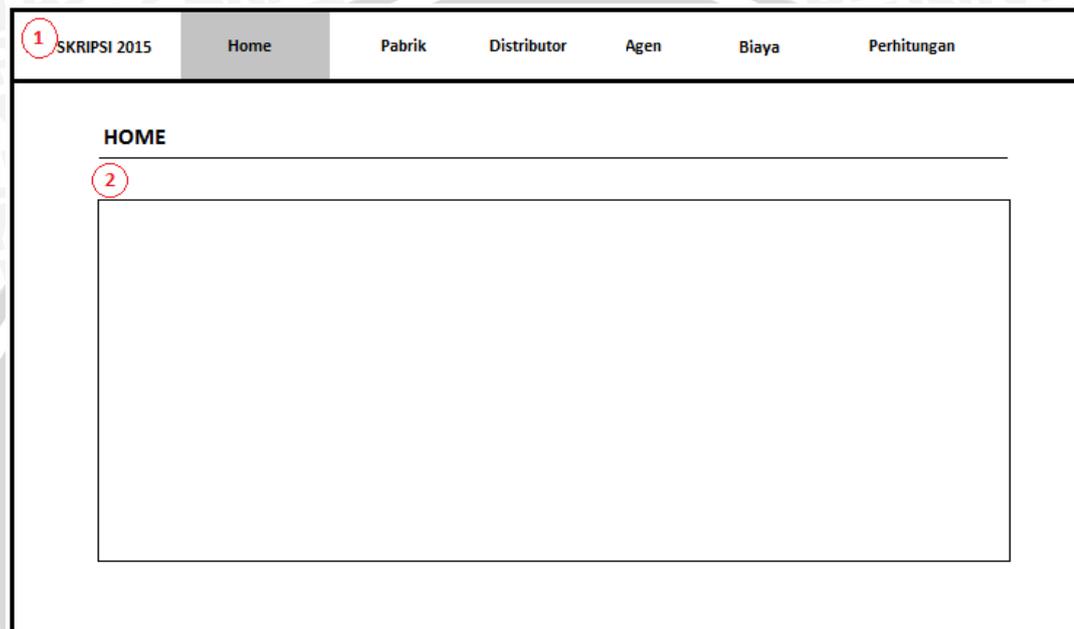
## 4.2 Perancangan Antarmuka

Perancangan antarmuka menunjukkan gambaran desain dari antarmuka sistem yang menerapkan algoritma genetika untuk optimasi distribusi barang dua tahap. Perancangan antarmuka digunakan untuk tahap implementasi dari metode yang diterjemahkan kedalam suatu kode program. Tahapan implementasi menghasilkan suatu aplikasi sebagai media representasi dari metode yang digunakan dalam penelitian. Antarmuka merupakan sarana interaksi antara pengguna dengan sistem. Perancangan antarmuka menggambarkan bentuk tampilan sistem, digunakan aplikasi *paint* dalam menggambarkan perancangan antarmuka pada penelitian ini.

Pada penelitian ini, perancangan antarmuka dibagi menjadi beberapa bagian sesuai dengan menu yang ditampilkan yaitu :

#### 4.2.1 Halaman Beranda

Fungsi halaman beranda adalah untuk menampilkan tampilan awal aplikasi. Pada menu ini terdapat informasi umum mengenai aplikasi dengan menerapkan algoritma genetika pada distribusi barang dua tahap. Gambar 4.5 menunjukkan perancangan halaman beranda dari sistem.



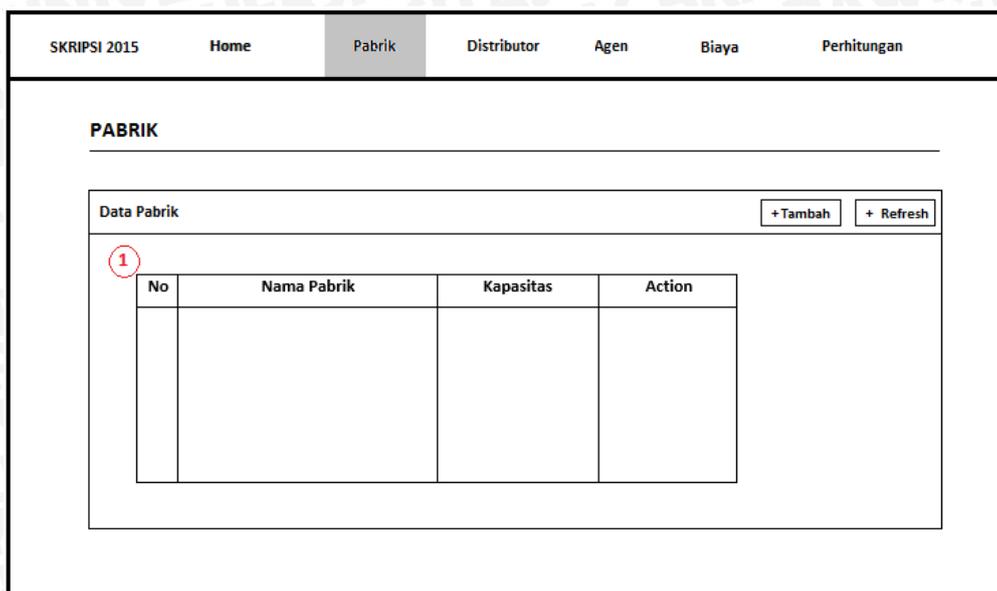
**Gambar 4.5** Perancangan Antarmuka Halaman Beranda

Keterangan gambar :

1. Merupakan tab-tab yang akan digunakan pada penerapan algoritma genetika berupa tab *home* atau halaman beranda, tab pabrik yang merupakan tab untuk mengisi data pabrik beserta kapasitas persediaan, tab distributor dan agen berisi data distributor dan agen beserta kapasitas permintaan, tab perhitungan yang berisi *inputan* untuk algoritma genetika.
2. Berisi gambar dan keterangan mengenai distribusi barang dua tahap.

#### 4.2.2 Menu Pabrik

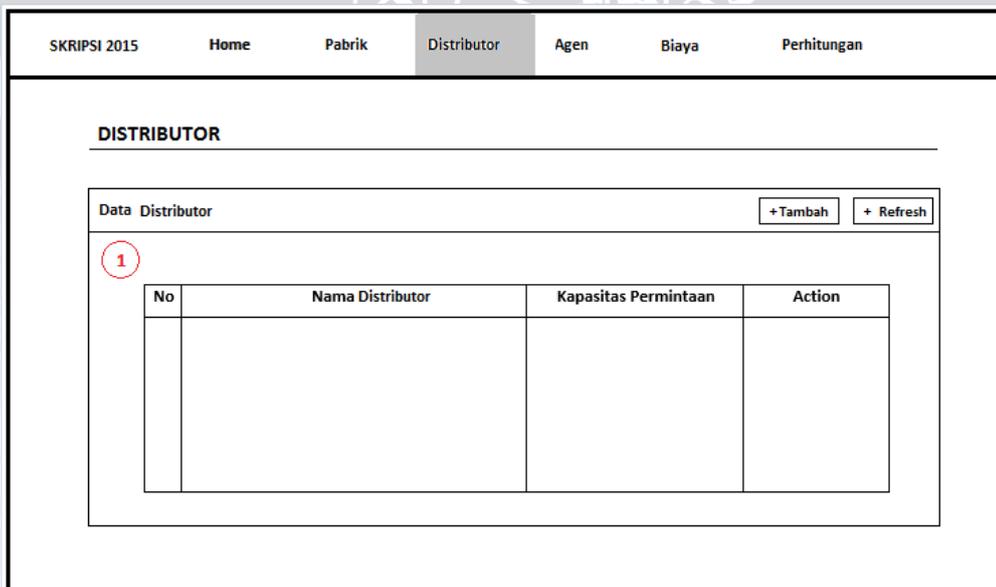
Menu pabrik digunakan untuk menampilkan data pabrik dan kapasitas persediaan dari pabrik dalam bentuk tabel. Dalam menu pabrik terdapat *action* berupa tambah data pabrik dan *edit* data pabrik. Tampilan menu pabrik ditunjukkan pada Gambar 4.6.



**Gambar 4.6** Perancangan Antarmuka Halaman Pabrik

### 4.2.3 Manu Distributor

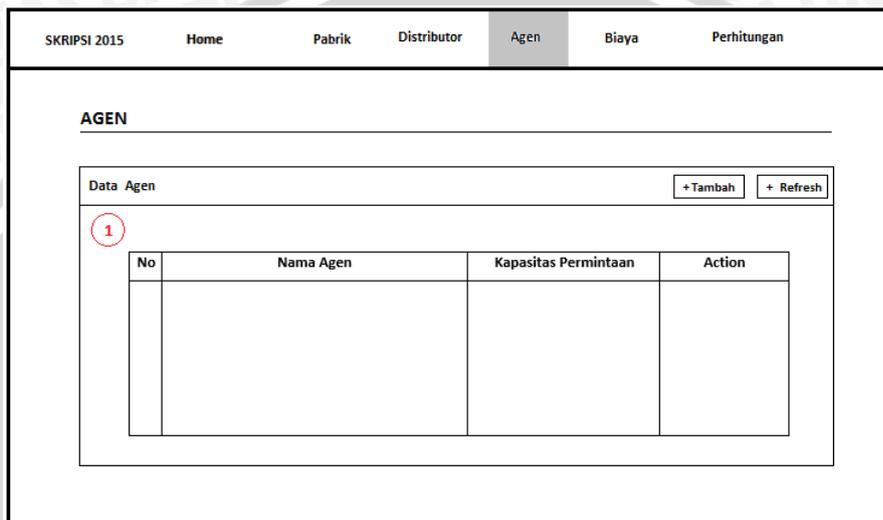
Menu distributor digunakan untuk menampilkan data distributor dan kapasitas permintaan dalam bentuk tabel. Terdapat *action* berupa tambah data distributor dan *edit* data distributor. Tampilan perancangan antarmuka untuk menu distributor ditunjukkan pada Gambar 4.7.



**Gambar 4.7** Perancangan Antarmuka Halaman Distributor

#### 4.2.4 Menu Agen.

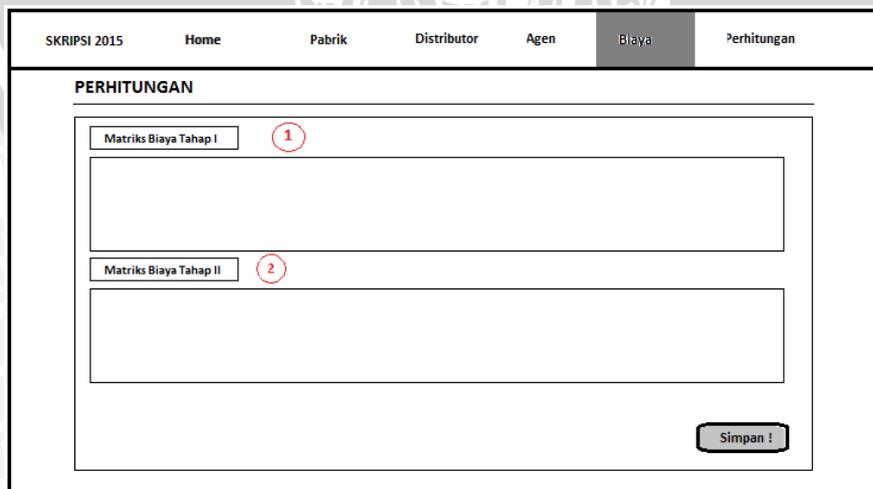
Menu agen digunakan untuk menampilkan data agen dan kapasitas permintaan dalam bentuk tabel. Tampilan perancangan antarmuka untuk menu agen ditunjukkan pada Gambar 4.8.



Gambar 4.8 Perancangan Antarmuka Halaman Agen

#### 4.2.5 Menu Biaya Transportasi

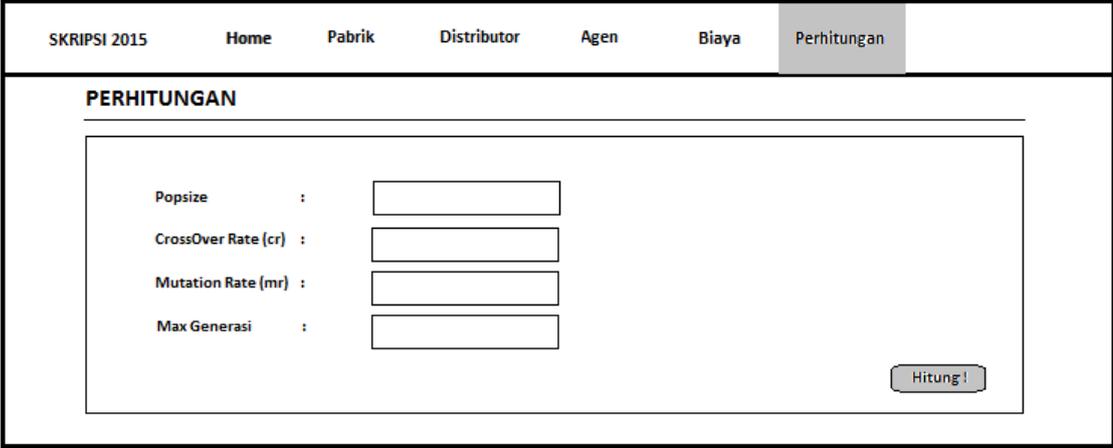
Menu biaya transportasi digunakan untuk menunjukkan biaya pada distribusi tahap 1 dan tahap 2. Bentuk perancangan biaya transportasi ditunjukkan pada gambar 4.9.



Gambar 4.9 Biaya Transportasi

#### 4.2.6 Menu Perhitungan

Setelah proses *input* data nama lokasi beserta kapasitas barang, maka akan dilakukan perhitungan manual. Menu perhitungan merupakan menu yang berisi *inputan* utama untuk parameter-parameter genetika berupa *inputan* jumlah populasi, *crossover rate*, *mutation rate* dan banyaknya generasi. Bentuk perancangan halaman antarmuka perhitungan ditunjukkan pada Gambar 4.10.



SKRIPSI 2015	Home	Pabrik	Distributor	Agen	Biaya	Perhitungan
--------------	------	--------	-------------	------	-------	-------------

**PERHITUNGAN**

Popsize	:	<input type="text"/>
CrossOver Rate (cr)	:	<input type="text"/>
Mutation Rate (mr)	:	<input type="text"/>
Max Generasi	:	<input type="text"/>

Hitung!

**Gambar 4.10** Perancangan Antarmuka Halaman Perhitungan

Setelah melakukan proses *inputan* parameter untuk algoritma genetika, selanjutnya dilakukan proses perhitungan dari data *inputan*. Gambar 4.11 menunjukkan rancangan antarmuka untuk perhitungan menggunakan algoritma genetika yang akan menampilkan hasil akhir berupa jalur distribusi optimum untuk distribusi barang dua tahap.

SKRIPSI 2015	Home	Pabrik	Distributor	Agen	Biaya	Perhitungan
--------------	------	--------	-------------	------	-------	-------------

HASIL PERHITUNGAN	
1	<input type="text"/>
2	<input type="text"/>
3	<input type="text"/>
4	<input type="text"/>
5	<input type="text"/>
6	<u>JALUR DISTRIBUSI OPTIMAL</u>

**Gambar 4.11** Perancangan Antarmuka Halaman Hasil Perhitungan

Keterangan gambar :

1. Menampilkan *input* parameter algoritma genetika yang sebelumnya ditunjukkan pada gambar 4.11 pada panel *inputan*
2. Menampilkan populasi awal dari distribusi tahap 1 dan tahap dua beserta nilai *fitness*.
3. Menampilkan individu hasil reproduksi dari proses *crossover* dan proses mutasi.
4. Menampilkan semua populasi baik dari populasi awal maupun individu baru hasil dari proses reproduksi. Setelah itu ditampilkan hasil sorting dari individu yang memiliki nilai *fitness* terbesar sampai individu yang memiliki nilai *fitness* terkecil.

5. Menampilkan individu baru yang akan digunakan pada proses generasi selanjutnya.

Menampilkan jalur distribusi optimum untuk distribusi barang dua tahap beserta total biaya minimum distribusi

### 4.3 Perancangan Pengujian

Perancangan pengujian pada sistem optimasi distribusi barang dua tahap menggunakan algoritma genetika dilakukan dengan mencoba berbagai parameter. Parameter yang akan diuji berupa ukuran *crossover rate* (cr), ukuran *mutation rate* (mr), jumlah generasi dan perancangan pengujian untuk mutasi per segmen. Pengujian ini bertujuan untuk mengetahui pengaruh parameter pada algoritma genetika terhadap optimasi jalur distribusi barang dua tahap berdasarkan nilai *fitness*-nya.

#### 4.3.1 Pengujian Ukuran Populasi

Perancangan pengujian ukuran populasi digunakan untuk mengetahui mengetahui kebaikan solusi berdasarkan jumlah ukuran populasi yang optimal. Pada pengujian ini digunakan  $Cr= 0,4$   $mr=0,6$  dan jumlah generasi sebanyak 50 generasi. Rancangan pengujian akan dilakukan sebanyak 10 kali percobaan yang akan diambil rata-rata nilai *fitness*-nya. Rancangan pengujian terhadap ukuran populasi ditunjukkan pada Tabel 4.1.

**Tabel 4.1** Pengujian Ukuran Populasi

Popsize	Percobaan ke-										Rata-rata <i>fitness</i>
	1	2	3	4	5	6	7	8	9	10	
20											
40											
60											
80											
100											

#### 4.3.2 Perancangan Pengujian *Crossover Rate* (cr) dan *Mutation Rate* (mr)

Perancangan pengujian pada subbab 4.3.2 dilakukan terhadap kombinasi parameter *crossover rate* dan *mutation rate* berdasarkan ukuran populasi

yang paling optimal yang ditunjukkan pada pengujian pada subbab 4.3.1. Pengujian ini dilakukan untuk mengetahui pengaruh kombinasi *crossover rate* dan *mutation rate* terhadap nilai *fitness*. Pada pengujian ini akan dilakukan uji coba sebanyak 10 kali dengan parameter yang ditentukan. Rancangan pengujian terhadap kombinasi *crossover rate* dan *mutation rate* ditunjukkan pada Tabel 4.2.

**Tabel 4.2** Perancangan Pengujian kombinasi *Crossover rate* (*cr*) dan *mutation rate* (*mr*)

Ukuran		Percobaan ke-										Rata – rata Nilai <i>Fitness</i>
<i>cr</i>	<i>mr</i>	1	2	3	4	5	6	7	8	9	10	
0,1	0,9											
0,2	0,8											
0,3	0,7											
0,4	0,6											
..	..											
0,9	0,1											

#### 4.3.3 Perancangan Pengujian Jumlah Generasi

Perancangan pengujian pada subbab 4.3.3 dilakukan terhadap jumlah generasi dengan menggunakan ukuran populasi terbaik pada subbab 4.3.1 dan ukuran kombinasi *crossover rate* dan *mutation rate* terbaik yang diperoleh dari rancangan pengujian subbab 4.3.2. Pengujian jumlah generasi dilakukan untuk mengetahui nilai *fitness* terbaik terhadap jumlah generasi. Rancangan pengujian terhadap jumlah generasi ditunjukkan pada Tabel 4.3.

**Tabel 4.3** Perancangan Pengujian Jumlah Generasi

Banyak Generasi	Percobaan ke-										Rata – rata Nilai <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
20											
40											
60											
80											
100											

#### 4.3.4 Perancangan Pengujian Jenis Segmen Mutasi

Perancangan pengujian jenis segmen dilakukan terhadap jenis segmen mutasi dengan ukuran populasi terbaik pada subbab 4.3.1, kombinasi *crossover* dan *mutation rate* pada subbab 4.3.2 dan jumlah generasi terbaik pada subbab 4.3.3 untuk mendapatkan nilai *fitness* paling optimal. Pada pengujian ini akan dilakukan dengan perbandingan terhadap penggunaan segmen mutasi. Rancangan pengujian terhadap jenis mutasi ditunjukkan pada tabel 4.4.

Tabel 4.4 Perancangan Pengujian Jenis Segmen Mutasi

Segmen			Percobaan ke-										Rata – rata Nilai <i>Fitness</i>
1,2	1	2	1	2	3	4	5	6	7	8	9	10	
100%	0	0											
0	100%	0											
0	0	100%											
40%	30%	30%											

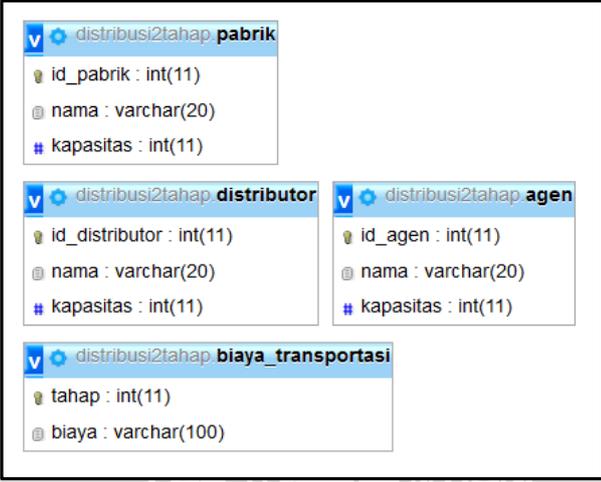
## BAB V

### IMPLEMENTASI

Pada bab ini implementasi dijelaskan mengenai implementasi dari rancangan sistem yang telah dibahas pada bab sebelumnya. Dalam bab ini akan menjelaskan mengenai implementasi tabel database, implementasi kode program dan hasil dari implementasi antarmuka pada penerapan algoritma genetika untuk optimasi distribusi barang dua tahap.

#### 5.1 Implementasi Tabel Database

Implementasi tabel database pada penerapan algoritma genetika untuk permasalahan distribusi barang dua tahap ditunjukkan pada gambar 5.1.



Tabel	Field	Tipe
distribusi2tahap.pabrik	id_pabrik	int(11)
	nama	varchar(20)
	kapasitas	int(11)
distribusi2tahap.distributor	id_distributor	int(11)
	nama	varchar(20)
	kapasitas	int(11)
distribusi2tahap.agen	id_agen	int(11)
	nama	varchar(20)
	kapasitas	int(11)
distribusi2tahap.biaya_transportasi	tahap	int(11)
	biaya	varchar(100)

Gambar 5.1 Implementasi Tabel dalam Sistem.

#### 5.3 Implementasi Kode Program

Implementasi kode program untuk penerapan algoritma genetika pada permasalahan optimasi distribusi barang dua tahap disusun berdasarkan metodologi penelitian pada bab III dan perancangan sistem pada bab IV. Pada subbab ini akan ditampilkan kode program sesuai dengan aplikasi yang telah dibuat.

##### 5.3.1 Implementasi Inisialisasi Populasi Awal

Inisialisasi populasi awal terdiri dari kromosom sejumlah *popsiz*e yang ditentukan. Panjang kromosom ditentukan oleh banyaknya *inputan* pabrik dikalikan jumlah distributor untuk kromosom distribusi tahap pertama, untuk tahap kedua

diperoleh dari banyaknya distributor dikali jumlah agen. Implementasi inialisasi populasi awal dilakukan proses untuk mendapatkan jumlah pabrik, distributor dan agen, kemudian menghitung banyak tahap pada distribusi. Proses implementasi untuk mengetahui panjang kromosom dijelaskan pada kode program 5.1.

```

1 // Variabel inputan user
2 $data['post'] = $this->input->post();
3 $data['post']['biayaTransportasi1'] = json_decode($this->mdb-
4 >getBiayaPerTahap(1));
5 $data['post']['biayaTransportasi2'] = json_decode($this->mdb-
6 >getBiayaPerTahap(2));
7 //Mendapatkan Pabrik
8 $data['pabrik'] = $this->mdb->getPabrik();
9 $jmlhPabrik = count($data['pabrik']);
10 // Mendapatkan Distributor
11 $data['distributor'] = $this->mdb->getDistributor();
12 $jmlhDistributor = count($data['distributor']);
13 // Mendapatkan Agen
14 $data['agen'] = $this->mdb->getAgen();
15 $jmlhAgen = count($data['agen']);
16 // Menghitung banyak tahap
17 $data['tahap'][0] = $jmlhPabrik * $jmlhDistributor;
18 $data['tahap'][1] = $jmlhDistributor * $jmlhAgen;
19 $jmlhTahap = count($data['tahap']);

```

#### Kode Program 5.1 Membentuk Individu

Setelah mendapatkan panjang kromosom maka dilakukan random terhadap kromosom untuk menentukan prioritas distribusi. Kromosom dibangkitkan sesuai dengan jumlah popsize yang telah ditentukan. Kode program untuk melakukan random terhadap kromosom ditunjukkan pada kode program 5.2.

```

1 for ($parent = 0; $parent < $data['post']['popsize']; $parent++) {
2     for ($thp = 0; $thp < $jmlhTahap; $thp++) {
3         // Membuat bilangan random
4         $data['parent'][0][$parent][$thp] = $this-
5 >UniqueRandomNumbersWithinRange(1,
6     $data['tahap'][$thp], $data['tahap'][$thp]+1);
7     }
8
9     $data['parent'][0][$parent]['indexName'] = 'parent';
10    $data['parent'][0][$parent]['indexNumber'] = $parent;
11 }
12
13 // Pengulangan sebanyak jumlah maxgenerasi
14 for ($generasi = 0; $generasi < $data['post']['maxgenerasi'];
15     $generasi++) {
16     $posisi = 0;
17 }

```

#### Kode Program 5.2 Pembangkitan Populasi Awal

Pada inialisasi populasi awal, sistem akan melakukan proses pembuatan individu sebanyak *popsiz* yang telah diinputkan. Kromosom yang telah diacak akan dimasukkan pada sebuah *array* yang selanjutnya akan dihitung nilai *fitness*-nya.

### 5.3.2 Implementasi Perhitungan Nilai *Fitness*

Proses implementasi perhitungan nilai *fitness* pada sistem diperoleh dari kapasitas persediaan pabrik, jumlah permintaan dari distributor dan agen serta biaya distribusi dari setiap tahap. Prioritas distribusi disesuaikan dengan kromosom yang telah dirandom. Prioritas distribusi adalah prioritas untuk meletakkan unit barang pada setiap lokasi yang nantinya akan dikalikan dengan biaya distribusi untuk perhitungan *fitness*. Kode program untuk perhitungan nilai *fitness* ditunjukkan pada kode program 5.3.

```

1 // Menghitung nilai fitness dari representasi kromosom (parent)
2 for ($parent = 0; $parent < $data['post']['popsiz']; $parent++) {
3     // Mencari distribusi dan biaya rute per tahap
4     $data['biayaRute'][$posisi] = 0;
5
6     for ($thp = 0; $thp < $jmlhTahap; $thp++) {
7         if ($thp == 0) {
8             $pembagi = $jmlhDistributor;
9             $batasOrdinatDistribusi = $jmlhPabrik;
10            $kapasitas = $data['distributor'];
11            $persediaan = $data['pabrik'];
12            $biayaTransportasi =
13            $data['post']['biayaTransportasi1'];
14        }
15        elseif ($thp == 1) {
16            $pembagi = $jmlhAgen;
17            $batasOrdinatDistribusi = $jmlhDistributor;
18            $kapasitas = $data['agen'];
19            $persediaan = $data['distributor'];
20            $biayaTransportasi =
21            $data['post']['biayaTransportasi2'];
22        }
23
24        for ($kromosom = 0; $kromosom < $data['tahap'][$thp];
25            $kromosom++) {
26            $nilaiKromosom =
27            $data['parent'][$generasi][$parent][$thp][$kromosom];
28
29            $hasilBagi = floor(($nilaiKromosom / $pembagi));
30            $sisabagi = $nilaiKromosom % $pembagi;
31
32            if ($sisabagi == 0) {
33                $y = $hasilBagi-1;
34                $x = $pembagi-1;

```

```

35     }
36     else {
37         $y = $hasilBagi;
38         $x = $sisabagi-1;
39     }
40
41     // Menghitung jumlah per kolom
42     $jumlah = 0;
43     for ($ordinat = 0; $ordinat < $batasOrdinatDistribusi;
44         $ordinat++) {
45         $jumlah = $jumlah +
46         @$data['distribusi'][$generasi][$posisi][$thp][$or
47         dinat][$x];
48     }
49     $kurang = $kapasitas[$x]['kapasitas'] - $jumlah;
50
51     if ($kurang > $persediaan[$y]['kapasitas'])
52     {$data['distribusi'][$generasi][$posisi][$thp][$y][
53     $x] = $persediaan[$y]['kapasitas'];
54     $persediaan[$y]['kapasitas'] = 0;
55     }
56
57     Else {
58         $data['distribusi'][$generasi][$posisi][$thp][$y][$
59     x] = $kurang;
60         $persediaan[$y]['kapasitas'] =
61         $persediaan[$y]['kapasitas'] - $kurang;
62     }
63
64     $data['biayaRute'][$posisi] =
65     $data['biayaRute'][$posisi] +
66     ($data['distribusi'][$generasi][$posisi][$thp][$y][$x]
67     * $biayaTransportasi[$y][$x]);
68 }
69 }
70 }
71 $data['parent'][$generasi][$parent]['totalBiaya'] =
72 $data['biayaRute'][$posisi];
73
74 $data['parent'][$generasi][$parent]['fitness'] = round(10000 /
75 ($data['biayaRute'][$posisi]), 5, PHP_ROUND_HALF_UP);
76 $posisi++;
77 }
78 $data['parent'][$generasi][$parent]['fitness'] = round(10000 /
79 ($data['biayaRute'][$posisi]), 5, PHP_ROUND_HALF_UP);
80 $posisi++;
81 }

```

**Kode Program 5.3** Implementasi Perhitungan Nilai *Fitness*

Kode program dimulai dengan menentukan letak distribusi barang pada tahap 1 untuk pabrik ke distributor sesuai dengan prioritas gen, begitu pula untuk tahap 2 untuk distribusi dari distributor ke agen. Setelah ditentukan posisi distribusi barang dari masing-masing tahap dihitung biaya sesuai dengan biaya dari masing-

masing lokasi distribusi yang kemudian dijumlahkan pertahapnya. Baris 68 dan 69 menunjukkan rumus perhitungan nilai *fitness* yaitu  $10000/(\text{biaya tahap1}+\text{biaya tahap2})$ .

### 5.3.3 Implementasi Crossover

Pada proses reproduksi dengan *crossover*, digunakan metode *one cut point crossover*. Proses *crossover* dimulai dengan menentukan dua kromosom yang akan menjadi induk pada persilangan. Setiap proses persilangan ini akan menghasilkan satu anak, proses persilangan akan diulang-ulang sampai jumlah *child* sesuai dengan jumlah *crossover rate* dikalikan *popsi*. Kode program proses *crossover* ditunjukkan pada kode program 5.4.

```

1 // Menghitung crossover
2 $data['jmlhChildCrossover'] = round($data['post']['cr'] *
3 $data['post']['popsi']);
4 $data['jmlhCrossover'] = $data['jmlhChildCrossover'] * 2;
5
6 // Inisialisasi awal
7 $isRandom = true;
8 $random[0] = rand(0, $data['post']['popsi']-1);
9 $cr = 1;
10
11 while ($isRandom == true) {
12     $angkaRandom = rand(0, $data['post']['popsi']-1);
13
14     if ($angkaRandom != $random[$cr-1]) {
15         $random[$cr] = $angkaRandom;
16         $cr++;
17     }
18
19     if ($cr == $data['jmlhCrossover']) {
20         $isRandom = false;
21     }
22 }
23 foreach ($random as $index => $value) {
24     $data['crossover'][$generasi][$index] =
25     $data['parent'][$generasi][$value];
26 }
27
28 // Mendapatkan random pemilihan gen untuk kromosom crossover
29
30 for ($ch = 0; $ch < $data['jmlhChildCrossover']; $ch++) {
31     for ($thp = 0; $thp < $jmlhTahap; $thp++) {
32         $data['randGenCrossover'][$generasi][$ch][$thp] = rand(1,
33         $data['tahap'][$thp]-2);
34     }
35 }
36

```

```

37 // Menghasilkan crossover
38 // ch = child, btsCr = batas crossover, cr = crossover
39 for ($ch = 0; $ch < $data['jmlhChildCrossover']; $ch++) {
40     for ($thp = 0; $thp < $jmlhTahap; $thp++) {
41         $btsCr = $ch * 2;
42
43         $kromosomChild = 0;
44
45         // Memasukkan random gen dari kromosom crossover ke child
46         $data['child'][$generasi][$ch][$thp][0] = 0;
47         for ($scr = $btsCr; $scr < $btsCr+2; $scr++) {
48             for ($kromosom = 0; $kromosom <
49                 $data['randGenCrossover'][$generasi][$ch][$thp
50                     ]; $kromosom++) {
51                 if(!in_array($data['crossover'][$generasi][$scr][
52                     $thp][$kromosom],
53                     $data['child'][$generasi][$ch][$thp]))
54                     {$data['child'][$generasi][$ch][$thp][$kromosomChild] =
55                         $data['crossover'][$generasi][$scr][$thp][$kromosom];
56                         $kromosomChild++;
57                     }
58                 }
59             }
60         }
61     }
62     // Memasukkan gen dari kromosom crossover ke child yang
63     // bukan random gen
64     $scr--;
65     for ($kromosom =
66         $data['randGenCrossover'][$generasi][$ch][$thp];
67         $kromosom < $data['tahap'][$thp]; $kromosom++) {
68         If(!in_array($data['crossover'][$generasi][$scr][$thp]
69             [$kromosom],
70             $data['child'][$generasi][$ch][$thp]))
71             {$data['child'][$generasi][$ch][$thp][$kromosomChild] =
72                 $data['crossover'][$generasi][$scr][$thp][$kromosom];
73             }
74         }
75     }
76     $kromosomChild++;
77 }
78 }
79 }
80 }
81 // Memberi label child
82 $data['child'][$generasi][$ch]['indexName'] = 'child';
83 $data['child'][$generasi][$ch]['indexNumber'] = $ch;
84
85 } //end for menghasilkan crossover

```

#### Kode Program 5.4 Implementasi Proses Crossover

Dari kode program 5.4 maka penjelasan proses *crossover* adalah :

1. Baris 1-4 menghitung banyaknya individu hasil *crossover* dengan mengalikan *cr* dengan *popsize*.

2. Baris 6-26 digunakan untuk menentukan individu yang digunakan pada proses *crossover*, dengan melakukan random kromosom.
3. Baris 30-36 digunakan untuk menentukan random gen yang akan digunakan untuk titik potong *crossover*.
4. Baris 39-62 digunakan untuk menghasilkan individu hasil *crossover*.
6. Baris 63-80 digunakan untuk memasukkan gen hasil random pada *child*.
7. Memberi label *child* pada individu baru hasil *crossover*

### 5.3.3 Implementasi Mutasi

Teknik mutasi yang digunakan dalam penelitian ini menggunakan metode *exchange mutation*. Pada implementasinya, teknik mutasi dimulai dengan memasukkan parameter *mutation rate* (*mr*). Jumlah individu hasil reproduksi dengan mutasi adalah sejumlah *popsi* dikalikan dengan *mutation rate*. Reproduksi dengan mutasi menggunakan 1 individu untuk menjadi *parent*, yang akan menghasilkan 1 anak. Penerapan proses reproduksi dengan mutasi ditunjukkan pada kode program 5.5.

```

1 // Menghitung mutasi
2 $data['jmlhMutasi'] = round($data['post']['mr'] *
3     $data['post']['popsi']);
4
5 // Mendapatkan random anggota dari representasi kromosom untuk
6     mutasi
7 for ($mt = 0; $mt < $data['jmlhMutasi']; $mt++) {
8     $data['mutasi'][$generasi][$mt] =
9     $data['parent'][$generasi][rand(0,
10        $data['post']['popsi']-1)];
11 }
12
13 // Mendapatkan random pemilihan gen untuk kromosom mutasi
14 // Jika mutasi campuran
15 if (@$data['post']['mutasiCamp'] != '') {
16     $mtsThp12 = round($data['post']['mtsThp12']
17         $data['jmlhMutasi']);
18     $mtsThp1 = round($data['post']['mtsThp1'] *
19         $data['jmlhMutasi']);
20     $mtsThp2 = $data['jmlhMutasi'] - $mtsThp12 - $mtsThp1;
21
22     for ($parentCamp = 0; $parentCamp < $mtsThp12; $parentCamp++){
23         for ($thp = 0; $thp < $jmlhTahap; $thp++){
24             $data['randGenMutasi'][$generasi][$parentCamp][$thp] =
25             $this->UniqueRandomNumbersWithinRange(0,
26                 $data['tahap'][$thp]-1, 2);
27         }

```

```
28 }
29 // untuk mutasi tahap 1
30 for ($parentThp1 = $parentCamp;
31     $parentThp1 < $parentCamp + $mtsThp1; $parentThp1++)
32     {$data['randGenMutasi'][$generasi][$parentThp1][0] =
33     $this->UniqueRandomNumbersWithinRange(0,
34     $data['tahap'][0]-1, 2);
35     }
36 // untuk mutasi tahap 2
37 for ($parentThp2 = $parentThp1; $parentThp2 <
38     $parentThp1 + $mtsThp2; $parentThp2++) {
39     $data['randGenMutasi'][$generasi][$parentThp2][1] = $this-
40     >UniqueRandomNumbersWithinRange(1, $data['tahap'][1]-1,
41     2);
42     }
43 } end if mutasi campuran
44
45 else { // Jika bukan mutasi campuran
46     for ($parent = 0; $parent < $data['jmlhMutasi']; $parent++) {
47         foreach ($data['post']['mutasi'] as $index => $thp) {
48             $data['randGenMutasi'][$generasi][$parent][$thp]=
49             $this->UniqueRandomNumbersWithinRange(0,
50             $data['tahap'][$thp]-1, 2);
51         }
52     }
53 }
54
55 // Menghasilkan mutasi
56 if (@$data['post']['mutasiCamp'] != '') {
57     // untuk mutasi campuran (tahap 1 dan 2)
58     for ($parentCamp = 0; $parentCamp < $mtsThp12;
59         $parentCamp++) {
60         $data['child'][$generasi][$ch]=
61         $data['mutasi'][$generasi][$parentCamp];
62
63         for ($thp = 0; $thp < $jmlhTahap; $thp++) {
64             $tempKromosom =
65             $data['child'][$generasi][$ch][$thp][$data['randGen
66             Mutasi'][$generasi][$parentCamp][$thp][0]];
67             $data['child'][$generasi][$ch][$thp][$data['randGen
68             Mutasi'][$generasi][$parentCamp][$thp][0]] =
69             $data['child'][$generasi][$ch][$thp][$data['randGen
70             Mutasi'][$generasi][$parentCamp][$thp][1]];
71             $data['child'][$generasi][$ch][$thp][$data['randGen
72             Mutasi'][$generasi][$parentCamp][$thp][1]] =
73             $tempKromosom;
74         }
75
76         $data['child'][$generasi][$ch]['indexName'] = 'child';
77         $data['child'][$generasi][$ch]['indexNumber'] = $ch;
78         $ch++;
79     }
80 }
81 // untuk mutasi tahap 1
82 for ($parentThp1 = $parentCamp; $parentThp1 <
83     $parentCamp + $mtsThp1; $parentThp1++)
84 }
```

```

85     {$data['child'][$generasi][$ch] =
86     $data['mutasi'][$generasi][$parentThp1];
87
88     $tempKromosom =
89     $data['child'][$generasi][$ch][0][$data['randGenMutasi'
90     ][$generasi][$parentThp1][0][0]];
91
92     $data['child'][$generasi][$ch][0][$data['randGenMutasi'
93     ][$generasi][$parentThp1][0][0]] =
94     $data['child'][$generasi][$ch][0][$data['randGenMutasi'
95     ][$generasi][$parentThp1][0][1]];
96
97     $data['child'][$generasi][$ch][0][$data['randGenMutasi'
98     ][$generasi][$parentThp1][0][1]] = $tempKromosom;
99
100    $data['child'][$generasi][$ch]['indexName'] = 'child';
101    $data['child'][$generasi][$ch]['indexNumber'] = $ch;
102    $ch++;
103  }
104  // untuk mutasi tahap 2
105  for ($parentThp2 = $parentThp1; $parentThp2 <
106    $parentThp1+$mtsThp2; $parentThp2++)
107    {$data['child'][$generasi][$ch] =
108    $data['mutasi'][$generasi][$parentThp2];
109
110    $tempKromosom =
111    $data['child'][$generasi][$ch][1][$data['randGenMutasi'
112    ][$generasi][$parentThp2][1][0]];
113
114    $data['child'][$generasi][$ch][1][$data['randGenMutasi'
115    ][$generasi][$parentThp2][1][0]] =
116    $data['child'][$generasi][$ch][1][$data['randGenMutasi'
117    ][$generasi][$parentThp2][1][1]];
118
119    $data['child'][$generasi][$ch][1][$data['randGenMutasi'
120    ][$generasi][$parentThp2][1][1]] = $tempKromosom;
121
122    $data['child'][$generasi][$ch]['indexName'] = 'child';
123    $data['child'][$generasi][$ch]['indexNumber'] = $ch;
124    $ch++;
125  } //end for mutasi tahap 2
126 } //end if menghasilkan mutasi
127 else {
128   for ($mt = 0; $mt < $data['jmlhMutasi']; $mt++) {
129     $data['child'][$generasi][$ch] =
130     $data['mutasi'][$generasi][$mt];
131
132     foreach ($data['post']['mutasi'] as $index => $thp) {
133       $tempKromosom =
134       $data['child'][$generasi][$ch][$thp][$data['randGenMu
135       tasi'][$generasi][$mt][$thp][0]];
136
137       $data['child'][$generasi][$ch][$thp][$data['randGenMu
138       tasi'][$generasi][$mt][$thp][0]] =
139       $data['child'][$generasi][$ch][$thp][$data['randGenMu
140       tasi'][$generasi][$mt][$thp][1]];

```

```

141
142     $data['child'][$generasi][$ch][$thp][$data['randGenMu
143     tasi'][$generasi][$mt][$thp][1]] = $tempKromosom;
144 }
145
146     $data['child'][$generasi][$ch]['indexName'] = 'child';
147     $data['child'][$generasi][$ch]['indexNumber'] = $ch;
148     $ch++;
149 } end for
150 } end else if

```

### Kode Program 5.5 Implementasi dengan Mutasi

Adapun penjelasan kode program 5.5 mengenai implementasi teknik mutasi adalah sebagai berikut :

1. Baris 1-3 digunakan untuk menghitung jumlah mutasi yang harus dilakukan.
2. Baris 5-11 digunakan untuk mendapatkan random anggota pada populasi awal untuk menjadi *parent* pada mutasi.
3. Baris 13-43 digunakan untuk mendapatkan random gen pada kromosom sebagai titik tukar pada mutasi untuk mutasi tahap 1 dan 2 bukan campuran
4. Baris 43-53 digunakan untuk mendapatkan random gen pada kromosom sebagai titik tukar pada mutasi untuk mutasi tahap 1 dan 2 untuk mutasi campuran
5. Baris 55-80 digunakan untuk menghasilkan individu hasil dari proses mutasi campuran.
6. Baris 81-103 digunakan untuk menghasilkan individu hasil dari proses mutasi tahap 1.
7. Baris 104-25 digunakan untuk menghasilkan individu hasil dari proses mutasi tahap 2.
8. Baris 104-150 digunakan untuk menghasilkan individu hasil dari proses mutasi tahap 1 dan 2.

### 5.3.5 Implementasi Proses Seleksi

Implementasi proses seleksi pada penelitian ini digunakan teknik *elitism selection*. Seleksi digunakan untuk memilih individu dari *parent* dan hasil reproduksi untuk digunakan pada proses selanjutnya. Implementasi pada proses seleksi ditunjukkan pada kode program 5.6.

```

1 // Sorting dari besar ke kecil
2 $data['seleksi'][$generasi] = $data['preSeleksi'][$generasi] =
3 array_merge($data['parent'][$generasi],
4 $data['child'][$generasi]);
5 for ($p = 0; $p < $posisi; $p++) {

```

```

6      for ($p2 = $p+1; $p2 < $posisi; $p2++) {
7          if ($data['seleksi'][$generasi][$p]['fitness'] <
8              $data['seleksi'][$generasi][$p2]['fitness']) {
9              $tempPreSeleksi = $data['seleksi'][$generasi][$p];
10             $data['seleksi'][$generasi][$p] =
11             $data['seleksi'][$generasi][$p2];
12             $data['seleksi'][$generasi][$p2] = $tempPreSeleksi;
13
14             $tempDistribusi =
15             $data['distribusi'][$generasi][$p];
16             $data['distribusi'][$generasi][$p] =
17             $data['distribusi'][$generasi][$p2];
18             $data['distribusi'][$generasi][$p2] =
19             $tempDistribusi;
20         }
21     }
22 }

```

#### Kode Program 5.6 Proses Sorting pada Seleksi Elitism

Dari kode program 5.6 digunakan untuk proses seleksi *elitism* yaitu mengurutkan kromosom induk dan anak hasil dari proses reproduksi sesuai dengan nilai *fitness* yang terbesar hingga nilai terkecil. Setelah mengurutkan, maka diambil individu baru yang memiliki nilai *fitness* terbesar sesuai dengan jumlah populasi awal. Proses selanjutnya adalah menjadikan individu hasil seleksi sebagai individu baru untuk generasi berikutnya. Kode program 5.7 menunjukkan individu baru hasil seleksi *elitism*.

```

1      // Membuat individu baru
2      for ($parent = 0; $parent < $data['post']['popsize']; $parent++){
3          $data['parent'][$generasi+1][$parent] =
4          $data['seleksi'][$generasi][$parent];
5
6          $data['parent'][$generasi+1][$parent]['indexName'] = 'parent';
7
8          $data['parent'][$generasi+1][$parent]['indexNumber'] =
9          $parent;
10     }

```

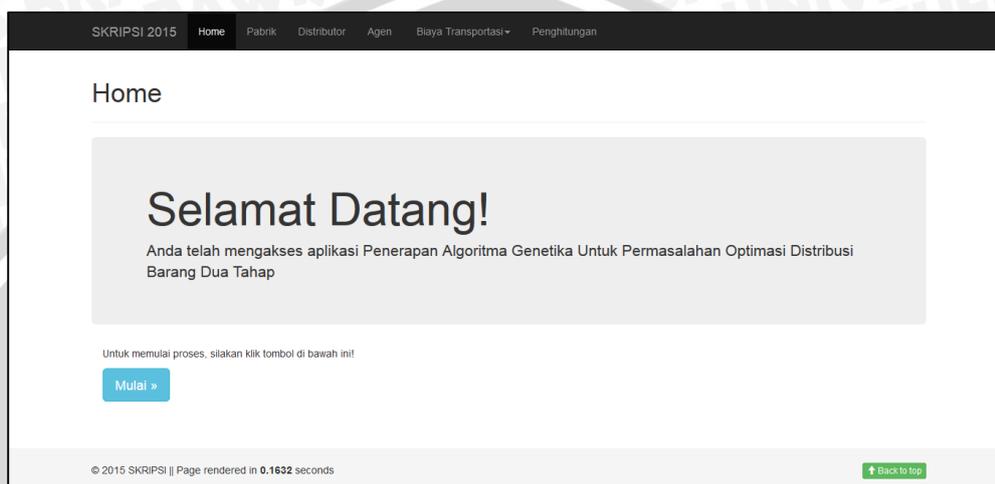
#### Kode Program 5.7 Individu Baru Hasil Seleksi *Elitism*

### 5.4 Implementasi Antar Muka

Berdasarkan penjelasan pada bab perancangan, setiap proses pada penerapan algoritma genetika untuk permasalahan optimasi distribusi barang dua tahap dapat dilihat pada tampilan antar muka sistem. Implementasi antarmuka pada sistem ini terdiri dari 6 bagian utama sebagai berikut :

## 1. Home

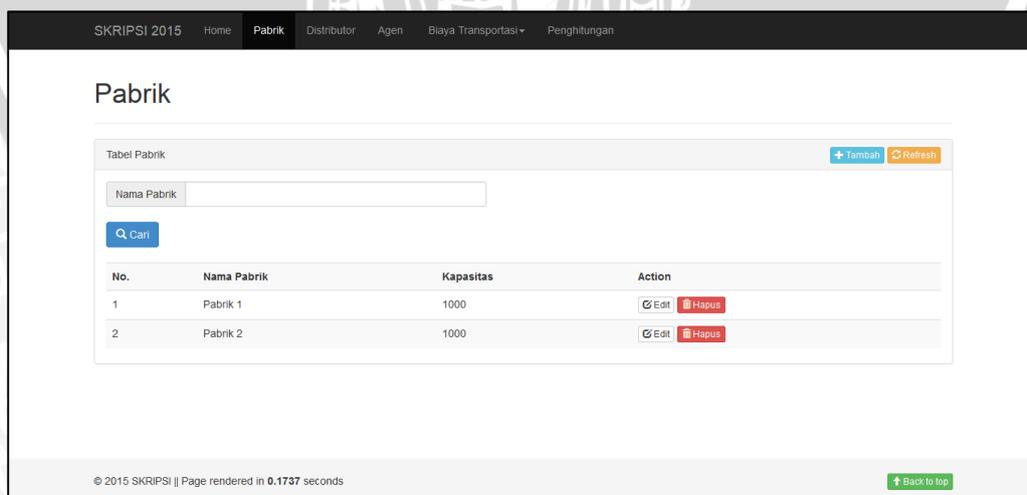
Pada halaman beranda atau *home* menampilkan informasi umum mengenai sistem dengan menerapkan algoritma genetika untuk permasalahan distribusi barang dua tahap. Halaman *home* ditunjukkan pada gambar 5.2.



**Gambar 5.2** Halaman *Home*

## 2. Pabrik

Halaman pabrik berisi mengenai informasi pabrik yang digunakan dalam obyek permasalahan. Terdiri dari kolom yang berisi informasi nama pabrik dan kapasitas persediaan pabrik. Halaman pabrik dan detailnya ditunjukkan pada gambar 5.3.



**Gambar 5.3** Implementasi Halaman Pabrik

### 3. Distributor

Halaman distributor berisi informasi mengenai distributor yang digunakan dalam saluran distribusi barang dua tahap. Dalam halaman ini terdapat kolom yang berisi mengenai informasi nama distributor dan kapasitas permintaan distributor. Pada antar muka distributor juga dapat dilakukan penambahan distributor, serta pada setiap baris terdapat tombol detail yang digunakan untuk melihat detail distributor. Terdapat *filter* tabel distributor yang berfungsi untuk masalah pencarian data berdasarkan nama distributor. Halaman distributor dan detailnya ditunjukkan pada gambar 5.4.

SKRIPSI 2015 Home Pabrik Distributor Agen Biaya Transportasi Penghitungan

## Distributor

Tabel Distributor [+ Tambah](#) [Refresh](#)

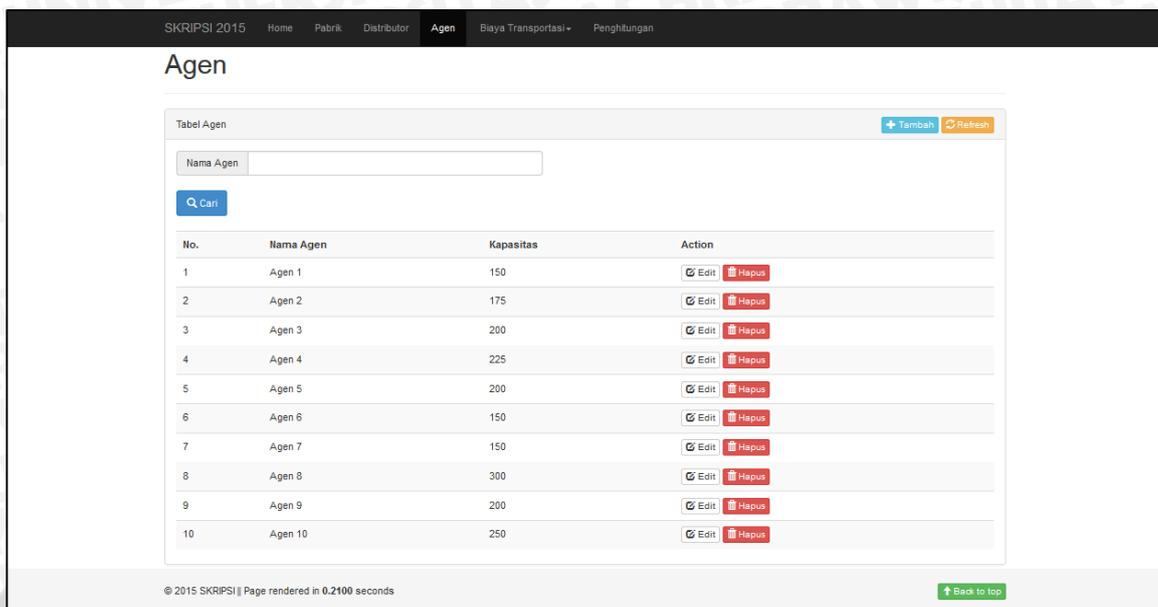
No.	Nama Distributor	Kapasitas	Action
1	Distributor 1	300	<a href="#">Edit</a> <a href="#">Hapus</a>
2	Distributor 2	400	<a href="#">Edit</a> <a href="#">Hapus</a>
3	Distributor 3	250	<a href="#">Edit</a> <a href="#">Hapus</a>
4	Distributor 4	600	<a href="#">Edit</a> <a href="#">Hapus</a>
5	Distributor 5	450	<a href="#">Edit</a> <a href="#">Hapus</a>

© 2015 SKRIPSI || Page rendered in 0.1720 seconds [Back to top](#)

**Gambar 5.4** Implementasi Halaman Distributor

### 4. Agen

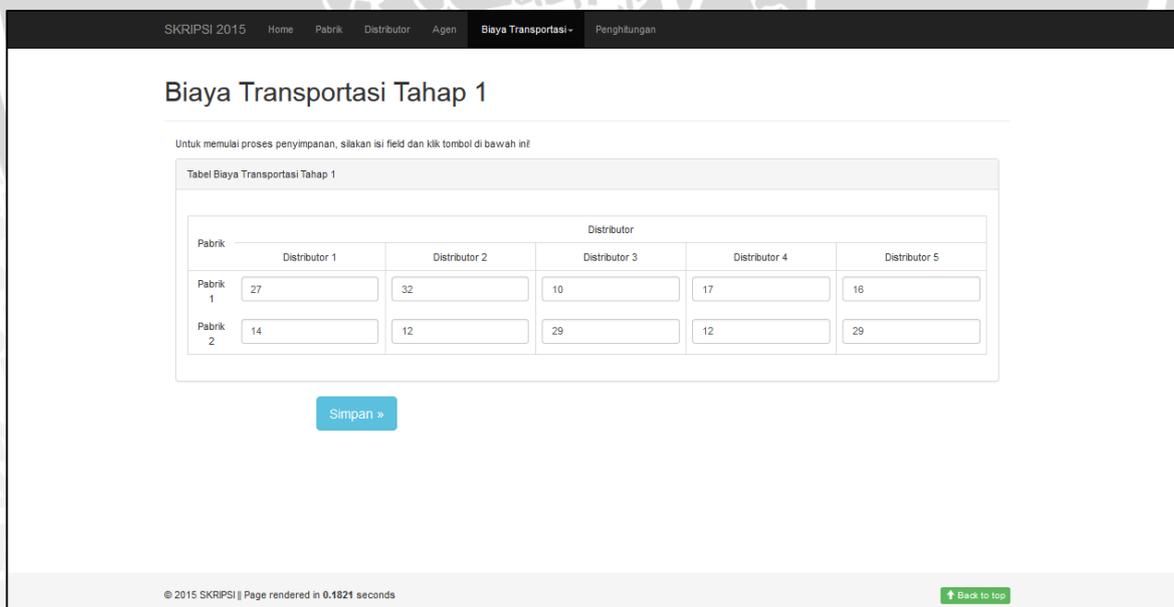
Halaman agen berisi informasi mengenai agen yang digunakan dalam saluran distribusi barang dua tahap. Dalam halaman ini terdapat kolom yang berisi mengenai informasi nama agen dan kapasitas permintaan agen. Pada antar muka agen juga dapat dilakukan penambahan agen, serta pada setiap baris terdapat tombol detail yang digunakan untuk melihat detail agen. Terdapat *filter* tabel agen yang berfungsi untuk masalah pencarian data berdasarkan nama agen. Halaman agen dan detailnya ditunjukkan pada gambar 5.5.



**Gambar 5.5** Implementasi Halaman Agen

5. Biaya Distribusi

Implementas biaya distribusi berisi mengenai informasi biaya distribusi pada tahap 1 untuk pabrik ke distributor dan tahap 2 untuk distributor ke agen. Halaman biaya distribusi ditunjukkan pada gambar 5.6.



**Gambar 5.6** Implementasi Halaman Biaya Distribusi

## 6. *Input* parameter

Halaman *input* parameter digunakan untuk memasukkan parameter genetika seperti jumlah populasi awal, maksimal generasi, *cr*, *mr* dan jenis mutasi yang digunakan. Halaman *input* parameter ditunjukkan pada gambar 5.7.

SKRIPSI 2015 Home Pabrik Distributor Agen Biaya Transportasi - Penghitungan

### Hitung

Untuk memulai proses penghitungan, silakan isi field dan klik tombol di bawah ini

Untuk memasukkan bilangan decimal, pisahkan dengan tanda titik (.)

Popsize

Max Generasi

Cross Over Rate

Mutation Rate

Mutasi  Tahap 1  Tahap 2  Campuran

Tahap 1 dan 2

Tahap 1

Tahap 2

© 2015 SKRIPSI | Page rendered in 0.1211 seconds

**Gambar 5.7** Halaman *Input* Parameter

## 7. Halaman Hasil Perhitungan

Halaman hasil perhitungan muncul ketika ditekan tombol hitung pada halaman *input* parameter. Halaman hasil perhitungan menampilkan hasil dari *inputan* biaya distribusi, *input* parameter genetika dan menampilkan proses penyelesaian masalah menggunakan algoritma genetika yang berupa representasi kromosom, proses reproduksi dengan *crossover* dan mutasi, hasil seleksi, jalur distribusi optimal dan individu yang lolos pada proses berikutnya. Untuk halaman hasil *input* parameter ditunjukkan pada gambar 5.8.

SKRIPSI 2015 Home Pabrik Distributor Agen Biaya Transportasi - Penghitungan

Hitung » hasil

Input User

Biaya Transportasi Tahap 1

Pabrik	Distributor		
	Distributor 1	Distributor 2	Distributor 3
Pabrik 1	27	32	10
Pabrik 2	14	12	29

Biaya Transportasi Tahap 2

Distributor	Agen					
	Agen 1	Agen 2	Agen 3	Agen 4	Agen 5	Agen 6
Distributor 1	20	23	14	2	68	16
Distributor 2	62	66	61	26	55	67
Distributor 3	13	59	39	71	66	19

Popsize   
 Max Generasi   
 Cross Over Rate   
 Mutation Rate

© 2015 SKRIPSI | Page rendered in 0.2723 seconds [Back to top](#)

**Gambar 5.8** Halaman Hasil *Input*-an Parameter

Representasi kromosom merupakan populasi awal yang dibentuk sejumlah masukkan parameter *popsize*. Representasi kromosom ditunjukkan pada gambar 5.9.

SKRIPSI 2015 Home Pabrik Distributor Agen Biaya Transportasi - Penghitungan

Generasi ke-1

Representasi Kromosom

Parent	Tahap 1						Tahap 2												Fitness	Total Biaya						
	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	11	12			13	14	15	16	17	18
Parent 1	5	6	2	4	3	1	14	3	5	2	13	1	15	11	12	8	9	6	4	16	7	18	17	10	0.07664	130475
Parent 2	1	4	5	3	6	2	4	14	9	15	17	13	12	8	2	10	5	18	1	3	6	11	16	7	0.07505	133250
Parent 3	2	1	4	3	6	5	14	5	15	11	13	17	3	12	8	6	7	18	2	9	10	1	4	16	0.06304	156400
Parent 4	1	3	2	5	4	6	2	12	3	18	9	6	14	17	13	4	16	7	10	11	5	15	8	1	0.08349	119775
Parent 5	6	2	5	4	1	3	14	9	17	18	7	1	15	3	11	8	13	10	5	12	6	4	2	16	0.06697	149325

**Gambar 5.9** Tampilan Antarmuka Representasi Kromosom

Setelah tampilan representasi kromosom pada populasi beserta tampilan nilai *fitnees*, maka seperti urutan proses dengan algoritma genetika adalah proses reproduksi dengan *crossover* dan mutasi. Untuk tampilan antarmuka proses reproduksi dengan *crossover* ditunjukkan pada gambar 5.10 dan tampilan antarmuka reproduksi dengan mutasi ditunjukkan pada gambar 5.11.

Crossover 1																										
Parent	Tahap 1						Tahap 2												Fitness	Total Biaya						
	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	11	12			13	14	15	16	17	18
Parent 1	5	6	2	4	3	1	14	3	5	2	13	1	15	11	12	8	9	6	4	16	7	18	17	10	0.07684	130475
Parent 2	1	4	5	3	6	2	4	14	9	15	17	13	12	8	2	10	5	18	1	3	6	11	16	7	0.07505	133250
Child	Tahap 1						Tahap 2												Fitness	Total Biaya						
	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	11	12			13	14	15	16	17	18
Child 1	5	1	4	3	6	2	14	3	5	2	13	1	15	11	12	8	9	6	4	16	7	18	17	10	0.07938	125675

© 2015 SKRIPSI || Page rendered in 0.2723 seconds [Back to top](#)

Gambar 5.10 Tampilan Halaman *Crossover*

Mutasi 1																										
Parent	Tahap 1						Tahap 2												Fitness	Total Biaya						
	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	11	12			13	14	15	16	17	18
Parent 4	1	3	2	5	4	6	2	12	3	18	9	6	14	17	13	4	16	7	10	11	5	15	8	1	0.08349	119775
Child	Tahap 1						Tahap 2												Fitness	Total Biaya						
	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	11	12			13	14	15	16	17	18
Child 3	1	3	2	5	6	4	2	12	3	18	9	6	14	4	13	17	16	7	10	11	5	15	8	1	0.08507	117550

© 2015 SKRIPSI || Page rendered in 0.2723 seconds [Back to top](#)

Gambar 5.11 Tampilan Halaman *Mutasi*

Proses yang terjadi setelah reproduksi merupakan tahap evaluasi, yaitu menampilkan seluruh individu awal baik dari populasi awal dan individu hasil reproduksi. Halaman evaluasi ditunjukkan pada gambar 5.12.



Parent	Tahap 1						Tahap 2												Fitness	Total Biaya						
	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	11	12			13	14	15	16	17	18
parent 1	5	6	2	4	3	1	14	3	5	2	13	1	15	11	12	8	9	6	4	16	7	18	17	10	0.07664	130475
parent 2	1	4	5	3	6	2	4	14	9	15	17	13	12	8	2	10	5	18	1	3	6	11	16	7	0.07505	133250
parent 3	2	1	4	3	6	5	14	5	15	11	13	17	3	12	8	6	7	18	2	9	10	1	4	16	0.06394	156400
parent 4	1	3	2	5	4	6	2	12	3	18	9	6	14	17	13	4	16	7	10	11	5	15	8	1	0.08349	119775
parent 5	6	2	5	4	1	3	14	9	17	18	7	1	15	3	11	8	13	10	5	12	6	4	2	16	0.06697	149325
child 1	5	1	4	3	6	2	14	3	5	2	13	1	15	11	12	8	9	6	4	16	17	10	18	7	0.07938	125975
child 2	2	1	4	3	6	5	14	5	15	11	13	17	3	12	8	6	7	9	18	1	10	4	2	16	0.06394	156400
child 3	1	3	2	5	6	4	2	12	3	18	9	6	14	4	13	17	16	7	10	11	5	15	8	1	0.08507	117550
child 4	1	4	5	6	3	2	4	14	9	15	17	13	2	8	12	10	5	18	1	3	6	11	16	7	0.07505	133250
child 5	6	2	3	4	1	5	2	9	17	18	7	1	15	3	11	8	13	10	5	12	6	4	14	16	0.07373	135625

**Gambar 5.12** Tampilan Halaman Evaluasi

Halaman seleksi menggunakan *elitism* menunjukkan barisan kromosom dari yang memiliki nilai *fitness* terbesar hingga terkecil. Tampilan halaman seleksi ditunjukkan pada gambar 5.13.

Parent	Tahap 1						Tahap 2												Fitness	Total Biaya						
	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	11	12			13	14	15	16	17	18
child 3	1	3	2	5	6	4	2	12	3	18	9	6	14	4	13	17	16	7	10	11	5	15	8	1	0.08507	117550
parent 4	1	3	2	5	4	6	2	12	3	18	9	6	14	17	13	4	16	7	10	11	5	15	8	1	0.08349	119775
child 1	5	1	4	3	6	2	14	3	5	2	13	1	15	11	12	8	9	6	4	16	17	10	18	7	0.07938	125975
parent 1	5	6	2	4	3	1	14	3	5	2	13	1	15	11	12	8	9	6	4	16	7	18	17	10	0.07664	130475
parent 2	1	4	5	3	6	2	4	14	9	15	17	13	12	8	2	10	5	18	1	3	6	11	16	7	0.07505	133250
child 4	1	4	5	6	3	2	4	14	9	15	17	13	2	8	12	10	5	18	1	3	6	11	16	7	0.07505	133250
child 5	6	2	3	4	1	5	2	9	17	18	7	1	15	3	11	8	13	10	5	12	6	4	14	16	0.07373	135625
parent 5	6	2	5	4	1	3	14	9	17	18	7	1	15	3	11	8	13	10	5	12	6	4	2	16	0.06697	149325
child 2	2	1	4	3	6	5	14	5	15	11	13	17	3	12	8	6	7	9	18	1	10	4	2	16	0.06394	156400
parent 3	2	1	4	3	6	5	14	5	15	11	13	17	3	12	8	6	7	18	2	9	10	1	4	16	0.06394	156400

**Gambar 5.13** Tampilan Halaman Seleksi

Setelah proses pengurutan pada seleksi, didapatkan individu terpilih sejumlah populasi awal yang dimasukkan dan akan digunakan pada proses

generasi selanjutnya. Setelah itu ditampilkan jalur distribusi yang optimal dari kromosom yang memiliki nilai *fitness* tertinggi. Jalur distribusi optimal ditunjukkan pada gambar 5.14.

Rute Alokasi	
No.	Hasil
1	Pabrik ke-1 menuju Distributor ke-1 kapasitas 700 biaya distribusi 18900
2	Pabrik ke-1 menuju Distributor ke-3 kapasitas 300 biaya distribusi 3000
3	Pabrik ke-2 menuju Distributor ke-2 kapasitas 700 biaya distribusi 8400
4	Pabrik ke-2 menuju Distributor ke-3 kapasitas 300 biaya distribusi 8700
5	Distributor ke-1 menuju Agen ke-3 kapasitas 250 biaya distribusi 3500
6	Distributor ke-1 menuju Agen ke-6 kapasitas 450 biaya distribusi 7200
7	Distributor ke-2 menuju Agen ke-1 kapasitas 125 biaya distribusi 7750
8	Distributor ke-2 menuju Agen ke-3 kapasitas 225 biaya distribusi 13725
9	Distributor ke-2 menuju Agen ke-4 kapasitas 300 biaya distribusi 7800
10	Distributor ke-2 menuju Agen ke-5 kapasitas 50 biaya distribusi 2750
11	Distributor ke-3 menuju Agen ke-1 kapasitas 200 biaya distribusi 2600
12	Distributor ke-3 menuju Agen ke-2 kapasitas 400 biaya distribusi 23800
<b>Total Biaya</b>	
107925	

**Gambar 5.14** Jalur Distribusi Optimal

Individu hasil seleksi sejumlah populasi awal akan digunakan pada generasi selanjutnya. Tampilan kromosom yang lolos pada generasi selanjutnya ditunjukkan pada gambar 5.15.

Parent	Tahap 1						Tahap 2												Fitness						
	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	11	12		13	14	15	16	17	18
parent 1	1	3	2	5	6	4	2	12	3	18	9	6	14	4	13	17	16	7	10	11	5	15	8	1	0.08507
parent 2	1	3	2	5	4	6	2	12	3	18	9	6	14	17	13	4	16	7	10	11	5	15	8	1	0.08349
parent 3	5	1	4	3	6	2	14	3	5	2	13	1	15	11	12	8	9	6	4	16	17	10	18	7	0.07938
parent 4	5	6	2	4	3	1	14	3	5	2	13	1	15	11	12	8	9	6	4	16	7	18	17	10	0.07664
parent 5	1	4	5	3	6	2	4	14	9	15	17	13	12	8	2	10	5	18	1	3	6	11	16	7	0.07505

**Gambar 5.15** Individu yang Lolos Seleksi



## BAB VI

### PENGUJIAN DAN ANALISIS

Pada bab pengujian dan analisis dijelaskan mengenai sistematika yang digunakan dalam pengujian, analisa hasil dan pembahasan dari pengujian serta hasil solusi terbaik yang didapatkan pada pengujian.

#### 6.1 Sistematika Pengujian

Sistematika pengujian pada penelitian ini dilakukan berdasarkan perancangan pengujian pada Bab IV. Pada perancangan pengujian dijelaskan bahwa pengujian dilakukan terhadap parameter-parameter genetika sebanyak empat jenis meliputi pengujian terhadap ukuran populasi, pengujian terhadap pengaruh parameter *crossover rate* (*cr*) dan *mutation rate* (*mr*), pengujian terhadap jumlah generasi dan pengujian terhadap segmen mutasi. Setiap pengujian akan dilakukan sebanyak 10 kali percobaan dan akan dihitung rata-rata nilai *fitness*-nya, sehingga diharapkan akan diperoleh parameter terbaik yang akan dapat digunakan untuk masalah optimasi distribusi barang dua tahap.

#### 6.2 Hasil dan Pembahasan

Berdasarkan perancangan pengujian pada Bab IV, berikut ini akan ditampilkan hasil pengujian terhadap parameter pengujian serta analisa hasil dan pembahasan terhadap hasil pengujian. Pengujian dilakukan seperti yang dijelaskan pada sistematika pengujian subbab sebelumnya. Adapun data yang digunakan pada pengujian sebanyak 2 pabrik, 5 distributor dan 10 agen serta data biaya distribusi per-tahap yang dapat dilihat pada formulasi permasalahan pada subbab 3.3.

##### 6.2.1 Hasil Pengujian Terhadap Ukuran Populasi

Pengujian terhadap ukuran populasi digunakan untuk mengetahui pengaruh ukuran populasi terhadap nilai *fitness* yang dihasilkan. Ukuran populasi yang akan diujia adalah 20, 40, 60, 80 dan 100. Parameter yang digunakan untuk melakukan pengujian ukuran populasi berupa :

1. Ukuran populasi : 20-100
2. Maksimal generai : 50

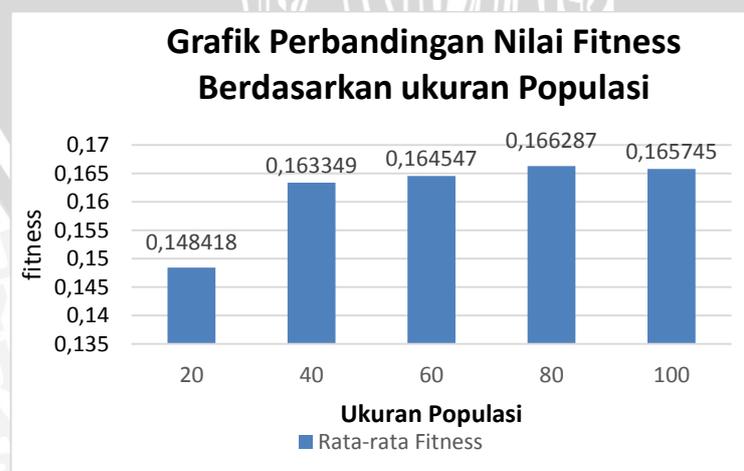
3. *Crossover rate* : 0,4
4. *Mutation rate* : 0,6 dengan segmen mutasi 1 dan 2

Setiap percobaan akan dilakukan sebanyak 10 kali, percobaan akan didata nilai *fitness* terbaik dan diambil rata-rata nilai *fitness*-nya. Hasil pengujian terhadap pengaruh jumlah populasi yang digunakan dapat dilihat pada Tabel 6.1.

**Tabel 6.1** Hasil Pengujian Pengaruh Jumlah Populasi

Popsize	Percobaan ke-										Rata-rata <i>fitness</i>
	1	2	3	4	5	6	7	8	9	10	
20	0,164	0,145	0,135	0,158	0,157	0,153	0,138	0,143	0,148	0,143	0,1484
40	0,159	0,159	0,167	0,166	0,167	0,161	0,152	0,169	0,173	0,16	0,1633
60	0,151	0,163	0,17	0,169	0,17	0,166	0,164	0,158	0,164	0,17	0,1645
80	0,173	0,167	0,16	0,167	0,169	0,162	0,165	0,158	0,169	0,173	0,1663
100	0,169	0,159	0,173	0,157	0,163	0,168	0,169	0,172	0,17	0,158	0,1657

Berdasarkan Tabel 6.1 diperoleh hasil pengujian pengaruh jumlah populasi terhadap nilai *fitness* yang menghasilkan solusi terbaik adalah ketika digunakan jumlah populasi sebanyak 80 dengan nilai *fitness* rata-rata 0,1663 dan nilai *fitness* terkecil 0,1484 ditunjukkan pada penggunaan jumlah populasi sebanyak 20. Tabel 6.1 menunjukkan rata-rata nilai *fitness* yang diperoleh berdasarkan ukuran populasi yang digunakan memiliki kenaikan pada rentang 20 sampai 80 dan penurunan ketika ukuran populasi 80 sampai 100. Untuk melihat secara lebih jelas mengenai pengaruh ukuran populasi, pada Gambar 6.1 ditunjukkan grafik pengaruh ukuran populasi terhadap nilai *fitness* yang dihasilkan.



**Gambar 6.1** Grafik Hasil Pengaruh Pengujian Ukuran Pupulasi

Berdasarkan grafik hasil pengaruh pengujian ukuran populasi terhadap nilai *fitness* dari rentang 20 sampai 80 mengalami kenaikan, pada ukuran 100 populasi mengalami penurunan kembali. Pada pengujian ini diperoleh kesimpulan bahwa ukuran populasi yang terlalu kecil maupun terlalu besar tidak akan menghasilkan solusi yang optimal. Ukuran populasi yang terlalu kecil maka area eksplorasi juga sempit sehingga peluang untuk mendapatkan solusi terbaik semakin kecil, sedangkan ukuran populasi yang terlalu besar mengakibatkan area eksplorasi menjadi semakin lebar sehingga memungkinkan terjadi konvergensi dan waktu komputasi juga akan lama. Eksplorasi yang sempit menyebabkan pencarian solusi optimal tidak berjalan dengan baik dan *child* yang dihasilkan nilai *fitness*-nya hampir sama dengan induknya (Mahmudy, 2013).

Pada pengujian ini diperoleh hasil nilai *fitness* terbesar pada ukuran populasi sebanyak 80. Jumlah populasi yang memiliki nilai *fitness* terbesar akan digunakan untuk pengujian parameter lainnya berupa parameter reproduksi, parameter jumlah maksimal generasi dan pengujian terhadap segmen mutasi.

### 6.2.2 Hasil Pengujian Terhadap *Crossover Rate* dan *Mutation Rate*

Pengujian terhadap pengaruh kombinasi nilai *crossover rate* (*cr*) dan *mutation rate* (*mr*) digunakan untuk mengetahui pengaruh kombinasi dari kedua parameter terhadap hasil yang optimal untuk kasus distribusi barang dua tahap. Pada pengujian ini digunakan parameter terbaik berdasarkan pengujian sebelumnya. Kombinasi parameter reproduksi yang digunakan berkisar 0,1 sampai 0,9. Parameter yang digunakan untuk melakukan pengujian kombinasi *cr* dan *mr* berupa :

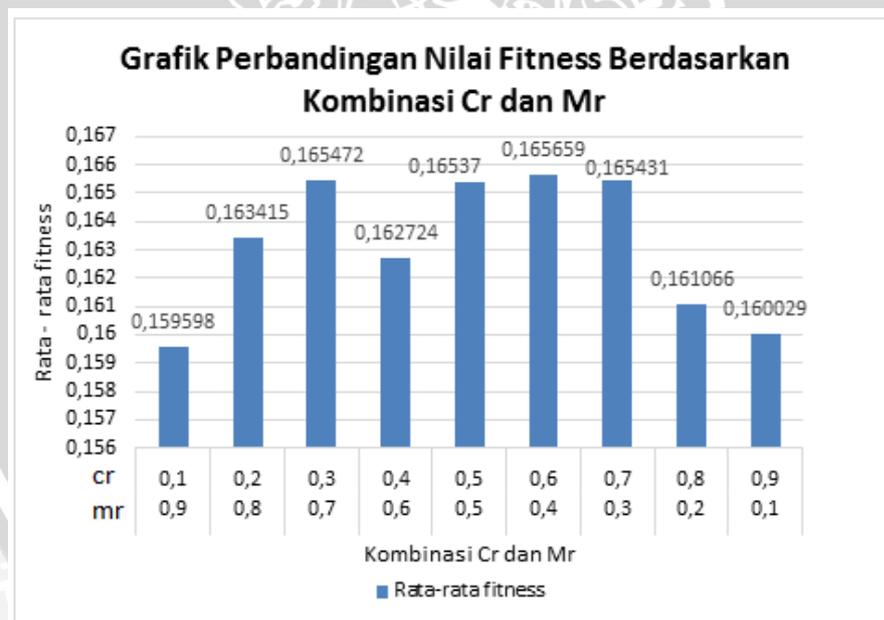
1. Ukuran populasi : 80
2. Kombinasi *cr* dan *mr* : rentang 0,1 – 0,9
3. Maksimal generasi : 50
4. Segmen mutasi : segmen 1 dan 2

Masing-masing percobaan dilakukan sebanyak 10 kali yang akan diambil nilai rata-rata *fitness*. Hasil pengujian terhadap pengaruh parameter reproduksi ditunjukkan pada Tabel 6.2.

**Tabel 6.2** Hasil Pengujian Terhadap Kombinasi *cr* dan *mr*

Ukuran		Percobaan ke-										Rata – rata Nilai Fitness
<i>cr</i>	<i>mr</i>	1	2	3	4	5	6	7	8	9	10	
0,1	0,9	0,158	0,165	0,164	0,157	0,152	0,155	0,163	0,152	0,16	0,1703	0,1596
0,2	0,8	0,173	0,163	0,167	0,157	0,173	0,16	0,157	0,162	0,16	0,1631	0,16342
0,3	0,7	0,163	0,169	0,172	0,169	0,162	0,168	0,16	0,167	0,161	0,1628	0,16547
0,4	0,6	0,169	0,162	0,163	0,168	0,159	0,156	0,166	0,168	0,159	0,1576	0,16272
0,5	0,5	0,17	0,164	0,169	0,156	0,165	0,17	0,167	0,161	0,163	0,1681	0,16537
0,6	0,4	0,169	0,166	0,169	0,167	0,16	0,17	0,16	0,17	0,16	0,164	0,16566
0,7	0,3	0,162	0,173	0,167	0,164	0,163	0,159	0,167	0,164	0,17	0,1662	0,16543
0,8	0,2	0,153	0,163	0,157	0,161	0,162	0,163	0,165	0,173	0,155	0,1584	0,16107
0,9	0,1	0,168	0,157	0,16	0,173	0,155	0,159	0,157	0,159	0,155	0,1573	0,16003

Berdasarkan Tabel 6.2 dapat dibuat grafik yang menyatakan pengaruh hasil pengujian kombinasi ukuran *cr* dan *mr* terhadap nilai *fitness*. Grafik tersebut ditunjukkan pada Gambar 6.2.

**Gambar 6.2** Grafik Hasil Pengaruh Pengujian Kombinasi *cr* dan *mr*

Berdasarkan Tabel 6.2 dan Gambar 6.2 diperoleh rata-rata nilai *fitness* terbesar adalah pada ukuran kombinasi *crossover rate* (*cr*) sebesar 0,6 dan ukuran *mutation rate* (*mr*) sebesar 0,4 dengan rata-rata nilai *fitness* 0,16566. Sedangkan rata-rata nilai *fitness* terkecil adalah pada ukuran kombinasi *cr* 0,1 dan *mr* 0,9

dengan rata-rata *fitness* 0,1596. Nilai *fitness* yang dihasilkan pada setiap kombinasi cenderung rendah ketika nilai *cr* dan *mr* yang tidak seimbang, yaitu *cr* yang terlalu kecil maupun *mr* yang terlalu kecil. Nilai *cr* yang terlalu tinggi dan *mr* yang rendah menyebabkan algoritma genetika tidak akan mampu memperlebar area pencarian dan mengakibatkan konvergensi dini yaitu tidak adanya perubahan terlalu besar dalam mendapatkan solusi ketika di uji coba pada beberapa generasi saja, sedangkan nilai *cr* yang rendah dan *mr* yang tinggi menyebabkan algoritma genetika akan bekerja seperti *random search* dan tidak mampu mengeksplorasi daerah pencarian secara efektif (Mahmudy, WF, Marlian, RM dan Luong, LHS dalam Pratiwi, 2014).

Pengujian kombinasi *cr* dan *mr* menunjukkan bahwa pada *cr* 0,6 dan *mr* 0,4 menghasilkan solusi optimal dibandingkan ukuran kombinasi yang lain. Hasil pengujian kombinasi *cr* dan *mr* yang optimal akan digunakan pada pengujian selanjutnya mengenai pengujian jumlah generasi dan pengujian jenis segmen mutasi.

### 6.2.3 Hasil Pengujian Terhadap Jumlah Generasi

Pengujian terhadap jumlah generasi digunakan untuk mengetahui pengaruh banyaknya jumlah generasi yang digunakan terhadap rata-rata nilai *fitness*. Pada pengujian ini digunakan jumlah generasi sebanyak 20, 40, 60, 80,100 dengan jumlah ukuran populasi sebanyak 80 dan kombinasi *cr* sebesar 0,6 dan *mr* sebesar 0,4. Percobaan terhadap pengaruh jumlah generasi dilakukan sebanyak 10 kali untuk diambil nilai rata-rata *fitness*, hasil pengujian ditunjukkan pada Tabel 6.3.

**Tabel 6.3** Hasil Pengujian Terhadap Parameter Jumlah Generasi

Banyak Generasi	Percobaan ke-										Rata – rata Nilai <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
20	0,148	0,1561	0,1552	0,1631	0,157	0,153	0,16	0,14	0,1624	0,1689	0,156344
40	0,152	0,1525	0,1523	0,1664	0,158	0,155	0,158	0,163	0,1584	0,1695	0,158542
60	0,163	0,1631	0,1629	0,1726	0,168	0,167	0,159	0,163	0,165	0,163	0,164622
80	0,168	0,1599	0,163	0,1726	0,159	0,166	0,17	0,16	0,172	0,1667	0,165652
100	0,161	0,1675	0,17	0,1673	0,162	0,171	0,155	0,165	0,1631	0,1675	0,1649

Berdasarkan Tabel 6.3 dapat dibuat grafik yang menyatakan pengaruh hasil pengujian jumlah generasi terhadap nilai *fitness*. Grafik tersebut ditunjukkan pada Gambar 6.3.



**Gambar 6.3** Grafik Hasil Pengaruh Pengujian Jumlah Generasi

Berdasarkan Tabel 6.3 dan Gambar 6.3 mengenai pengujian jumlah generasi diperoleh solusi terbaik pada generasi ke 80. Hasil *fitness* tertinggi pada generasi tersebut adalah 0,165652 dan nilai *fitness* terendah dihasilkan ketika menggunakan jumlah generasi yang sedikit pada jumlah generasi 20. Pada grafik yang ditunjukkan oleh Gambar 6.3 generasi 20 sampai dengan 80 mengalami kenaikan rata-rata nilai *fitness* yang signifikan karena dengan bertambahnya generasi maka akan terbentuk individu yang lebih beragam sehingga terdapat peluang untuk menghasilkan individu baru yang memiliki nilai *fitness* yang tinggi. Jumlah generasi sebanyak 20 memiliki rata-rata nilai *fitness* cenderung kecil dikarenakan eksplorasi terhadap ruang pencarian cenderung lebih kecil sehingga individu yang dihasilkan kurang beragam. Sedangkan pada jumlah generasi 80 menghasilkan rata-rata nilai *fitness* yang tinggi karena eksplorasi terhadap ruang pencarian cenderung lebih besar dan menghasilkan individu yang beragam namun diikuti dengan waktu eksekusi yang lebih lama (Permadi & Subanar, 2010). Pada generasi ke 100 rata-rata nilai *fitness* cenderung mengalami penurunan meskipun ruang eksplorasi cukup besar, hal ini dikarenakan konsep random pada algoritma genetika yang menyebabkan individu dengan nilai *fitness* yang rendah pada

generasi sebelumnya dicoba pada generasi selanjutnya. Karena pada generasi ke-80 memiliki rata-rata nilai *fitness* yang lebih tinggi dari generasi yang lain, maka untuk generasi tersebut digunakan pada pengujian segmen mutasi berserta dengan parameter yang telah diuji pada subbab sebelumnya.

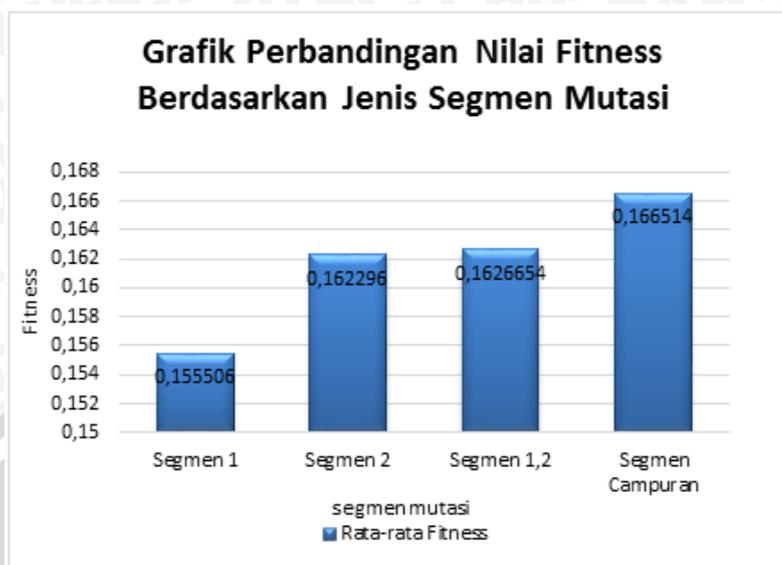
#### 6.2.4 Hasil Pengujian Terhadap Perbandingan Segmen Mutasi

Pengujian terhadap segmen mutasi digunakan untuk mengetahui perbandingan rata-rata nilai *fitness* berdasarkan segmen mutasi yang digunakan. Pada pengujian segmen mutasi digunakan beberapa parameter terbaik dari pengujian sebelumnya yaitu ukuran populasi sebanyak 80, kombinasi *cr* 0,6 dan *mr* 0,4 serta jumlah generasi 80. Pengujian segmen mutasi diterapkan dengan menggunakan segmen mutasi pada distribusi tahap 1, segmen mutasi pada distribusi tahap 2, menggunakan dua segmen sekaligus dan menggunakan kombinasi antara ketiga jenis segmen. Percobaan pada pengujian segmen mutasi dilakukan sebanyak 10 kali untuk diambil rata-rata nilai *fitness*, hasil pengujian terhadap segmen mutasi ditunjukkan pada Tabel 6.4.

**Tabel 6.4** Hasil Pengujian Terhadap Segmen Mutasi

Segmen			Percobaan ke-										Rata – rata Nilai <i>Fitness</i>
1,2	1	2	1	2	3	4	5	6	7	8	9	10	
100%	0	0	0,166	0,148	0,154	0,153	0,144	0,172	0,134	0,159	0,162	0,164	0,1555
0	100%	0	0,161	0,17	0,152	0,161	0,166	0,1681	0,158	0,157	0,173	0,158	0,1623
0	0	100%	0,166	0,163	0,173	0,156	0,166	0,1556	0,163	0,159	0,163	0,162	0,1627
40%	30%	30%	0,167	0,168	0,164	0,168	0,159	0,1729	0,168	0,168	0,169	0,163	0,1665

Berdasarkan Tabel 6.4 dapat dibuat grafik yang menyatakan pengaruh hasil pengujian segmen mutasi terhadap nilai *fitness*. Grafik tersebut ditunjukkan pada Gambar 6.4.



**Gambar 6.4** Grafik Hasil Pengaruh Pengujian Segmen Mutasi

Berdasarkan Tabel 6.4 dan grafik pada Gambar 6.4 mengenai pengujian segmen mutasi terhadap rata-rata nilai *fitness*, menunjukkan bahwa segmen campuran menghasilkan hasil optimal dibandingkan dengan menggunakan segmen yang lain. Hasil rata-rata nilai *fitness* optimal ditunjukkan oleh segmen mutasi campuran dengan hasil 0,166514, sedangkan untuk hasil rata-rata nilai *fitness* minimal ditunjukkan oleh penggunaan segmen mutasi distribusi tahap 1. Hal ini membuktikan bahwa mutasi dengan segmen campuran lebih optimal karena lebih banyak solusi yang dicoba pada setiap segmennya, sedangkan pada mutasi dengan segmen 1 saja menghasilkan hasil yang lebih rendah dibandingkan dengan yang lain mengakibatkan area eksplorasi lebih kecil dikarenakan pada segmen pertama panjang kromosomnya lebih pendek. Sehingga kecenderungan suatu kromosom untuk lebih mirip dengan induknya lebih besar.

### 6.3 Solusi Terbaik yang Pernah Didapatkan

Algoritma genetika merupakan salah satu algoritma yang mampu memecahkan masalah dengan meniru mekanisme biologis, dari beberapa penelitian algoritma genetika mampu menghasilkan solusi optimal untuk banyak kasus. Walaupun solusi yang dihasilkan tidak selalu optimal namun selalu mendekati optimal. Pada penerapan algoritma genetika untuk permasalahan optimasi distribusi

barang dua tahap, solusi optimal yang pernah dihasilkan adalah 0,17294 diterapkan pada 2 buah pabrik dengan 5 distributor dan 10 agen dengan panjang kromosom sebanyak 60. Solusi ini diperoleh dengan parameter berupa ukuran populasi sebanyak 80, kombinasi  $cr$  0,6 dan  $mr$  0,4 dengan jumlah generasi sebanyak 80 menggunakan jenis mutasi segmen campuran. Rincian dari nilai *fitness* ini adalah sebagai berikut :

Rute distribusi :

Tahap I : 3-5-4-1-10-2-8-7-9-6

Tahap II : 45-3-27-47-7-4-14-1-30-20-2-18-17-39-13-16-23-21-25-38-28-37-19-31-9-44-10-46-36-29-50-12-8-11-5-43-33-49-48-26-6-35-40-15-34-42-24-41-32-22

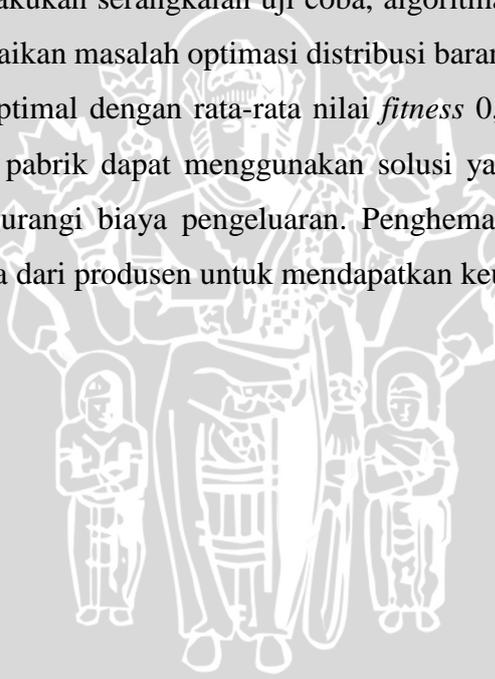
Detail rute distribusi :

- Pabrik ke-1 menuju Distributor ke-3 kapasitas 250 biaya distribusi 2500
- Pabrik ke-1 menuju Distributor ke-4 kapasitas 300 biaya distribusi 5100
- Pabrik ke-1 menuju Distributor ke-5 kapasitas 450 biaya distribusi 7200
- Pabrik ke-2 menuju Distributor ke-1 kapasitas 300 biaya distribusi 4200
- Pabrik ke-2 menuju Distributor ke-2 kapasitas 400 biaya distribusi 4800
- Pabrik ke-2 menuju Distributor ke-4 kapasitas 300 biaya distribusi 3600
- Distributor ke-1 menuju Agen ke-3 kapasitas 200 biaya distribusi 2800
- Distributor ke-1 menuju Agen ke-4 kapasitas 100 biaya distribusi 200
- Distributor ke-2 menuju Agen ke-4 kapasitas 125 biaya distribusi 3250
- Distributor ke-2 menuju Agen ke-8 kapasitas 25 biaya distribusi 600
- Distributor ke-2 menuju Agen ke-10 kapasitas 250 biaya distribusi 6500
- Distributor ke-3 menuju Agen ke-7 kapasitas 150 biaya distribusi 1650
- Distributor ke-3 menuju Agen ke-8 kapasitas 100 biaya distribusi 1100
- Distributor ke-4 menuju Agen ke-1 kapasitas 150 biaya distribusi 1200
- Distributor ke-4 menuju Agen ke-2 kapasitas 75 biaya distribusi 1050
- Distributor ke-4 menuju Agen ke-8 kapasitas 175 biaya distribusi 4025
- Distributor ke-4 menuju Agen ke-9 kapasitas 200 biaya distribusi 1200

- Distributor ke-5 menuju Agen ke-2 kapasitas 100 biaya distribusi 2300
- Distributor ke-5 menuju Agen ke-5 kapasitas 200 biaya distribusi 3000
- Distributor ke-5 menuju Agen ke-6 kapasitas 150 biaya distribusi 1650

Dari beberapa pengujian yang telah dilakukan menggunakan parameter yang berbeda algoritma genetika mampu menyelesaikan permasalahan optimasi untuk distribusi barang dua tahap. Dari sekian kasus, algoritma genetika mampu memberikan solusi yang optimal dibandingkan dengan algoritma *brute force* (Ningrum, 2010). Pada penelitian ini menggunakan seleksi *elitism* yang merupakan jenis seleksi terbaik untuk kasus optimasi, karena seleksi *elitism* terbukti lebih baik untuk memberikan solusi optimal.

Setelah melakukan serangkaian uji coba, algoritma genetika dianggap mampu untuk menyelesaikan masalah optimasi distribusi barang dua tahap dengan memberikan hasil ter-optimal dengan rata-rata nilai *fitness* 0,17294 dengan total biaya minimum. Pihak pabrik dapat menggunakan solusi yang telah ditemukan sehingga mampu mengurangi biaya pengeluaran. Penghematan biaya distribusi merupakan tujuan utama dari produsen untuk mendapatkan keuntungan yang lebih besar.



## BAB VII

### PENUTUP

#### 7.1 Kesimpulan

Berdasarkan penelitian mengenai penerapan algoritma genetika untuk menyelesaikan permasalahan optimasi distribusi barang dua tahap, diperoleh beberapa kesimpulan sebagai berikut :

1. Algoritma genetika mampu menyelesaikan permasalahan optimasi distribusi barang dua tahap dengan memberikan solusi berupa pencarian jalur distribusi yang tepat dan menghasilkan biaya total minimum dari satu jenis barang.
2. Penyelesaian masalah distribusi barang dua tahap dengan algoritma genetika digunakan representasi kromosom berupa permutasi dengan jumlah gen didapat dari mengalikan jumlah produsen dengan distributor untuk tahap 1, sedangkan untuk tahap 2 panjang kromosom diperoleh dari distributor dikali dengan agen.
3. Digunakan teknik *crossover* berupa *one cut point crossover*, mutasi berupa *exchange mutation* dan seleksi *elitism*. Teknik mutasi menggunakan *exchange mutation* dengan segmen campuran yang memperoleh hasil rata-rata *fitness* tertinggi sebesar 0,1665. Teknik mutasi jenis ini menyebabkan area eksplorasi lebih besar sehingga memungkinkan untuk menemukan solusi terbaik.
4. Parameter algoritma genetika yang baik pada permasalahan optimasi distribusi barang dua tahap ditentukan dengan melakukan pengujian terhadap ukuran populasi, kombinasi *crossover rate* dan *mutation rate*, jumlah generasi dan segmen mutasi yang digunakan.
5. Nilai parameter *popsize* yang baik untuk penelitian ini adalah 80, kombinasi *cr* 0,4 dan *mr* 0,6, jumlah generasi sebanyak 80 dan menggunakan segmen mutasi campuran memberikan rata-rata nilai *fitness* tertinggi yaitu 0,1665.

6. Pengujian dilakukan terhadap 2 pabrik, 5 distributor dan 10 agen dengan individu terbaik yang pernah diperoleh adalah pada rute distribusi 3-5-4-1-10-2-8-7-9-6 untuk tahap 1 dan 45-3-27-47-7-4-14-1-30-20-2-18-17-39-13-16-23-21-25-38-28-37-19-31-9-44-10-46-36-29-50-12-8-11-5-43-33-49-48-26-6-35-40-15-34-42-24-41-32-22 untuk tahap 2.

## 7.2 Saran

Pada penelitian ini, terdapat beberapa hal yang dapat dikembangkan dan dijadikan bahan untuk penelitian lebih lanjut yaitu :

1. Aplikasi dapat diterapkan pada kasus distribusi yang lebih nyata dengan memperhatikan kapasitas distribusi dan biaya distribusi. Pada penelitian ini masih menggunakan jenis distribusi yang seimbang pada kapasitas persediaan sesuai dengan kapasitas permintaan, pada kenyataannya kapasitas persediaan tidak selalu sama dengan permintaan bisa jadi lebih atau kurang.
2. Biaya distribusi yang diperoleh pada penelitian ini menggunakan data *dummy* yang disesuaikan dengan kondisi permasalahan nyata , belum memperhatikan faktor biaya tetap pada pengiriman barang misalnya jarak distribusi, standar harga bahan bakar, beban gaji supir, jumlah kendaraan yang dipakai dan jadwal distribusi.
3. Menggunakan jumlah pabrik, distributor dan agen yang lebih banyak untuk membuktikan daya tahan (*robustness*) pada data yang berukuran besar.
4. Melakukan proses hibrid dengan algoritma lain untuk memperoleh solusi yang lebih baik. Terbukti bahwa teknik hibrid cukup sukses diterapkan pada kasus yang kompleks (Mahmudy, WF, Marian, RM, & Luong, 2014)

## DAFTAR PUSTAKA

- Gen, M., & Cheng, R. (2000). *Genetic Algorithms and Engineering Optimization*. Wiley. New York.
- Gen, M., Altiparmak, F., & Lin, A. (2006). A genetic algorithm for two-stage transportation problem using priority-based encoding. *OR Spektrum*, 28: 337-354.
- Haupt, S. (2004). *Practical Genetic Algorithm*. Wiley: New Jersey.
- Harun, IA, Mahmudy, WF & Yudistira, N. 2014. Implementasi Evolution Strategies untuk Penyelesaian Vehicle Routing Problem With Time Windows pada Distribusi Minuman Soda XYZ. *DORO: Repository Jurnal Mahasiswa PTIK Universitas Brawijaya*, vol 4, no. 1.
- Kotler, P. (2003). *Marketing Management*. Eleventh Edition. Prentice Hall.
- Kumar, T. (2004, November 8). 106. Retrieved from Cornerstone: <https://cornerstone.lib.mnsu.edu/cgi/viewcontent.cgi?article=1162&context=jur>[5 Februari 2015]
- Mahmudy, W. (2013). *Algoritma Evolusi. Modul Kuliah Semester Ganjil 2013-2014*. Malang: Universitas Brawijaya.
- Mahmudy, WF, Marian, RM, & Luong. (2014). Hybrid genetic algorithms for part type selection and machine loading problems with alternative production plans in flexible manufacturing system. *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 8, no. 1, pp. 80-93.
- Mayasari, N. (2012). *Penerapan Algoritma Genetika untuk Permasalahan Distribusi Rantai Pasok Dua Tingkat yang Dipengaruhi oleh Biaya Tetap*. Surabaya: Skripsi. Sistem Informasi. Institut Teknologi Sepuluh Nopember.
- Michalewicz, Z. (1996). *Genetic Algorithms+data Structures = Evolution programs*. Germany: Springer.
- Natalisa, D. (2007). *Dasar-dasar Manajemen Pemasaran*. Unsri. Palembang.
- Ningrum, L. A. (2010). *Pencarian Biaya Minimum Pendistribusian Suatu Komoditi dengan Transportasi Dua Tahap Menggunakan Algoritma Genetik*. Skripsi. FMIPA. Universitas Brawijaya. Malang.

- Nirwansyah, H., & Widowati. (2007). Efisiensi Biaya Distribusi dengan Metode Transportasi. *Universitas Diponegoro*, hal. 2.
- Permadi, I., & Subanar. (2010). Penerapan Algoritma Genetika untuk Penjadwalan Tebangan Hutan . *JUITA*, 23. vol.1 no. 1.
- Pitaloka, D., Mahmudy, W., & Sutrisno. (2012). Penerapan Hybrid Algoritma Genetika untuk Permasalahan Vehicle Routing Problem with Time Windows (VRPTW). *DORO*, vol. 4 no. 11.
- Prabowo, K. A. (2010). *Penerapan Algoritma Genetik Dua Populasi pada Kasus Transportasi Dua Tahap*. Malang: Skripsi. FMIPA. Universitas Brawijaya.
- Pratiwi, M. (2014). Implementasi Algoritma Genetika pada Optimasi Biaya Pemenuhan Kebutuhan Gizi . *DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya*, vol.4, no 6. hal. 7.
- Pujawan, I., & Mahendrawati. (2010). *Supply Chain Management*. Guna Wdya. Surabaya.
- Sari, P. A. (2008). *Optimasi Rute Pendistribusian Komoditi Berdasarkan Jarak dan Kondisi Jalan Untuk Meminimalkan Biaya Bahan Bakar dengan Algoritma Genetika*. Skripsi. FMIPA. Universitas Brawijaya. Malang.
- Sutojo, T., Mulyanto, E., & Suhartanto, V. (2011). *Kecerdasan Buatan*. ANDI. Yogyakarta.
- Toni, D. M. (2009). *Penentuan Rute Optimum Jalur Distribusi dalam Suplly Chain Network Menggunakan Algoritme Genetika*. FMIPA. IPB. . Bogor.