

**ALGORITMA GENETIKA UNTUK PERMASALAHAN *ROUTING*
ARMADA DENGAN PEMERATAAN BEBAN (STUDI KASUS PT. PATRA
NIAGA ANAK PERUSAHAAN PT PERTAMINA DEPOT REWULU)**

Skripsi

Untuk Memenuhi Sebagian Persyaratan

Mencapai Derajat Sarjana S-1

Program Studi Teknik Industri



Diajukan Oleh:

TRIO YONATHAN TEJA KUSUMA

07660021

Kepada

PROGRAM STUDI TEKNIK INDUSTRI

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA

YOGYAKARTA

2011



PENGESAHAN SKRIPSI/TUGAS AKHIR

Nomor : UIN.02/D.ST/PP.01.1/1205/2011

Skripsi/Tugas Akhir dengan judul : Algoritma Genetika untuk Permasalahan *Routing* Armada dengan Pemerataan Beban (Studi Kasus PT. Patra Niaga Anak Perusahaan PT. Pertamina Depot Rewulu)

Yang dipersiapkan dan disusun oleh :
Nama : Trio Yonathan Teja Kusuma
NIM : 07660021
Telah dimunaqasyahkan pada : 27 Juni 2011
Nilai Munaqasyah : A

Dan dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga

TIM MUNAQASYAH :

Ketua Sidang

Taufiq Aji, M.T
NIP. 19800715 200604 1 002

Penguji I

Cahyono Sigit Pramudyo, M.T
NIP.19801025 200604 1 001

Penguji II

Arya Wirabhuana, M.Sc
NIP. 19770127 200501 1 002

SUNAN KALIJAGA
YOGYAKARTA

Yogyakarta, 1 Juli 2011

UIN Sunan Kalijaga

Fakultas Sains dan Teknologi
Dekan



Prof. Drs. H. Akh. Minhaji, M.A, Ph.D
NIP.19580919 198603 1 002

SURAT PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama : Trio Yonathan Teja Kusuma

NIM : 07660021

Jurusan : Teknik Industri

Fakultas : Sains dan Teknologi

Menyatakan dengan sesungguhnya dan sejujurnya, bahwa skripsi saya yang berjudul:

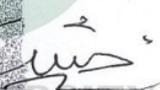
“Algoritma Genetika Untuk Permasalahan Routing Armada Dengan Pemerataan Beban (Studi Kasus PT. Patra Niaga Anak Perusahaan PT Pertamina Depot Rewulu)”

Adalah asli hasil penelitian saya sendiri dan bukan plagiasi hasil karya orang lain.

Yogyakarta, 10 Juni 2011

Yang menyatakan




Trio Yonathan Teja Kusuma
07660021

STATE ISLAM
SUNAN KALIJAGA
YOGYAKARTA



SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Surat Persetujuan Skripsi/ Tugas Akhir
Lamp :

Kepada
Yth. Dekan Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta
di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Trio Yonathan Teja Kusuma
NIM : 07660021
Judul skripsi : Algoritma Genetika Untuk Permasalahan Routing Armada Dengan
Pemerataan Beban (Studi Kasus PT. Patra Niaga Anak Perusahaan PT
Pertamina Depot Rewulu)

Sudah dapat diajukan kembali kepada Program Studi Teknik Industri Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Jurusan Teknik Industri.

Dengan ini kami mengharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapkan terima kasih.

Yogyakarta, 09 Juni 2011

Pembimbing I

TaufiqAji S. T, M. T
NIP. 19800715 200604 1 002

Pembimbing II

Yandra Rahadian Perdana, M.T
NIP. 19811025 200912 1 002

KATA PENGANTAR

Alhamdulillah, puji syukur kehadiran Allah SWT atas segala limpahan rahmat serta hidayah-Nya, sehingga penulis dapat melakukan penelitian dan menyelesaikan Laporan Skripsi ini sesuai dengan jadwal yang telah ditentukan. Skripsi ini merupakan salah satu syarat akademis yang harus dipenuhi oleh mahasiswa untuk menyelesaikan studi di Program Studi Teknik Industri, Fakultas Teknik, Universitas Islam Negeri Sunan Kalijaga Yogyakarta.

Pada kesempatan ini, dengan hati yang tulus penulis ingin menyampaikan terima kasih atas segala bimbingan dan bantuan yang telah diberikan selama penyusunan laporan skripsi ini. Ucapan terima kasih, penyusun sampaikan kepada:

1. Allah SWT, Sholawat dan salam selalu tercurah kepada Beliau, kepada para sahabat, kepada para alim ulama, dan kepada seluruh kaum muslim di seluruh dunia.
2. Bapak Arya Wirabhuana, S.T, M.Sc, selaku Ketua Program Studi Teknik Industri, Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga Yogyakarta.
3. Bapak Taufiq Aji, M. T dan bapak Yandra Rahadian Perdana, M. T selaku Dosen Pembimbing atas kesabaran, perhatian, dan bimbingan yang telah banyak diberikan dari awal hingga terselesaikannya Skripsi ini.
4. Bapak Arya Wirabhuana, S.T, M.Sc dan bapak Cahyono Sigit Pramudyo selaku dosen penguji.

5. Seluruh Dosen dan Mahasiswa Teknik Industri, Fakultas Sains dan Teknologi, UIN Sunan Kalijaga.
6. Bapak Riyadi Atmono (site Supervisor Patra Niaga) yang telah memberi ijin untuk penelitian serta banyak membantu saya dalam pengambilan data.
7. Semua karyawan PT. Patra Niaga atas bantuannya selama penelitian.
8. Babeh dan Ibu tercinta atas doa, kasih sayang dan nasehat yang sangat berarti yang selalu terlantun untukku.
9. Mas sigit, Mbak Yuli, Amik, Fitri yang selalu memberi nasihat dan dukungannya selama ini.
10. Kiyandra Aldia Arka Purnomo keponakanku yang nakal abis tapi pintar dan bikin kangen.
11. Keluarga EYANG SOEDIRMAN yang selalu memberi dukungan dan doa.
12. Sulistia Khasanah yang dengan sabar membantu dan membangunkan dari mimpi buruk dan terus memotivasiku.
13. Keluarga Brebes yang selalu mendoakanku.
14. My Brother Renno (pakbo), agus (mas' copet), dito, mas budi, maksy banyak atas doa dukungan dan nasihat-nasihatnya serta menjadi tempat untukku berkeluh kesah serta turut menyadarkan ku saat mimpi buruk datang.
15. Budi 08 yang telah membantuku hingga bergadang saat membuat data base.
16. Si buluk (Smash) yang selalu setia menemani dan mengantarkanku sampai ke tempat tujuan dengan selamat, serta tempat pelampiasan saat aku mulai gelap mata.

17. Blaccky, tikus, putih, serta cah bayi yang membuatku tertawa saat kepala dipenuhi kepenatan.
18. Mbak Hasti dan mas Saipul yang telah memberi masukan dan dukungan
19. Teman-teman seperjuangan di Teknik Industri khususnya angkatan 2007, serta seluruh keluarga besar Teknik Industri UIN Sunan Kalijaga.
20. Untuk ex caca' yang memberiku pelajaran tentang arti hidup.
21. Seluruh karyawan Fakultas Sains dan Teknologi, UIN Sunan Kalijaga.

Penulis menyadari bahwa Skripsi ini jauh dari sempurna dan tak luput dari kesalahan dikarenakan keterbatasan waktu dan pengetahuan penulis. Oleh karena itu, penulis mengharapkan kritik dan saran dari pihak manapun guna perbaikan karya kebaikan Skripsi ini. Semoga laporan Skripsi ini bermanfaat bagi kalangan bagi kita semua.

Yogyakarta, 22 Juni 2011

Penulis

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

Trio Yonathan Teja Kusuma

Karya ini sebagai baktiku, atas air susu, perlindungan, serta kasih sayang yang kalian berikan.

IBU dan BAPAK

Terima Kasih

Ini bukan sebuah hadiah pengganti tapiu ini adalah karya dari hatiku untuk kalian.

“Jangan pernah mengharapkan keajaiban tapi jadilah keajaiban itu maka kamu akan diharapkan”

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

DAFTAR ISI

HALAMAN JUDUL	I
HALAMAN PENGESAHAN	ii
SURAT PERNYATAAN	iii
SURAT PERSETUJUAN SKRIPSI	vi
KATA PENGANTAR	v
HALAMAN PERSEMBAHAN	viii
DAFTAR ISI	ix
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xiv
ABSTRAK	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah dan Asumsi	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
BAB II STUDI PUSTAKA DAN LANDASAN TEORI.....	5
2.1 StudiPustaka	5
2.2 Landasan Teori	5
2.2.1 Distribusi.....	11

2.2.2 <i>Vehicle Routing Problem</i>	12
2.2.3 <i>Split Delivery Vehicle Routing Problem</i>	14
2.2.4 Pemerataan Beban Kerja..	16
2.2.5 Optimasi.....	17
2.2.6 Penyelesaian Permasalahan.....	18
2.2.7 Metode Heuristik.....	18
2.2.8 Algoritma Genetika.....	21
BAB III METODE PENELITIAN	40
3.1 Objek Penelitian	40
3.2 Pengumpulan Data	40
3.2.1 Jenis Data yang Digunakan.....	40
3.2.2 Sumber Data.....	41
3.2.3 Teknik Pengumpulan Data.....	41
3.3 Variabel Data	42
3.4 Tahap Penelitian.....	46
BAB IV PENGUMPULAN DAN PENGOLAHAN DATA	51
4.1 Pengumpulan Data.....	51
4.1.1 Sekilas PT Patra Niaga.....	51
4.1.2 Proses Bisnis Patra Niaga.....	52
4.1.3 Dasar Proses Bisnis.....	54
4.1.4 Tujuan Perusahaan.....	55
4.1.5 Visi dan Misi Perusahaan	55
4.2 Pengolahan Data	56

4.2.1 Genetik Algoritma.....	56
4.2.2 Proses <i>Coding</i>	57
BAB V ANALISA DAN PEMBAHASAN	105
5.1 Pembangunan Populasi Awal.....	105
5.2 Fitness.....	106
5.3 Seleksi.....	107
5.4 Crossover.....	109
5.5 Mutasi.....	109
5.6 Evolusi.....	110
5.7 Pemerataan Beban dan Minimasi Biaya.....	111
5.8 Perbandingan Biaya dengan Perusahaan.....	112
BAB VI KESIMPULAN DAN SARAN.....	114
6.1 Kesimpulan	114
6.1.1 Langkah Proses Genetik Algoritma dalam Sistem.....	114
6.1.2 Solusi (Rute) Hasil Pengolahan.....	114
6.1.3 Perbandingan dengan Rute Rancangan Perusahaan.....	114
6.2 Kontribusi Penelitian.....	115
6.2.1 Vehicle Based Representation.....	115
6.2.2 Block Crossover.....	115
6.2.3 Constrained Insertion Mutation.....	115
6.3 Saran.....	116
DAFTAR PUSTAKA	117
LAMPIRAN	119

DAFTAR TABEL

	Perbandingan Penelitian yang Dilakukan dengan Penelitian	
Tabel 2.1	terdahulu.....	6
Tabel 4.1	Kode Persiapan Data.....	58
Tabel 4.2	Kode Pembentukan Populasi Pertama.....	61
Tabel 4.5	Kode Perhitungan Nilai Fitness.....	67
Tabel 4.6	Kode Untuk Fitnes Minimasi Biaya.....	68
Tabel 4.7	Kode Fitnes Pemerataan Beban.....	72
	Data Historis Tabel Jarak dan Biaya Pengiriman Tanggal 11	
Tabel 4.8	Maret 2011.....	74
	Data Urutan Biaya dan Hasil Pengolahan Jarak dengan	
Tabel 4.9	Pendekatan MAD.....	76
Tabel 4.10	Kode Penentuan Fitness Total.....	77
Tabel 4.11	Kode Pengurutan Fitness.....	78
Tabel 4.12	Penentuan Probabilitas Crossover.....	79
Tabel 4.13	Pemberian Nilai Random.....	80
Tabel 4.14	Penyimpanan Nilai Random.....	81
Tabel 4.15	Penentuan Kromosom Terpilih.....	81
Tabel 4.16	Penentuan Urutan Genom Yang Akan di Crossover.....	84
Tabel 4.17	Proses Crossover.....	84
Tabel 4.18	Penanggulangan Genome yang Sama.....	85
Tabel 4.19	Proses Mengenal Genome.....	87

Tabel 4.20	Pencarian Genome yang hilang.....	88
Tabel 4.21	Proses Penyisipan Genome Terpilih.....	88
Tabel 4.22	Proses Evolusi Awal.....	91
Tabel 4.23	Evolusi Lanjutan.....	92
Tabel 4.24	Data Permintaan SPBU.....	94
Tabel 4.25	Perhitungan Fitness untuk Rute Terbaik pada Percobaan 10....	97
Tabel 4.26	Perhitungan Fitness untuk Rute Terbaik pada Percobaan 20....	97
Tabel 4.27	Perhitungan Fitness untuk Rute Terbaik pada Percobaan 30....	98
Tabel 4.28	Data SPBU Pemesan Tanggal 17 April 2011.....	102
	Hasil Perbandingan Rute dan Biaya Distribusi Tanggal 17	
	April 2011.....	103
Tabel 4.30	Perbandingan Total Biaya Distribusi.....	104

DAFTAR GAMBAR

Gambar 2.1	Split Delivery Vehicle Routing Problem.....	15
Gambar 2.2	Seleksi Elitisme.....	30
Gambar 2.3	Diagram Alir Algoritma Genetik.....	38
Gambar 3.1	Diagram Alir Tahapan Penelitian.....	46
Gambar 3.2	Antar Muka Sistem.....	49
Gambar 4.1	Struktur Organisasi Patra Niaga.....	51
Gambar 4.2	Proses Bisnis Patra Niaga.....	52
Gambar 4.3	Diagram Alir Proses Algoritma Genetik.....	57
	Simulasi Proses Pembentukan Populasi Awal untuk	
Gambar 4.4	Kromosom Pertama	65
	Simulasi Proses Pembentukan Populasi Awal untuk	
Gambar 4.5	Kromosom Kedua.....	65
Gambar 4.6	Simulasi Proses Seleksi.....	82
Gambar 4.7	Simulasi Proses Crossover.....	85
Gambar 4.8	Contoh Kromosom Hasil Crossover.....	90
Gambar 4.9	Proses Mengenolkan Genome.....	90
Gambar 4.10	Proses Mengenolkan Angka Indeks.....	90
Gambar 4.11	Proses menyisipkan Genome yang Tersisa Pada Indeks.....	91
Gambar 4.12	Tampilan Sistem.....	93
Gambar 4.13	Tampilan Percobaan Ke-1 hingga Ke-30.....	95

DAFTAR LAMPIRAN

Lampiran 1. Data Mobil Tangki.....	119
Lampiran 2. Data Wilayah SPBU.....	120
Lampiran 3. Data Biaya Variabel	126
Lampiran 4. Grafik Nilai Fitness Tiap Generasi tiap Percobaan.....	127
Lampiran 5. Data Permintaan SPBU tanggal 17 April dan 3 Mei-7 Mei	128
Lampiran 6. Data Perbandingan Rute.....	132

**Algoritma Genetika Untuk Permasalahan *Routing* Armada Dengan
Pemerataan Beban (Studi Kasus PT. Patra Niaga Anak Perusahaan PT
Pertamina Depot Rewulu)**

Trio Yonathan Teja Kusuma
07660021

Program Studi Teknik Industri Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Sunan Kalijaga Yogyakarta

ABSTRAK

Vehicle Routing Problem (VRP) adalah sebuah permasalahan optimasi kombinatorial yang kompleks yang termasuk dalam kategori permasalahan non polinomial hard (NP-Hard Problem), Tujuan dari VRP adalah meminimalkan biaya yang diperlukan, dengan meminimasi jarak yang ditempuh, dan jarak berkaitan dengan penentuan rute. Penentuan rute juga harus memikirkan pemerataan beban kerja, agar permasalahan pengemudi yang berkaitan dengan jarak perjalanan dapat dihindari. Dalam VRP klasik diasumsikan bahwa permintaan dari pelanggan kurang dari atau sama dengan kapasitas kendaraan dan setiap pelanggan harus dilayani tepat satu kendaraan. Namun dalam kenyataannya pada PT Patra Niaga permintaan pelanggan terkadang lebih dari kapasitas mobil tangki. Untuk itu varian VRP yang digunakan adalah Split Delivery Vehicle Routing Problem. Karena varian ini memperbolehkan kendaraan mengunjungi pelanggan yang sama, lebih dari sekali dalam satu kali pemesanan. Lingkup VRP pada praktiknya sangatlah luas, sehingga menurut Thot and Vigo (1999) metode eksak sulit digunakan untuk menyelesaikannya. Penelitian akhirnya menggunakan metode heuristik, dan beberapa eksperimen telah membuktikan kehebatan Genetik Algorithm untuk menyelesaikan VRP. Dengan adanya kapasitas kendaraan maka proses pembentukan kromosom dalam Genetik Algorithm diharuskan untuk memperhatikannya, agar tidak terjadi mutan yang berlebih. Untuk itu muncul ide menggunakan teknik Vehicle Based Representation, Block Crossover, serta Constrained Insertion Mutation. Sistem yang dibuat dalam laporan ini mampu menemukan rute yang dapat pemeratakan beban kerja dengan minimasi biaya kirim. Dari hasil pengolahan data disimpulkan bahwa rute yang dihasilkan sistem, memiliki total error yang lebih rendah dari pada rute rancangan perusahaan dimana selisih error sebesar 33,31. Dan total biaya lebih mahal dari pada rute rancangan perusahaan dimana selisih biaya sebesar Rp 928.399,68. Dengan pemerataan beban kerja yang lebih rata diharapkan kondisi tiap mobil dapat lebih stabil, umur mobil tangki lebih panjang, mobil tangki tidak mudah rusak, serta ketidakpuasan pengemudi yang berkaitan dengan pemerataan beban kerja dapat dihindari.

Kata kunci : Split Delivery VRP, Genetik Algorithm, Vehicle Based Representation, Block Crossover, Constrained Insertion Mutasi



STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

BAB I

PENDAHULUAN

1.1 Latar Belakang

Patra Niaga adalah sebuah anak perusahaan Pertamina, yang salah satu tugasnya adalah mendistribusikan Bahan Bakar Minyak (BBM) ke SPBU-SPBU yang memesan. Dalam pengoperasiannya Patra Niaga harus mendistribusikan BBM dengan mobil tangki yang tersedia menuju SPBU-SPBU yang letaknya terpencar dan berjarak jauh dari depot. Untuk itu maka perusahaan harus teliti dalam penentuan rute tiap kendaraan, agar biaya pengiriman BBM dapat diminimalkan.

Salah satu masalah yang biasa muncul dalam pendistribusian barang adalah masalah penentuan rute kendaraan atau biasa disebut *vehicle routing problem* (VRP). Menurut Bräysy dan Gendreau (2005) tipe VRP dapat digambarkan sebagai suatu kasus dengan tujuan mencari rute terpendek dari suatu depot menuju sekumpulan titik-titik yang tersebar secara geografis (misal kota, toko, sekolah, SPBU).

Menurut Thot and Vigo (2002) VRP termasuk dalam kelas NP-hard problem dalam *combinatorial optimization*, sehingga sulit diselesaikan dengan metode eksak yang berlaku secara umum. Menurut Mutakhiroh et al (2007) pada umumnya solusi VRP dapat diperoleh dengan menggunakan metode *heuristik* dan metode *konvensional*. Meskipun penentuan rute dapat dicari dengan 2 metode namun dijelaskan lagi oleh Mutakhiroh et al (2007) bahwa dengan metode

heuristik hasil yang diperoleh lebih variatif dan waktu proses menjadi lebih singkat hingga 30 %.

Di dalam metode *heuristik* terdapat beberapa metode, salah satu yang dapat digunakan adalah Algoritma Genetika. Algoritma genetik merupakan metode *heuristik* yang berdasarkan pada mekanisme seleksi alam dan proses evolusi alam, menurut Goldberg dalam Sarwadi dan Anjar (2004). Menurut Faradian (2007) Algoritma genetik akan menghasilkan solusi yang lebih optimal pada setiap generasinya.

Diharapkan dengan metode algoritma genetik ini, rute pengiriman bahan bakar minyak (BBM) yang dihasilkan, dapat meminimalkan biaya kirim dan dapat pemeratakan beban kerja kendaraan maupun sopir.

1.2 Perumusan Masalah

Penelitian dilakukan untuk mendapatkan rute pengiriman BBM yang dapat pemeratakan beban kerja dan meminimalkan biaya kirim. Dengan beban kerja yang merata maka kinerja setiap mobil tangki dan sopir akan lebih mudah dikelola, dan dengan pengelolaan tersebut maka biaya-biaya yang berkaitan dengan pendistribusian BBM dapat diminimalkan. Untuk itu maka perumusan masalah yang dituangkan adalah

- 1) Bagaimana tahapan proses Algoritma Genetik pada sistem yang dibuat?
- 2) Bagaimana pemerataan beban dengan minimasi biaya pada rute hasil proses Algoritma genetik?

- 3) Bagaimana perbandingan biaya distribusi dan pemerataan beban antara rute hasil pengolahan sistem dengan rute rancangan perusahaan?

1.3 Batasan Masalah dan Asumsi

1.3.1 Batasan Masalah

Untuk menghindari permasalahan yang lebih luas dan agar tujuan pembahasan semakin terarah maka dilakukan pembatasan masalah sebagai berikut:

- 1) Data yang dibahas dalam penelitian ini mengenai data pengelolaan mobil tangki yang diperoleh dari PT Patra Niaga di PT Pertamina Unit Pemasaran Rewulu.
- 2) Masalah utama dalam pengendalian distribusi yang diatasi adalah menentukan pemerataan beban kerja dan minimasi biaya variabel untuk melakukan pengiriman bahan bakar minyak (BBM) jenis Premium.
- 3) Biaya yang dipakai adalah biaya variabel dalam pendistribusian
- 4) Armada yang digunakan adalah mobil tangki berkapasitas.
- 5) Faktor kepentingan antara minimasi biaya dengan pemerataan beban kerja didapat dari Supervisor Patra Niaga.
- 6) Metode yang digunakan untuk pencarian rute adalah Algoritma Genetik.
- 7) Program yang digunakan dalam perancangan sistem adalah Ms.Visual Basic 6.0 dan untuk perancangan data base menggunakan Ms.Access 2003.

1.3.2 Asumsi

- 1) Biaya tetap diabaikan, karena biaya yang berpengaruh pada perubahan jarak tempuh adalah biaya variabel.
- 2) Transportasi dalam keadaan normal.
- 3) Kendaraan dalam kondisi tidak rusak.

1.4 Tujuan

Tujuan yang hendak dicapai dalam penelitian ini adalah menentukan rute pendistribusian Bahan Bakar Minyak (BBM) yang mampu meratakan beban kerja dengan minimasi biaya kirim dan visualisasi yang nyata.

1.5 Manfaat penelitian

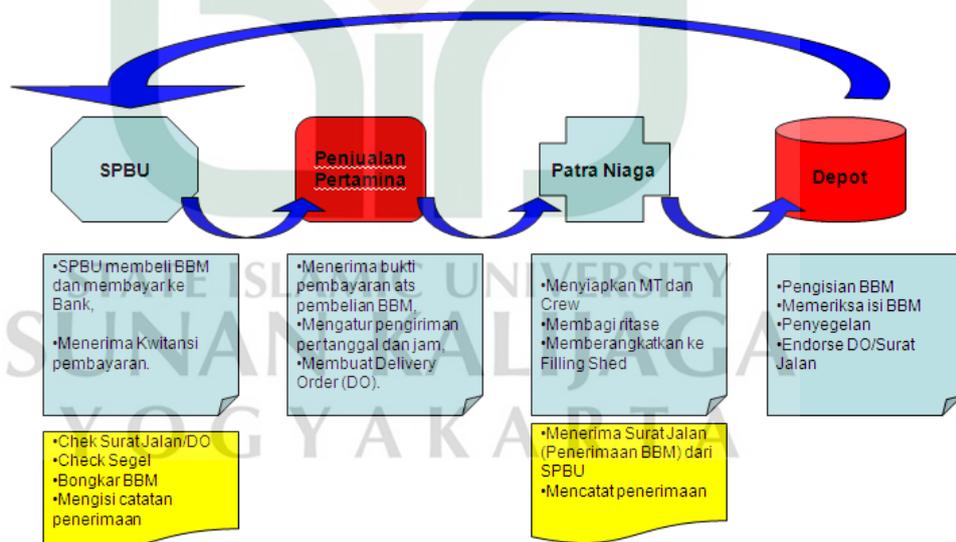
Dari hasil perancangan ini diharapkan mampu membantu perusahaan dalam penentuan rute untuk pendistribusian BBM, dan manfaat yang dapat diperoleh dari penelitian ini adalah:

- a. Dengan mengetahui rute yang baik dalam pengiriman BBM maka dapat meminimumkan biaya transportasi, meningkatkan efisiensi perawatan mobil tangki, pemerataan kinerja mobil tangki, hingga peningkatan kesejahteraan karyawan.
- b. Dengan adanya sistem penentuan rute maka akan membantu perusahaan dalam pengelolaan distribusi BBM dengan terotomatisasi.

Sebagai anak perusahaan yang sepenuhnya dimiliki oleh Pertamina, Indonesia's Oil nasional dan Perusahaan Gas, PT. Patra Niaga memiliki peranan yang unik dalam sektor Industri. Keunikannya adalah PT. Patra Niaga memiliki semua sektor yang dibutuhkan oleh induk perusahaan.

4.1.2 Proses Bisnis Patra Niaga pada Pendistribusian BBM

PT Patra Niaga disini bertugas dalam transportasi untuk menyalurkan BBM dan LPG dari depot kepada konsumen. Dengan selalu memberikan solusi transportasi yang terbaik kepada prinsipal dengan mendesign, mengimplementasikan, mengoperasikan serta mengelola kompleksitas rantai pasokan yang didukung oleh teknologi yang tepat guna. Proses bisnis Patra Niaga terangkum dalam gambar 7 dibawah ini.



Gambar 4.2 Proses Bisnis Patra Niaga

Sumber: Patra Niaga

Kegiatan pemenuhan order yang dilakukan PT PATRANIAGA ini diawali dari SPBU yang bersangkutan menebus (membeli BBM) dengan sejumlah uang di Bank. Di sana perwakilan dari SPBU tersebut harus mengisi aplikasi yang berisi pembagian BBM (dengan sejumlah uang tersebut SPBU menginginkan BBM dalam jenis apa saja dan dengan jumlah berapa).

Selanjutnya Bank melakukan pengiriman data (House to House) kepada Pertamina dengan bantuan *OSDS (Online Sale Distribution System)*, dan Bank memberikan nomer SO (Sales Order) kepada perwakilan SPBU tersebut. Data yang dikirim Bank diterima Pertamina pada bagian penjualan, kemudian setiap harinya SPBU yang bersangkutan melaporkan stock terakhir dan apakah SPBU tersebut menginginkan pengiriman BBM atau tidak melalui MS2.

Bila SPBU tersebut menginginkan pengiriman maka bagian penjualan mencetak data-data yang masuk (jumlah yang diinginkan untuk dikirim) dalam bentuk surat LO (Loading Order). Surat LO ini sebagai bukti bahwa SPBU yang bersangkutan telah melakukan penebusan dan masih memiliki persediaan BBM di Depot Rewulu, setelah itu tepat jam 02.00 siang MS2 ditutup. Selanjutnya dilakukan penjadwalan dengan menggunakan sistem.

Penjadwalan berdasarkan ketahanan stock dari masing-masing SPBU, sistem yang ada menjadwalkan pengiriman ke dalam 4 shift, yaitu shift 1 hingga shift 4. LO kemudian dikirim menuju bagian Distribusi

untuk ditindak lanjuti (dilayani) dalam bentuk uang menjadi barang (BBM), semua LO direkap kemudian diketahui berapa BBM yang akan keluar untuk esok hari, dan data tersebut digunakan sebagai informasi pada bagian penimbunan.

Setelah LO direkap kemudian LO disalurkan menuju PATRANIAGA untuk pengantaran sampai ke SPBU yang bersangkutan. Pada PATRANIAGA dilakukan penentuan rute setiap shiftnya (dari penjadwalan yang dilakukan OSDS) untuk setiap mobil tangki. Mobil-mobil tangki yang telah selesai melakukan pengisian muatan, selanjutnya menuju PATRANIAGA untuk mengetahui rute yang ditentukan dan mengambil LO sebagai bukti pengiriman menuju SPBU yang dimaksud. Setelah mobil tangki sampai di SPBU yang dimaksud LO dan segel dicek selanjutnya BBM dibongkar.

4.1.3 Dasar Proses Bisnis Patra Niaga

Dalam menjalankan proses bisnisnya Patra Niaga memegang dasar yang telah disepakati oleh pihak-pihak yang terkait, dimana dasar tersebut adalah :

- 1) Keputusan Rapat Evaluasi Pengelolaan Mobil Tangki Pola Baru PT. PERTAMINA (PERSERO) di Hotel Aryaduta, Medan pada tanggal 17-18 Desember 2007,
- 2) Surat Manager *Quality & Quantity* Suplai & Distribusi Direktorat Pemasaran & Niaga PT. PERTAMINA (PERSERO) nomor

176/F30130/2007-S0 tanggal 28 Desember 2008 perihal Persyaratan Pengelolaan Mobil Tangki,

- 3) Surat Deputi Direktur Distribusi Direktorat Pemasaran & Niaga nomor 018/F30000/2008-S3 tanggal 23 Maret 2008 perihal RFQ Jasa Kelola *All-In* Mobil Tangki,
- 4) Surat *Vice President* Suplai & Distribusi Direktorat Pemasaran & Niaga nomor 0661/F30100/2008-S3 tanggal 16 April 2008 perihal Jasa Kelola *All-In* Mobil Tangki & TAS Instalasi Tanjung Perak, dan
- 5) Surat Keputusan Direktur Pemasaran & Niaga nomor Kpts-0661/F00000/2008-S0 tanggal 22 April 2008 perihal Tarif *All-In* Jasa Kelola Mobil Tangki Angkutan BBM/BBK ke SPBU/Industri.

4.1.4 Tujuan Perusahaan

Tujuan atau target yang ingin dicapai oleh PT Patra Niaga dalam menjalankan tugasnya adalah :

- 1) Peningkatan *service level*, dan efisiensi biaya pengelolaan mobil tangki,
- 2) *Zero Loss* (Tepat Mutu, Tepat Jumlah, Tepat Waktu dan Tempat Tempat),
- 3) *Fuel Consumption at cost*,
- 4) Pelatihan SDM (sopir, kernet, mekanik,dll),
- 5) Peningkatan kesejahteraan awak dan *supporting* Mobil Tangki.

4.1.5 Visi dan Misi Perusahaan

Visi yang dimiliki oleh PT Patra Niaga adalah

“Menjadi pelopor perubahan terdepan dalam inovasi transportasi Migas di Indonesia”

Misi yang dimiliki PT Patra Niaga adalah

Patra Niaga akan mencapai Visi perusahaan dengan selalu memberikan solusi transportasi yang terbaik kepada prinsipal dengan mendesign, mengimplementasikan, mengoperasikan serta mengelola kompleksitas rantai pasokan yang didukung oleh teknologi yang tepat guna.

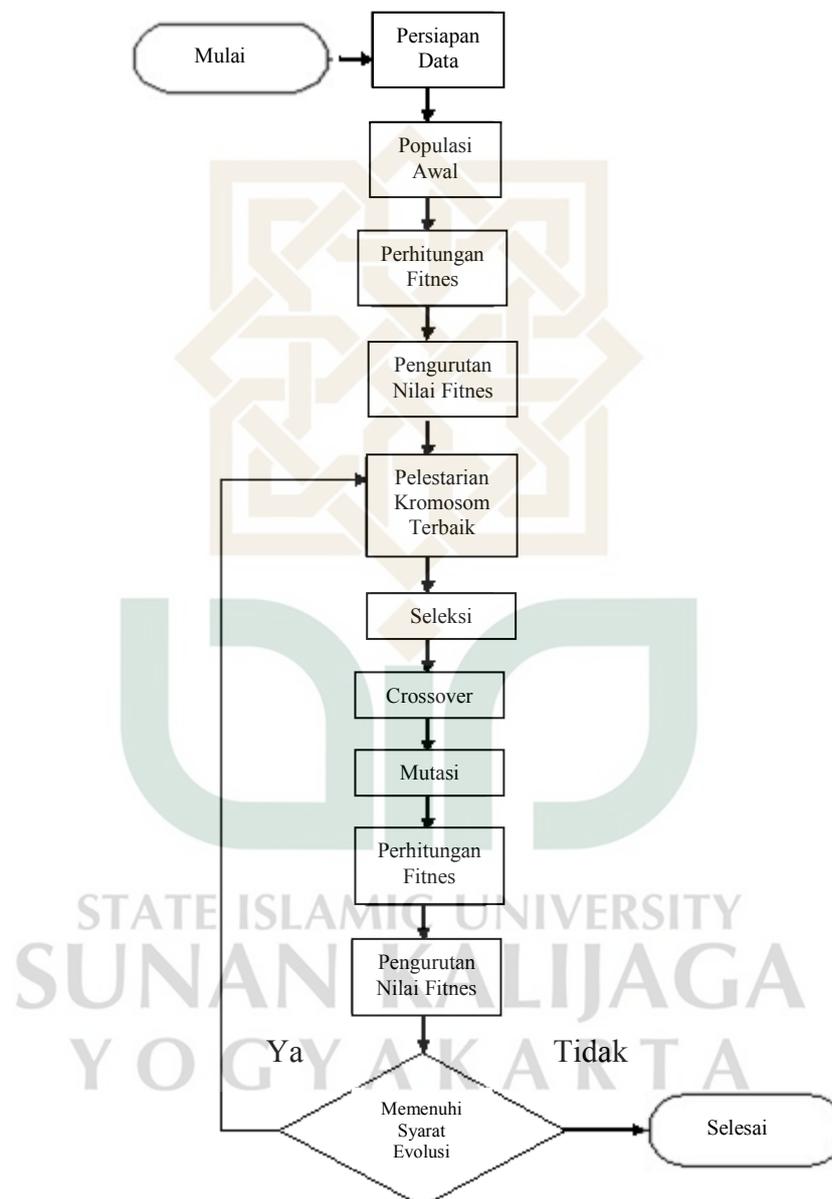
4.2 Pengolahan Data

4.2.1 Genetik Algoritma

Metode ini dipilih karena bila dibandingkan dengan metode heuristik yang lain metode ini lebih simpel dalam melakukan proses. Karena kromosom yang diproses adalah kromosom hasil populasi awal dan memperbolehkan adanya kromosom yang sama. Dalam GA nantinya ada proses penyilangan yang membuat kromosom menjadi berbeda. Bila pada metode heuristik yang lain misal pada tabu search, proses berjalan dengan melihat kromosom pada iterasi sebelumnya. Jika kromosom yang terbentuk sudah ada maka kromosom tersebut diganti. Hal ini membuat proses lebih lama. Pada Simulated Annealing proses berjalan dengan penurunan suhu. Suhu merepresentasikan jumlah kromosom pada populasi. Proses berjalan dengan selalu mengubah jumlah populasi setelah

populasi sebelumnya selesai diproses. Hal ini membuat proses berjalan lama karena harus memproses populasi baru.

4.2.2 Proses *Coding*



Gambar 4.3 Diagram Alir Proses Algoritma Genetik dalam Sistem

1) Persiapan Data

Data yang telah ada disiapkan dalam sistem, dengan mengkodekan setiap data. Dimana kode yang dibuat tercantum pada tabel 2 di bawah ini.

Tabel 4.1 Kode Persiapan Data

```

JumTruk = 31
JumSPBU = List1.ListCount

For i = 1 To JumTruk
  TrukKapasitas(i) = List3.List(i - 1)
Next i

For i = 1 To JumSPBU
  SPBUJarak(i) = List1.List(i - 1)
Next i

For i = 1 To JumSPBU
  SPBU(i) = List6.List(i - 1)
Next i

For a = 1 To JumSPBU
  SPBUMinta(a) = List2.List(a - 1)
Next a

For a = 1 To List4.ListCount
  selkode(a) = List4.List(a - 1)
  jarsel(a) = List5.List(a - 1)
Next a

```

JumTruk adalah kode yang menunjukkan jumlah mobil tangki yang dimiliki Patra Niaga, dimana jumlah mobil tangki sebanyak 31 mobil. JumSPBU kode yang menunjukkan Jumlah SPBU yang memesan atau yang akan diproses, jumlah SPBU tersebut diambil dari jumlah SPBU yang muncul di List1.

TrukKapasitas adalah kode yang menunjukkan kapasitas mobil tangki yang dimiliki Patra Niaga, kapasitas setiap mobil tangki tersebut diambil dari List3. Proses perulangan sebanyak jumlah mobil tangki akan menunjukkan kapasitas setiap mobil tangki, sebagai contoh saat $i = 1$ maka proses akan bekerja untuk TrukKapasitas (1) dan List3.list (1-1), sehingga TrukKapasitas (1) atau kapasitas mobil tangki pertama yang diperoleh pada List3.list (0).

SPBUJarak adalah kode yang menunjukkan jarak SPBU dengan depot rewulu, jarak SPBU tersebut diambil dari List1. Proses perulangan sebanyak jumlah SPBU yang memesan akan menunjukkan jarak setiap SPBU, sebagai contoh saat $i = 1$ maka proses akan bekerja untuk SPBUJarak (1) dan List1.list (1-1), sehingga SPBUJarak (1) atau jarak SPBU pertama diperoleh pada List1.list (0).

SPBU () adalah kode untuk menunjukkan kode SPBU yang memesan, dan kode SPBU tersebut didapat pada List6. Proses perulangan sebanyak jumlah SPBU yang memesan akan menunjukkan kode SPBU, sebagai contoh saat $i = 1$ maka proses akan bekerja untuk SPBU (1) dan List6.list (1-1), sehingga SPBU (1) atau kode SPBU pertama diperoleh pada List6.list (0).

SPBUMinta () adalah kode untuk menunjukkan kapasitas permintaan setiap SPBU yang memesan, dan kapasitas permintaan

tersebut didapat pada List2. Proses perulangan sebanyak jumlah SPBU yang memesan akan menunjukkan kapasitas permintaan setiap SPBU, sebagai contoh saat $a = 1$ maka proses akan bekerja untuk SPBU (1) dan List2.list (1-1), sehingga kapasitas permintaan SPBU (1) atau kapasitas permintaan SPBU pertama diperoleh pada List2.list (0). Kapasitas permintaan SPBU ditetapkan hanya 8000 liter dan kelipatannya hingga maksimal 42000 liter (aturan perusahaan).

Selkode () adalah kode yang menunjukkan kode perjalanan dari SPBU satu ke SPBU selanjutnya, dan jarsel () menunjukkan jarak antar SPBU yang memesan. Proses perulangan sebanyak kode yang ada di List4 akan menunjukkan kode perjalanan dari SPBU satu ke SPBU selanjutnya dan jarak antar SPBU yang memesan, sebagai contoh saat $a = 1$ maka proses akan bekerja untuk selkode (1) dan List4.list (1-1), jarsel (1) dan List5(1-1), sehingga selkode (1) atau kode perjalanan dari SPBU satu ke SPBU selanjutnya yang pertama diperoleh pada List4.list (0), serta jarsel (1) atau jarak dari SPBU 1 ke SPBU selanjutnya yang pertama didapat pada list5.list (0).

2) Pembentukan populasi pertama

Proses pembentukan populasi awal ini menggunakan metode *Vehicle Based Representation*. Metode ini merupakan pengembangan dari metode *random generator*. Kromosom yang

akan dibuat dengan metode ini adalah kromosom yang memiliki *block-block*. *Block-block* tersebut bertindak sebagai mobil tangki yang digunakan. Dengan adanya *block* yang berkapasitas dan *genome* yang berkapasitas maka secara otomatis terdapat batasan jumlah muatan (*genome*) yang bisa ditampung dalam satu *block*. Proses pengisian *genome* kedalam kromosom tetap menggunakan bilangan acak namun berbeda dari random generator yang pengisiannya tidak memperhitungkan kapasitas. Dalam metode ini pengisian *genome* kedalam kromosom dengan memperhitungkan kapasitas *Block*. Jadi kromosom hasil pengolahan dengan metode *Part in block* ini adalah *kromosom* yang memiliki N buah *block* yang didalam satu *block* terdapat N buah *genome*.

Proses pada tabel 3 adalah proses untuk membuat populasi awal, yaitu populasi yang akan diproses, kode yang ada sebagai berikut :

Tabel 4.2 Kode Pembentukan populasi pertama

```

For i = 1 To PopuMain.NumOfIndivid 'To generate random genome for every
individual
  DoEvents
  ReDim PopuMain.Individuals(i).Genome(PopuMain.GenomeLen)

PosKromosom = 1
KumKap = 0
kumjar = 0
SisaSPBU = JumSPBU
For j = 1 To JumSPBU
  SPBUBool(j) = False
Next j

For j = 1 To JumSPBU
  DoEvents
  BilRand = random(1, SisaSPBU)
  C1 = 0
  C2 = 0

```

Lanjutan tabel 4.3 Kode Pembentukan populasi pertama

```

Do While C1 < BilRand
DoEvents
  C2 = C2 + 1
  If SPBUBool(C2) = False Then
    C1 = C1 + 1
  End If
Loop

if C1 = BilRand Then
  SPBUBool(C2) = True
End If

KumKap = KumKap + SPBUMinta(C2)
If KumKap <= TrukKapasitas(Round((PosKromosom + 1.5) / 4)) Then
  PosKromosom = PosKromosom
  KumKap = KumKap
Else
lagi:
  jum = Round((PosKromosom + 1.5) / 4) * 4 + 1
  PosKromosom = jum
  KumKap = SPBUMinta(C2)

  If KumKap <= TrukKapasitas(Round((PosKromosom + 1.5) / 4)) Then
    PosKromosom = PosKromosom
    KumKap = KumKap
  Else
    GoTo lagi
  End If
End If
PopuMain.Individuals(i).Genome(PosKromosom) = C2

PosKromosom = PosKromosom + 1

SisaSPBU = SisaSPBU - 1
Next j

```

Kode pada tabel 4.3 di atas adalah kode yang akan membentuk populasi pertama, Populasi pertama berjumlah sebanyak $\text{PopuMain.NumOfIndivid}$ (kode yang menunjukkan jumlah kromosom), dimana jumlahnya ditentukan oleh pengguna sistem. Kromosom pada populasi pertama dibuat dengan teknik random. Maksudnya adalah pemilihan SPBU yang akan masuk proses penentuan rute mobil tangki, dilakukan secara acak, proses ini dimulai dari kromosom pertama hingga terakhir.

Setelah didapat, SPBU terpilih diproses dengan tetap memperhatikan kapasitas mobil tangki. Proses dimulai dari mobil tangki pertama, bila kapasitas mobil tangki pertama mencukupi untuk mengirim ke SPBU terpilih maka SPBU tersebut dikirim dengan mobil tangki pertama. Bila mobil tangki pertama tidak mencukupi kapasitasnya maka proses dilanjutkan ke mobil tangki selanjutnya hingga ditemukan mobil tangki yang berkapasitas mencukupi. Begitu seterusnya hingga semua SPBU terproses.

Untuk menanggulangi SPBU yang telah terpilih kembali terpilih lagi, maka digunakan fungsi Boolean. Dimana pada awalnya setiap SPBU bernilai "False" dan jumlah angka random sebanyak jumlah SPBU yang memesan. Setiap angka menunjukkan SPBU pemesan yang bernilai "False" (contoh = angka 1 untuk SPBU pertama yang bernilai "False"). Bila sebuah SPBU terpilih maka nilai "False" pada SPBU tersebut berubah menjadi "True" dan jumlah SPBU yang dirandom dikurangi 1.

Proses pemilihan SPBU hanya dilakukan pada SPBU yang bernilai "False" sehingga SPBU yang telah terpilih tidak dapat terpilih lagi. Bila semua SPBU telah terproses maka masuk ke kromosom selanjutnya hingga semua kromosom terproses.

Setiap mobil tangki maksimal melakukan pengiriman menuju empat SPBU, hal ini dikarenakan terdapat mobil tangki yang berkapasitas 32.000 liter dan setiap SPBU hanya boleh memesan

pada kapasitas 8000 liter dan kelipatan 8000 liter (aturan dari perusahaan). Di dalam mobil tangki tersebut terdapat sekat yang membagi kapasitas mobil tangki menjadi per- 8000 liter jadi maksimal mobil tangki dapat mendistribusikan BBM menuju 4 SPBU yang memesan dengan kapasitas 8000 liter yaitu untuk mobil tangki dengan kapasitas 32000 liter.

Proses pembentukan populasi pertama berjalan dengan simulasi pada gambar 9 dan gambar 10.

jumlah kromosom dalam populasi = 2

jumlah SPBU pemesan = 2

kapasitas SPBU 1 = 8 KL

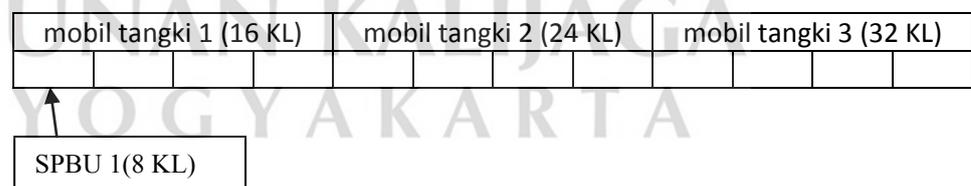
kapasitas SPBU 2 = 16 KL

(1) Kromosom pertama

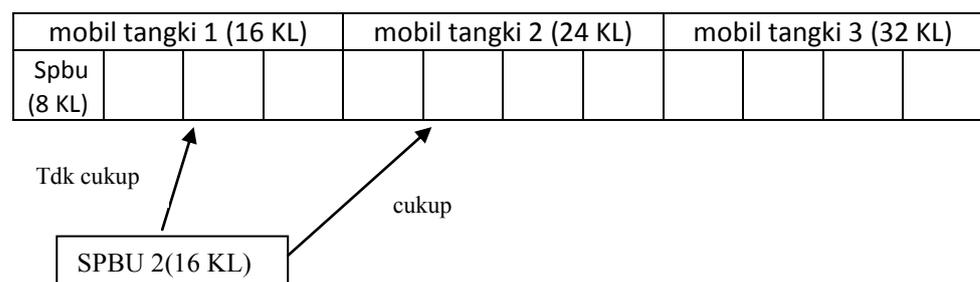
Random pertama = 1 (SPBU pertama diproses)

Random kedua = 2 (SPBU kedua diproses)

a) Peletakkan SPBU pertama



b) Peletakkan SPBU kedua



mobil tangki 1 (16 KL)				mobil tangki 2 (24 KL)				mobil tangki 3 (32 KL)			
SPBU (8 KL)				SPBU (16)							

Gambar 4.4 Simulasi Proses Pembentukan Populasi Awal Untuk
Kromosom Pertama

(2) Kromosom kedua

Random pertama = 2 (SPBU kedua diproses)

Random kedua = 1 (SPBU kedua diproses)

a) Peletakkan SPBU kedua



b) Peletakkan SPBU pertama



mobil tangki 1 (16 KL)				mobil tangki 2 (24 KL)				mobil tangki 3 (32 KL)			
SPBU (16)				SPBU (8 KL)							

Gambar 4.5 Simulasi Proses Pembentukan Populasi Awal Untuk Kromosom

Kedua

3) Fitness

Terdapat dua macam fitness yang akan diproses yaitu :

a) Fitness untuk minimasi biaya

Dalam buku Ekonomi Teknik (Giatman,2005) menyatakan

$$\text{Biaya Total} = \text{Biaya tetap} + \text{Biaya Variabel} \quad (11)$$

Dalam penelitian ini Biaya tetap = 0, ini dikarenakan dalam pendistribusian biaya yang sangat berpengaruh adalah biaya variabel. Biaya ini selalu berubah setiap perubahan jarak yang terjadi, sehingga dalam penelitian ini yang akan diteliti adalah hanya biaya variabel saja.

$$\text{Biaya Variabel} = S + P + B + G_s + G_k \quad (12)$$

$$S = T_j \times S_k \times J \quad (13)$$

$$P = T_j \times P_k \quad (14)$$

$$B = T_j / J_1 \times L \quad (15)$$

$$G_s = T_j - 60 \times G_{sk} \quad (16)$$

$$G_k = T_j - 60 \times G_{kk} \quad (17)$$

Pengiriman lebih dari satu SPBU

$$T_j = J_{da} + J_{ab} + J_{db} \quad (18)$$

Pengiriman satu SPBU

$$T_j = J_{da} \times 2 \quad (19)$$

Dimana :

S = Biaya Sewa Ban

P = Biaya Pemakaian Pelumas

B = Biaya Bahan Bakar

G_s = Gaji Sopir yang Harus Dibayar

G_k = Gaji Kondaktur yang Harus Dibayar

T_j = Total Jarak yang Ditempuh

Sk	=	Biaya Sewa Ban untuk 1 km Perjalanan
J	=	Jumlah Ban yang Dipakai Mobil Tangki
Pk	=	Biaya Pelumas untuk 1 km Perjalanan
Jl	=	Jarak yang Ditempuh dengan 1 Liter Solar
L	=	Biaya 1 Liter Solar
Gsk	=	Gaji Sopir Untuk 1 Km Perjalanan
Gkk	=	Gaji kondektur untuk 1 Km Perjalanan
Jda	=	Jarak Depot dengan SPBU Pertama Dituju
Jab	=	Jarak SPBU Satu dengan SPBU lanjutnya
Jdb	=	Jarak Depot dengan SPBU terakhir

Tabel 4.5 Kode perhitungan nilai fitness

```

For i = 1 To PopuMain.NumOfIndivid
  For j = 1 To JumTruk * 4
    jarkum(i, Round((j + 1.5) / 4)) = 0
  Next j
Next i

  For i = 1 To 1
    jum = 0
    jum = jumlahkodesel(i)
  Next i

  For i = 1 To PopuMain.NumOfIndivid
    kumjar = 0
    For j = 1 To JumTruk * 4
      kodesel = ""
      If SPBUJarak(PopuMain.Individuals(i).Genome(j)) <> 0 Then
        If Round((j + 1.5) / 4) = Round(((j - 1) + 1.5) / 4) Then
          kumjar = kumjar +
            SPBUJarak(PopuMain.Individuals(i).Genome(j))
          kodesel = SPBU(PopuMain.Individuals(i).Genome(j - 1)) & "-" &
            SPBU(PopuMain.Individuals(i).Genome(j))
        End If
      End If
    Next j
  Next i

```

Lanjutan tabel 4.5 Kode perhitungan nilai fitness

```

For s = 1 To jum 'dipanggil dari listcount
If kodesel = selkode(s) Then
    seljar = 0
    seljar = jarsel(s)
End If
Next s
truk = Round((j + 1.5) / 4)
If j = truk * 4 Then
    kumjar = kumjar + seljar
    jarkum(i, Round((j + 1.5) / 4)) = kumjar
Else
    kumjar = kumjar - SPBUJarak(PopuMain.Individuals(i).Genome(j)) + seljar
    jarkum(i, Round((j + 1.5) / 4)) = kumjar
End If
Else
    kumjar = 0
    kumjar = kumjar + SPBUJarak(PopuMain.Individuals(i).Genome(j))
    jarkum(i, Round((j + 1.5) / 4)) = kumjar
End If
Else
    If j = Round((j + 1.5) / 4) + 1 Then
        kumjar = kumjar * 2
        jarkum(i, Round((j + 1.5) / 4)) = kumjar
    ElseIf Round((j + 1.5) / 4) <> Round(((j - 1) + 1.5) / 4) Then
        kumjar = 0
        jarkum(i, Round((j + 1.5) / 4)) = kumjar
    Else
        kumjar = kumjar + SPBUJarak(PopuMain.Individuals(i).Genome(j - 1))
        jarkum(i, Round((j + 1.5) / 4)) = kumjar
    End If
End If
Next j
Next i

```

Kode pada tabel 4.5 di atas adalah kode untuk menghasilkan

total jarak pada setiap mobil tangki. Dimana kode untuk total jarak adalah $\text{jarkum}(i, \text{Round}(j+1.5)/4)$. Kode di atas bekerja dengan proses sebagai berikut :

(1) Untuk pengiriman ke lebih dari satu SPBU dalam satu mobil tangki

(a) Pengiriman untuk dua SPBU

$$\text{Jarak} = J_{d1} + J_{d2} + J_{d2} \quad (20)$$

(b) Pengiriman untuk tiga SPBU

$$\text{Jarak} = J_{d1} + J_{12} - J_{d2} + J_{23} + J_{d3} \quad (21)$$

(c) Pengiriman untuk empat SPBU

$$\text{Jarak} = J_{d1} + J_{12} - J_{d2} + J_{23} - J_{d3} + J_{34} + J_{d4} \quad (22)$$

dimana :

J_{d1} = Jarak Depot dengan SPBU pertama

J_{12} = Jarak SPBU Pertama dengan SPBU Kedua

J_{d2} = Jarak Depot dengan SPBU Kedua

J_{23} = Jarak SPBU Kedua dengan SPBU Ketiga

J_{d3} = Jarak Depot dengan SPBU Ketiga

J_{34} = Jarak SPBU Ketiga dengan SPBU Empat

J_{d4} = Jarak Depot dengan SPBU Keempat

Untuk pengiriman hanya ke satu SPBU

$$\text{Jarak} = J_{d1} \times 2 \quad (23)$$

dimana :

J_{d1} = Jarak depot dengan SPBU pertama

Tabel 4.6 Kode untuk Fitness Minimasi Biaya

```

For i = 1 To PopuMain.NumOfIndivid
  kumjar = 0
  For j = 1 To JumTruk

    If jarkum(i, j) <> 0 Then

      totalbiaya(i, j) = 0

      If TrukKapasitas(j) = 16000 Then
        ban = 0
        pelumas = 0
        ban = 46 * 10 * jarkum(i, j)
        pelumas = 135 * jarkum(i, j)
        bensin = (jarkum(i, j) / 3.2) * 4300
    
```

Lanjutan tabel 4.6 Kode untuk Fitness Minimasi Biaya

```

ElseIf TrukKapasitas(j) = 24000 Then
ban = 0
pelumas = 0
ban = 46 * 14 * jarkum(i, j)
pelumas = 135 * jarkum(i, j)
bensin = (jarkum(i, j) / 2.7) * 4300
ElseIf TrukKapasitas(j) = 32000 Then
ban = 0
pelumas = 0
ban = 46 * 16 * jarkum(i, j)
pelumas = 135 * jarkum(i, j)
bensin = (jarkum(i, j) / 2.5) * 4300
End If
If jarkum(i, j) > 60 Then
gajisopir = 0
gajikenek = 0
gajisopir = (jarkum(i, j) - 60) * 160
gajikenek = (jarkum(i, j) - 60) * 96
Else
gajisopir = 0
gajikenek = 0
End If

totalbiaya(i, j) = ban + pelumas + bensin + gajisopir + gajikenek
kumjar = kumjar + totalbiaya(i, j)
End If
Next j
arrfitnes(i) = kumjar
arrkumprob(i) = kumjar
'str9 = str9 & arrfitnes(i) & "-"
Next i

```

Kode pada tabel 4.6 di atas adalah kode untuk menentukan biaya keseluruhan dari seluruh mobil tangki yang beroperasi.

Dimana terdapat proses percabangan pada biaya ban dan bahan bakar, yaitu :

(1) Untuk mobil tangki yang berkapasitas 16000 liter

Ban yang digunakan = 10 buah

1 liter bahan bakar untuk 3,2 kilometer

(2) Mobil tangki yang berkapasitas 24000 liter

Ban yang digunakan = 14 buah

1 liter bahan bakar untuk 2,7 kilometer

(3) Mobil tangki yang berkapasitas 32000 liter

Ban yang digunakan = 16 buah

1 liter bahan bakar untuk 2.5 kilometer

Biaya total untuk satu kromosom dikodekan dengan Arrfitnes

(i). Proses berjalan dengan cara menjumlahkan biaya setiap mobil pada setiap kromosom. Biaya setiap mobil dikodekan dengan Totalbiaya (i,j), total biaya ini didapat dari :

(1) Jika jarak yang ditempuh melebihi 60 kilometer maka

$$\begin{aligned} \text{Totalbiaya} &= (T_j \times S_k \times J) + (T_j \times P_k) + (T_j / J_l \times L) \\ &+ (T_j - 60 \times G_{sk}) + (T_j - 60 \times G_{kk}) \end{aligned} \quad (24)$$

(2) Jika jarak yang ditempuh tidak lebih dari 60 kilometer maka

$$\begin{aligned} \text{Total biaya} &= (T_j \times S_k \times J) + (T_j \times P_k) + (T_j / J_l \times L) \\ &+ 0 + 0 \end{aligned} \quad (25)$$

Biaya untuk satu kromosom (Arrfitnes (i))

$$T_b = \sum \text{Total Biaya}$$

T_b = Total Biaya untuk Satu Kromosom

b) Fitness untuk pemerataan beban kerja

Fitnes ini ditunjukkan untuk pemerataan beban kerja dengan kode sebagai berikut :

Tabel 4.7 Kode Fitnes Pemerataan Beban

```

For i = 1 To PopuMain.NumOfIndivid
  kumjar = 0
  For j = 1 To JumTruk
    If jarkum(i, j) <> 0 Then
      kumjar = kumjar + jarkum(i, j)
    End If
  Next j
  totaljarak(i) = 0
  totaljarak(i) = kumjar
Next i

For i = 1 To PopuMain.NumOfIndivid
  kumjar = 0
  trukpakai(i) = 0
  For j = 1 To JumTruk
    If jarkum(i, j) <> 0 Then
      jumpakai(j) = 1
      kumjar = kumjar + jumpakai(j)
    End If
  Next j
  trukpakai(i) = kumjar
Next i

For i = 1 To PopuMain.NumOfIndivid
  ratarata(i) = 0
  ratarata(i) = totaljarak(i) / trukpakai(i)
Next i

For i = 1 To PopuMain.NumOfIndivid
  kumsel = 0
  selisihkum(i) = 0
  For j = 1 To JumTruk
    selsih2(i, j) = 0
    If jarkum(i, j) <> 0 Then
      selsih = 0
      selsih = jarkum(i, j) - ratarata(i)
      If selsih < 0 Then
        selsih = selsih * -1
      End If
      selsih2(i, j) = selsih / trukpakai(i)
      kumsel = kumsel + selsih2(i, j)
    End If
  Next j
  arrfitnesbar(i) = kumsel
  arrkumprob1(i) = kumsel
Next i

```

Proses pada tabel 4.7 di atas akan berjalan sebagai berikut :

(1) Menentukan total jarak (kode = totaljarak(i)) dalam setiap

kromosom

$$T_j = \sum_{i=1}^n J_{mi} \quad (26)$$

J_m = jarak total setiap mobil tangki

- (2) Menentukan jumlah mobil tangki yang dipakai

Dengan jalan mencari jarak mobil yang tidak bernilai nol, kemudian memberikan angka 1 pada setiap mobil yang tidak bernilai nol. Selanjutnya menjumlahkan angka pada setiap mobil tangki yang tidak bernilai nol dan hasilnya dikodekan dengan $trukpakai(i)$.

- (3) Menentukan rata-rata jarak (Ratarata (i)) setiap kromosom

$$R = T_j / M_p \quad (27)$$

R = Rata-rata jarak

M_p = Jumlah mobil tangki yang dipakai

- (4) Menentukan selisih (selsih) jarak total setiap mobil ($jarkum(i, j)$) dengan rata-rata jarak (rata-rata(i)) pada kromosom yang sama, dengan algoritma

$$S = J_m - R \quad (28)$$

S = Selisih jarak total mobil dengan jarak rata-rata

jika hasil berharga negative maka dikalikan dengan -1 agar nilai positif

- (5) Membagi hasil dari proses 4 (selsih) dengan jumlah truk yang dipakai, $selsih2(i, j)$, dengan algoritma sebagai berikut

$$E = S / M_p \quad (29)$$

E = Selisih setiap jarak mobil tangki dengan rata-rata (*error*)

- (6) Menentukan total hasil ($arrfitnessbar(i)$) setiap kromosom

$$T_h = \sum_{i=1}^n E \quad (30)$$

$T_h = \text{Total hasil}$

c) *Fitness* gabungan dari minimasi biaya dengan pemerataan beban kerja

Untuk menjumlahkan antara *fitness* pemerataan beban kerja dengan minimasi biaya, maka harus dicari faktor perkalian untuk menyamakan satuan. Faktor perkalian tersebut dicari dari data historis total jarak pengiriman dan biaya total setiap mobil tangki selama 15 hari (data terlampir) sebagai contoh data pada tabel.2.

Data total jarak selanjutnya diproses untuk pencarian error dengan pendekatan *Mean Absolut Deviation* (MAD). Setelah itu data jarak hasil MAD tertinggi dibandingkan dengan data total biaya tertinggi. Hasil perbandingan setiap hari dikumulatitkan dan dicari rata-rata, hasil dari rata-rata tersebut digunakan sebagai faktor perkalian. Proses dilakukan dengan bantuan *microsoft Excel*. Tabel 4.8 adalah tabel data yang digunakan dalam perhitungan dan hasil perhitungan tercantum pada tabel 4.9.

Tabel 4.8 Data historis total jarak dan biaya pengiriman tanggal 11

Maret 2011

Tanggal	NOPOL	Jenis Sewa	Ritase	Total KM	biaya (Rp)
11/03/2011	AB8031AS	Allin	3	0	0
11/03/2011	AB9142EH	Allin	2	258	513510
11/03/2011	AB90012EH	Allin	3	0	0
Lanjutan tabel 8		Allin	4	248	489560
11/03/2011	AB9142FH	Allin	3	0	0
11/03/2011	AB9246YH	Allin	4	212	418640

11/03/2011	AB9286EH	Allin	3	206	406070
11/03/2011	AB9317EH	Allin	4	284	560480
11/03/2011	AB9349EH	Allin	3	0	0
11/03/2011	AB9351EH	Allin	3	270	538650
11/03/2011	AB9423FH	Allin	3	298	591310
11/03/2011	AB9642EH	Allin	3	256	507820
11/03/2011	AB9643EH	Allin	3	290	573050
11/03/2011	AD1534MC	Allin	2	0	0
11/03/2011	AD1649MC	Allin	4	196	436120
11/03/2011	AD1962MC	Allin	4	270	615150
11/03/2011	G1907AE	Allin	4	218	444710
11/03/2011	G1927AE	Allin	4	260	577700
11/03/2011	H1429VW	Allin	3	0	0
11/03/2011	H1430VW	Allin	3	306	605070
11/03/2011	H1430ZS	Allin	3	320	635900
11/03/2011	H1439ZS	Allin	4	218	431210
11/03/2011	H1443ZS	Allin	4	310	634450
11/03/2011	H1446ZS	Allin	3	0	0
11/03/2011	H1449ZS	Allin	4	250	490750
11/03/2011	H1488YW	Allin	4	288	567360
11/03/2011	H1520YY	Allin	4	310	611950
11/03/2011	H1521YY	Allin	3	0	0
11/03/2011	H1644YW	Allin	3	288	567360
11/03/2011	H1783BY	Allin	3	252	664992
11/03/2011	H1848DH	Allin	3	0	0
11/03/2011	H1863BY	Allin	3	314	623330
11/03/2011	H1869DH	Allin	4	252	632808
11/03/2011	H1955GA	Allin	3	282	559290
11/03/2011	R1753AK	Allin	3	212	562652
11/03/2011	R1763CB	Allin	3	282	559290
11/03/2011	R1766CB	Allin	2	0	0
11/03/2011	R1769CB	Allin	3	0	0
11/03/2011	R1773CB	Allin	3	256	507820
11/03/2011	R1914BB	Allin	4	0	0
11/03/2011	AB2301CA	Allin	3	320	635900
11/03/2011	AB2298CA	Allin	4	214	531206
11/03/2011	AB2297CA	Allin	4	0	0
11/03/2011	AB2293CA	Allin	4	194	466126

Untuk mobil tangki yang berjarak nol (0) dan biaya yang bernilai (0) adalah mobil tangki untuk pengiriman solar.

Tabel 4.9 Data urutan biaya dan hasil pengolahan jarak dengan pendekatan MAD (persamaan 4)

Urutan Total biaya	Urutan Total jarak	Urutan Hasil MAD
664992	320	1,858481
635900	320	1,858481
635900	314	1,664932
634450	310	1,5359
632808	310	1,5359
623330	306	1,406868
615150	298	1,148803
611950	290	0,890739
605070	288	0,826223
591310	288	0,826223
577700	284	0,69719
573050	282	0,632674
567360	282	0,632674
567360	270	0,245578
562652	270	0,245578
560480	260	-0,077
559290	258	-0,14152
559290	256	-0,20604
538650	256	-0,20604
531206	252	-0,33507
513510	252	-0,33507
507820	250	-0,39958
507820	248	-0,4641
490750	218	-1,43184
489560	218	-1,43184
466126	214	-1,56087
444710	212	-1,62539
436120	212	-1,62539
431210	206	-1,81894
418640	196	-2,14152
406070	194	-2,20604

Kolom pertama pada tabel 4.9 di atas adalah kolom yang menyatakan urutan total biaya dari yang terbesar hingga terkecil. Kolom kedua menunjukkan urutan total jarak dari yang terbesar hingga terkecil. Kolom ke tiga menunjukkan urutan hasil pendekatan MAD dari yang terbesar hingga terkecil.

Dari data tersebut selanjutnya dicari perbandingan yaitu hasil pendekatan MAD terbesar dibagi dengan total biaya terbesar.

Faktor perkalian tanggal 11 Maret 2011 = Rp 664.992,00 / 1,858481

Faktor perkalian tanggal 11 Maret 2011 = Rp 357.814,8

Begitu seterusnya hingga hari ke 15 dan selanjutnya dicari rata-rata dan dihasilkan faktor perkalian sebesar Rp 283.108,9

Fitness pemerataan beban kerja ($arrfitnessbar(i)$) selanjutnya dikalikan dengan faktor perkalian yang didapat.

Setelah dilakukan perkalian selanjutnya dicari fitness total dengan cara menjumlahkan fitness untuk minimasi biaya dengan

fitness pemerataan beban yang telah dikalikan dengan faktor perkalian.

Tabel 4.10. Kode Penentuan Fitness Total

```

For i = 1 To PopuMain.NumOfIndivid
  arrfitnestotal(i) = 0
  arrfitnesdes(i) = 0
  arrfitnestotal(i) = arrfitnes(i) + arrfitnesbar(i)
  arrfitnesdes(i) = arrfitnes(i) + arrfitnesbar(i)
Next i

```

Arrfitnestotal (i) adalah fitness yang digunakan untuk proses selanjutnya. Fitness total ini selanjutnya diurutkan dari fitness terkecil hingga terbesar.

Tabel 4.11 Kode Pengurutan Fitness

```

For i = 1 To PopuMain.NumOfIndivid - 1
  For l = PopuMain.NumOfIndivid To 2 Step -1
    If arrfitnestotal(l) < arrfitnestotal(l - 1) Then
      temp = arrfitnestotal(l)
      arrfitnestotal(l) = arrfitnestotal(l - 1)
      arrfitnestotal(l - 1) = temp
    End If
  Next l
Next i

```

Pengurutan dilakukan dengan metode *Bubble*. Metode ini akan membawa fitness terkecil menuju bagian atas dan fitness terbesar ke bagian bawah. Untuk pelestarian kromosom terbaik dilakukan dengan cara kromosom yang memiliki fitness terbesar digantikan dengan kromosom yang memiliki fitness terkecil namun kromosom tersebut tetap diletakkan pada bagian bawah dan tidak dikenakan proses seleksi.

4) Seleksi

Seleksi yang digunakan adalah metode elitisme. Dibandingkan dengan metode yang lain metode ini lebih simpel karena tidak ada proses perhitungan. Meskipun tidak ada perhitungan namun metode ini tetap memberikan peluang pada kromosom yang bernilai fitness baik mendapat kesempatan terpilih lebih besar dari pada kromosom dengan fitness buruk. Peluang tersebut didapat dari nilai random

yang diberikan. Dimana untuk kromosom dengan nilai fitness yang dianggap baik diberikan nilai random yang lebih rendah dari pada kromosom dengan nilai fitness yang dianggap buruk. Selanjutnya dengan jumlah kromosom yang dianggap baik berjumlah lebih sedikit dari pada kromosom yang dianggap buruk, diharapkan konvergensi prematur dapat dihindari.

Proses ini dilakukan dengan kode pada tabel 4.12 - tabel 4.14.

Tabel 4. 12 Penentuan Probabilitas *Crossover*

```

probcrossover = 0.5
jumcros = Round(probcrossover * PopuMain.NumOfIndivid)
atas = Round((40 / 100) * jumcros)
bawah = jumcros - atas

```

Proses ini diawali dengan penentuan *probabilitas crossover* (*pc*), *pc* ditentukan sebesar 0,5, yang berarti bahwa setengah dari jumlah kromosom yang ada pada populasi tersebut akan *dicrossover*. Kromosom dibagi menjadi dua bagian yaitu bagian atas dan bagian bawah. Bagian atas adalah kromosom yang menempati urutan pertama hingga urutan pada jumlah yang akan *dicrossover*. Bagian bawah adalah kromosom yang berada pada urutan setelah jumlah yang akan *dicrossover* hingga kromosom terakhir. Sebanyak 40 % dari jumlah kromosom yang akan *dicrossover* dipilih dari bagian atas dan sisanya dipilih pada bagian bawah.

Setelah kromosom dibagi menjadi dua bagian dan diketahui jumlah kromosom yang harus *dicrossover* pada setiap bagian, maka

dilanjutkan dengan proses pemilihan kromosom mana yang harus *dicrossover* pada setiap bagiannya. Proses ini dijalankan dengan kode pada tabel 4.13.

Tabel 4.13. Pemberian Nilai *Random*

```

'UNTUK BAGIAN ATAS
For l = 1 To jumeros
random1:
  Rand = random(1, 1000)
  Arrrand(urut(l)) = Rand
  selisih(urut(l)) = Rand
  If l > 1 Then
    For k = 1 To l - 1
      If Arrrand(urut(l)) = Arrrand(urut(k)) Then
        Arrrand(urut(l)) = 0
        selisih(urut(l)) = 0
        GoTo random1
      End If
    Next k
  End If
End If
Next l

```

Lanjutan tabel 4.13. Pemberian Nilai *Random*

```

'UNTUK BAGIAN BAWAH
For l = 1 To PopuMain.NumOfIndivid
  If Arrrand(l) = 0 Then
    Rand = random(1000, 2000)
    Arrrand(l) = Rand
    selisih(l) = Rand
  End If
  If l = PopuMain.NumOfIndivid Then
    Rand = 2001
    Arrrand(l) = Rand
    selisih(l) = Rand
  End If
  'str9 = str9 & selisih(l) & "-"
Next l

```

Proses pemilihan kromosom dilakukan dengan pemberian nilai random pada setiap kromosom. Pada bagian atas pemberian nilai random dilakukan dengan mengacak nilai dari 1 hingga 1000 kemudian untuk bagian bawah dilakukan dengan mengacak nilai dari 1000 hingga 2000 dan khusus kromosom terakhir diberikan nilai

2001 agar tidak terpilih. Proses pengurutan menjadi proses lanjutan untuk memilih kromosom yang harus *dicrossover*. Proses pengurutan menggunakan metode *bubble* yang membawa random terkecil ke bagian atas dan random terbesar ke bagian bawah. Untuk menentukan mana kromosom yang memiliki random terkecil maka digunakan kode pada tabel 4.14.

Tabel 4.14. Penyimpanan Nilai Random

```
'bagian atas
For l = 1 To atas
sim(l) = random (l)
Next l
'bagian bawah
For l = jumcros + 1 To jumcros + bawah
sim(l - jumcros + atas) = random(l)
Next l
```

Pertama-tama dilakukan proses penyimpanan nilai random pada kode sim (l). Bagian atas, sim(l) akan menyimpan sebanyak 40% dari jumlah yang harus *dicrossover*. Dan bagian bawah menyimpan sebanyak sisa dari jumlah *crossover*. Dan sim(l) akan menyimpan angka random dari *kromosom* terpilih, sebanyak jumlah yang harus *dicrossover* pada setiap bagian. Dengan kode di atas maka sim (l) akan berjumlah sebanyak jumlah kromosom yang harus *dicrossover*. Kode pada tabel 4.15 di bawah ini adalah langkah terakhir untuk menentukan kromosom mana yang terpilih

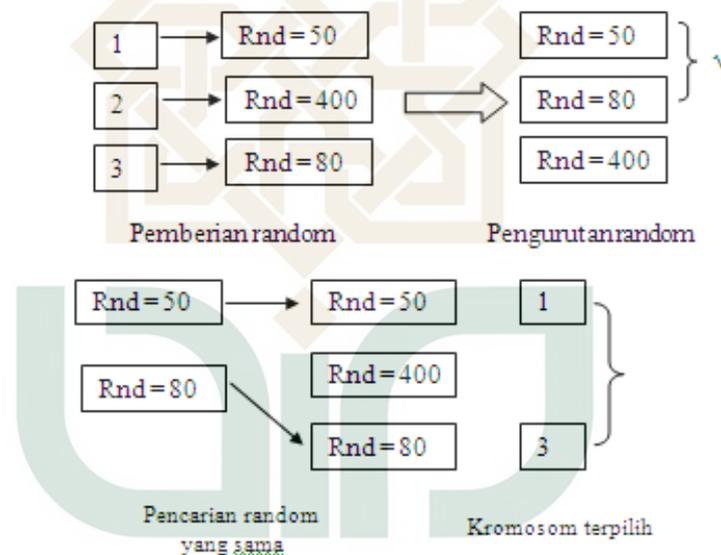
Tabel 4.15 Penentuan Kromosom Terpilih

```
For k = 1 To jumcros
If sim(k) > 0 Then
For j = 1 To PopuMain.NumOfIndivid
If Arrrand(j) = sim(k) Then
simp(k) = j
End If
Next j
End If
Next k
```

Pencarian dilakukan dengan cara mencari nilai yang sama dari simpanan ($\text{sim}(k)$) dengan random yang ada pada setiap kromosom sebelum diurutkan. Dengan demikian maka kita dapat mengetahui kromosom keberapa yang harus *dicrossover*. Selanjutnya kromosom yang terpilih disimpan dalam $\text{simp}(k)$.

Simulasi jalannya proses seleksi pada gambar 4.6 :

Dipilih 2 kromosom yang akan *dicrossover*



Gambar 4.6 Simulasi Proses Seleksi

5) Crossover

Proses *crossover* dijalankan dengan metode *Block Crossover*. Metode ini adalah pengembangan dari metode *Order Crossover*. Pada metode *Order Crossover* pemindahan *genome* dari kromosom lawan ke kromosom yang *dicrossover* dilakukan dengan memindahkan *genome* yang belum ada pada kromosom tanpa memperhatikan kapasitas mobil tangki. Dengan pemindahan seperti

ini maka akan terjadi muatan yang melebihi kapasitas mobil tangki dalam pengiriman. Untuk itu penulis membuat cara lain dalam proses *crossover*, yaitu metode *Block Crossover*.

Block Crossover adalah metode *crossover* yang digunakan untuk mengatasi kasus pemindahan kromosom yang harus mempertimbangkan kapasitas mobil tangki yang tersedia. Proses metode ini adalah sebagai berikut, setelah diketahui *probabilitas crossover* dan diketahui mulai *genome* ke berapa yang akan dipindah silangkan (*random*), pemindahan *genome* dilakukan secara sekumpulan (dalam satu *block*) tidak per-*genome*. Dan proses persilangan dilakukan dengan *block* yang memiliki kapasitas sama pada *kromosom* lawan. Sebenarnya kunci pada metode *block crossover* ini adalah pada proses *random* untuk menentukan mulai *genome* keberapa akan dipindah silangkan. *Random* tersebut diarahkan ke *genome* terakhir dari setiap *block*, dan *genome* yang dipindahkan adalah *genome* setelah bilangan *random*. Sehingga pemindahan *genome* selalu dari awal *genome* setiap *block*. Dengan demikian secara otomatis *genome* tetap berada pada mobil tangki yang berkapasitas mencukupi.

Proses *crossover* diawali dengan pencarian mulai *genome* keberapa yang akan dipindah. Dengan menggunakan angka *random* yang diacak dari 1 hingga panjang *genome* (sebanyak 31 x 4). Karena metode yang digunakan adalah metode *Block crossover* maka setelah

didapat angka random selanjutnya dimasukkan dalam proses *Round* atau pembulatan dan selanjutnya dikalikan 4. Proses ini dilakukan agar bilangan random berada pada akhir blok yang terpilih. Sehingga pemindahan selalu dari awal *blok*. Proses di atas dapat dilihat pada kode yang terdapat pada tabel 4.16 di bawah ini

Tabel 4.16 Penentuan Urutan Genome
yang Akan di Crossover

```

For k = 1 To jumcros
  Rand = random(1, JumTruk * 4)
  hasil = Round((Rand + 1.5) / 4)
  jum = hasil * 4
  Arrrand(k) = jum
  'str11 = str11 & Arrrand(k) & "-"
Next k

```

Selanjutnya masuk pada proses pemindahan atau kawin silang

Tabel 4.17 Proses Crossover

```

For k = 1 To jumcros
  For j = 1 To JumTruk * 4
    If j > arrand(j)
      krombar2(simp(k)) = krombar(simp(a))
      For l = 1 To JumTruk * 4
        silang(simp(k)) = PopuMain.Individuals(simp(k)).Genome(l)
        If krombar2(simp(k)) = silang(simp(k)) And l <= Arrrand(k) Then
          krombar2(simp(k)) = 0
        End If
      Next l
    End if
  Next j
Next k

```

Kode pada tabel. 4.17 di atas menunjukkan ketika j lebih dari nilai random (arrand (k)) maka nilai genome yang ada dipindah silangkan dengan kromosom lawan yang *dicrossover*. Untuk

menanggulangi terdapatnya nilai genome yang sama maka dilakukan proses pada tabel 4.18 dibawah ini.

Tabel 4.18 Penanggulangan Genome yang Sama

```

For l = 1 To JumTruk * 4
  silang(simp(k)) = PopuMain.Individuals(simp(k)).Genome(l)
  If krombar2(simp(k)) = silang(simp(k)) And l <= Arrrand(k) Then
    krombar2(simp(k)) = 0
  End If
Next l

```

Proses pada tabel 4.18 di atas akan mengecek apakah nilai genome yang akan dipindahkan sudah ada pada kromosom atau belum (*genome* yang berada pada urutan sebelum nilai random), jika nilai *genome* sudah ada maka *genome* tersebut di-nol kan. Dengan proses ini maka tidak terdapat *genome* yang sama namun kemungkinan akan kehilangan *genome* pada kasus tertentu.

Simulasi proses *crossover* tercantum pada gambar 4.7 di bawah ini :

Genome yang dipindah mulai dari urutan ke -5

Parent 1 vs parent 2

mobil tangki 1 (16 KL)				mobil tangki 2 (24 KL)				mobil tangki 3 (32 KL)			
1	0	0	0	2	3	0	0	5	4	6	0
ada											
mobil tangki 1 (16 KL)				mobil tangki 2 (24 KL)				mobil tangki 3 (32 KL)			
2	6	0	0	1	4	0	0	3	5	0	0
Anak											
mobil tangki 1 (16 KL)				mobil tangki 2 (24 KL)				mobil tangki 3 (32 KL)			
1	0	0	0	0	4	0	0	3	5	0	0

Gambar 4.7 Simulasi Proses Crossover

6) Mutasi (*Constrained Insertion Mutation*)

Proses *mutasi* ini digunakan sebagai langkah untuk melengkapi *genome* yang hilang setelah dilakukan *crossover*. Dan juga sebagai

langkah untuk mengacak posisi *genome* yang ada pada *kromosom*. Proses mutasi ini adalah pengembangan dari metode *Insertion Mutasi*. Dimana pada *insertion mutasi* pemilihan *genome* yang akan dimutasi dilakukan secara acak pada *kromosom* yang memiliki *genome* lengkap dan penyisipan *genome* yang terpilih pun dilakukan secara *random*.

Dalam kasus ini bila pemilihan *genome* dilakukan demikian, hanya akan mengulang pekerjaan, yaitu programmer harus mengisikan *genome* yang belum ada pada *kromosom* (pada *block* yang mencukupi kapasitasnya) kemudian memilih lagi dan menyisipkan lagi pada *block* lain. Bila proses penyisipan dilakukan secara *random* akan terjadi muatan mobil tangki melebihi kapasitasnya. Untuk itu maka pada metode ini *genome* yang termutasi adalah *genome* yang hilang pada *kromosom* setelah dilakukan *crossover*. Dan *programer* memberikan tambahan dalam proses penyisipan, yaitu dengan menambahkan proses pencarian *block* yang memiliki kapasitas mencukupi, bila *genome* yang terpilih diletakkan pada *block* tersebut. Metode ini dinamakan *Insertion with Capacitated*.

Dalam proses mutasi ini pula, *blok* yang memiliki bagian (*part*) pertama bernilai nol tetapi *part* selanjutnya terisi, maka *part* yang terisi tersebut akan di-nol-kan. Hal ini digunakan sekaligus untuk

menempatkan dan mengurutkan *genome*. Kode yang dibuat untuk proses mutasi ini tercantum pada tabel 4.19 di bawah :

Tabel 4.19 Proses Mengenkilkan Genome

```

For k = 1 To jumcros
  For j = 1 To JumTruk * 4
    krom(simp(k)) = PopuMain.Individuals(simp(k)).Genome(j)
    If Round((j + 1.5) / 4) = Round(((j - 1) + 1.5) / 4) And krom(simp(k)) <> 0 Then
      For l = j - 1 To j - 1
        silang(simp(k)) = PopuMain.Individuals(simp(k)).Genome(l)
        If silang(simp(k)) = 0 Then
          krom(simp(k)) = 0
        End If
      Next l
    End If
    PopuMain.Individuals(simp(k)).Genome(j) = krom(simp(k))
  Next j
Next k

```

Proses pada tabel 4.19 di atas adalah proses untuk mengenkilkan *genome* pada *block* yang memiliki *part* pertama bernilai nol tetapi *part* selanjutnya terisi. Hal ini digunakan sekaligus untuk menempatkan nilai selalu dari urutan awal. Proses ini diawali dengan pencarian *genome* bernilai (bukan nol). Ketika didapat proses akan menentukan posisi nilai tersebut pada *part* (urutan) beberapa. Jika posisi nilai tersebut bukan pada *part* pertama maka proses dilanjutkan untuk melihat apakah *part* (urutan) sebelumnya pada mobil yang sama bernilai nol atau tidak. Jika urutan sebelumnya bernilai nol maka nilai yang ada pada urutan tersebut di-nol kan. Proses selanjutnya adalah proses untuk mencari nilai (angka) yang hilang pada setiap *kromosom*. Proses ini tercantum pada tabel 4.20.

Tabel 4.20 Pencarian Genome yang Hilang

```

For k = 1 To jumcros
  For j = 1 To JumSPBU
    ind(k, j) = 0 + j
  Next j
Next k

'mengenolkan indeks yang sama
For k = 1 To jumcros
  For j = 1 To JumTruk * 4
    krom(simp(k)) = PopuMain.Individuals(simp(k)).Genome(j)
    For l = 1 To JumSPBU
      If krom(simp(k)) = ind(k, l) Then
        ind(k, l) = 0
      End If
    Next l
  Next j
Next k

```

Proses diawali dengan memberikan angka pada sebuah indeks dengan kode ind (k,j) sebanyak jumlah SPBU yang memesan. Setelah itu indeks digunakan untuk mencari angka yang hilang dengan cara mengecek setiap angka pada indeks ke setiap *kromosom*. Bila pada *kromosom* tersebut terdapat angka yang sama maka angka pada indeks tersebut di-nol kan. Sehingga pada akhirnya angka yang tidak ada pada *kromosom* dapat diketahui dari angka sisa yang terdapat pada indeks.

Tabel 4.21 Proses Penyisipan Genome Terpilih

```

'proses pemindahan mutasi yang indeks sisa
For k = 1 To jumcros
  KumKap = 0
  'str13 = ""
  For j = 1 To JumTruk * 4
    krom(simp(k)) = PopuMain.Individuals(simp(k)).Genome(j)
    For l = 1 To JumSPBU
      If ind(k, l) <> 0 Then
        If krom(simp(k)) = 0 Then
          krom(simp(k)) = ind(k, l)
        End If
      End If
    Next l
  Next j
Next k

```

Lanjutan tabel 4.21 Proses Penyisipan Genome Terpilih

```

If krom(simp(k)) <> 0 Then
KumKap = jumkap(k, Round((j + 1.5) / 4)) + SPBUMinta(krom(simp(k)))
If KumKap <= TrukKapasitas(Round((j + 1.5) / 4)) Then
krom(simp(k)) = krom(simp(k))
ind(k, l) = 0
jumkap(k, Round((j + 1.5) / 4)) = KumKap
Else
krom(simp(k)) = 0
End If
End If
End If
End If
Next l
PopuMain.Individuals(simp(k)).Genome(j) = krom(simp(k))

Next j
Next k

```

Proses pada tabel 4.21 di atas digunakan untuk menempatkan nilai (angka) yang hilang pada setiap *kromosom*. Penempatan angka yang hilang tersebut diletakkan pada *genome* yang bernilai nol, setelah diletakkan proses selanjutnya adalah proses pengecekan kapasitas mobil tangki, apabila kapasitas muatan melebihi batas (setelah peletakkan angka yang hilang) maka angka tadi kembali dinol-kan, kemudian proses dilanjutkan untuk mencari *genome* yang bernilai nol selanjutnya, hingga didapat kapasitas mobil tangki yang mencukupi. Dengan proses demikian maka *kromosom* anak memiliki *genome* yang lengkap, setiap pendistribusian BBM tidak ada yang melebihi kapasitas, dan proses mutasi pun berjalan.

Simulasi proses mutasi seperti pada gambar 4.8 – gambar 4.11.

kromosom hasil *crossover* dengan jumlah SPBU pemesan sebanyak 6

mobil tangki 1 (16 KL)				mobil tangki 2 (24 KL)				mobil tangki 3 (32 KL)			
1	0	0	0	0	4	0	0	3	5	0	0

Gambar 4.8 Contoh *Kromosom* Hasil *Crossover*

Proses meng-nol-kan genome yang memiliki urutan awal pada mobil tangki yang bernilai nol.

mobil tangki 1 (16 KL)				mobil tangki 2 (24 KL)				mobil tangki 3 (32 KL)			
1	0	0	0	0	4	0	0	3	5	0	0

mobil tangki 1 (16 KL)				mobil tangki 2 (24 KL)				mobil tangki 3 (32 KL)			
1	0	0	0	0	0	0	0	3	5	0	0

Gambar 4.9 Proses Meng-nol-kan Genome

Proses pencarian *genome* yang hilang pada *kromosom*

Indeks = (1,2,3,4,5,6)

Angka indeks yang sudah ada pada kromosom di-nol kan

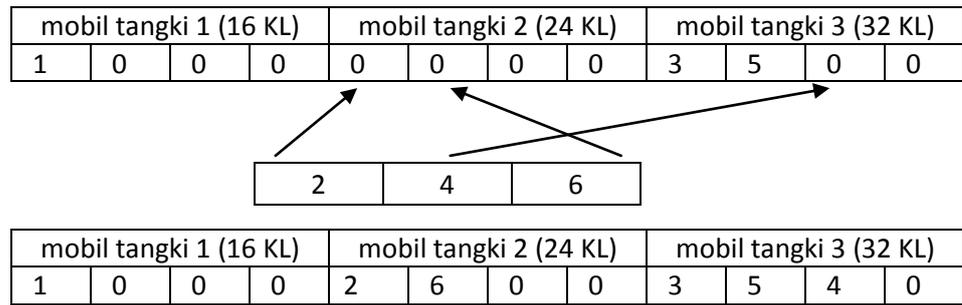
mobil tangki 1 (16 KL)				mobil tangki 2 (24 KL)				mobil tangki 3 (32 KL)			
1	0	0	0	0	0	0	0	3	5	0	0

Indeks					
1	2	3	4	5	6

Indeks sisa		
2	4	6

Gambar 4.10 Proses Meng-nol-kan Angka Indeks

Penempatan indeks sisa pada mobil tangki yang memiliki kapasitas cukup.



Kromosom hasil mutasi

Gambar 4.11 Proses Menyisipkan Genome yang Tersisa pada Indeks

7) Evolusi

Proses ini akan berjalan hingga mencapai generasi yang telah ditentukan. Pada generasi yang telah ditentukan, fitness telah mendekati optimal. Fitness yang mendekati optimal diketahui jika nilai fitness bernilai sama disetiap generasi.

Tabel 4.22 Proses Evolusi Awal

```

Do While min2 >= min1
DoEvents
evolusi
  For i = 1 To 1
  min2 = arrfitnesstotal(i)
  Next i
  l = l + 1
  If l = 100 Then
  If min2 >= min1 Then
  GoTo henti
  End If
  End If
Loop

```

Proses ini diawali dengan proses pencarian nilai fitness yang lebih kecil dari generasi awal (populasi awal). Min1 adalah kode nilai fitness yang paling kecil pada populasi awal atau sebelumnya dan

min2 adalah kode untuk nilai fitness terkecil pada populasi yang sedang diproses. Kode pada tabel 4.22 di atas akan melakukan evolusi hingga ditemukan nilai fitness yang lebih kecil dari populasi awal. Namun bila hingga 100 generasi nilai fitness tetap sama maka dapat dikatakan fitness telah mendekati optimal dan proses evolusi berhenti. Namun bila pada proses tersebut didapat fitness yang lebih kecil dari populasi awal maka proses masuk ke proses selanjutnya

Tabel 4.23 Evolusi Lanjutan

```

For z = 1 To 100
evolusi
min2 = arrfitnestotal(1)

If min2 < min1 Then
  For k = 1 To 1
  For j = 1 To JumTruk * 4
  krom(k) = PopuMain.Individuals(urut(k)).Genome(j)

  For b = 1 To JumTruk
  If Round((j + 1.5) / 4) = b Then
  If Round((j + 1.5) / 4) <> Round(((j - 1) + 1.5) / 4) Then

  End If

  If j = Round((j + 1.5) / 4) * 4 Then

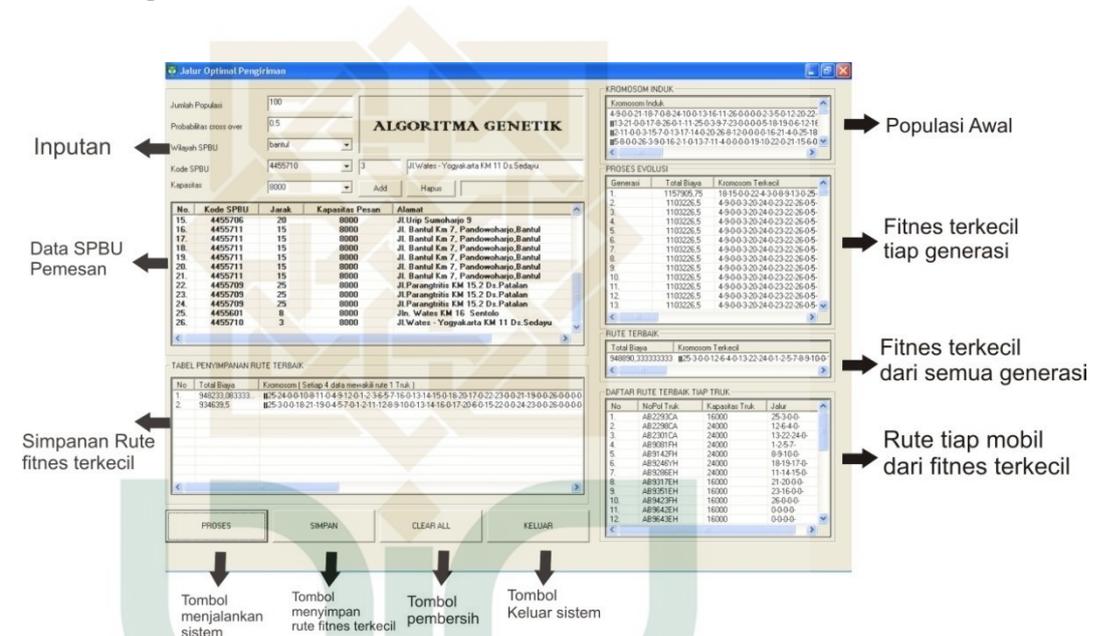
  End If
  End If
  Next b
  Next j
  Next k
  min1 = min2
  jumrubah(1) = jumrubah(1) + 1
  End If
  Next z

```

Proses pada tabel 23 berjalan setelah ditemukan nilai fitness yang lebih kecil dari populasi awal. Proses itu akan mengulang proses evolusi hingga 100 generasi. Jika pada proses tersebut

ditemukan nilai fitness yang lebih kecil maka min1(fitness pembandingan) akan berubah menjadi nilai fitness yang paling kecil. Kemudian dilanjutkan hingga generasi ke-100. Dan nilai fitness yang paling kecil akan ditampilkan pada pengguna sistem.

4.2.3 Tampilan Sistem



Gambar 4.12 Tampilan Sistem

Dengan adanya kolom simpanan diharapkan nilai fitness yang paling optimal dapat diketahui. Karena kolom tersebut akan menyimpan nilai fitness terkecil pada setiap kali proses dan pengguna dapat membandingkan nilai fitness terkecil pada setiap kali percobaan.

4.2.4 Eksperimentasi

Eksperimentasi dilakukan pada data perusahaan pada tanggal 17 April 2011 :

Jumlah populasi = 100 populasi

Probabilitas *Crossover* = 0.5

SPBU wilayah Bantul

Jumlah Data = 20

Banyak Eksperimen = 30 kali

Tabel 4.24. Data Permintaan SPBU

Kode SPBU	Permintaan (Liter)
4455102	8000
4455103	8000
4455103	16000
4455106	16000
4455112	8000
4455112	16000
4455701	8000
4455701	8000
4455701	8000
4455704	8000
4455704	8000
4455707	16000
4455707	16000
4455709	8000
4455709	8000
4455712	8000
4455712	8000
4455803	8000
4455803	8000
4455803	8000

Tabel 4.24 Menunjukkan data yang digunakan dalam eksperimentasi sistem dimana kolom pertama adalah kode SPBU yang memesan dan Kolom kedua adalah kapasitas permintaan masing-masing SPBU.

Jalur Optimal Pengiriman

Jumlah Populasi: 100
 Probabilitas cross over: 0.5
 Wilayah SPBU: bankul
 Kode SPBU: 4455803
 Kapasitas: 8000

ALGORITMA GENETIK

No.	Kode SPBU	Jarak	Kapasitas Pesan	Alamat
9.	4455701	20	8000	Jati Wonokromo Pleret Bantul
10.	4455704	21	8000	Jl. Wonosari Km. 10,5
11.	4455704	21	8000	Jl. Wonosari Km. 10,5
12.	4455707	18	16000	Jl. Imogiri Barat km 7 Kec. Sewon
13.	4455707	18	16000	Jl. Imogiri Barat km 7 Kec. Sewon
14.	4455709	25	8000	Jl. Parangtritis KM 15.2 Ds. Patalan
15.	4455709	25	8000	Jl. Parangtritis KM 15.2 Ds. Patalan
16.	4455712	25	8000	Galur Pal Bapang Bantul
17.	4455712	25	8000	Galur Pal Bapang Bantul
18.	4455803	20	8000	Jl. Raya Sewon Parangtritis KM 11 Bantul
19.	4455803	20	8000	Jl. Raya Sewon Parangtritis KM 11 Bantul
20.	4455803	20	8000	Jl. Raya Sewon Parangtritis KM 11 Bantul

PROSES EVOLUSI

Generasi	Total Biaya	Kromosom Terbaik
1.	2523716.1350...	17-20-0-9-7-18-0-19-4-0-0-12
2.	2455167.9751...	6-0-0-9-4-0-0-12-9-0-0-3-0-0-0
3.	2455167.9751...	6-0-0-9-4-0-0-12-9-0-0-3-0-0-0
4.	2455167.9751...	6-0-0-9-4-0-0-12-9-0-0-3-0-0-0
5.	2408641.1218...	6-0-0-9-4-0-0-12-9-0-0-3-0-0-0
6.	2408641.1218...	6-0-0-9-4-0-0-12-9-0-0-3-0-0-0
7.	2408641.1218...	6-0-0-9-4-0-0-12-9-0-0-3-0-0-0
8.	2408641.1218...	6-0-0-9-4-0-0-12-9-0-0-3-0-0-0
9.	1953941.3424...	6-0-0-9-4-0-0-12-9-0-0-3-0-0-0
10.	1953941.3424...	6-0-0-9-4-0-0-12-9-0-0-3-0-0-0
11.	1953941.3424...	6-0-0-9-4-0-0-12-9-0-0-3-0-0-0
12.	1953941.3424...	6-0-0-9-4-0-0-12-9-0-0-3-0-0-0
13.	1953941.3424...	6-0-0-9-4-0-0-12-9-0-0-3-0-0-0

TABEL PENYIMPANAN RUTE TERBAIK

No	Total Biaya	Kromosom [Setiap 4 data mewakili rute 1 Truk.]
21.	1659453.46551...	00-2-0-0-12-9-0-0-4-7-0-0-3-0-0-1-5-0-0-6-0-0-13-0-0-0-19-18-0-0-16-17-0-0-15-14-0-0-10-11-0-
22.	1722183.52370...	01-3-0-0-1-5-7-0-2-3-0-0-16-17-0-0-4-0-0-0-12-0-0-0-10-11-0-0-6-9-0-0-14-15-0-0-6-0-0-18-20-0-0
23.	1616255.28796...	01-2-0-0-3-9-0-0-2-4-0-0-1-5-0-0-11-1-0-0-16-17-0-0-6-0-0-7-8-0-0-15-14-0-0-19-20-0-0-13-0-0-0
24.	1597623.40688...	04-0-0-0-7-8-9-0-3-2-0-0-1-5-0-0-12-0-0-6-0-0-0-15-14-0-0-10-11-0-0-16-17-0-0-18-19-0-0-20-0-0-0
25.	1557337.16314...	01-8-19-0-0-1-6-0-0-2-4-0-0-12-0-0-5-9-0-0-14-15-0-0-10-11-0-0-3-0-0-0-16-17-0-0-7-8-0-0-13-0-0-0
26.	1510861.48259...	05-0-0-0-9-13-0-0-2-4-0-0-6-0-0-0-1-5-0-0-10-11-0-0-12-0-0-0-7-8-0-0-14-15-0-0-16-17-0-0-18-19-0-0
27.	1510861.48259...	01-4-15-0-0-4-2-0-0-12-9-0-0-6-0-0-0-7-8-0-0-13-0-0-0-1-5-0-0-3-0-0-0-10-11-0-0-18-19-0-0-16-17-0-0
28.	1510861.48259...	05-1-0-0-12-9-0-0-2-4-0-0-7-8-0-0-16-17-0-0-3-0-0-0-6-0-0-0-10-11-0-0-13-0-0-0-14-15-0-0-18-19-0-0
29.	1583834.24425...	06-0-0-0-12-8-0-0-1-4-0-0-3-0-0-0-16-17-0-0-14-15-0-0-2-5-0-0-13-0-0-0-10-11-0-0-7-9-0-0-18-19-0-0
30.	1562834.24425...	05-0-0-0-1-4-0-0-17-8-0-0-3-0-0-0-10-11-0-0-13-0-0-0-7-8-0-0-14-15-0-0-16-17-0-0-18-19-0-0-1

DAFTAR RUTE TERBAIK TIAP TRUK

No	NoPel Truk	Kapasitas Truk	Jalur
1.	AB2293CA	16000	6-0-0-0-
2.	AB2298CA	24000	1-4-0-0-
3.	AB2301CA	24000	12-8-0-0-
4.	AB9081FH	16000	3-0-0-0-
5.	AB9142FH	16000	2-5-0-0-
6.	AB9246YH	16000	10-11-0-0-
7.	AB9286EH	16000	13-0-0-0-
8.	AB9317EH	16000	7-9-0-0-
9.	AB9351EH	16000	14-15-0-0-
10.	AB9423FH	16000	16-17-0-0-
11.	AB9642EH	16000	18-19-0-0-
12.	AB9643EH	16000	20-0-0-0-

Gambar 4.13 Tampilan Percobaan ke-1 hingga ke-30

Gambar 4.13 di atas menunjukkan jalannya proses eksperimentasi sistem. Dimana gambar menunjukkan jalannya proses eksperimentasi (percobaan) dari pertama hingga percobaan ke-tiga puluh.

1) Perhitungan Fitness (Manual)

Perhitungan fitness dengan cara manual ini dilakukan pada data yang sama dengan data yang digunakan untuk proses eksperimentasi di atas, yaitu data pada tabel 4.24. Perhitungan fitness dilakukan untuk fitness terbaik pada percobaan kesepuluh, kedua puluh, dan ketiga puluh.

Perhitungan untuk rute terbaik pada proses kesepuluh pada mobil tangki pertama (AB2293CA)

Pengantaran untuk data ke-17 dan data ke-16 (SPBU 4455712 dan 4455712)

$$\text{Total jarak} = J_{d1} + J_{d2} + J_{d3}$$

$$\text{Total Jarak} = 25 + 0 + 25 = 50 \text{ km}$$

$$\text{Rata-rata Jarak} = 41,08$$

$$\text{Truk yang dipakai} = 12$$

$$\text{MAD} = \frac{50-41,08}{12} = 0,743$$

$$\text{MAD x Faktor perkalian} = 0,743 \times 283108,9 \times 0,4$$

$$\text{Fitnes Pemerataan} = 84146,2564$$

$$\begin{aligned} \text{Totalbiaya} &= (T_j \times S_k \times J) + (T_j \times P_k) + (T_j / J_1 \times L) \\ &+ G_s + G_k \end{aligned}$$

$$\text{Gaji sopir dan kondektur} = 0 \text{ karena } T_j < 60$$

$$\text{Total Biaya} = (50 \times 46 \times 10) + (50 \times 135) + (50/3.2 \times 4300) + 0 + 0$$

$$\text{Fitnes Minim Biaya} = 96937,5$$

$$\text{Kumulatif Fitnes} = 84146,2564 + 96937,5$$

$$= 181083,76$$

Perhitungan selanjutnya dibantu dengan *software Excel*, tetapi tetap menggunakan algoritma yang sama seperti di atas, dan hasilnya terdapat pada tabel 4.25 - tabel 4.26 :

Tabel 4.25 Perhitungan Fitnes Untuk Rute Terbaik pada Percobaan ke-

10

NO.POLISI	Rute (Data ke-)	T.Jarak	Ft.Pemerataan	Ft. Total Biaya	Ft.P +FT.M
AB2293CA	17-16	50	84146,25639	96937,5	181083,7564
AB2298CA	2-4	47	55835,36639	111464,8519	167300,2182
AB2301CA	7-8-9	40	10223,37694	94863,7037	105087,0806
AB9081FH	18 – 20	40	10223,37694	77550	87773,37694
AB9142FH	10 -11	42	8650,549722	81427,5	90078,04972
AB9246YH	1-5	44	27524,47639	85305	112829,4764
AB9286EH	3	32	85719,08361	62040	147759,0836
AB9317EH	15 -14	50	84146,25639	96937,5	181083,7564
AB9351EH	6	36	47971,23028	69795	117766,2303
AB9423FH	12	36	47971,23028	69795	117766,2303
AB9642EH	13	36	47971,23028	69795	117766,2303
AB9643EH	19	40	10223,37694	77550	87773,37694
	rata-rata	41,08333	Total fitness		1514066,866

Tabel 4.26 Perhitungan Fitnes Untuk Rute Terbaik pada Percobaan ke-20

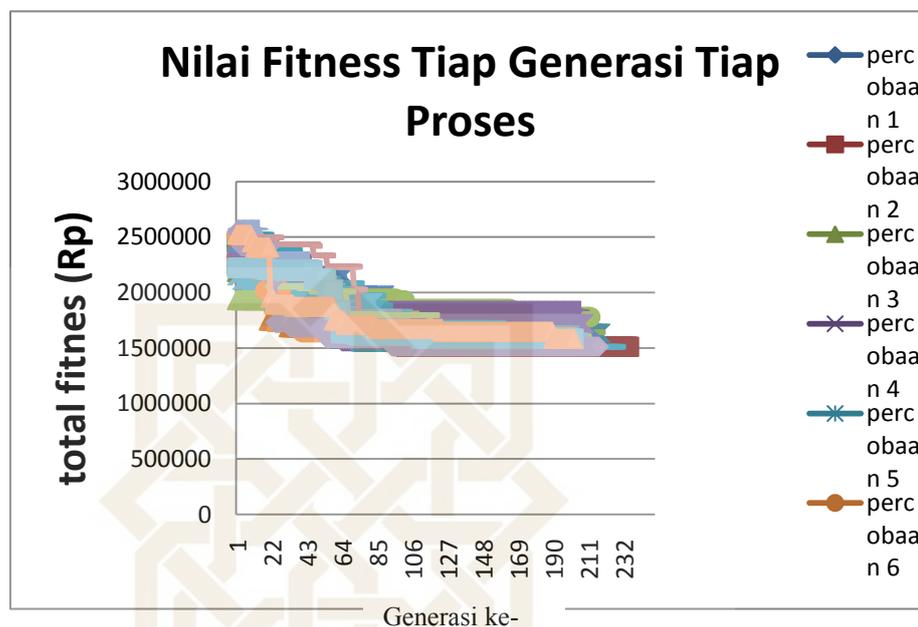
NO.POLISI	Rute (Data ke-)	T.Jarak	Ft.Pemerataan	Ft. total Biaya	Ft.P +FT.M
AB2293CA	20-2	46	41679,92139	89182,5	130862,4214
AB2298CA	12-7	44	22805,99472	104350,0741	127156,0688
AB2301CA	4-1	47	51116,88472	111464,8519	162581,7366
AB9081FH	3	32	90437,56528	62040	152477,5653
AB9142FH	8-9	40	14941,85861	77550	92491,85861
AB9246YH	5	36	52689,71194	69795	122484,7119
AB9286EH	14-15	50	79427,77472	96937,5	176365,2747
AB9317EH	10-11	42	3932,068056	81427,5	85359,56806
AB9351EH	13	36	52689,71194	69795	122484,7119
AB9423FH	16-17	50	79427,77472	96937,5	176365,2747
AB9642EH	19-18	40	14941,85861	77550	92491,85861
AB9643EH	6	36	52689,71194	69795	122484,7119
	Rata2	41,58333	Total fitness		1563605,763

Tabel 4.27 Perhitungan Fitnes Untuk Rute Terbaik pada Percobaan ke-30

NO.POLISI	Rute (Data ke-)	T.Jarak	Ft.Pemerataan	Ft. total Biaya	Ft.P +FT.M
AB2293CA	5-1	36	58981,02083	69795	128776,0208
AB2298CA	12-9	47	44825,57583	111464,8519	156290,4277
AB2301CA	2-4	44	16514,68583	104350,0741	120864,7599
AB9081FH	7-8	32	96728,87417	62040	158768,8742
AB9142FH	16-17	50	73136,46583	96937,5	170073,9658
AB9246YH	3	42	2359,240833	81427,5	83786,74083
AB9286EH	6	36	58981,02083	69795	128776,0208
AB9317EH	10-11	40	21233,1675	77550	98783,1675
AB9351EH	13	50	73136,46583	96937,5	170073,9658
AB9423FH	14-15	50	73136,46583	96937,5	170073,9658
AB9642EH	18-19	40	21233,1675	77550	98783,1675
AB9643EH	20	40	21233,1675	77550	98783,1675
	rata-rata	41,58333	Total fitness		1583834,244

Tabel 4.25 - tabel 4.27 adalah tabel hasil perhitungan nilai fitness total untuk rute terbaik pada percobaan berturut-turut percobaan 10, percobaan 20, dan percobaan 30. Dimana kolom pertama adalah kolom yang menunjukkan no.polisi mobil tangki yang dipakai dalam pendistribusian, kolom kedua adalah kolom yang menunjukkan rute pendistribusian setiap mobil, kolom ketiga adalah kolom yang menunjukkan total jarak yang ditempuh setiap mobil (berdasar perhitungan), kolom keempat adalah kolom yang menunjukkan fitness pemerataan beban setiap mobil, kolom kelima menunjukkan fitness total biaya setiap mobil tangki, kolom keenam adalah kolom yang menunjukkan total fitness setiap mobil tangki. Hasil perhitungan ditunjukkan pada baris terakhir dengan nama total fitness.

2) Grafik Ekperimentasi Nilai Fitness

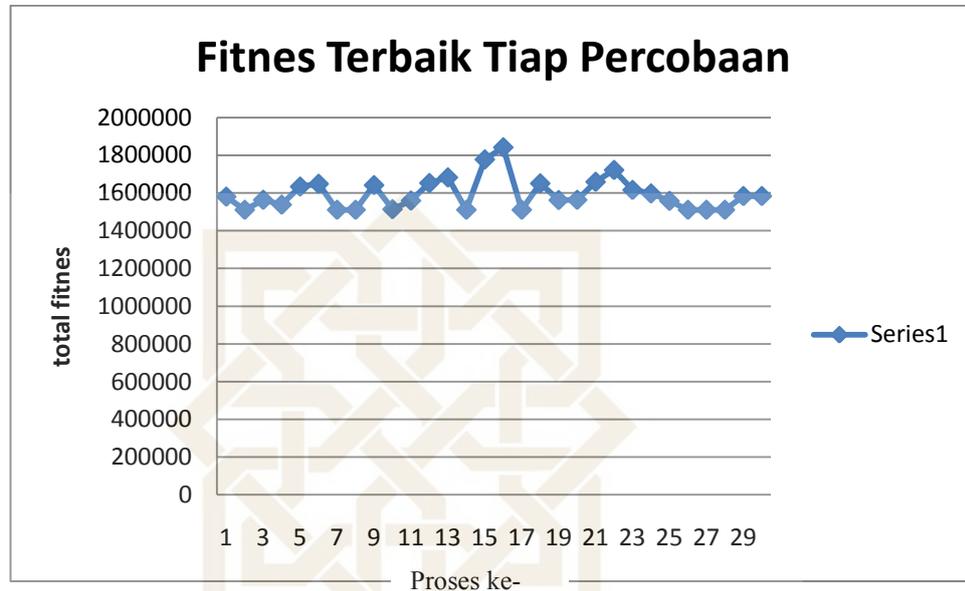


Grafik. 4.1 Grafik nilai fitness tiap generasi untuk 30 percobaan

Grafik.4.1 di atas adalah grafik yang menggambarkan nilai fitness yang dihasilkan tiap generasi selama 30 kali percobaan. Sumbu Y menyatakan total fitness setiap generasi selama 30 kali percobaan dan sumbu x menyatakan banyaknya generasi yang dihasilkan setiap kali percobaan. Warna yang berbeda-beda pada garis yang terbentuk sebagai penunjuk atau pembeda setiap percobaan. Warna untuk setiap percobaan ditunjukkan pada sisi kanan grafik dengan nama percobaan (dari 1-30). Untuk hasil lebih lengkap grafik dan data hasil eksperimen terlampir pada lampiran 4 dan lampiran 5.

Grafik.1 tersebut semakin lama generasi menunjukkan penurunan nilai fitness. Hal tersebut dikarenakan fungsi tujuan adalah minimasi biaya dan pemerataan beban kerja (minimasi selisih jarak tiap mobil tangki yang

beroperasi) yang semakin rendah fungsi tujuan yang dihasilkan maka semakin baik.



Grafik 4.2. Nilai fitnes terbaik untuk tiap kali percobaan

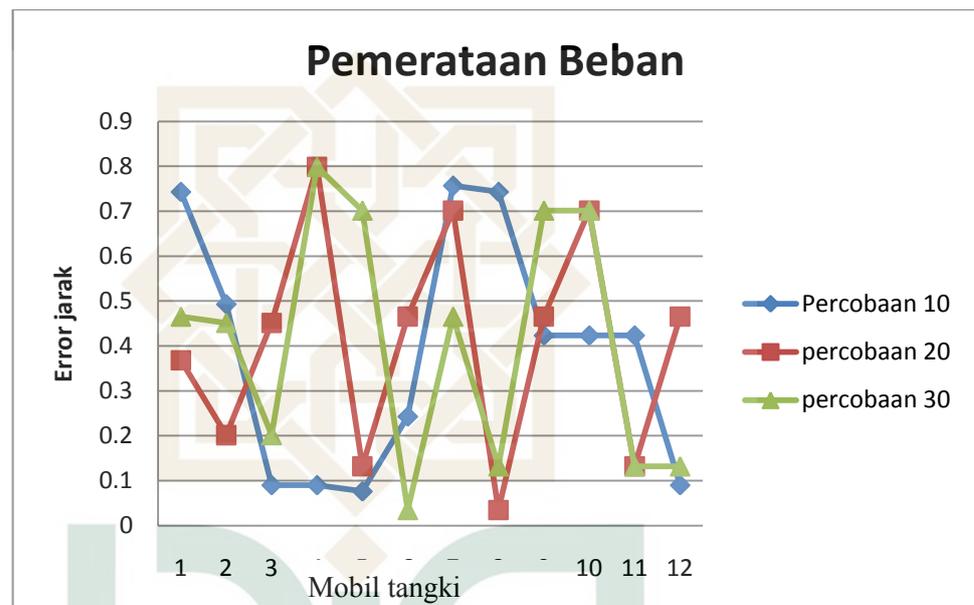
Grafik 4.2 di atas menunjukkan nilai fitness terbaik untuk setiap percobaan dari percobaan pertama hingga ke-30. Sumbu y adalah sumbu untuk total fitness dan sumbu x adalah sumbu untuk jumlah percobaan (urutan percobaan).

Dari grafik 4.2 didapatkan sebuah garis yang hampir rata, itu menunjukkan perbedaan fitness terkecil dari setiap kali percobaan mendekati sama. Kecuali grafik untuk percobaan ke-16 yang mencuat paling tinggi.

3) Pemerataan Beban dan Minimasi Biaya

Eksperimentasi pemerataan beban dan minimasi biaya adalah eksperimentasi yang bertujuan untuk melihat kinerja sistem dalam menghasilkan rute, apakah sistem dapat menghasilkan rute

pendistribusian BBM yang dapat memmeratakan beban kerja dan meminimasi biaya atau belum. Eksperimentasi ini menggunakan data total jarak pada tabel 25- tabel 27. Hasil eksperimentasi dapat dilihat pada grafik 4.3 di bawah.



Grafik 4.3. Grafik Pemerataan Beban Kerja

Grafik 4.3 di atas adalah gambaran dari pemerataan beban hasil pengolahan dengan sistem. Dimana sumbu X menyatakan mobil tangki yang digunakan dan sumbu Y menyatakan nilai *error* jarak dari masing-masing mobil tangki. Dari grafik tersebut dapat dilihat bahwa garis yang berwarna biru (percobaan 10) memiliki garis yang tingkat fluktuasinya paling rendah dari pada garis lain. Dan percobaan 10 adalah percobaan yang memiliki nilai fitness paling kecil dari ketiga percobaan yang ada.

4) Perbandingan total biaya (Validasi)

Pada eksperimentasi ini biaya yang dihasilkan dari pengolahan sistem akan dibandingkan dengan biaya yang dihasilkan dari rute rancangan Patra Niaga. Data yang diambil adalah data pengiriman tanggal 17 April 2011 dan 3 Mei – 7 Mei 2011 untuk wilayah bantul dan sekitarnya.

Perbandingan menggunakan data pada tabel 4.28 dan hasil perbandingan dapat dilihat pada tabel 4.29 di bawah ini

Tabel 4.28. Data SPBU Pemesan Tanggal 17 April 2011

Kode SPBU	Permintaan (Liter)
4455102	8000
4455103	8000
4455103	16000
4455106	16000
4455112	8000
4455112	16000
4455701	8000
4455701	8000
4455701	8000
4455704	8000
4455704	8000
4455707	16000
4455707	16000
4455709	8000
4455709	8000
4455712	8000
4455712	8000
4455803	8000
4455803	8000
4455803	8000

Tabel 28 di atas adalah tabel yang menunjukkan data SPBU pemesan untuk wilayah Bantul tanggal 17 April 2011. Kolom kedua menunjukkan kode SPBU pemesan, dan kolom ketiga menunjukkan jumlah kapasitas BBM yang dipesan.

Tabel 4.29 Hasil Perbandingan Rute dan Biaya Distribusi Tanggal 17 April 2011

Mobil Tangki	Rute Patra Niaga		Rute GA	
16000	4455112-4455102		4455103	
24000	4455106-4455103		4455701-4455707	
24000	4455112-4455701		4455103-4455106	
16000	4455701-4455701		4455112	
16000	4455704-4455704		4455102-4455112	
16000	4455103		4455704-4455704	
16000	4455712-4455712		4455707	
16000	4455707		4455701-4455701	
16000	4455709-4455709		4455709-4455709	
16000	4455103		4455712-4455712	
16000	4455803-4455803		4455803-4455803	
16000	4455803		4455803	
Fitness	Pemerataan beban = 4,17	Minimasi biaya = Rp 1.087.820,4	Pemerataan beban = 3,72	Minimasi biaya = Rp1.089.343,8

Tabel 4.29 di atas adalah tabel perbandingan rute yang dihasilkan sistem dengan rute yang dibuat Patra Niaga untuk melakukan pendistribusian pada tanggal 17 April 2011. Kolom pertama adalah kolom yang menunjukan kapasitas mobil tangki yang digunakan dalam melakukan pengiriman. Kolom kedua menunjukkan rute tiap mobil tangki yang dibuat Patra Niaga untuk melakukan pendistribusian BBM ke SPBU pemesan. Kolom ketiga adalah kolom yang menunjukkan rute yang dihasilkan sistem dalam melakukan pendistribusian BBM tiap mobil tangki. Baris terakhir adalah total biaya dan jumlah *error* (pemerataan beban)

yang dihasilkan dari rute yang dibuat oleh Patra Niaga dengan rute yang dibuat oleh sistem Genetik Algoritma.

Tabel 4.30. Perbandingan Total Biaya Distribusi

Tanggal Distribusi	Patra Niaga		Sistem GA	
	Error	Biaya	Error	Biaya
17-Apr-11	4,17	Rp1.087.820,40	3,72	Rp1.089.343,80
03-Mei-11	14,72	Rp 1.056.835,42	1,2	Rp 1.155.740,16
04-Mei-11	5,34	Rp1.114.315,58	3,9	Rp 1.241.122, 42
05-Mei-11	5,03	Rp1.093.060,92	2,2	Rp 1.251.813, 92
06-Mei-11	8,56	Rp1.192.344,17	1,96	Rp 1.415.973, 50
07-Mei-11	10,3	Rp1.095.395,33	1,83	Rp 1.414.177, 70
Total	48,12	Rp6.639.771,82	14,81	Rp7.568.171,50

Tabel 4.30 di atas adalah tabel yang menunjukkan biaya total pendistribusian BBM wilayah Bantul dari tanggal 17 April dan 3 Mei-7 Mei 2011. Kolom pertama menunjukkan tanggal pendistribusian BBM. Kolom kedua menunjukkan biaya dan *error* (pemerataan beban) yang dihasilkan dari rute yang dibuat oleh Patra Niaga. Kolom ketiga adalah kolom yang menunjukkan biaya dan *error* (pemerataan beban) yang dihasilkan dari rute yang dibuat oleh sistem Genetik Algoritma. Baris terakhir menunjukkan total *error* dan total biaya tanggal 17 April 2011 dan tanggal 3 Mei hingga 7 Mei 2011.

BAB V

ANALISA DAN PEMBAHASAN

Dari hasil pengolahan data di atas yang menggunakan metode Algoritma Genetika maka dihasilkan rute dengan nilai fitness yang kecil (mendekati optimal). Dengan nilai fitness yang kecil diharapkan rute yang didapat optimal.

5.1. Pembentukan populasi awal

Dari pengolahan (pembuatan kode untuk populasi awal) dapat diketahui bahwa kode yang dibuat, dapat membuat *kromosom* yang didalamnya tidak terdapat *genome* yang sama dan peletakan *genome* tepat pada mobil tangki yang mencukupi kapasitasnya.

Hal ini dapat ditunjukkan dari rute yang dihasilkan untuk setiap generasi (pada kolom 4 ditampilkan sistem) yang memiliki rute yang lengkap (tidak ada *genome* yang hilang) sesuai dengan hasil eksperimentasi yang telah dilakukan.

Dapat dikatakan fungsi *Boolean* yang digunakan dalam pembuatan *coding* tepat, karena proses pencarian nilai (*genome*) yang akan diletakkan pada mobil tangki hanya berjalan pada *genome* yang bernilai “*False*” yaitu pada *genome* yang belum terpilih. Sedangkan *genome* yang telah terpilih berubah nilainya menjadi “*True*”. Dengan demikian maka hasil pengolahan yaitu *kromosom* pada populasi awal tidak memiliki *genome* yang sama (ganda) dan jumlah *genome* lengkap (sebanyak SPBU yang memesan).

Peletakan *genome* pada mobil tangki dengan kapasitas yang mencukupi, dapat dilihat pada kolom 6 di tampilan sistem (kolom rute mobil terbaik) saat sistem dijalankan. Meskipun pada kolom tersebut rute yang

dihasilkan adalah rute hasil evolusi, namun dengan tidak terdapatnya mobil tangki yang membawa muatan melebihi kapasitas selama proses eksperimentasi yang berlangsung dengan 30 kali proses, menandakan bahwa pada populasi awal pun tidak terdapat mobil yang membawa muatan melebihi kapasitas. Seandainya pada populasi awal terdapat mobil tangki yang membawa muatan melebihi kapasitas pastilah hasil evolusi pun akan demikian.

Metode percabangan yang dibuat pada proses peng-kode-an pun berjalan baik. Proses tersebut berjalan saat *genome* diletakkan, setelah *genome* diletakkan proses perhitungan kapasitas berjalan, jika kapasitas melebihi batas maka *genome* ditarik lagi dan dipindahkan ke mobil tangki selanjutnya, hingga menemukan mobil tangki dengan kapasitas yang mencukupi. Dengan terbentuknya *genome* pada semua kromosom terletak pada mobil tangki yang kapasitasnya mencukupi maka dikatakan bahwa proses percabangan berjalan sesuai yang diharapkan.

5.2 Fitness

Proses *coding* perhitungan nilai fitness berjalan dengan baik karena dapat menghitung nilai fitness sesuai dengan algoritma yang diterapkan. Hal tersebut dapat dilihat pada pengolahan data perhitungan nilai fitness secara manual yang memiliki hasil sama dengan perhitungan dengan sistem.

Dari sini dapat diketahui bahwa proses pengkodean untuk perhitungan nilai fitness berjalan dengan benar. Algoritma untuk perhitungan MAD dan perhitungan biaya total pun berjalan dengan benar. Nilai yang cukup besar yaitu diatas 1 juta adalah nilai gabungan antara nilai total biaya dengan pemerataan beban yang telah dikalikan dengan faktor perkalian.

Faktor perkalian adalah sebuah angka yang didapat dari perbandingan (dengan skala) antara nilai pemerataan beban tertinggi dengan total biaya distribusi tertinggi. Data itu diambil dari data historis selama 15 hari. Faktor perkalian yang dihasilkan cukup besar yaitu sebesar 283108,9 yang bila dikalikan akan menghasilkan fitness yang mungkin nilainya tidak lazim. Untuk itu maka dibutuhkan nilai kepentingan antara minimasi biaya dengan pemerataan beban dan didapat nilai kepentingan sebesar 70 (dari skala 100) untuk minimasi biaya, dan 30 (dari skala 100) untuk pemerataan beban, dan perbandingan dari keduanya sebesar 0,4 (didapat dari nilai objektif kepala Patra Niaga). Dengan nilai fitness dikalikan 0,4 ini maka nilai fitness tidak terlalu besar dan sudah dapat digunakan sebagai pembanding antar minimasi biaya dengan pemerataan beban.

Untuk mendapatkan nilai fitness yang lebih baik maka diperlukan data yang valid. Data yang digunakan untuk proses ini kurang valid, yaitu data pada jarak antar SPBU karena pihak perusahaan tidak memiliki data tersebut. Sehingga untuk mendapatkan data dilakukan dengan memanfaatkan skala pada peta wilayah kerja perusahaan, jarak tersebut dihitung, sehingga data kurang valid. Namun proses telah berjalan baik sehingga nantinya jika telah didapat data yang valid maka data pada database tinggal diganti saja.

5.3 Seleksi

Proses seleksi dilakukan dengan mengambil 40% dari jumlah *kromosom* yang harus *dicrossover* dari bagian atas dan sisanya diambil dari bagian bawah. Hal ini dilakukan agar tidak terjadi *konvergensi premature*. Karena jika proses seleksi hanya diambil dari bagian atas saja (memiliki fitness yang bagus) maka proses

generasi akan berlangsung cepat (mendapatkan nilai fitness yang sama) yang mungkin sebenarnya nilai tersebut masih jauh dari optimal. Untuk itu maka perlu dilakukan proses *crossover* antara kromosom yang baik dengan yang kurang baik.

Dari data yang dihasilkan pada eksperimentasi yang dilakukan sebanyak 30 kali percobaan data fitness paling kecil yang didapat sebesar 1510861,48 yaitu pada proses ke-2, ke-7, ke-8, ke-14, ke-17, ke-26, ke-27, ke-28. Dari skala nilai fitness terkecil adalah 1510861,48, ternyata masih terdapat nilai fitness yang besarnya diatas 1600000 telah bertahan sejak generasi awal hingga terakhir.

Pada percobaan ke-9 didapat nilai fitness terkecil sebesar 1641820,9012 (dimulai sejak generasi ke-63 hingga 215) dan percobaan ke-16 didapat nilai fitness terkecil sebesar 1842562,29 (dimulai sejak generasi ke-64 hingga 206). Kedua kasus tadi mungkin dikarenakan pada populasi awal nilai fitness pada masing-masing *kromosom* terlalu tinggi, atau mungkin juga dikarenakan nilai presentase bagian atas kurang (harus >40%) atau bisa dikarenakan keduanya. Dengan populasi awal yang memiliki skala fitness cukup tinggi maka nilai fitness anak hasil persilangan (*crossover*) pun menjadi tinggi.

Jika nilai fitness pada populasi awal berskala cukup tinggi dan presentase pengambilan bagian atas tetap 40% maka dapat terjadi *konvergensi premature*, karena kromosom bagian atas yang bernilai cukup besar akan memiliki peluang *dicrossover* dengan kromosom bagian bawah yang bernilai lebih besar dari bagian atas, sehingga dihasilkan anak yang nilai fitnessnya besar. Salah satu cara untuk mengatasi hal tersebut yaitu dengan meningkatkan presentase pengambilan kromosom pada bagian atas di setiap evolusinya.

Dengan ditingkatkannya presentase kromosom bagian atas terpilih, maka peluang proses crossover antara kromosom bagian atas dengan kromosom bagian bawah lebih kecil sehingga nantinya dihasilkan anak yang memiliki fitness kecil pula.

5.4 Proses *Crossover*

Proses pengkodean untuk *crossover* berjalan dengan benar. Hal tersebut terbukti dari hasil eksperimentasi. Setiap generasi pada setiap percobaan (sebanyak 30 kali percobaan) tidak terdapat kromosom yang memiliki *genome* ganda dan setiap *genome* tersebut menempati mobil tangki dengan kapasitas yang mencukupi.

Proses *crossover* dilakukan dengan metode *block crossover*, dengan metode tersebut dapat mengurangi kinerja programmer dalam meletakkan *genome* pada mobil tangki (*block*) yang muatannya mencukupi dan mengurangi kesalahan dalam penataan *genome*. Hal tersebut dikarenakan proses *crossover* dilakukan secara keseluruhan (dalam satu *block*). *Genome* dipindahkan ke mobil tangki *kromosom* lawan yang memiliki kapasitas muatan sama besar, sehingga secara pasti *genome* terletak pada mobil tangki dengan kapasitas yang mencukupi. Namun kelemahan dari metode *block crossover* adalah sering terdapatnya *genome* yang hilang, untuk itu perlu ditambah proses melengkapi *genome* yang hilang. Dalam proses ini melengkapi *genome* yang hilang dilakukan sekaligus dengan proses *mutasi*. Dengan adanya proses mutasi ini maka *genome* setiap *kromosom* lengkap.

5.5 *Mutasi*

Proses mutasi ini dilakukan selain untuk membuat sebuah kromosom baru yang mungkin belum ada pada kromosom-kromosom sebelumnya juga digunakan sebagai pelengkap genome yang hilang karena proses *crossover*. Proses ini berjalan dengan benar karena dari eksperimentasi tidak terdapat genome yang hilang.

Dengan terbentuknya kromosom baru tersebut diharapkan memiliki fitness yang lebih mendekati optimal. Pada eksperimentasi yang telah dilakukan proses mutasi berjalan, hal ini terlihat dari nilai fitness pada semua proses yang selalu terjadi penurunan. Apalagi pada percobaan ke-2 mutasi sangat jelas terlihat karena membawa secara langsung dari fitness sebesar Rp 1.712.259,60 menjadi fitness terkecil yaitu sebesar Rp 1.510.861,48. Namun terkadang proses peletakkan genome yang termutasi kembali ke posisi awal (sebelum dilakukan mutasi). Hal ini dikarenakan muatan yang ada pada mobil tangki lain sudah penuh atau tidak mencukupi jika ditambah dengan muatan yang ada pada genome tersebut. Mungkin hal ini juga yang mengakibatkan terperangkapnya kromosom pada percobaan ke-9 dan ke-16 yang nilai fitnessnya tetap besar hingga akhir evolusi.

Namun secara keseluruhan proses mutasi telah berjalan karena mengubah nilai fitness yang besar menurun hingga mendekati optimal.

5.6 Evolusi

Proses evolusi adalah proses mengulang-mengulang proses seleksi, proses *crossover*, proses *mutasi*, proses perhitungan fitness. Proses tersebut akan berlangsung hingga mendapatkan fitness paling mendekati optimal. Fitness yang dinyatakan mendekati optimal adalah jika nilai fitness sama untuk waktu yang

lama, jika demikian proses dapat dihentikan. Untuk proses evolusi ini terdapat proses pelestarian *kromosom* terbaik. Proses ini bertujuan untuk menambah peluang *kromosom* terbaik terpilih dalam proses *crossover* di *evolusi* selanjutnya. Proses ini pun telah berjalan, hal ini dapat dilihat pada setiap generasi pada setiap proses bahwa nilai fitness tidak ada yang naik (menjadi lebih besar) namun nilai tetap atau malah menurun.

Dari eksperimentasi yang telah dilakukan, proses evolusi untuk mendapatkan fitness yang mendekati optimal (untuk 100 populasi) dilakukan antara rentang 100 generasi hingga 250 generasi (berdasar data hasil eksperimentasi). Pada rentang generasi tersebut nilai fitness telah menunjukkan nilai yang kecil dan telah sama pada waktu yang cukup lama. Selain itu juga dengan dilakukan evolusi selama rentang 150 generasi hingga 250 generasi nilai fitness terkecil setiap kali proses kebanyakan memiliki perbedaan yang tipis (hampir sama). Hal ini dapat dilihat dari grafik.2 yang grafiknya hampir datar atau sebaran titik-titik tidak jauh, sekitar angka 1500000. Sehingga untuk populasi sebanyak 100 generasi maka proses evolusi yang optimal dilakukan sebanyak rentang antara 150 generasi hingga 250 generasi.

5.7 Pemerataan Beban dan Minimasi Biaya

Dari hasil eksperimentasi pemerataan beban dan minimasi biaya di atas, terlihat jelas pada grafik 3 bahwa semakin kecil nilai fitness yang diperoleh, garis yang tergambar pada grafik memiliki fluktuatif yang rendah. Hal ini dapat dilihat dari garis yang berwarna biru (data pada percobaan 10), disana jelas terlihat garis tersebut adalah garis yang memiliki tingkat fluktuatif yang paling kecil diantara

garis-garis lain. Dan nilai fitness pada percobaan 10 adalah nilai fitness terkecil dari ke-3 percobaan yang digunakan.

Meskipun percobaan ke-10 memiliki tingkat pemerataan beban kerja terbaik, namun belum menjamin fungsi minimasi biaya turut menjadi yang terbaik juga (biaya paling rendah). Hal ini dikarenakan sistem menyelesaikan permasalahan *multi objective*. Yang biasanya terjadi konflik pada tiap fungsi tujuannya.

5.8 Perbandingan dengan Perusahaan

Dari hasil perbandingan yang telah dilakukan dapat diketahui bahwa rute yang dihasilkan oleh sistem memiliki total biaya lebih mahal dan total *error* (semakin kecil *error* biaya beban kerja semakin rata) yang lebih kecil dari pada rute yang dirancang oleh Patra Niaga. Dimana selisih total biaya dalam 6 hari sebesar Rp 928.399,68 dan selisih *error* dalam 6 hari sebesar 33,31. Hubungan dari keduanya adalah berbanding terbalik dimana semakin kecil *error* (pemerataan beban) semakin besar (mahal) biaya distribusi.

Kasus ini merupakan konflik dari kedua fungsi tujuan. Dimana untuk pemerataan beban kerja maka harus ada penambahan objek tujuan (SPBU pemesan) yang berarti terdapat penambahan jarak tempuh pada mobil tangki yang beban kerjanya dianggap masih rendah. Objek tujuan diambil dari mobil tangki yang memiliki objek berlebih. Pembagian objek tujuan (jarak tempuh) inilah yang mengakibatkan beban kerja menjadi rata namun biaya menjadi mahal. Sebagai gambaran pada kasus ini sebagai berikut, terdapat 3 mobil tangki yaitu A, B, C dan SPBU pemesan D, E, F, G, H, I. Mobil tangki A mengirim ke SPBU D, E, F, G, mobil tangki B mengirim ke SPBU H, dan mobil tangki C mengirim ke SPBU

I. Karena mobil tangki A memiliki objek tujuan yang berlebih maka objek tujuan yang ada dikurangi dan ditambahkan pada mobil tangki B dan C. Sehingga pada akhirnya mobil tangki A mengirim ke SPBU D dan E, mobil tangki B mengirim ke SPBU F dan H, serta mobil tangki C mengirim ke SPBU G dan I. Dengan demikian beban kerja pada setiap mobil tangki menjadi rata namun terdapat penambahan biaya kirim. Karena 4 SPBU yang seharusnya dapat dikirim dengan satu mobil tangki akhirnya dikirim dengan 3 mobil tangki. Penambahan objek pada SPBU B dan C mengakibatkan penambahan jarak tempuh dan berdampak pada naiknya biaya distribusi.

Meskipun rute rancangan sistem memiliki biaya yang lebih mahal dan tingkat *error* (pemerataan beban kerja) kecil. Namun inilah yang menunjukkan bahwa sistem telah berjalan dengan benar. Yaitu sistem yang mampu memecahkan permasalahan *multi objective*. Karena bila peneliti fokus pada satu tujuan misal menghasilkan biaya yang kecil seperti tujuan pada perusahaan dapat dilihat bahwa tingkat *error* pada pemerataan beban akan besar (beban kerja jauh tidak rata).

Dengan beban kerja yang jauh lebih rata maka diharapkan kondisi tiap mobil dapat lebih stabil, umur mobil tangki lebih panjang, mobil tangki tidak mudah rusak, serta kesejahteraan pekerja terjamin. Sehingga dalam jangka panjang biaya yang dikeluarkan perusahaan untuk perbaikan mobil tangki dapat ditekan dan pengeluaran pun semakin kecil.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dari hasil pengolahan data dan analisa data di atas dapat ditarik kesimpulan bahwa :

6.1.1 Langkah Proses Algoritma Genetik dalam sistem

Dalam penelitian ini metode algoritma genetik berhasil dikembangkan menjadi sebuah sistem (*software*). Proses algoritma genetik dalam sistem berjalan dengan beberapa langkah, yaitu pembentukan populasi awal, perhitungan nilai *fitness*, seleksi, *crossover*, mutasi, pelestarian kromosom terbaik, evolusi.

6.1.2 Pemerataan beban dengan Minimasi Biaya Hasil Algoritma Genetik

Pemerataan beban yang dihasilkan telah mendekati rata. Ini dilihat dari grafik 3 dimana semakin kecil nilai *fitness*, garis yang digambarkan semakin memiliki tingkat fluktuatif yang kecil. Namun meskipun semakin kecil pemerataan beban, biaya yang dihasilkan belum tentu semakin rendah. Tetapi proses ini dapat membawa hasil mendekati optimal. Ini dilihat dari nilai *fitness* yang semakin menurun (fungsi tujuan minimasi) pada setiap generasi.

6.1.3 Perbandingan dengan Rute Rancangan Perusahaan

Dari hasil pengolahan data disimpulkan bahwa rute yang dihasilkan sistem, memiliki total *error* yang lebih rendah dari pada rute rancangan perusahaan dimana selisih *error* sebesar 33,31. Dan total biaya lebih mahal dari pada rute rancangan perusahaan dimana selisih biaya sebesar Rp 928.399,68.

6.2 Kontribusi Penelitian

Dalam penelitian yang telah dilakukan penulis berhasil mengembangkan beberapa metode yaitu :

6.2.1 *Vehicle Based Representation*

Dalam penelitian ini penulis berhasil mengembangkan metode *Vehicle Based Representation* . *Vehicle Based Representation* adalah pengembangan dari metode *random generator*. Kromosom hasil pengolahan dengan metode ini adalah *kromosom* yang memiliki N buah *block* yang didalam satu *block* terdapat N buah *genome*.

6.2.2 *Block Crossover*

Block Crossover adalah pengembangan dari metode *order crossover*. Metode ini digunakan untuk mengatasi kasus pemindahan kromosom yang harus mempertimbangkan kapasitas mobil tangki yang tersedia.

6.2.3 *Constrained Insertion Mutation*

Constrained Insertion Mutation merupakan pengembangan dari metode *insertion mutation*. Metode ini bertujuan untuk mengisi *genome* yang hilang pada solusi anak hasil *block crossover* dengan tetap memperhatikan kapasitas kendaraan.

6.3 Saran

Dari hasil analisa di atas terdapat beberapa saran yang mungkin dapat digunakan agar hasil lebih baik lagi, yaitu :

- 1) Faktor perkalian untuk menyamakan satuan antara pemerataan beban dengan minimasi biaya, harus dicari dengan metode yang lebih baik lagi dan dengan analisa yang lebih detail lagi agar mendapatkan hasil yang lebih optimal.
- 2) Untuk mengetahui apakah dengan Algoritma Genetik sudah mendapat hasil yang baik maka diharapkan ada pengembangan pencarian rute dengan menggunakan metode lain, sehingga dapat dilakukan perbandingan.

DAFTAR PUSTAKA

- Ai-ling, C., Gen-ke, Y., Zhi-ming, WU. 2005. *Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem*. Journal of Zhejiang University SCIENCE A, (10) : 607-609
- Archetti, C., Salvendy, M., W, P., Speranza, M, G. 2001. *Worst-Case Analysis for Split Delivery Vehicle Routing Problems*. Transportation Science, (13) : 226-228
- Bräysy, O., Gendreau, M. 2005. *Vehicle Routing Problem with Time Windows, Part II: Metaheuristics*. Transportation Science, (1) : 119-124
- Faradian, M, I. 2007. *Perbandingan Penggunaan Algoritma Genetika dengan Algoritma Konvensional pada Traveling Salesman Problem*. Teknik Informatika, Institut Teknologi Bandung, (5) : 3-10
- Heizer, J., Render, B. 2006. *Operation Management*. Salemba Empat, (14): 177
- Hendrawan, B, E. 2007. *IMPLEMENTASI ALGORITMA PARALEL GENETIC ALGORITHM UNTUK PENYELESAIAN HETEROGENEOUS FLEET VEHICLE ROUTING PROBLEM*. Teknik Informatika Institut Teknologi Sepuluh Nopember, (17) : 12-19
- Kritikos, M, N., Loannou, G. 2007. *The balanced cargo vehicle routing problem with time windows*. Int. J. Production Economics, (9) : 42-44
- Kusumadewi, S., dan Purnomo, H. 2005. *Penyelesaian Masalah Optimasi dengan Teknik-teknik Heuristik*. Graha Ilmu, (6) : 231-247
- Laporte, G. 1991. *The Vehicle Routing Problem: An overview of exact and approximate algorithms*. European Journal of Operational Research, (11) : 345-347
- Leksono, Agus. 2009. *ALGORITMA ANT COLONY OPTIMIZATION (ACO) UNTUK MENYELESAIKAN TRAVELING SALESMAN PROBLEM (TSP)*. Fakultas MIPA Universitas Diponegoro Semarang, (16) : 14-18
- Lukas, S., Anwar, T., Yuliani, W. 2005. *PENERAPAN ALGORITMA GENETIKA UNTUK TRAVELING SALESMAN PROBLEM DENGAN MENGGUNAKAN METODE ORDER CROSSOVER DAN INSERTION MUTATION*. Seminar Nasional Aplikasi Teknologi Informasi 2005, (7) : 1-5

- Mutakhirah,I.,Saptono,F.,Hasanah,N.,Wiryadinata,R. 2007. *Pemanfaatan Metode Heuristik dalam Pencarian Jalur Terpendek dengan Algoritma Semut dan Algoritma Genetika*. Seminar Nasional Aplikasi Teknologi Informasi 2007, (3) : 34-39
- Nugraha, I. 2008. *APLIKASI ALGORITMA GENETIK UNTUK OPTIMASI PENJADWALAN KEGIATAN BELAJAR MENGAJAR*. Teknik Informatika ITB, (18) : 1-6
- PHK, TIK1. 2008. *Algoritma Genetik*. Universitas Widyagama Malang,(15) : 1-16
- Prana, R. 2007. *Aplikasi Kombinatorial pada Vehicle Routing Problem*. Teknik Informatika Intitute Teknologi Bandung, (12) : 1-6
- Sarwadi, Anjar,KSW.2004. *Algoritma Genetika Untuk Penyelesaian Masalah Vehicle Routing*. Jurnal Matematika dan Komputer, (4) : 1-9
- Thot,P.,Vigo,D.2002. *Vehicle Routing Problem*. Society for Industrial and Applied Mathematics,(2) : 1-10
- Wirdianto, E., Jonrinaldi, Surya,B. 2007. *PENERAPAN ALGORITMA SIMULATED ANNEALING PADA PENJADWALAN DISTRIBUSI PRODUK*. Jurnal Optimasi Sistem Industri, (8) : 8-20